ระบบวางบิลหลายแพลตฟอร์มอัจฉริยะสำหรับร้านซูชิสายพาน

น.ส.พัชริยา ปิยะอารมณ์รัตน์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Intelligent Multi-platform Billing System for Conveyor Belt Sushi Restaurant

Miss Patchariya Piya-aromrat

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ระบบวางบิลหลายแพลตฟอร์มอัจฉริยะสำหรับร้านซูชิสายพาน
โดย	น.ส.พัชริยา ปิยะอารมณ์รัตน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

พัชริยา ปิยะอารมณ์รัตน์ : ระบบวางบิลหลายแพลตฟอร์มอัจฉริยะสำหรับร้านซูชิสายพาน. (Intelligent Multi-platform Billing System for Conveyor Belt Sushi Restaurant) อ.ที่ปรึกษาหลัก : รศ. ดร.ญาใจ ลิ่มปิยะกรณ์

ร้านซูชิสายพานเป็นที่นิยมทั่วโลก เนื่องจากราคาถูกกว่าร้านซูชิที่มีบริการ รวมทั้งรายการอาหารมีความหลากหลาย เพื่อประหยัดเวลาในการรับประทานอาหารธุรกิจซูชิสายพานจะแสดงราคาอาหารจากสีของจานรองที่แตกต่างกัน ขึ้นอยู่กับราคา การคำนวณการชำระบิลสามารถทำได้โดยการนับจำนวนจานแต่ละประเภท การตรวจจับวัตถุพัฒนาโดยใช้ปัญญาประดิษฐ์ เพื่อตรวจจับเพื่อลดระยะเวลาในการนับจานเพื่อคำนวณการชำระบิลแบบเดิม ระบบวางบิลอัจฉริยะพัฒนาโดยใช้ฟลัตเตอร์ สามารถทำงานได้บนหลายแพลตฟอร์ม โดยผู้ใช้สามารถถ่ายรูปกองจานซูชิเพื่อใช้เป็นภาพอินพุตเพื่อให้แบบจำลองที่พัฒนาด้วยโยโลวีสี่จำแนกจานสีต่างๆ ในวิทยานิพนธ์นี้พัฒนาในส่วนของการปรับปรุงภาพและแอปพลิเคชัน เนื่องจากระบบสามารถทำงานได้บนระบบปฏิบัติการไอโอเอส และแอนดรอยด์ เทคนิคการถ่ายโอนสีด้วยช่วงสีเอลเอบี คอนทราสต์ลิมิตอเดปทีฟอีควอไลเซชัน และเอสอาร์-ซีเอ็นเอ็น ถูกนำมาใช้เพื่อปรับปรุงคุณภาพของภาพ ผลการทดลองรายงานด้วยการตรวจจับวัตถุสำเร็จด้วยค่าความมั่นในการตรวจจับวัตถุที่สูงขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6370187721 : MAJOR COMPUTER SCIENCE

KEYWORD: image enhancement, Object detection, multi-platform, color transfer, CLAHE, Conveyer belt sushi

Patchariya Piya-aromrat : Intelligent Multi-platform Billing System for Conveyor Belt Sushi Restaurant. Advisor: Assoc. Prof. Yachai Limpiyakorn, Ph.D.

Conveyor belt sushi restaurants have become worldwide popular. The customers are drawn to the low prices and accessibility of the rotation sushi to save their dining time. These small dishes are typically served on different kinds of plates depending on price. Counting the number of each type of plates can then be computed for bill payment. An AI-based approach for sushi plate detection to overcome the time-consuming plate count is introduced in this work. A smart billing system is developed using flutter to run on multi-platform mobile application. The user can take a photo of stacks of consumed dishes as the input image to the YOLOv4 object detector for classifying the different colored plates. This paper focuses on parts of image enhancement as the system can work on iOS and Android systems. The techniques of color transfer with LAB color space ,CLAHE and SR-CNN are applied for image quality improvement. The experimental results reported the achievement of higher confidence and the successful detection of the objects from out of color space images.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Field of Study: Computer Science Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จสมบูรณ์ได้ด้วยดีเพราะได้รับความกรุณาอย่างดียิ่งจาก รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ความรู้ ให้คำแนะนำทั้งคำปรึกษา คอยผลักดัน และตรวจแก้ไขข้อบกพร่องมาโดยตลอด ผู้วิจัยขอกราบขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ดร.สุกรี สินธุภิญโญ ประธานกรรมการสอบวิทยานิพนธ์ และ อาจารย์ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ที่กรุณาเสียสละเวลาอันมีค่า ในการตรวจสอบและให้คำแนะนำที่เป็นประโยชน์ในการทำวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณบิดามารดาและเพื่อนๆ พี่ๆ น้องๆ ที่สนับสนุนและให้กำลังใจ ให้คำแนะนำความช่วยเหลือ รวมถึงขอบคุณตัวเองที่ไม่ยอมแพ้และพยายามในการทำวิทยาพนธ์ ซึ่งทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจต่อไปในภายภาคหน้า

พัชริยา ปิยะอารมณ์รัตน์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค
บทคัดย่อภาษาอังกฤษ ..... ง
กิตติกรรมประกาศ ..... จ
สารบัญ ..... ฉ
สารบัญตาราง 1
สารบัญรูปภาพ 1
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตการดำเนินงาน 2
1.4 ขั้นตอนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2
1.6 ลำดับการจัดเรียงวิทยานิพนธ์ 3
1.7 ผลงานที่ได้รับการตีพิมพ์ 3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
2.1 ทฤษฎีที่เกี่ยวข้อง 4
2.1.1 Flutter Opensource Framework 4
2.1.2 Color Space 4
2.1.3 Color Transfer 5
2.1.4 CLAHE (Contrast Limited Adaptive Histogram Equalization) 6
2.1.5 Super-Resolution Convolutional Neural Network (SR-CNN) 6

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 งานวิจัยที่เกี่ยวข้อง 8
บทที่ 3 แนวคิดและวิธีการวิจัย 13
3.1 การปรับความคมชัดของภาพ (SR-CNN) 15
3.2 การเปลี่ยนช่วงสี. (Convert color space) 18
3.3 การถ่ายโอนสี (Color transfer) 20
3.4 Contrast Limited Adaptive Histogram Equalization (CLAHE) 21
บทที่ 4 การทดลองและผลการทดลอง 24
4.1 สภาพแวดล้อมที่ใช้ในการทดลอง 24
4.2 ขั้นตอนการถ่ายโอนสี 25
4.3 Contrast Limited Adaptive Histogram Equalization (CLAHE) 27
4.4 SR-CNN 27
4.5 Flutter 33
4.6 ผลการทดลอง 37
บทที่ 5 สรุปผลการวิจัย 39
ภาคผนวก ก 40
ภาคผนวก ข 48
บรรณานุกรม 56
ประวัติผู้เขียน 59

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

ตารางที่ 1 ค่าเปรียบเทียบคุณภาพของภาพ 18

ตารางที่ 2 เปรียบเทียบค่าความมั่นใจเฉลี่ยของแบบจำลองก่อนและหลังการปรับคุณภาพของภาพ 38

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูปภาพ

ภาพที่ 1 RGB Color Space [3] 5
ภาพที่ 2 LAB Color Space [4] 5
ภาพที่ 3 ผลลัพธ์การทำ Color transfer [5] 6
ภาพที่ 4 ผลลัพธ์การปรับคุณภาพของภาพโดยการใช้แบบจำลอง SRCNN [7] 7
ภาพที่ 5 โครงสร้างการทำงานของ SRCNN [7] 7
ภาพที่ 6 ผลลัพธ์การทำ CLAHE โดยใช้ช่องสีเขียว [8] 9
ภาพที่ 7 ผลลัพธ์การปรับคุณภาพของภาพด้วย Laplacian filter และ CLAHE [11] 10
ภาพที่ 8 เปรียบเทียบค่า PSNR ของการทำงานอัลกอริทึมเก่า และอัลกอริทึมที่นำเสนอ[11] 10
ภาพที่ 9 สรุปเปรียบเทียบการปรับภาพใต้น้ำด้วยวิธีต่าง ๆโดยใช้ค่า Edge Contrast [10] 11
ภาพที่ 10 สรุปเปรียบเทียบการปรับภาพใต้น้ำด้วยวิธีต่าง ๆโดยใช้ค่า BRISQUE [10] 12
ภาพที่ 11 สถาปัตยกรรมของแอปพลิเคชัน 13
ภาพที่ 12 ขั้นตอนการทำ Data Preprocessing 14
ภาพที่ 13 ตัวอย่างภาพชุดข้อมูล 15
ภาพที่ 14 โครงสร้าง Hierarchical Data Format version 5 [12] 16
ภาพที่ 15 ภาพก่อนเข้าแบบจำลอง SRCNN 17
ภาพที่ 16 ภาพผลลัพธ์หหลังจากเข้าแบบจำลอง SRCNN 18
ภาพที่ 17 ขั้นตอนการถ่ายโอนสี [14] 20
ภาพที่ 18 กราฟแสดงการกระจายค่าความส่องสว่างของภาพก่อนและหลังการทำ CLAHE 22
ภาพที่ 19 โปรแกรม Android Studio 24
ภาพที่ 20 ไลบารีที่จำเป็นในการทำการถ่ายโอนสี 25
ภาพที่ 21 การหาค่าเฉลี่ยของภาพเป้าหมาย 25
ภาพที่ 22 การเปลี่ยนช่วงสี 26
ภาพที่ 23 การนำเอาสีพื้นออก 26

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาพที่ 24 การทำภาพอินพุตให้เป็นบรรทัดฐาน 26
ภาพที่ 25 การเพิ่มสี (Add source color) 27
ภาพที่ 26 การรวมช่องสี (merge color space) 27
ภาพที่ 27 การทำ Contrast Limited Adaptive Histogram Equalization 27
ภาพที่ 28 ไลบารีที่จำเป็นในการทำ SR-CNN 28
ภาพที่ 29 ขั้นตอนการเชื่อม Google Colab กับ Google drive 28
ภาพที่ 30 ขั้นตอนการแบ่งชุดข้อมูล 28
ภาพที่ 31 ขั้นตอนการเตรียมชุดข้อมูลสอนแบบจำลอง 29
ภาพที่ 32 ขั้นตอนการเตรียมชุดข้อมูลทดสอบแบบจำลอง 30
ภาพที่ 33 ขั้นตอนการสร้างโมเดล 31
ภาพที่ 34 ขั้นตอนการสอนโมเดล 31
ภาพที่ 35 ขั้นตอนการลดคุณภาพของภาพ 32
ภาพที่ 36 ขั้นตอนการนำภาพเข้าแบบจำลอง 32
ภาพที่ 37 ฟังก์ชันการคำนวณค่า PSNR 33
ภาพที่ 38 ฟังก์ชันการคำนวณค่า MSE 33
ภาพที่ 39 ฟังก์ชันการเปรียบเทียบภาพ 33
ภาพที่ 40 ไลบารี่ที่ใช้ใน Flutter 34
ภาพที่ 41 หน้าจอแสดงผลหน้าเมนู 35
ภาพที่ 42 หน้าจอแสดงผลรายการค่าบริการและ QR Code ชำระค่าบริการ 36
ภาพที่ 43 หน้าจอแสดงผลการชำระเงินสำเร็จและ QR Code สะสมแต้ม 36
ภาพที่ 44 ผลลัพธ์การถ่ายโอนสีของภาพ 37
ภาพที่ 45 กราฟแสดงค่าสีของภาพก่อนและหลังการปรับคุณภาพของภาพ 37

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบัน ร้านซูชิสายพานเริ่มเป็นที่นิยมในประเทศไทย เนื่องจากราคาถูกกว่าร้านซูชิที่มีบริกร รวมทั้งรายการอาหารมีความหลากหลาย จุดขายของร้านซูชิสายพาน คือ การที่ลูกค้าสามารถหยิบซูชิที่อยากรับประทานจากสายพานที่เคลื่อนที่ไปเรื่อยๆ โดยไม่จำเป็นจะต้องสั่งรายการอาหารกับพนักงาน ทำให้ลูกค้าสามารถบริหารเวลาในการรับประทานอาหารได้ ทั้งนี้ ธุรกิจซูชิสายพานจะแสดงราคาอาหารจากสีของจานรองที่แตกต่างกัน อ้างอิงจากร้านซูชิสายพานแฟรนไชส์ญี่ปุ่นแห่งหนึ่งที่ได้รับความนิยมในประเทศไทย ลูกค้าสามารถทราบราคาอาหารได้จากสีของจานรอง กล่าวคือ จานสีแดง ราคา 40 บาท จานสีเทา ราคา 60 บาท จานสีทอง ราคา 80 บาท และจานสีดำ ราคา 120 บาท สังเกตว่า ลูกค้าที่มารับประทานซูชิสายพานส่วนมากจะรับประทานกันเป็นจำนวนมาก ส่งผลให้ขั้นตอนชำระเงิน พนักงานจะต้องมาแยกสีของจาน และนับจำนวนจาน ยิ่งปริมาณมากก็จะใช้เวลามาก และเพิ่มโอกาสการเกิดความผิดพลาดในการคิดราคาค่าบริการมากขึ้น เมื่อชำระค่าบริการแล้ว ทางร้านจะมีการสะสมคะแนนเพื่อแลกของรางวัล โดยจะมีบัตรสะสมแจกให้กับลูกค้าในครั้งแรก และลูกค้าจำเป็นจะต้องพกบัตรสะสมมาด้วยทุกครั้งเพื่อสะสมคะแนน หากลูกค้าไม่ได้นำบัตรสะสมมาก็จะพลาดโอกาสในการสะสมคะแนนในครั้งนั้นไป

ผู้วิจัยจึงได้นำเสนอและพัฒนาแอปพลิเคชันการวางบิลค่าบริการร้านอาหารซูชิสายพาน แอปพลิเคชันที่พัฒนาขึ้น สามารถทำงานได้บนหลายแพลตฟอร์มทั้งบนโทรศัพท์มือถือ แท็บเล็ต และเว็บแอปพลิเคชัน โดยการเขียนโค้ดครั้งเดียว ทั้งนี้ เพื่อให้ระบบสามารถรองรับการคำนวณค่าอาหารได้อย่างถูกต้องมากยิ่งขึ้นจากภาพถ่ายจานซูชิที่ถ่ายจากกล้องของอุปกรณ์ที่หลากหลายแตกต่างกัน ผู้วิจัยจึงได้พัฒนาส่วนการปรับคุณภาพของภาพถ่าย เพื่อเพิ่มประสิทธิภาพการตรวจหาวัตถุ (ประเภทจานรองที่ใช้บอกราคาสินค้า) ประกอบด้วย การปรับความคมชัดของภาพ โดยพัฒนาแบบจำลอง Super-Resolution Convolutional Neural Network (SR-CNN) เพื่อเพิ่มความละเอียดคมชัดของภาพถ่าย รวมทั้งการปรับสีภาพด้วยวิธี Color Transfer ให้อยู่ในช่วงสีของภาพที่ใช้เรียนรู้การสร้างแบบจำลองตรวจหาวัตถุ ข้อดีของระบบที่พัฒนาขึ้นคือ สามารถคำนวณค่าอาหารแบบอัตโนมัติ โดยไม่ต้องใช้พนักงานมาคัดแยกสีจานและนับจำนวนจานรอง ทำให้การคำนวณค่าอาหารและคำนวณรายรับของร้านในแต่ละวันมีความรวดเร็วและถูกต้องมากยิ่งขึ้นเป็นการเพิ่มความสะดวกในการทำบัญชีของร้านค้า นอกจากนี้ ยังสามารถช่วยลดระยะเวลาในขั้นตอนการชำระเงินผ่าน QR code โดยลูกค้าไม่ต้องต่อแถวคอยนาน ลดความเสี่ยงการสัมผัสโรค เป็นการส่งเสริมสังคมไร้สัมผัส

(contactless society) ลดการใช้กระดาษ เป็นมิตรกับสังคมจากการลดขยะพิษจากการใช้เทคโนโลยี RFID ในปัจจุบัน นอกจากนี้ ระบบยังมีฟังก์ชันสะสมคะแนน เพื่ออำนวยความสะดวกให้กับลูกค้าในการไม่ต้องพกบัตรสะสมคะแนนมาทุกครั้ง ช่วยให้ลูกค้าไม่เสียโอกาสในการสะสมคะแนน

1.2 วัตถุประสงค์

งานวิจัยได้ออกแบบและพัฒนาระบบวางบิลร้านซูชิสายพานที่ทำงานได้ทั้งบนระบบปฏิบัติการ iOS และ Android โดยใช้เทคโนโลยี Flutter และมีการพัฒนาฟีเจอร์การเสริมแต่งภาพถ่ายข้อมูลนำเข้าเพื่อลดผลกระทบจากปัจจัยตัวแปรที่มีต่อสมรรถนะตัวตรวจหาวัตถุ รวมทั้งพัฒนาฟีเจอร์สะสมแต้มของสมาชิกเพื่อส่งเสริมการขายของธุรกิจ

1.3 ขอบเขตการดำเนินงาน

1.3.1 พัฒนาแอปพลิเคชันโดยใช้ Flutter

1.3.2 เปรียบเทียบสมรรถนะของตัวตรวจหาวัตถุกรณีไม่ใช้ฟีเจอร์การเสริมแต่งภาพ กับกรณีที่ใช้ฟีเจอร์การเสริมแต่งภาพ บนมือถือ 3 รุ่นที่แตกต่างกัน

1.3.3 ประเมินความถกต้องการทำงานของฟีเจอร์สะสมแต้มบนข้อกำหนดของร้าน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาค้นคว้าทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.4.2 ออกแบบและพัฒนาแอปพลิเคชัน

1.4.3 ทดสอบแอปพลิเคชัน

1.4.4 ประเมินผลงานวิจัย

1.4.5 เผยแพร่ผลงานวิชาการ

1.4.6 สรุปผลและเรียบเรียงวิทยานิพนธ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้แอปพลิเคชันหลายแพลตฟอร์มในการคิดค่าบริการอาหารประเภทซูชิสายพานจากภาพถ่ายภาชนะอาหาร ที่มีฟีเจอร์สะสมแต้มของลูกค้าที่เป็นสมาชิก เพื่อส่งเสริมการขาย

1.5.2 ได้นวัตกรรมที่เหมาะกับวิถีชีวิตปกติใหม่ ซึ่งลดสัมผัส และลดความเสี่ยงการติดเชื้อในสถานการณ์การแพร่ระบาดของโรคติดต่อ

1.6 ลำดับการจัดเรียงวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์แบ่งออกเป็น 5 บท ได้แก่

บทที่ 1 บทนำ

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทที่ 3 แนวคิดและวิธีการวิจัย

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 สรุปผลการวิจัย และแนวทางการวิจัยในอนาคต

1.7 ผลงานที่ได้รับการตีพิมพ์

- Piya-aromrat, P. & Limpiyakorn, Y. (2022). Image Enhancement for Rotation Sushi Plates Detection in Smart Multi-Platform Billing System. In 7th International Workshop on Pattern Recognition. Xiamen, China.
- Maitriboriruk, R., Piya-aromrat, P. & Limpiyakorn, Y. (2022). Smart Conveyor Belt Sushi Bill Payment with a Mobile Shot. In 4th International Conference on Information Technology and Computer Communications (ITCC). Guangzhou, China.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

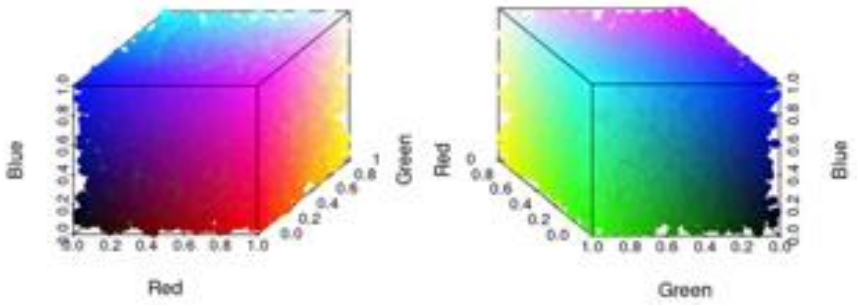
2.1.1 Flutter Opensource Framework

Flutter [1] เป็น Opensource Framework พัฒนาโดย Google สำหรับสร้างแอปพลิเคชันแแบบหลายแพลตฟอร์ม โดยการเขียนโค้ดเพียงครั้งเดียว ทำให้ลดเวลาในการพัฒนา นอกจากนี้ Flutter มีฟังก์ชันการทำงานที่เรียกว่า Hot Reload ที่ทำให้การ reload หน้าจอแสดงผลเมื่อมีการแก้ไขสามารถแสดงผลได้ในระยะเวลาที่สั้น ภาษาที่ใช้ในการพัฒนา Flutter คือ ภาษา Dart โดยภาษา Dart เป็น opensource ที่พัฒนาโดย Google เป็นภาษาโปรแกรมที่ใช้การเขียนแบบการเขียนโปรแกรมเชิงวัตถุ (object oriented programming) ซึ่งมีการออกแบบให้มีโครงสร้างที่ยืดหยุ่นและแก้ปัญหาโปรแกรมทำงานช้าและใช้หน่วยความจำ (memory) มากเกินไป ซึ่งเป้าหมายของภาษา Dart คือเป็นภาษาที่เรียนรู้ง่าย และทำงานได้บนอุปกรณ์พกพาขนาดเล็กและสามารถทำงานร่วมกับไลบารีต่างๆได้ ทำให้สะดวกต่อการพัฒนาแอปพลิเคชัน [2]

2.1.2 Color Space

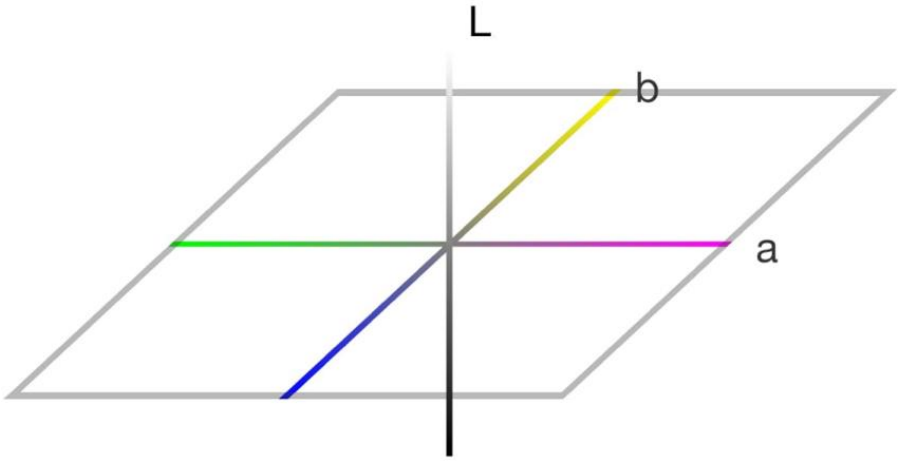
เป็นช่วงของสีที่ถูกสร้างขึ้นโดยโมเดลของสี ซึ่งแต่ละโมเดลจะมีวิธีการสร้างสีต่างๆ จากสีหลัก (Primary Color) แตกต่างกัน ทำให้ช่วงของสีที่สามารถสร้างได้แตกต่างกัน ในวิทยานิพนธ์นี้ได้อธิบายถึงโมเดลของสี 2 โมเดลประกอบด้วย โมเดลสี RGB. และโมเดลสี LAB

- RGB ย่อมาจาก Red, Green และ Blue คือ ระบบสีของแสง เกิดจากการหักเหของแสงกลายเป็นสีรุ้ง ประกอบด้วยด้วย 7 สี ซึ่งเป็นช่วงแสงที่มนุษย์สามารถมองเห็นได้ โดยแสงสีม่วงจะมีความถี่สูงสุด เรียกว่า อัลตราไวโอเล็ต (Ultraviolet)และแสงสีแดงจะมีความถี่ต่ำสุด เรียกว่า อินฟราเรด (Infrared) สำหรับคลื่นแสงที่มีความถี่สูงกว่าสีม่วงและต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถมองเห็นได้ แสงสีทั้งหมดเกิดจาก แสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำแม่สีของแสงทั้ง 3 มาผสมกัน ในปริมาณแสงสว่างเท่ากันก็จะได้เป็นแสงที่สีขาว แต่ถ้าผสมกันระหว่างแสงระดับความสว่างต่างกัน จะได้ผลลัพธ์เป็นแสงสีต่างๆ [3] ดังภาพที่ 1



ภาพที่ 1 RGB Color Space [3]

- LAB เป็น Color space ที่แยกระหว่างค่าสีกับค่าแสงสว่าง รูปแบบของ LAB จะสามารถอธิบายแยกได้เป็น 3 แกน ดังภาพที่. 2 แกนแรกคือแกน a เป็นแกนนอนแบบผลต่างของสีเขียวและม่วงแดง แกนที่สองคือแกน b เป็นแกนตั้งแบบผลต่างของสีน้ำเงินและเหลือง ซึ่งการเพิ่มขึ้นทั้งสองแกนจะได้สีแดง และถ้าลดลงทั้งสองแกนจะได้สีฟ้า แกนสุดท้ายซึ่งเป็นแกนของความสว่างตัดผ่านแกนนอนและแกนตั้ง เรียกว่าแกน L หรือ Light จะทำให้เราอธิบายความเป็นไปของขอบเขตรูปแบบการแสดงผลสีได้อย่างเห็นภาพมากยิ่งขึ้น ซึ่งในความเป็นจริงแล้วสีที่ผลิตได้ในแต่ละช่วงความสว่างยังมีความไม่เท่ากันเมื่อเทียบกับอุดมคติ [4]

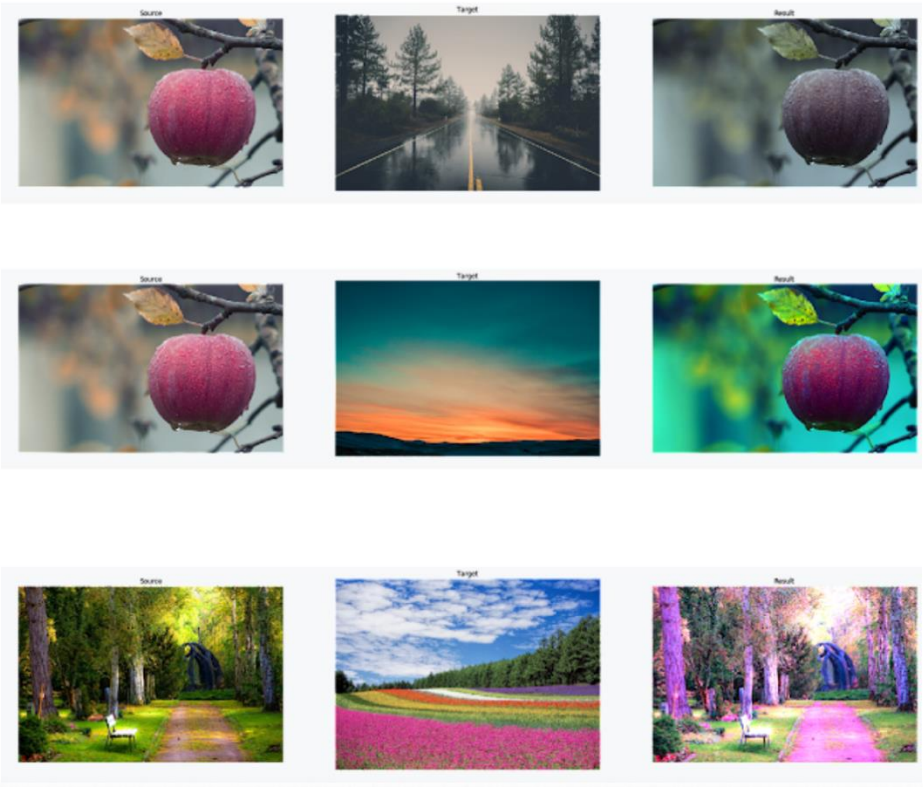


ภาพที่ 2 LAB Color Space [4]

2.1.3 Color Transfer

เป็นอัลกอริทึมในการย้ายสีจากภาพหนึ่งไปยังอีกภาพหนึ่ง ในการทำงานจะต้องมีรูปต้นทาง (source image) และรูปเป้าหมาย (target image) โดยรูปต้นทางจะเป็นรูปที่ต้องการนำเสนอ และรูปเป้าหมายเป็นรูปที่จะถูกถ่ายโอนสีให้กับรูปต้นทาง ดังภาพที่ 3 คอลัมน์แรกจะเป็นรูปต้นทางที่ต้องการนำเสนอ คอลัมน์ที่ 2 จะเป็นรูปที่นำสีถ่ายโอนไปยังรูปต้นทาง

และคอละมน์สุดท้ายเป็นผลลัพธ์ของการถ่ายโอนสี ในงานวิจัยนี้ใช้รูปต้นทางเป็นรูปที่ถ่ายด้วย iPhone 12 Promax เพื่อให้สอดคล้องกับสีของรูปใน dataset [5]



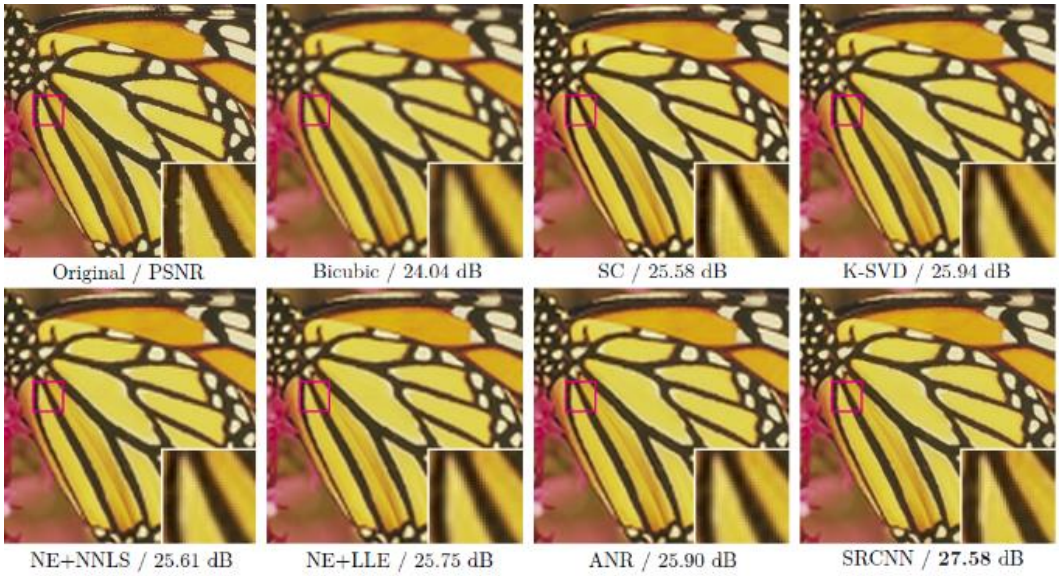
ภาพที่ 3 ผลลัพธ์การทำ Color transfer [5]

2.1.4 CLAHE (Contrast Limited Adaptive Histogram Equalization)

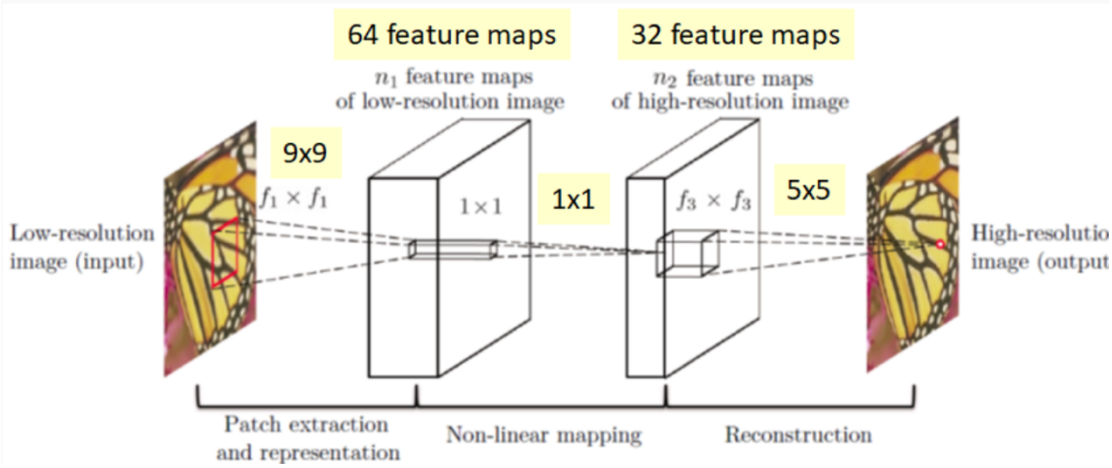
คือ วิธีการเพิ่มคุณภาพภาพ ที่ได้รับการพัฒนาจาก HE (HistogramEqualization) ซึ่งวิธีการดังกล่าวได้พิจารณาถึงรายละเอียด ข้อมูลจาก HE ในแต่ละค่าพิกเซลบนบริเวณพื้นส่วนกลางของภาพต้นฉบับ โดยค่า Histogram ที่มีระดับสูงกว่าค่าเฉลี่ย จะถูกนำมากระจายให้กับทุกพิกเซลในภาพ เพื่อให้เกิดความสมดุลกันของภาพมากยิ่งขึ้น [6]

2.1.5 Super-Resolution Convolutional Neural Network (SR-CNN)

SR-CNN มักใช้ในการประมวลผลภาพที่มีคุณภาพต่ำเพื่อให้ภาพมีคุณภาพดีขึ้น ตัวอย่างดังภาพที่ 4 ซึ่งอาจสังเกตไม่ได้ด้วยตาเปล่า แต่สามารถวัดได้จาก PSNR เป็นค่าที่บอกคุณภาพของภาพ ในการเตรียมภาพอินพุต



ภาพที่ 4 ผลลัพธ์การปรับคุณภาพของภาพโดยการใช้แบบจำลอง SRCNN [7]



ภาพที่ 5 โครงสร้างการทำงานของ SRCNN [7]

ภาพที่ 5 แสดงโครงสร้างการทำงานของ SRCNN [7] ประกอบด้วย 3 ขั้นตอน ดังนี้

- Patch Extraction and Representation

ในขั้นแรกจะต้องนำภาพอินพุตที่มีความละเอียดต่ำมาเพิ่มขนาดให้ได้ขนาดตามที่ต้องการโดยใช้ ไบคิวบิก (bicubic) ในการแก้ไขก่อนที่จะเข้าไปประมวลผลใน SRCNN โดยในชั้นแรกจะทำการกำหนดค่า Convolution ด้วย Relu ดังสมการที่ 1 โดยค่า Y คือภาพตัวอย่างความละเอียดต่ำที่ถูกประมวลผลโดยไบคิวบิก ค่า B_1 คือ ค่าขนาดมิติของ Bias ใช้สำหรับเพิ่ม

ระดับความอิสระ ค่า W_1 สามารถคำนวณได้ดังสมการที่ 2 โดยที่ค่า c คือ จำนวนช่องของรูป ค่า f1 คือขนาดของตัวกรอง (filter) ค่า n1 คือ จำนวนของตัวกรอง

$$F_1(Y) = max(0, w_1 \times Y + B_1) \quad (1)$$

$$W_1 = c \times f1 \times f1 \times n1 \quad (2)$$

- การทำแผนผังค่าแบบไม่เชิงเส้น (Non-Linear Mapping)

ขั้นตอนที่สองจะคำนวณค่า ดังสมการที่ 3 โดยจะทำการเทียบ (mapping) ค่า n1 และ n2 เมื่อค่า n1 > n2 W_2 สามารถคำนวณได้ ดังสมการที่ 4

$$F_2(Y) = max(0, w_2 \times F_1(Y) + B_2) \quad (3)$$

$$W_2 = n1 \times 1 \times 1 \times n2 \quad (4)$$

- การสร้างใหม่ (Reconstruction)

ขั้นตอนที่สามจะคำนวณค่า ดังสมการที่ 5 และค่า W_3 สามารถคำนวณได้ ดังสมการที่ 6

$$F_3(Y) = max(0, w_3 \times F_2(Y) + B_3) \quad (5)$$

$$W_3 = n2 \times f3 \times f3 \times c \quad (6)$$

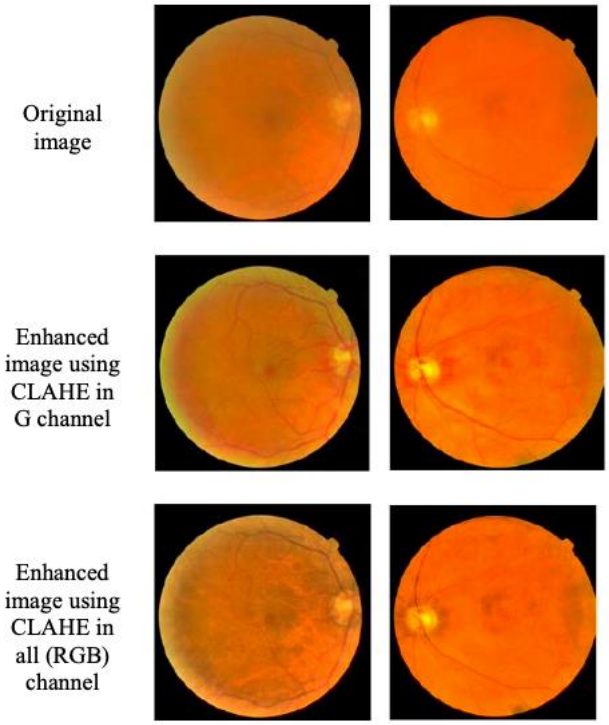
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Color Retinal Image Enhancement using CLAHE [8]

Agung W. Setiawan ได้นำเสนองานวิจัยการเพิ่มประสิทธิภาพของสีภาพจอประสาทตาโดยใช้ CLAHE (Contrast Limited Adaptive Histogram Equalization) เพื่อปรับสีและความคมชัดของหลอดเลือดในจอประสาทตา โดยเริ่มจากการนำภาพจอประสาทตาที่ใช้โมเดลสี RGB มาแบ่งเป็นช่องสี 3 ช่องได้แก่ ช่อง R คือช่องสีแดง ช่อง G คือช่องสีเขียวและช่อง B คือช่องสีน้ำเงิน และใช้เทคนิค CLAHE ปรับโดยใช้ช่อง R ช่อง G ช่อง B ตามลำดับ และปรับโดยใช้ทั้ง 3 ช่องพร้อมกัน ผล

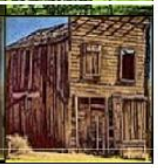
การทดลองในงานวิจัยนี้การปรับโดยใช้ช่องสีเขียวในการปรับสีของภาพทำให้ผลลัพธ์ของภาพจอประสาทตาดีที่สุด เพราะช่องสีเขียวทำให้สามารถเห็นหลอดเลือดได้ชัดที่สุดเมื่อเปรียบเทียบกับทุกช่องสี ดังภาพที่ 6



ภาพที่ 6 ผลลัพธ์การทำ CLAHE โดยใช้ช่องสีเขียว [8]

2.2.2 Color Image Enhancement using Laplacian filter and Contrast Limited Adaptive Histogram Equalization [9]

Satish Bhairannawar และคณะได้นำเสนอวิธีการเพิ่มประสิทธิภาพของภาพโดยใช้ Laplacian filter และ CLAHE โดยเริ่มจากการนำภาพที่เป็นที่มีช่วงสี RGB มาแปลงเป็นช่วงสี HSV จากนั้น apply Laplacian filter ให้กับค่า V และ S โดย Laplacian filter เป็นการหาบริเวณของภาพที่มีความไม่ต่อเนื่องของความคมชัดเพื่อนำมาปรับปรุงด้วยตัวดำเนินการอนุพันธ์เชิงเส้น เมื่อ apply Laplacian filter แล้วจะใช้เทคนิค CLAHE ปรับปรุงภาพโดยใช้ค่า V ขั้นตอนสุดท้ายจะแปลงภาพกลับไปเป็นช่วงสี จะได้ผลลัพธ์ดังภาพที่ 7 และจะประเมินผลด้วยค่า PSNR เพื่อเปรียบเทียบคุณภาพของภาพระหว่างอัลกอริทึมที่นำเสนอกับอัลกอริทึมแบบเก่า ดังภาพที่ 8

Sl no.	Image name	Original image	Proposed method	Sl no.	Image name	Original image	Proposed method
1	Butterfly			5	Tulips		
2	Clin mill			6	Peppers		
3	Flower			7	Barnfall		
4	Voit			8	Bodie		

ภาพที่ 7 ผลลัพธ์การปรับคุณภาพของภาพด้วย Laplacian filter และ CLAHE [11]

Sl No	**Image**	**Technique**	**Existing methods**	**Proposed method (Laplacian + CLAHE)**
1	Man image [15]	HSV + Geometric mean	34.0000	35.8534
2	Shed image [19]	CLAHE + DWT	31.8385	35.2900
3	Satellite image [19]		23.4730	25.1209
4	Low contrast image [20]	Gamma distribution	25.2100	30.7400

ภาพที่ 8 เปรียบเทียบค่า PSNR ของการทำงานอัลกอริทึมเก่า และอัลกอริทึมที่นำเสนอ[11]

2.2.3 Comparative Analysis of Underwater Image Enhancement Methods in Different Color Spaces [10]

Siaw-Lang Wong และคณะได้นำเสนอการเปรียบเทียบวิธีการเพิ่มประสิทธิภาพของภาพใต้น้ำทะเลที่แตกต่างกันทั้งหมด 3 วิธี ประกอบด้วย CS (Contrast stretching), HE (Histogram equalization) และ CLAHE (Contrast limited adaptive histogram equalization) ในช่วงสี RGB และ ช่วงสี HSV วิธีแรก Contrast Stretching เป็นวิธีพื้นฐานในการปรับคุณภาพของภาพ วิธีนี้จะทำการปรับค่า contrast โดยการยืดช่วงค่าของเข้ม เพื่อขยายช่วงของค่าสีที่ต้องการ วิธีถัดมาคือ Histogram Equalization เป็นวิธีที่ได้รับความนิยมเป็นอย่างมากในการใช้ปรับคุณภาพของภาพ วิธีนี้จะทำการเพิ่ม contrast โดยการปรับค่า Histogram โดยจะปรับโดยใช้ค่า Histogram ของสีเทาเพื่อให้มีการกระจายของสีเทาอย่างสม่ำเสมอ วิธีสุดท้ายคือ Contrast Limited Adaptive Histogram Equalization โดยวิธีการนี้จะแบ่งส่วนของรูปออกเป็นขอบเขตข้อมูลขนาดเล็กเพื่อที่จะค่อยๆปรับค่า Histogram ไปทีละขอบเขต ทำให้การกระจายของสีเทามีความสม่ำเสมอและทำให้การมองเห็นคุณสมบัติที่ซ่อนอยู่ในภาพดีขึ้น โดยประเมินคุณภาพของภาพจะประเมินจากค่า Edge Contrast (EC) และ BRISQUE จากการทดลองค่า BRISQUE อาจไม่ใช่ตัววัดที่ดีที่สุดในการวัดคุณภาพของภาพใต้น้ำทะเล จากการเปรียบเทียบวิธีการปรับประสิทธิภาพของภาพโดยใช้ CLAHE ดีที่สุด ดังภาพที่ 9 และภาพที่ 10

Images	Color Space	Original	CS	HE	CLAHE
Image 1	*RGB*	0.0456	0.0490	0.1165	**0.1399**
	HSV		0.0577	0.0761	0.1218
Image 2	*RGB*	0.0458	0.0477	0.1140	**0.1414**
	HSV		0.0563	0.0712	0.1231
Image 3	*RGB*	0.0260	0.0285	0.0796	**0.0827**
	HSV		0.0323	0.0395	0.0716
Image 4	*RGB*	0.0522	0.0561	0.1168	**0.1551**
	HSV		0.0597	0.0827	0.1374

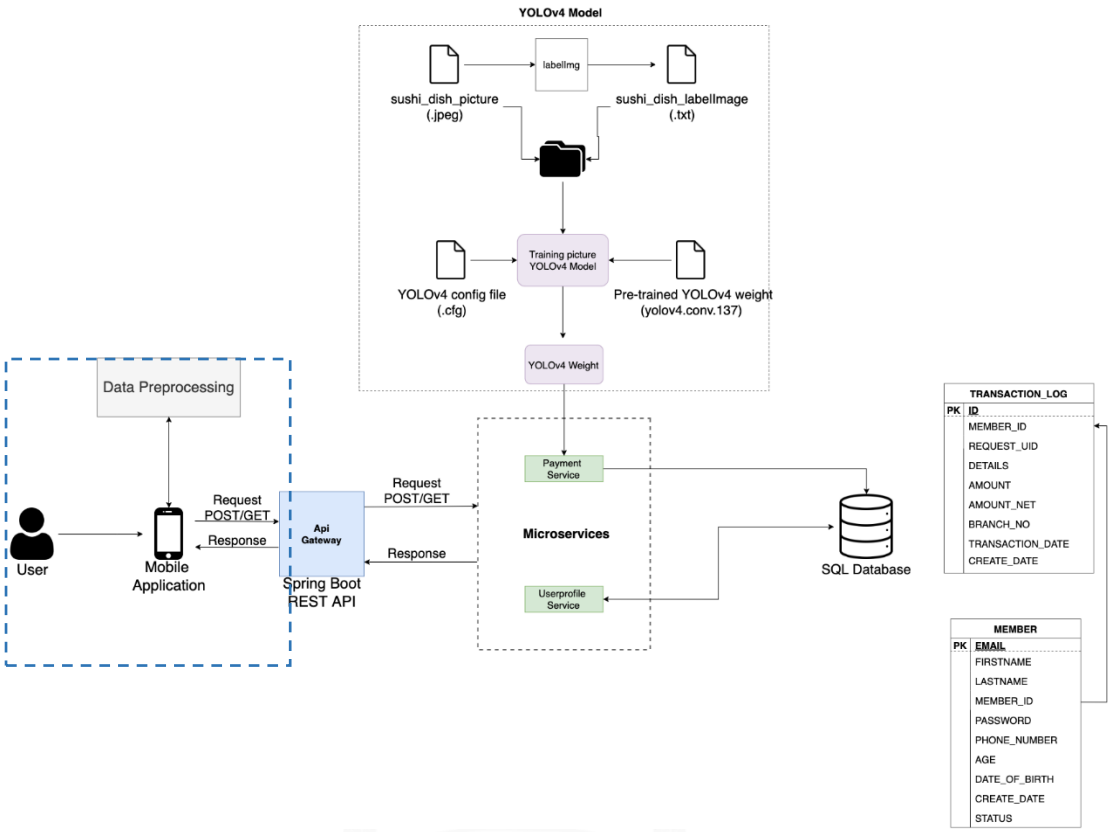
ภาพที่ 9 สรุปเปรียบเทียบการปรับภาพใต้น้ำด้วยวิธีต่าง ๆโดยใช้ค่า Edge Contrast [10]

Images	Color Space	Original	CS	HE	CLAHE
Image 1	*RGB*	7.5303	6.3520	15.5010	11.7646
	HSV		**6.1173**	11.9450	11.1244
Image 2	*RGB*	10.0392	9.0371	19.9947	14.2351
	HSV		**7.2689**	15.8671	12.7784
Image 3	*RGB*	26.8976	25.3699	24.5627	21.0511
	HSV		24.9521	32.4112	**20.6224**
Image 4	*RGB*	16.0517	14.6676	16.0709	10.5030
	HSV		12.9406	14.1900	**9.4000**

ภาพที่ 10 สรุปเปรียบเทียบการปรับภาพใต้น้ำด้วยวิธีต่าง ๆโดยใช้ค่า BRISQUE [10]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3
แนวคิดและวิธีการวิจัย



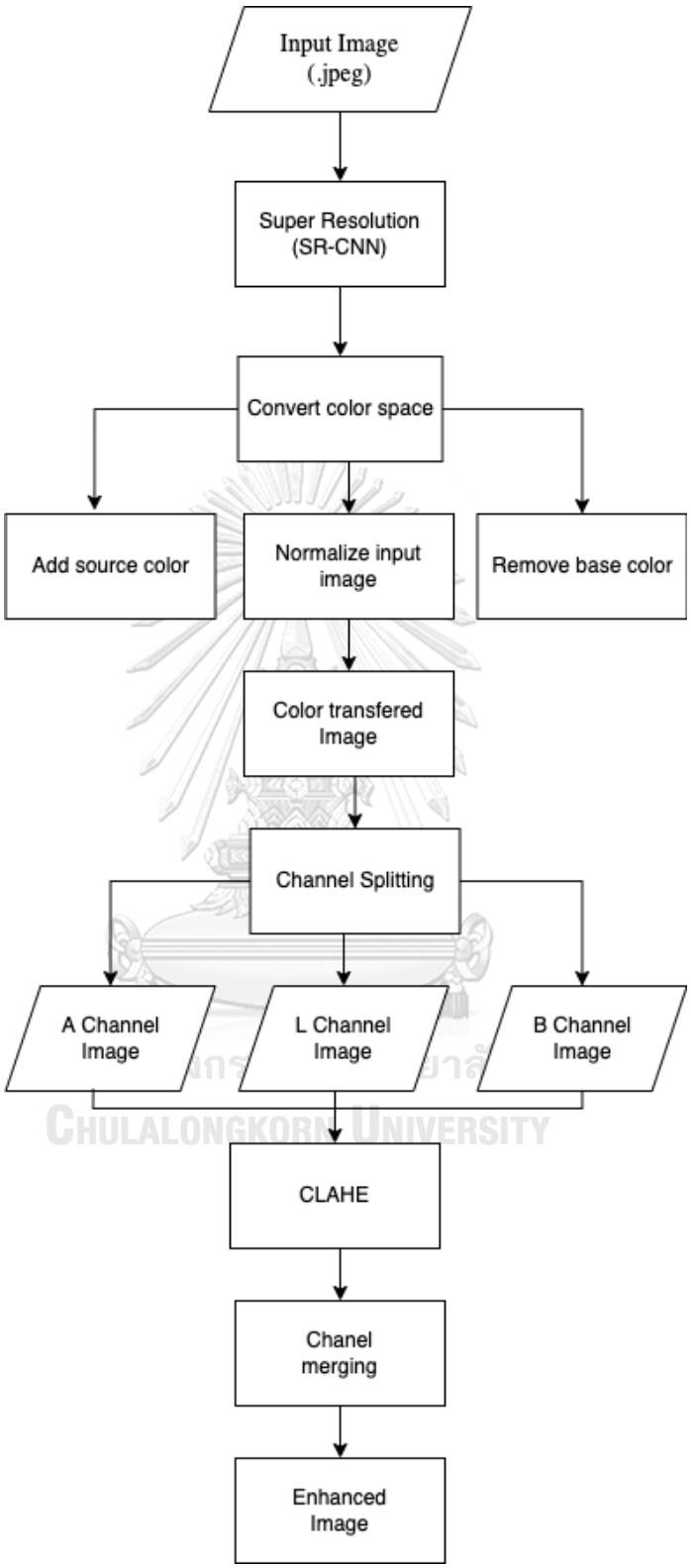
ภาพที่ 11 สถาปัตยกรรมของแอปพลิเคชัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การออกแบบสถาปัตยกรรมระบบวางบิลหลายแพลตฟอร์มอัจฉริยะสำหรับร้านซูชิสายพานมีโครงสร้างและส่วนต่างๆ ดังแสดงในภาพที่ 11 ในงานวิจัยนี้จะครอบคลุมเฉพาะ:

1) การพัฒนาแอปพลิเคชันบนมือถือด้วย Flutter ซึ่งรองรับการทำงานหลายแพลตฟอร์มจากการเขียนโค้ดครั้งเดียว

2) การพัฒนา Data Preprocessing component ประกอบด้วยขั้นตอนต่างๆ ดังภาพที่ 12 เพื่อปรับปรุงคุณภาพของภาพถ่ายจากหลากหลายอุปกรณ์ให้รองรับการทำงานส่วนการตรวจหาวัตถุซึ่งพัฒนาขึ้นในงานวิจัย [11] ได้อย่างมีประสิทธิภาพ



ภาพที่ 12 ขั้นตอนการทำ Data Preprocessing

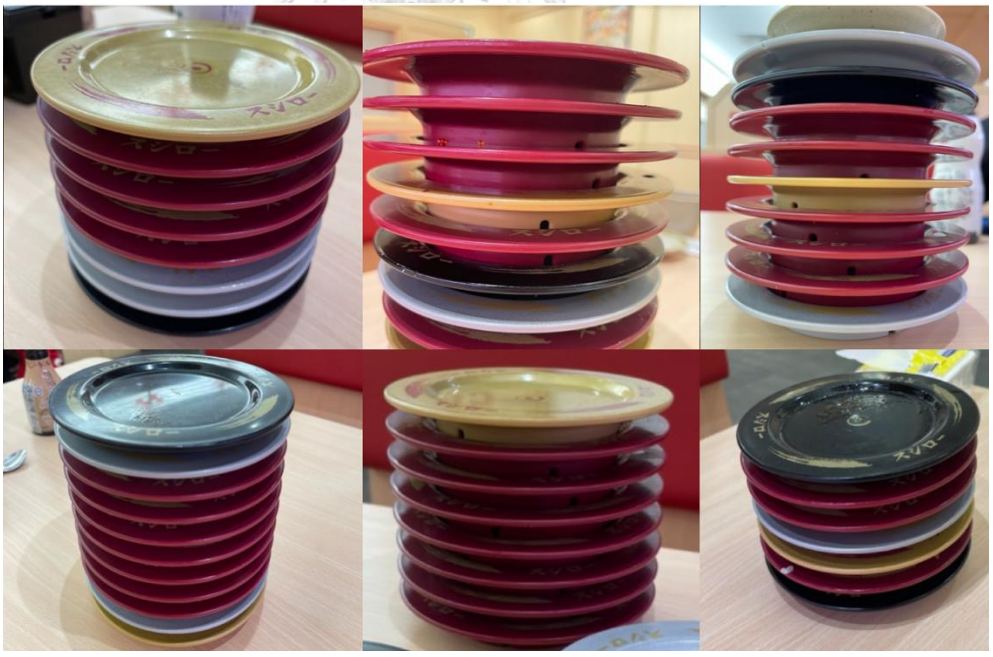
CHULALONGKORN UNIVERSITY

3.1 การปรับความคมชัดของภาพ (SR-CNN)

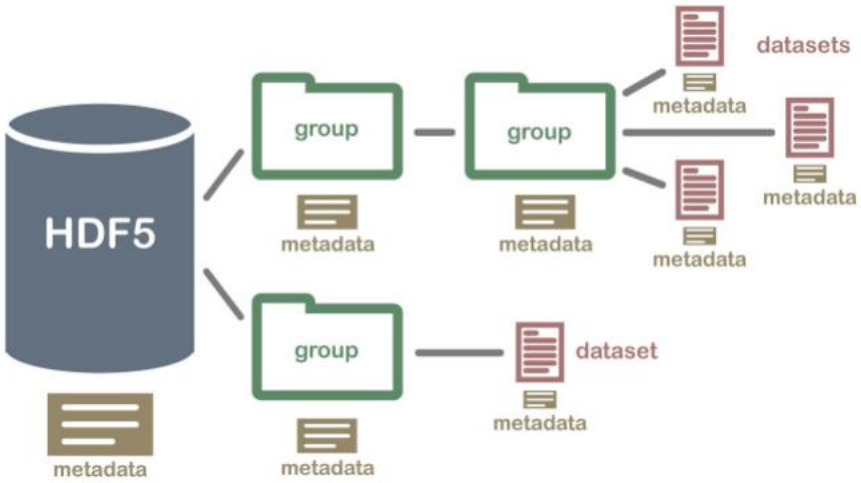
ในการปรับความคมชัดของภาพ ในงานวิจัยนี้จะใช้โมเดล Super-Resolution Convolutional Neural Network (SR-CNN)

- การเตรียมข้อมูล

ในการเตรียมข้อมูลจะต้องมีชุดข้อมูลภาพที่มีความละเอียดสูง และภาพที่มีความละเอียดต่ำ ในงานวิจัยนี้ชุดข้อมูลจะเป็นจานจานซูชิที่ถ่ายด้วยกล้อง iPhone 12 Promax ดังภาพที่ 13 เป็นภาพที่มีความละเอียดสูง และจะใช้ฟังก์ชันในโอเพนซีวีในการปรับภาพให้มีความละเอียดต่ำเพื่อใช้เป็นชุดข้อมูลที่มีความละเอียดต่ำ เพื่อให้ได้ชุดข้อมูลในการเตรียมสอนโมเดลในรูปแบบไฟล์ Hierarchical Data Format version 5 (HDFS5) ซึ่งเป็นระบบการจัดเก็บข้อมูลชนิดหนึ่ง ซึ่งรองรับการเก็บข้อมูลต่าง ๆ เช่น ข้อมูลภาพ .jpg หรือข้อมูลในรูปแบบตาราง .csv โดยข้อมูลที่เก็บในโครงสร้าง HDF5 จะเป็นข้อมูลชนิดเดียวกัน หรือต่างชนิดกันก็ได้ โดยข้อมูลดังกล่าวจะถูกรวมไว้ในไฟล์เพียงไฟล์เดียว โครงสร้างของ HDF5 ดังภาพที่ 14 จะประกอบด้วยกลุ่ม (Group) เปรียบเสมือนโฟลเดอร์ (folder) มีไว้สำหรับเก็บชุดข้อมูล และ ชุดข้อมูล เป็นกลุ่มของข้อมูลที่ถูกเก็บในไฟล์ .h5 [12]



ภาพที่ 13 ตัวอย่างภาพชุดข้อมูล



ภาพที่ 14 โครงสร้าง Hierarchical Data Format version 5 [12]

- การสอนโมเดล (Train model)

ในการสอนโมเดล ในงานวิจัยนี้ใช้โมเดล SR-CNN และแบ่งชุดข้อมูลในการสอนเป็นร้อยละ 80 และแบ่งข้อมูลในการทดสอบเป็นร้อยละ 20 จากภาพในชุดข้อมูลทั้งหมด 400 ภาพ ตั้งค่า batch size เป็น 64 และจำนวน epoch เป็น 200

- การประเมินผลโมเดล

ในการประเมินผลของโมเดลจะใช้ค่า PSNR (Peak Signal to Noise Ratio) เป็นค่าที่บอกคุณภาพของภาพ โดยค่า PSNR จะสามารถคำนวณได้ดังสมการที่ 7 ค่า MAX_I คือค่าพิกเซลสูงสุดของภาพ และค่า MSE (Mean Square Error) คำนวณได้ดังสมการที่ 8

Chulalongkorn University

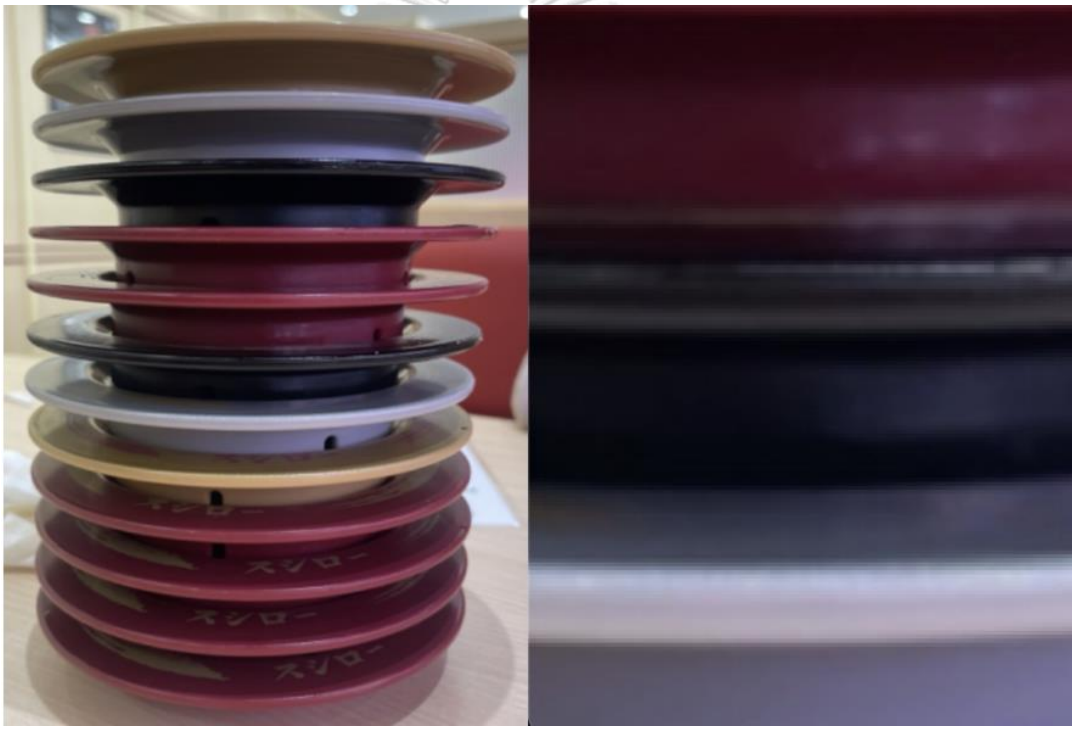
$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

$$PSNR = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (7)$$

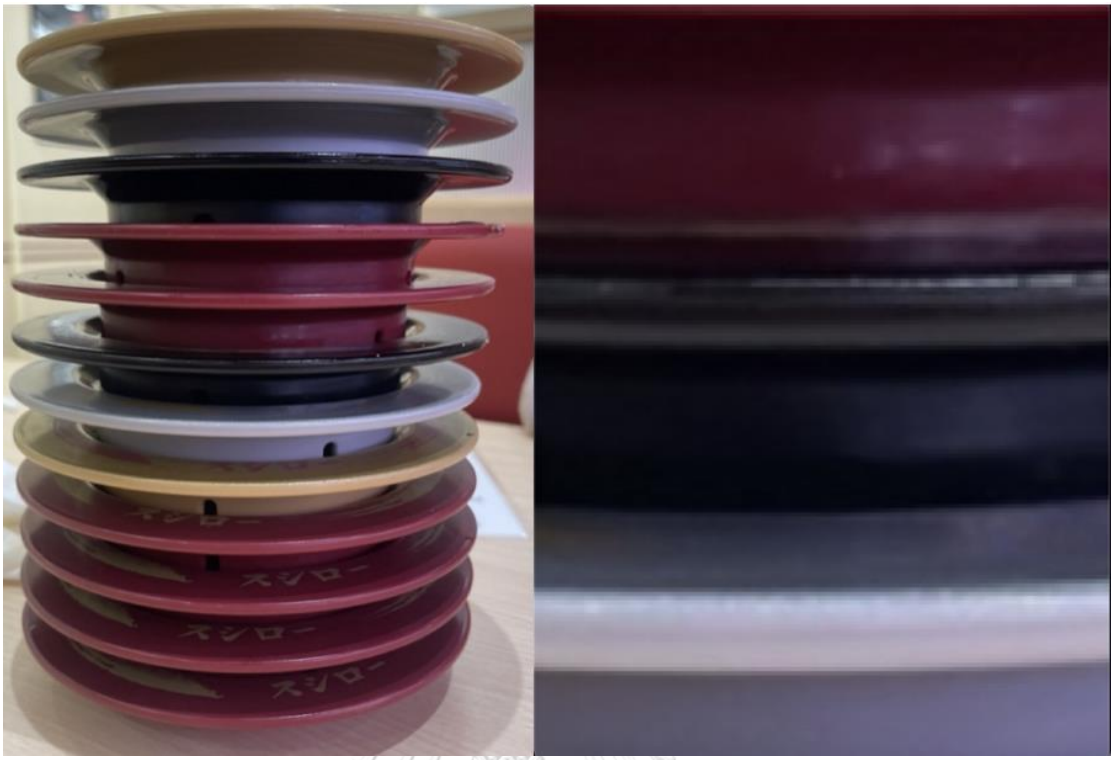
$$PSNR = 20log_{10}(MAX_I) - 10log_{10}(MSE)$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (8)$$

จากการนำภาพเข้าแบบจำลอง SR-CNN เพื่อที่จะปรับคุณภาพของภาพให้ดีขึ้น ภาพต้นฉบับดังภาพที่ 15 เป็นภาพต้นฉบับที่ยังไม่ผ่านการปรับปรุงคุณภาพของภาพ จะเห็นได้จากภาพทางด้านขวาที่ซูม จะมีความแตกและเห็นขอบจานไม่ชัด เมื่อเทียบกับภาพที่ผ่านการปรับปรุงคุณภาพของภาพแล้วดังภาพที่ 16 ภาพที่ซูมจะมีความละเอียดที่ดีขึ้นและเห็นขอบจานชัดขึ้น จากการเปรียบเทียบค่าตัววัดดังตารางที่ 1 พบว่า ค่า PSNR เพิ่มขึ้นจาก 40.215798 เป็น 43.667236 ยิ่งค่า PSNR มาก หมายความว่าภาพยิ่งมีคุณภาพดี และค่า MSE จาก18.561870 ลดลงเป็น 8.384516 ยิ่งค่า MSE ลดลง หมายความว่าค่าความผิดพลาดน้อยลง และสุดท้ายค่า SSIM เป็นค่าความเหมือนกับภาพต้นฉบับจะเห็นได้ว่าค่าเพิ่มขึ้นจาก 0.979200 เป็น 0.987906 ผลลัพธ์การปรับความชัดดังภาคผนวก ก



ภาพที่ 15 ภาพก่อนเข้าแบบจำลอง SRCNN



ภาพที่ 16 ภาพผลลัพธ์หหลังจากเข้าแบบจำลอง SRCNN

ตารางที่ 1 ค่าเปรียบเทียบคุณภาพของภาพ

ภาพ/ค่าตัววัด	PSNR	MSE	SSIM
ภาพต้นฉบับ	40.215798	18.561870	0.979200
ภาพผลลัพธ์	43.667236	8.384516	0.987906

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 การเปลี่ยนช่วงสี. **(Convert color space)**

การเปลี่ยนช่วงสีเป็นการเพิ่มความเป็นอิสระต่อกันของสี เริ่มจากการเปลี่ยนจากช่วงสี RGB ซึ่งเป็นช่วงสีส่วนมากของรูปภาพดิจิทัล โดย RGB จะมีสหสัมพันธ์ (correlation) กันระหว่างช่วงสีแดง เขียว และน้ำเงิน ในการลดสหสัมพันธ์จะทำการเปลี่ยนช่วงสี RGB เป็นช่วงสี LAB โดยช่วงสี LAB เป็นช่วงสีที่แยกค่าแสงกับค่าสีอย่างชัดเจนทำให้เกิดอิสระต่อกัน โดย L คือค่าของแสง ค่า A เป็นค่าของช่วงสีเขียวถึงแดง ค่า B. เป็นค่าของช่วงสี น้ำเงินถึงเหลือง โดยการเปลี่ยนช่วงสีจาก RGB เป็น LAB เราจะทำการเปลี่ยน จากช่วงสี RGB เป็นช่วงสี. LMS ดังสมการที่ 9 ก่อนจากนั้นจึงนำช่วงสี LMS มาเปลี่ยนเป็นช่วงสี LAB [13] ดังสมการที่ 10

$$\begin{bmatrix} L \\ S \\ M \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

(9)

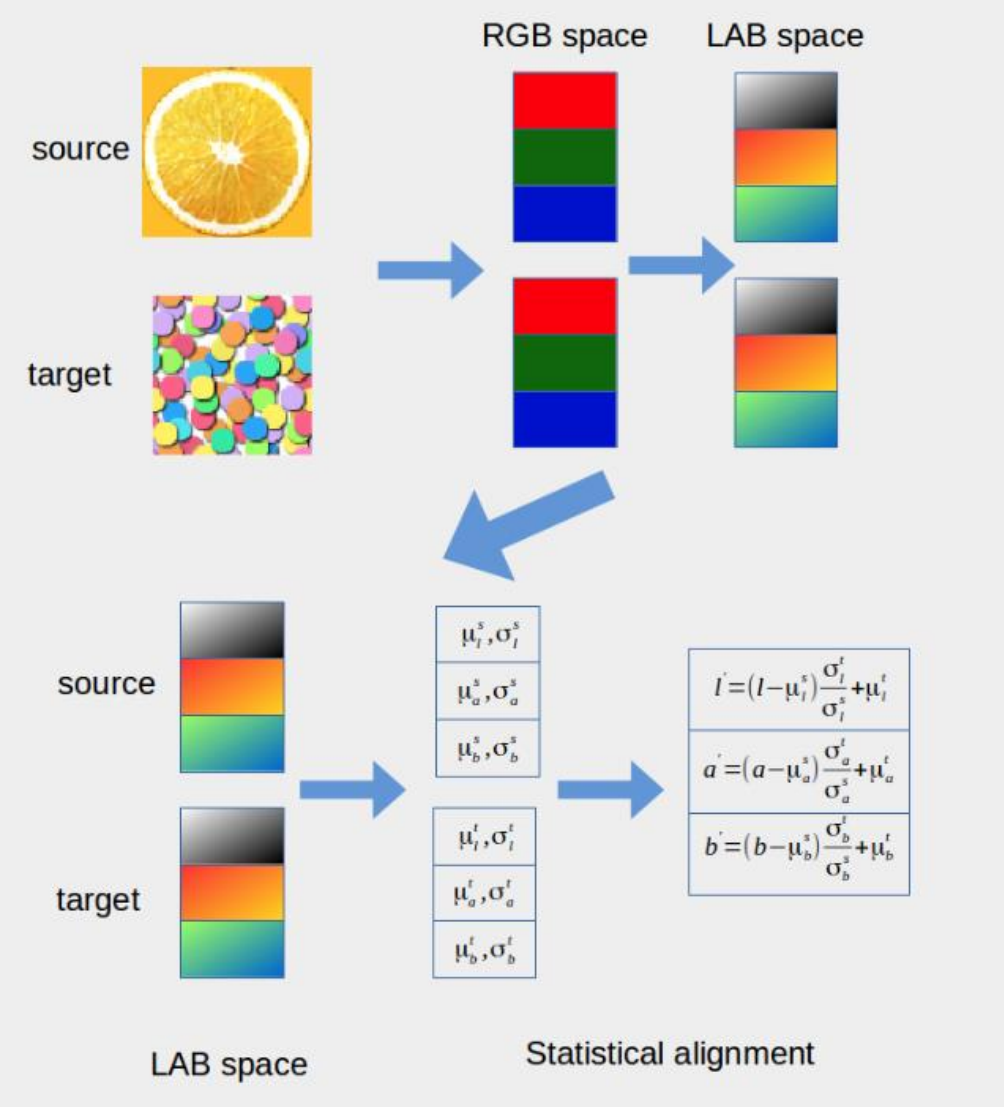
$$\begin{aligned} L &= \log 10(L) \\ M &= \log 10(M) \\ S &= \log 10(S) \end{aligned}$$

$$\begin{bmatrix} L \\ A \\ B \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (10)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3 การถ่ายโอนสี (Color transfer)

ภาพที่ 17 แสดงการถ่ายโอนสี ประกอบด้วย 3 ขั้นตอน ดังนี้



ภาพที่ 17 ขั้นตอนการถ่ายโอนสี [14]

- การนำเอาสีพื้นออก (Remove base color)

การนำเอาสีพื้นออกจากภาพ จะทำโดยการหาผลต่างระหว่างค่าสีของแต่ละช่องกับค่าเฉลี่ย (mean) ดังสมการที่ 11 โดยค่า μ แทนค่าเฉลี่ยของแต่ละช่องสี

$$l = l - \mu_l \tag{11}$$

$$a = a - \mu_a$$
$$b = b - \mu_b$$

- การทำภาพอินพุตให้เป็นบรรทัดฐาน (Normalize input image)

การทำภาพอินพุตให้เป็นบรรทัดฐานจะทำโดยการสเกล (scale)
ค่าที่ได้มาจากการนำเอาสีพื้นออกในขั้นตอนก่อนหน้าดังสมการที่ 12 โดย σ เป็นค่าการทำ standard deviation ของแต่ละช่องสี

$$l' = \frac{\sigma_l^t}{\sigma_l^s} - l$$
$$a' = \frac{\sigma_a^t}{\sigma_a^s} - a \quad (12)$$
$$b' = \frac{\sigma_b^t}{\sigma_b^s} - b$$

- การเพิ่มสี (Add source color)

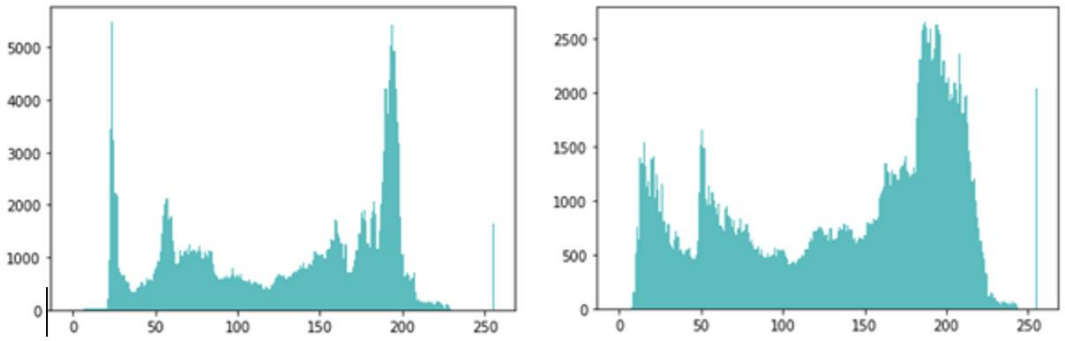
การเพิ่มสีจะเพิ่มโดยใช้บวกค่าเฉลี่ยของภาพที่เราต้องการไปยังภาพที่ต้องการเปลี่ยนสี ดังสมการที่ 13

$$l' = l' + \mu_l^t$$
$$a' = a' + \mu_a^t \quad (13)$$
$$b' = b' + \mu_b^t$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.4 Contrast Limited Adaptive Histogram Equalization (CLAHE)

โดยการทำ Contrast Limited Adaptive Histogram Equalization จะต้องทำการแบ่งสีแต่ละช่องก่อน แล้วค่าที่ใช้ในการกระจายค่าเฉลี่ยจะใช้ค่า L (Luminance) เพื่อให้ไม่กระทบกับสีที่ผ่านกระบวนการเปลี่ยนแปลงช่วงสี ในการทำ Contrast Limited Adaptive Histogram Equalization จะใช้โอเพนซีวี. (OpenCV) เมื่อภาพผ่านกระบวนการ Contrast Limited Adaptive Histogram Equalization ดังภาพที่ 18 แกน x แสดงค่าความส่องสว่างและแกน y แสดงจำนวนพิกเซลที่มีความส่องสว่างนั้น จากภาพกราฟทางด้านซ้ายจะเห็นว่ามีการกระจายกันของความส่องสว่างมาก ภาพทางด้านขวาเป็นกราฟที่แสดงภาพที่ผ่านกระบวนการ Contrast Limited Adaptive Histogram Equalization แล้วจะเห็นได้ว่ามีการกระจายตัวกันของค่าความส่องสว่างลดลง



ภาพที่ 18 กราฟแสดงการกระจายค่าความส่องสว่างของภาพก่อนและหลังการทำ CLAHE

3.5 การพัฒนาแอปพลิเคชันด้วย Flutter

การพัฒนาแอปพลิเคชันด้วย Flutter เป็นการพัฒนาที่เขียนโค้ด 1 ครั้งสามารถทำงานได้ทั้งบนระบบปฏิบัติการ iOS และ Android ในการพัฒนาจะใช้โปรแกรม Android Studio และใช้ภาษา Dart มีการใช้ไลบารี (Library) ในการเชื่อม Flutter กับ โครงสร้างแบบโครงคร่าว (framework)

- ภาษา Dart

เป็นภาษาที่ใช้ในการพัฒนาแอปพลิเคชัน Flutter โครงสร้างของภาษา DART คล้ายกับ C/C++ และ Java โดยที่จะมีความเป็นภาษาแบบ Structure Programming แต่ก็ยังมีความสามารถแบบภาษาประเภท Object Oriented Programmingภาษา Dart มีโครงสร้างที่ยืดหยุ่นมากพอ (structured yet flexible language) และเป็นการออกแบบตัวภาษาไปพร้อมกับตัวเครื่องประมวลผล (Engine) สำหรับรันภาษาเลยเพื่อแก้ปัญหาโปรแกรมทำงานช้าและใช้หน่วยความจำ (memory) มากเกินไป ซึ่งเป้าหมายของภาษา Dart คือเป็นภาษาที่เรียนรู้ง่าย และทำงานได้บนอุปกรณ์พกพาขนาดเล็ก มือถือไปจนถึงเครื่องบริการ (Server)

- การออกแบบส่วนต่อขยายกับผู้ใช้. (User Interface)

ในการสร้างส่วนต่อขยายกับผู้ใช้ Flutter จะเรียกองค์ประกอบต่างๆว่า Widget โดยนำมาประกอบเรียงกันเป็นลำดับขั้นขึ้นเป็นโครงสร้าง แต่ละ widget จะถูกวางซ้อนอยู่ภายใน Parent widget และได้รับการส่งต่อสืบทอดคุณสมบัติต่างๆ จาก Parent แม้กระทั้ง application object ก็ถือเป็น widget ซึ่งเราเรียกว่า root widget Widget แบ่งออกเป็น 2 ประเภทได้แก่ stateless และ stateful

  - Stateless เป็น widget ที่ไม่จำเป็นต้องมีการเปลี่ยนแปลงสถานะ (State) โดยเราจะใช้ stateless widget สำหรับสร้าง widget แบบคงที่ เหมาะสำหรับใช้

ในการสร้าง และกำหนดส่วนของส่วนต่อขยายกับผู้ใช้ ซึ่งจะปรับแต่งเฉพาะค่าข้อมูลของ ตัว widget เท่านั้น เช่น Text widget

- Stateful เป็น widget ที่มีการเปลี่ยนแปลงของสถานะ โดยจะมีการใช้งานคำสั่ง setState() เพื่อกำหนดการเปลี่ยนแปลง โดยการเรียกใช้คำสั่ง setState() เป็นการบอกให้ flutter รู้ว่ามีการเปลี่ยนแปลงเกิดขึ้นกับ state และ App ต้องทำการ rerun หรือทำคำสั่ง build() ใหม่ ดังนั้นตัว App จึงได้รับผลจากการเปลี่ยนแปลงที่เกิดขึ้น สถานะเป็นข้อมูลที่สามารถนำมาใช้งานได้ต่อเนื่องในขณะที่ widget ถูกสร้าง และอาจจะมีการเปลี่ยนแปลงในทุกช่วงเวลาที่มีการใช้งานของ widget ดังนั้นจำเป็นต้องกำหนดการทำงานรองรับเมื่อมีการเปลี่ยนแปลงของ state เกิดขึ้นStateful Widget จะถูกใช้งานเมื่อต้องการให้ widget รองรับการเปลี่ยนแปลงที่เกิดขึ้นอัตโนมัติ เช่น สถานะมีการเปลี่ยนแปลงเมื่อทำการพิมพ์ข้อความลงไปในฟอร์มหรือ สถานะมีการเปลี่ยนแปลงจากข้อมูลที่ได้รับมาหรือมีการอัพเดท เป็นต้นตัวอย่าง widget ที่เป็น stateful widget ที่เราน่าจะคุ้นกับรูปแบบการใช้งานก็เช่น Checkbox, Radio, Slider, Form และ TextField

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

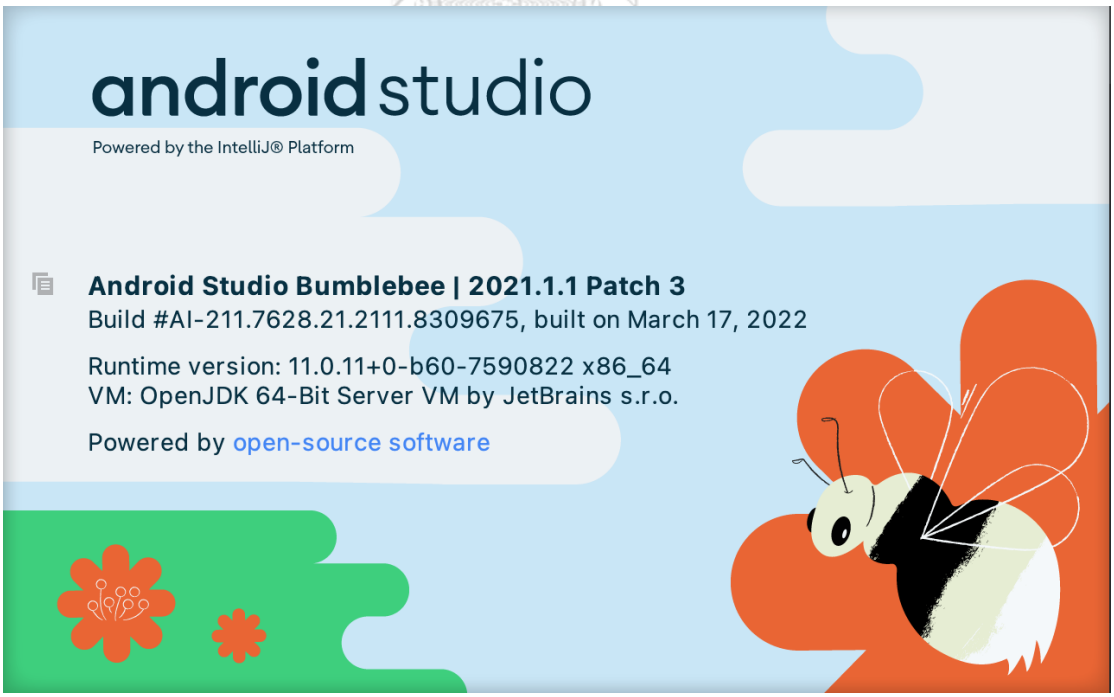
บทที่ 4
การทดลองและผลการทดลอง

ในงานวิจัยนี้นำเสนอแนวทางการปรับสีของภาพด้วยวิธีการเปลี่ยนช่วงสีและ Contrast Limited Adaptive Equalization การปรับความคมชัดของภาพด้วยวิธีการ Super-Resolution Convolutional Neural Network และมีการพัฒนาแอปพลิเคชันโดยการใช้ Flutter ขั้นตอนการทดลองอธิบายในหัวข้อต่อไปนี้

4.1 สภาพแวดล้อมที่ใช้ในการทดลอง

ใช้ Google Colab ซึ่งGoogle colab เป็นโฮสต์โปรแกรม Jupyter notebook บน Cloud ของ Google ชื่อเต็ม คือ Google Colaboratory โดยใช้ภาษา python3 เป็นภาษาหลักที่ใช้ในการเขียนและรันบน colab การสร้างโมเดลจะต้องใช้คอมพิวเตอร์ที่มีความเร็วหรือประสิทธิ์ภาพสูง เพื่อลดระยะเวลาในการ ประมวลผลของเครื่อง

ใช้โปรแกรม Android Studio ดังภาพที่ 19 ในการพัฒนาแอปพลิเคชัน Flutter



ภาพที่ 19 โปรแกรม Android Studio

4.2 ขั้นตอนการถ่ายโอนสี

- นำเข้าไลบารี (import library)

ขั้นตอนนี้จะทำการนำเข้าไลบารีที่จำเป็นในการทำการถ่ายโอนสี ดังภาพที่ 20

```
import matplotlib.pyplot as plt
import numpy as np
import cv2
```

ภาพที่ 20 ไลบารีที่จำเป็นในการทำการถ่ายโอนสี

- การหาค่าเฉลี่ยของภาพเป้าหมาย

การหาค่าเฉลี่ยของภาพเป้าหมายดังภาพที่ 21 เพื่อที่จะได้นำสี จากชุดข้อมูลที่ถ่ายด้วย iPhone 12 Promax มาถ่ายโอนให้กับภาพที่ถ่ายด้วยอุปกรณ์อื่น เพื่อที่จะให้ภาพมีสีเหมือนภาพที่ถูกถ่ายด้วย iPhone 12 Promax

```
def Average(lst):
    return sum(lst) / len(lst)
sourceL=[]
sourceA=[]
sourceB=[]
for i in range(1,100):
  path = "/content/drive/MyDrive/SushiDish/iPhone/iphone_"+str(i)+".jpg"
  img = cv2.imread(path)
  sourceLab = cv2.cvtColor(img, cv2.COLOR_BGR2Lab)
  sourceLab = np.float32(sourceLab)
  l, a, b = cv2.split(sourceLab)
  sourceL.append(l)
  sourceA.append(a)
  sourceB.append(b)
img = cv2.imread(path)
sourceLab = cv2.cvtColor(img, cv2.COLOR_BGR2Lab)
l, a, b = cv2.split(sourceLab)
sourceL.append(l)
sourceA.append(a)
sourceB.append(b)
meanSourceL = np.average(sourceL)
meanSourceA = np.average(sourceA)
meanSourceB = np.average(sourceB)
stdSourceL = np.std(sourceL)
stdSourceA = np.std(sourceA)
stdSourceB = np.std(sourceB)
```

ภาพที่ 21 การหาค่าเฉลี่ยของภาพเป้าหมาย

- การเปลี่ยนช่วงสี

ในการทำการถ่ายโอนสีขั้นแรกจะต้องทำการเปลี่ยนช่วงสีจากช่วงสี BGR เป็น ช่วงสี LAB โดยใช้ฟังก์ชัน cvtColor ของโอเพนซีวี และทำการแบ่งช่วงสีโดยใช้ฟังก์ชัน split ดังภาพที่ 22

```
destinationLab = cv2.cvtColor(destination, cv2.COLOR_BGR2LAB)
resultLab = cv2.cvtColor(result, cv2.COLOR_BGR2Lab)
destinationLab = np.float32(destinationLab)
resultLab = np.float32(resultLab)
destinationL, destinationA, destinationB = cv2.split(destinationLab)
resultL, resultA, resultB = cv2.split(resultLab)
```

ภาพที่ 22 การเปลี่ยนช่วงสี

- การนำเอาสีพื้นออก (Remove base color)

การนำเอาสีพื้นออกทำได้โดยการนำเอาค่าเฉลี่ยของแต่ละช่วงสีของภาพอินพุตมาลบออกจากค่าแต่ละช่วงสีของภาพอินพุตดังภาพที่ 23

```
resultL = destinationL - destinationL.mean()
resultA = destinationA - destinationA.mean()
resultB = destinationB - destinationB.mean()
```

ภาพที่ 23 การนำเอาสีพื้นออก

- การทำภาพอินพุตให้เป็นบรรทัดฐาน (Normalize input image)

การทำภาพอินพุตให้เป็นบรรทัดฐานทำได้โดยการนำค่าเบี่ยงเบนมาตราฐานของรูปเป้าหมายที่ได้มาจากการคำนวณจากรูปในชุดข้อมูลมาหารกับค่าเบี่ยงเบนมาตราฐานของภาพอินพุต แล้วนำผลลัพธ์ที่ได้จากการหารมาคูณกับผลลัพธ์ที่ได้จากขั้นตอนก่อนหน้าดังภาพที่ 24

```
resultL = resultL * (stdSourceL/ destinationL.std())
resultA = resultA * (stdSourceA/ destinationA.std())
resultB = resultB * (stdSourceB/ destinationB.std())
```

ภาพที่ 24 การทำภาพอินพุตให้เป็นบรรทัดฐาน

- การเพิ่มสี (Add source color)

ในขั้นตอนนี้จะเป็นการเพิ่มสีของภาพเป้าหมายให้กับภาพอินพุตโดยการนำค่าเฉลี่ยที่คำนวณได้จากการหาค่าเฉลี่ยของภาพในชุดข้อมูล มาบวกเข้ากับผลลัพธ์ที่ได้จากขั้นตอนก่อนหน้าดังภาพที่ 25

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
resultL = resultL + meanSourceL
resultA = resultA + meanSourceA
resultB = resultB + meanSourceB
```

ภาพที่ 25 การเพิ่มสี (Add source color)

- การรวมช่องสี (merge color space)

เป็นขั้นตอนการรวมช่องสีที่ได้ทำการแบ่งเอาไว้ในตอนแรก สำหรับการคำนวณเพื่อถ่ายโอนสีจากภาพเป้าหมายไปยังภาพอินพุตดังภาพที่ 26

```
resultL = np.clip(resultL, 0, 255)
resultA = np.clip(resultA, 0, 255)
resultB = np.clip(resultB, 0, 255)
resultLab = cv2.merge([resultL, resultA, resultB])
```

ภาพที่ 26 การรวมช่องสี (merge color space)

4.3 Contrast Limited Adaptive Histogram Equalization (CLAHE)

การทำ Contrast Limited Adaptive Histogram Equalization จะต้องทำการแบ่งสีแต่ละช่องก่อน แล้วค่าที่ใช้ในการกระจายค่าเฉลี่ยจะใช้ค่า L (Luminance) เพื่อให้ไม่กระทบกับสีที่ผ่านกระบวนการเปลี่ยนแปลงช่วงสี ในการทำ Contrast Limited Adaptive Histogram Equalization จะใช้ฟังก์ชัน createCLAHE ดังภาพที่ 27

```
clahe = cv2.createCLAHE(clipLimit = 1.0, tileGridSize = (3,3))
img = cv2.imread('/content/img.jpg')

lab_img = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
lab_img = cv2.resize(lab_img,dim)
l, a, b = lab_img[:,:,0], lab_img[:,:,1], lab_img[:,:,2]
l = clahe.apply(l)
lab_img = np.dstack((l,a,b))
rgb = cv2.cvtColor(lab_img, cv2.COLOR_LAB2RGB)
```

ภาพที่ 27 การทำ Contrast Limited Adaptive Histogram Equalization

4.4 SR-CNN

- นำเข้าไลบารี (import library)

ขั้นตอนนี้จะทำการนำเข้าไลบารีที่จำเป็นในการทำ SR-CNN ดังภาพที่ 28

```
import sys, os
import math
import tensorflow as tf
import numpy as np
import pandas as pd
import cv2
import matplotlib as mpl
import matplotlib.pyplot as plt
import skimage
from PIL import Image
```

ภาพที่ 28 ไลบารีที่จำเป็นในการทำ SR-CNN

- เชื่อม Google Colab กับ Google drive

ขั้นตอนนี้จะทำการเชื่อม Google Colab กับ Google drive ดังภาพที่ 29 เพื่อใช้เก็บข้อมูลต่างๆ

```
#mount drive
%cd ..
from google.colab import drive
drive.mount('/content/gdrive')

# this creates a symbolic link so that now the path /content/gdrive/My\ Drive/ is equal to /mydrive
!ln -s /content/gdrive/My\ Drive/ /mydrive

# list the contents of /mydrive
!ls /mydrive
```

ภาพที่ 29 ขั้นตอนการเชื่อม Google Colab กับ Google drive

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- แบ่งชุดข้อมูล (Split data)

ในขั้นตอนนี้จะทำการแบ่งชุดข้อมูลออกเป็นชุดข้อมูลในการสอนร้อยละ 80 และชุดข้อมูลในการทดสอบร้อยละ 20 ดังภาพที่ 30

```
import splitfolders
input_folder = "./content/gdrive/MyDrive/obj"
output = "/content/gdrive/My Drive/dataset/output"

splitfolders.ratio(input_folder, output=output, seed=42, ratio=(.8, 0, .2))
```

ภาพที่ 30 ขั้นตอนการแบ่งชุดข้อมูล

- การเตรียมชุดข้อมูลในการสอนแบบจำลอง

ขั้นตอนนี้เป็นการเตรียมชุดข้อมูลสอนแบบจำลอง โดยใช้ภาพชุดข้อมูลทั้งหมด 320 ภาพ เริ่มจากการนำภาพในชุดข้อมูลมาปรับให้มีคุณภาพสูง (High resolution) และคุณภาพต่ำ (Low resolution) และทำการปรับขนาดดังภาพที่ 31

```
import h5py

names = sorted(os.listdir('./content/gdrive/My Drive/dataset/output/train'))

data = []
label = []

for name in names:
    fpath = './content/gdrive/My Drive/dataset/output/train/' + name
    hr_img = cv2.imread(fpath, cv2.IMREAD_COLOR)
    hr_img = cv2.cvtColor(hr_img, cv2.COLOR_BGR2YCrCb)
    hr_img = hr_img[:, :, 0]
    shape = hr_img.shape

    # resize operation to produce training data and labels
    lr_img = cv2.resize(hr_img, (int(shape[1] / 2), int(shape[0] / 2)))
    lr_img = cv2.resize(lr_img, (shape[1], shape[0]))
    width_range = int((shape[0] - 16 * 2) / 16)
    height_range = int((shape[1] - 16 * 2) / 16)


    for k in range(width_range):
        for j in range(height_range):
            x = k * 16
            y = j * 16

            hr_patch = hr_img[x: x + 32, y: y + 32]
            lr_patch = lr_img[x: x + 32, y: y + 32]

            hr_patch = hr_patch.astype(np.float32) / 255.
            lr_patch = lr_patch.astype(np.float32) / 255.

            hr = np.zeros((1, 20, 20), dtype=np.double)
            lr = np.zeros((1, 32, 32), dtype=np.double)

            hr[0, :, :] = hr_patch[6:-6, 6: -6]
            lr[0, :, :] = lr_patch

            label.append(hr)
            data.append(lr)


data = np.array(data, dtype=np.float32)
label = np.array(label, dtype=np.float32)
```

ภาพที่ 31 ขั้นตอนการเตรียมชุดข้อมูลสอนแบบจำลอง

- การเตรียมชุดข้อมูลในการทดสอบแบบจำลอง

ขั้นตอนนี้เป็นการเตรียมชุดข้อมูลทดสอบแบบจำลอง โดยใช้ภาพชุดข้อมูลทั้งหมด 320 ภาพ เริ่มจากการนำภาพในชุดข้อมูลมาปรับให้มีคุณภาพสูง (High resolution) และคุณภาพต่ำ (Low resolution) และทำการปรับขนาดดังภาพที่ 32

```
names = sorted(os.listdir('./content/gdrive/My Drive/dataset/output/test'))
nums = len(names)

data_test = np.zeros((nums * 30, 1, 32, 32), dtype=np.double)
label_test = np.zeros((nums * 30, 1, 20, 20), dtype=np.double)

for i, name in enumerate(names):
    fpath = './content/gdrive/My Drive/dataset/output/test/' + name
    hr_img = cv2.imread(fpath, cv2.IMREAD_COLOR)
    hr_img = cv2.cvtColor(hr_img, cv2.COLOR_BGR2YCrCb)
    hr_img = hr_img[:, :, 0]
    shape = hr_img.shape

    # resize operation to produce training data and labels
    lr_img = cv2.resize(hr_img, (int(shape[1] / 2), int(shape[0] / 2)))
    lr_img = cv2.resize(lr_img, (shape[1], shape[0]))

    # Produce random crop
    x = np.random.randint(0, min(shape[0], shape[1]) - 32, 30)
    y = np.random.randint(0, min(shape[0], shape[1]) - 32, 30)

    for j in range(30):
        lr_patch = lr_img[x[j]:x[j] + 32, y[j]:y[j] + 32]
        hr_patch = hr_img[x[j]:x[j] + 32, y[j]:y[j] + 32]

        lr_patch = lr_patch.astype(np.float32) / 255.
        hr_patch = hr_patch.astype(np.float32) / 255.

        data_test[i * 30 + j, 0, :, :] = lr_patch
        label_test[i * 30 + j, 0, :, :] = hr_patch[6: -6, 6: -6]
```

ภาพที่ 32 ขั้นตอนการเตรียมชุดข้อมูลทดสอบแบบจำลอง

- การสร้างแบบจำลอง

ในการสร้างแบบจำลอง SR-CNN จะใช้ไลบารี่ tensorflow ในการสร้าง โดยกำหนดค่า optimizer เป็น Adam และค่า loss เป็น MSE ดังภาพที่ 33

```
def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=128, kernel_size=(9, 9),
                                     padding='VALID',
                                     use_bias=True,
                                     input_shape=(None, None, 1),
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(3, 3),
                                     padding='SAME',
                                     use_bias=True,
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=1, kernel_size=(5, 5),
                                     padding='VALID',
                                     use_bias=True,
                                     kernel_initializer='glorot_uniform',
                                     activation='linear'))
    # Optimizer
    optimizer = tf.keras.optimizers.Adam(learning_rate=0.0003)

    # Compile model
    SRCNN.compile(optimizer=optimizer, loss='mean_squared_error', metrics=['mean_squared_error'])

    return SRCNN
```

ภาพที่ 33 ขั้นตอนการสร้างโมเดล

- การฝึกสอนแบบจำลอง (Trained model)

ในการสอนโมเดลจะกำหนดจำนวน epochs เป็น 300 และ batch size เป็น 64 ดังภาพที่ 34

จุฬาลงกรณ์มหาวิทยาลัย

```
srcnn_model.fit(X_train, y_train, batch_size=64, validation_data=(X_test, y_test),
                callbacks=[checkpoint], shuffle=True, epochs=300, verbose=True)
```

ภาพที่ 34 ขั้นตอนการสอนโมเดล

- การทดสอบแบบจำลอง

ในการทดสอบแบบจำลอง เริ่มจากการที่นำภาพในชุดทดสอบมาปรับคุณภาพให้ลดลงด้วยฟังก์ชัน bicubic ดังภาพที่ 35 แล้วนำเข้าแบบจำลอง ดังภาพที่ 36

```
# Make output directory
try:
    os.listdir('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output')
except:
    os.mkdir('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output')

target = cv2.imread('./content/gdrive/MyDrive/dataset/output/test/cluster2_samsung_1.jpg', cv2.IMREAD_COLOR)
target = cv2.cvtColor(target, cv2.COLOR_BGR2YCrCb)
shape = target.shape

# Resize down by scale of 2
Y_img = cv2.resize(target[:, :, 0], (int(shape[1] / 2), int(shape[0] / 2)), cv2.INTER_CUBIC)

# Resize up to orignal image
Y_img = cv2.resize(Y_img, (shape[1], shape[0]), cv2.INTER_CUBIC)
target[:, :, 0] = Y_img
target = cv2.cvtColor(target, cv2.COLOR_YCrCb2BGR)
cv2.imwrite('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output/input5.jpg', target)

fig, ax = plt.subplots(figsize=(15, 10))
ax.imshow(Image.open('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output/input5.jpg'))
ax.title.set_text("Distorted Image")
plt.show()
```

ภาพที่ 35 ขั้นตอนการลดคุณภาพของภาพ

```
Y = np.zeros((1, target.shape[0], target.shape[1], 1), dtype=np.float32)
# Normalize
Y[0, :, :, 0] = Y_img.astype(np.float32) / 255.

# Predict
pre = srcnn_model.predict(Y, batch_size=1) * 255.

# Post process output
pre[pre[:] > 255] = 255
pre[pre[:] < 0] = 0
pre = pre.astype(np.uint8)

# Copy y channel back to image and convert to BGR
output = cv2.cvtColor(target, cv2.COLOR_BGR2YCrCb)
output[6: -6, 6: -6, 0] = pre[0, :, :, 0]
output = cv2.cvtColor(output, cv2.COLOR_YCrCb2BGR)

# Save image
cv2.imwrite('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output/output5.jpg', output)

fig, ax = plt.subplots(figsize=(15, 10))
ax.imshow(Image.open('./content/gdrive/MyDrive/dataset/SRCNN_dataset/output/output5.jpg'))
ax.title.set_text("Predicted Image")
plt.show()
```

ภาพที่ 36 ขั้นตอนการนำภาพเข้าแบบจำลอง

- การประเมินผล

ในการประเมินผลจะใช้ค่า PSNR ดังภาพที่ 37 ในการเปรียบเทียบคุณภาพของภาพก่อนเข้าแบบจำลองและหลังเข้าแบบจำลอง และใช้ค่า MSE ดังภาพที่ 38 ในการเปรียบค่าความผิดพลาด โดยทำการเปรียบเทียบดังภาพที่ 39

```python
def psnr(target, ref):
    # Assume target is RGB/BGR image
    target_data = target.astype(np.float32)
    ref_data = ref.astype(np.float32)

    diff = ref_data - target_data
    diff = diff.flatten('C')

    rmse = np.sqrt(np.mean(diff ** 2.))

    return 20 * np.log10(255. / rmse)
```

ภาพที่ 37 ฟังก์ชันการคำนวณค่า PSNR

```python
def mse(target, ref):
    target_data = target.astype(np.float32)
    ref_data = ref.astype(np.float32)
    err = np.sum((target_data - ref_data) ** 2)

    err /= np.float(target_data.shape[0] * target_data.shape[1])
    return err
```

ภาพที่ 38 ฟังก์ชันการคำนวณค่า MSE

```python
def compare_images(target, ref):
    scores = []
    scores.append(psnr(target, ref))
    scores.append(mse(target, ref))
    scores.append(ssim(target, ref, multichannel=True))
    return scores
```

CHULALONGKORN UNIVERSITY

ภาพที่ 39 ฟังก์ชันการเปรียบเทียบภาพ

4.5 Flutter

การพัฒนาแอปพลิเคชันด้วย Flutter เป็นการพัฒนาที่เขียนโค้ด 1 ครั้งสามารถทำงานได้ทั้งบนระบบปฏิบัติการ iOS และ Android ในการพัฒนาจะใช้โปรแกรม Android Studio BubbleBee และใช้ภาษา Dart ในการพัฒนา

- ไลบรารี

ในการพัฒนาแอปพลิเคชัน จำเป็นต้องใช้ไลบรารีต่างๆมาช่วยในการพัฒนา ในงานวิจัยนี้มีการใช้ไลบารีในการพัฒนาดังภาพที่. 40 ไลบารีที่ใช้ได้แก่

  - camera ใช้ในการถ่ายภาพจานซูชิ
  - path_provider และ path ใช้ในการตั้งค่าเส้นทาง (path) ในการจัดเก็บไฟล์ต่างๆ
  - flutter_color_model ใช้ในการเปลี่ยนช่วงสีของภาพที่ถ่ายเข้ามาจากช่วงสี RGB เป็นช่วงสี LAB
  - image ใช้ในการจัดการภาพต่างๆ
  - flutter_svg ใช้การจัดแต่งส่วนต่อประสานกับผู้ใช้ (User Interface)
  - tflite_fultter ใช้ในการตั้งค่าให้แบบจำลองSR-CNN สามารถทำงานบน flutter ได้
  - tflite_fultter_helper ใช้ในการช่วยให้ใช้งานฟังก์ชันต่างๆของ tensorflow lite ได้มากขึ้น
  - opencv_4 ใช้ในการช่วยให้ flutter สามารถใช้งานฟังก์ชันของโอเพนซีวี

```yaml
dependencies:
  flutter:
    sdk: flutter
  camera:
  path_provider:
  path:
  flutter_color_models: ^1.3.0
  image: ^2.1.19
  flutter_svg: ^0.19.3
  scidart: ^0.0.2-dev.9
  tflite_flutter: ^0.5.0
  tflite_flutter_helper: ^0.1.2
  opencv_4: ^1.0.0
```

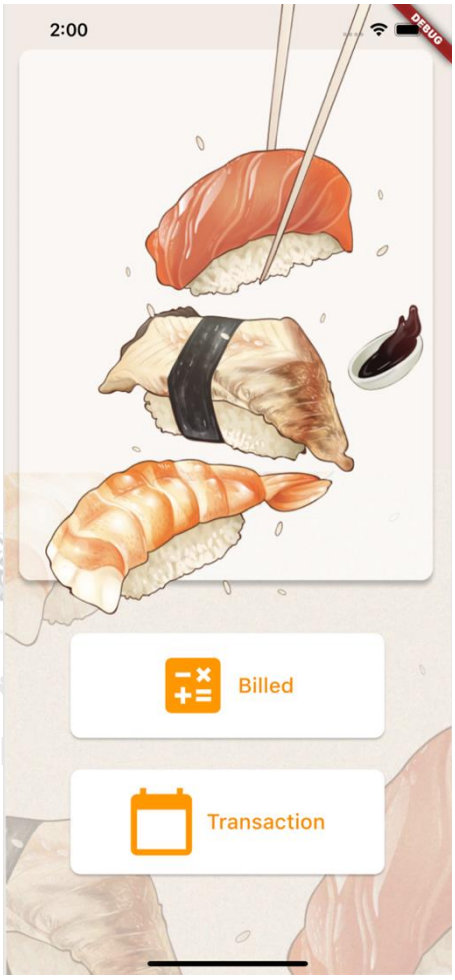
ภาพที่ 40 ไลบารีที่ใช้ใน Flutter

- ขั้นตอนการทำงานของแอปพลิเคชัน

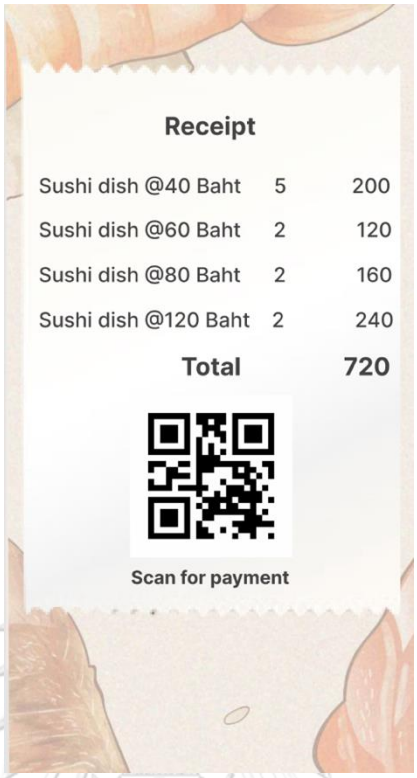
  การทำงานของแอปพลิเคชันเริ่มจากหน้าเมนูประกอบด้วย 2 เมนูได้แก่ การวางบิล (Billing) และการสรุปรายรับ (Transaction) ดังภาพที่ 41

  - การเก็บเงิน

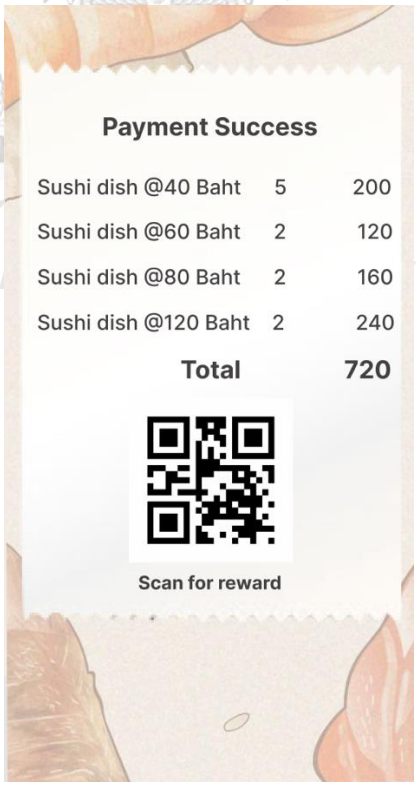
ขั้นตอนการเก็บเงินเริ่มจากการถ่ายภาพจานซูชิที่เรียงซ้อนกัน จากนั้นระบบจะทำ image preprocessing ดังได้กล่าวมาในข้างต้น แล้วส่งภาพที่ได้ แปลงเป็นไฟล์ base64 แล้วส่งเข้าแบบจำลองเพื่อคำนวณค่าบริการ จากนั้นระบบจะทำการสร้างรายการแจกแจงค่าบริการ และ QR Code เพื่อให้ลูกค้าสามารถสแกนและชำระเงิน ดังภาพที่ 42 เมื่อลูกค้าชำระเงินเสร็จแล้วระบบจะสร้างใบเสร็จและQR Code เพื่อสะสมแต้ม ดังภาพที่ 43



ภาพที่ 41 หน้าจอแสดงผลหน้าเมนู



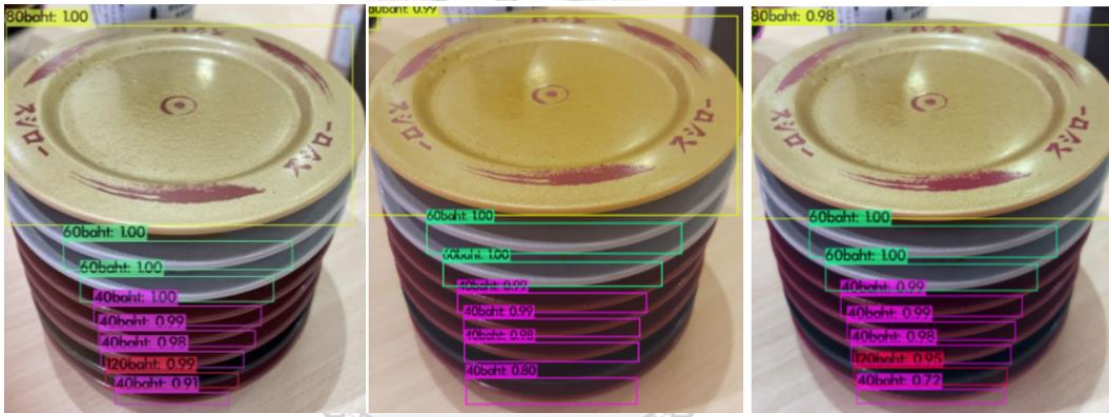
ภาพที่ 42 หน้าจอแสดงผลรายการค่าบริการและ QR Code ชำระค่าบริการ



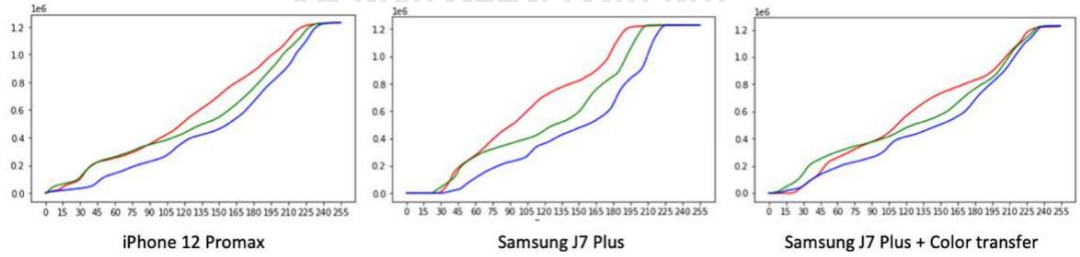
ภาพที่ 43 หน้าจอแสดงผลการชำระเงินสำเร็จและ QR Code สะสมแต้ม

4.6 ผลการทดลอง

จากการที่ทำ image preprocessing ในการปรับสีโดยใช้เทคนิค Color Transfer และ CLAHE ผลลัพธ์ที่ได้ดังภาพที่ 44 และภาคผนวก ข พบว่าประสิทธิภาพในการตรวจจับจานซูชิของแบบจำลองดีขึ้น ภาพที่ 45 แสดงกราฟที่สอดคล้องกับภาพที่ 44 โดยแกน X แสดงค่าสี RGB ตั้งแต่ 0-255 และแกน Y แสดงจำนวนพิกเซลสำหรับแต่ละสี จากการสังเกตในช่วงที่ 105-150 เส้นสีแดงและสีเขียวของกราฟแรกอยู่ใกล้กัน และแยกออกจากกัน ในกราฟที่สอง ส่วนในกราฟที่สาม เส้นสีแดงและสีเขียวอยู่ใกล้กันและดูคล้ายกับกราฟแรกมากกว่าเมื่อเทียบกับกราฟที่สอง กราฟที่สามมีความคล้ายกับกราฟแรกมากกว่ากราฟที่สองเนื่องจากการมีใช้เทคนิคการถ่ายโอนสีเพื่อปรับสีให้ใกล้เคียงกับสีภาพที่ถ่ายโดย iPhone 12 Pro Max



ภาพที่ 44 ผลลัพธ์การถ่ายโอนสีของภาพ



ภาพที่ 45 กราฟแสดงค่าสีของภาพก่อนและหลังการปรับคุณภาพของภาพ

จากผลการทดลองการปรับคุณภาพของภาพโดยการปรับความคมชัดด้วยแบบจำลองเอสอาร์-ซีเอ็นเอ็น และปรับสีโดยเทคนิคการถ่ายโอนสี และ CLAHE สามารถสรุปผลได้ดังตารางที่ 2 โดยตารางจะแสดงค่าเฉลี่ยของค่าความมั่นใจของแบบจำลองก่อนและหลังการปรับคุณภาพของภาพ โดยคำนวณจากภาพทดสอบจำนวน 240 ภาพ จากทั้งหมด 3 อุปกรณ์ อุปกรณ์ละ 80 ภาพ โดยอุปกรณ์ที่ 1 เป็นโทรศัพท์มือถือ android อุปกรณ์ที่ 2 เป็นแท็บเล็ต iOS และอุปกรณ์ที่ 3 เป็นโทรศัพท์ iOS ต่างรุ่นจากที่ใช้ถ่ายภาพชุดข้อมูลสำหรับการสร้างโมเดลตรวจหาวัตถุในวิทยานิพนธ์ [14] ผลการทดลองโดยภาพรวมค่าความมั่นใจเฉลี่ยของแบบจำลองหลังการปรับคุณภาพของภาพมีค่าสูงกว่าก่อนการปรับคุณภาพของภาพ โดยผลการทดลองอุปกรณ์ที่ 1 จานสีดำมีค่าความมั่นใจเฉลี่ยเพิ่มขึ้นสูงสุด จาก 0.56 เป็น 0.92 เนื่องจากก่อนปรับคุณภาพของภาพมีการตรวจไม่พบวัตถุจานสีดำเป็นจำนวนมาก แต่หลังจากปรับคุณภาพของภาพแล้วสามารถตรวจพบวัตถุจานสีดำ อุปกรณ์ที่ 2 จานสีแดงมีค่าความมั่นใจเฉลี่ยเพิ่มขึ้นสูงสุดจาก 0.87 เป็น 0.93 และอุปกรณ์ที่ 3 จานสีทองมีค่าความมั่นใจเฉลี่ยเพิ่มขึ้นสูงสุดจาก 0.84 เป็น 0.95

ตารางที่ 2 เปรียบเทียบค่าความมั่นใจเฉลี่ยของแบบจำลองก่อนและหลังการปรับคุณภาพของภาพ

สี/อุปกรณ์	อุปกรณ์ที่ 1		อุปกรณ์ที่ 2		อุปกรณ์ที่ 3	
ผลลัพธ์	ก่อน	หลัง	ก่อน	หลัง	ก่อน	หลัง
จานสีแดง	0.73	0.92	0.87	0.93	0.81	0.90
จานสีเทา	0.62	0.82	0.91	0.93	0.91	0.92
จานสีทอง	0.94	0.95	0.90	0.92	0.84	0.95
จานสีดำ	0.56	0.92	0.84	0.88	0.73	0.93

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 5
สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้นำเสนอเทคนิคการเตรียมรูปภาพก่อนนำเข้าแบบจำลอง โดยใช้เทคนิคการถ่ายโอนสี (Color transfer) และ Contrast Limited Adaptive Equalization และได้พัฒนาแอปพลิเคชันวางบิลหลายแพลตฟอร์มสำหรับซูชิสายพานที่พัฒนาโดยใช้ Flutter ทำให้สามารถทำงานได้หลายแพลตฟอร์มทั้ง iOS และ Android ทำให้ขั้นตอนการเตรียมภาพก่อนเข้าโมเดลมีส่วนสำคัญในการเพิ่มประสิทธิภาพของแบบจำลอง เนื่องจากชุดข้อมูลที่ใช้ในการสอนแบบจำลองถูกถ่ายด้วยกล้อง iPhone 12 Promax ทำให้เกิดปัญหาสีของภาพแตกต่างกันตามรุ่นของโทรศัพท์ที่ใช้ แต่ด้วยเทคนิคการเตรียมภาพ ที่สามารถปรับสีของรูปที่ถูกถ่ายด้วยอุปกรณ์ต่างๆสามารถปรับสีให้คล้ายกับสีของชุดข้อมูลมากที่สุด ทำให้การทำงานของแบบจำลองมีประสิทธิภาพที่สุด และยังมีการปรับคุณภาพของภาพโดยใช้แบบจำลอง SR-CNN เพื่อช่วยให้ภาพมีความละเอียดมากขึ้น โดยตัววัดที่ใช้ในการวัดแบบจำลอง SR-CNN ในวิทยานิพนธ์นี้ใช้ค่า PSNR เป็นการบอกคุณภาพของภาพยิ่งค่ามาก คุณภาพของภาพก็จะดี และใช้ค่า MSE ในการบอกค่าความผิดพลาดของภาพ ยิ่งค่าน้อย ภาพก็จะมีคุณภาพมาก

จากผลการทดลองจะแสดงให้เห็นว่าเทคนิคต่างๆที่ใช้ในวิทยานิพนธ์นี้สามารถทำให้แบบจำลองตรวจจับจานซูชิสามารถทำงานได้มีประสิทธิภาพมากยิ่งขึ้น และสามารถเป็นประโยชน์ต่อผู้ประกอบการร้านอาหารประเภทซูชิสายพาน ให้สามารถลดระยะเวลาในการชำระเงินและลดขยะอิเล็กทรอนิกส์ที่ใช้ในการนับจานซูชิในปัจจุบัน

สำหรับแนวทางในการวิจัยในอนาคต เนื่องจากแอปพลิเคชันที่พัฒนาด้วย Flutter ยังมีฟังก์ชันการทำงานไม่สมบูรณ์ และยังสามารถเพิ่มฟังก์ชันที่เป็นประโยชน์ต่อผู้ประกอบการและผู้บริโภคได้ การเพิ่มฟังก์ชันการชำระค่าบริการด้วยตนเอง โดยที่พนักงานไม่จำเป็นต้องเป็นคนคิดค่าบริการ จึงเป็นสิ่งที่น่าสนใจในการพัฒนาแอปพลิเคชันต่อไปในอนาคต

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก

ผลลัพธ์การปรับความคมชัดด้วย SR-CNN

ภาพที่ 46 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 1

ภาพที่ 47 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 2

ภาพที่ 48 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 3

ภาพที่ 49 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 4

ภาพที่ 50 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 5

ภาพที่ 51 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 1

ภาพที่ 52 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 2

ภาพที่ 53 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 3

ภาพที่ 54 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 4

ภาพที่ 55 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 5

ภาพที่ 56 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 1

ภาพที่ 57 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 2

ภาพที่ 58 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 3

ภาพที่ 59 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 4

ภาพที่ 60 เปรียบเทียบก่อนและหลังการปรับความคมชัดของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 5

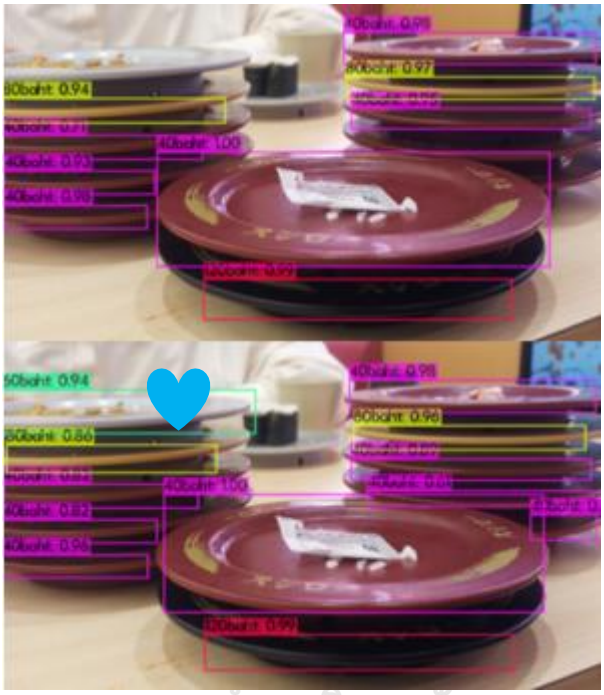
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข

ผลลัพธ์การปรับคุณภาพของภาพ

แสดงการตรวจพบวัตถุภายหลังการปรับคุณภาพของภาพ

แสดงค่าความมั่นใจเพิ่มขึ้นภายหลังการปรับคุณภาพของภาพ

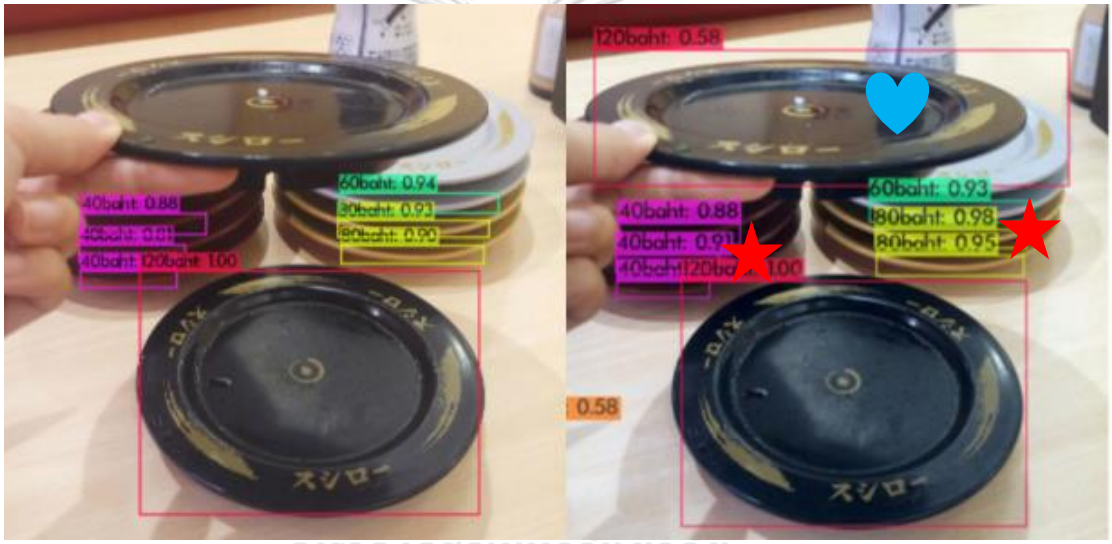


จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาพที่ 61 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 1

ภาพที่ 62 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 2

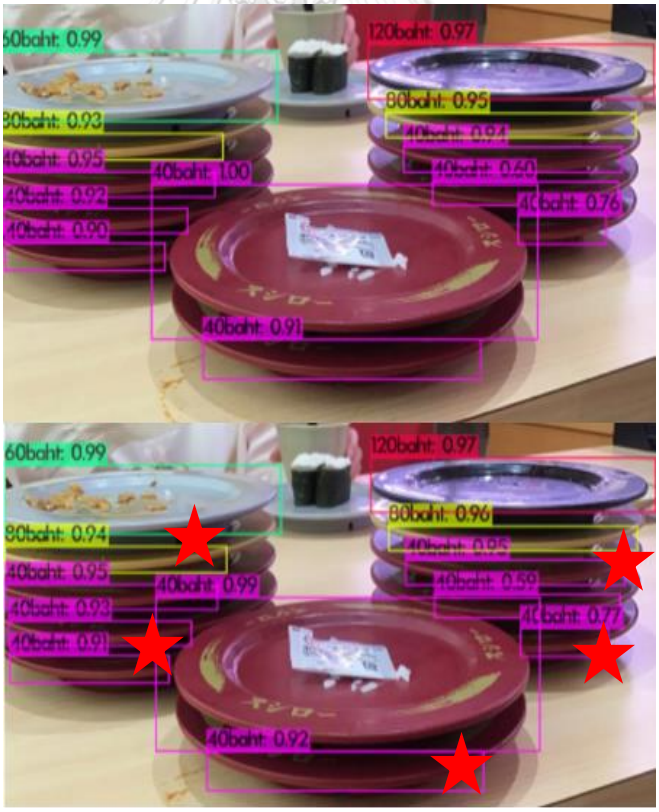
ภาพที่ 63 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 3



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

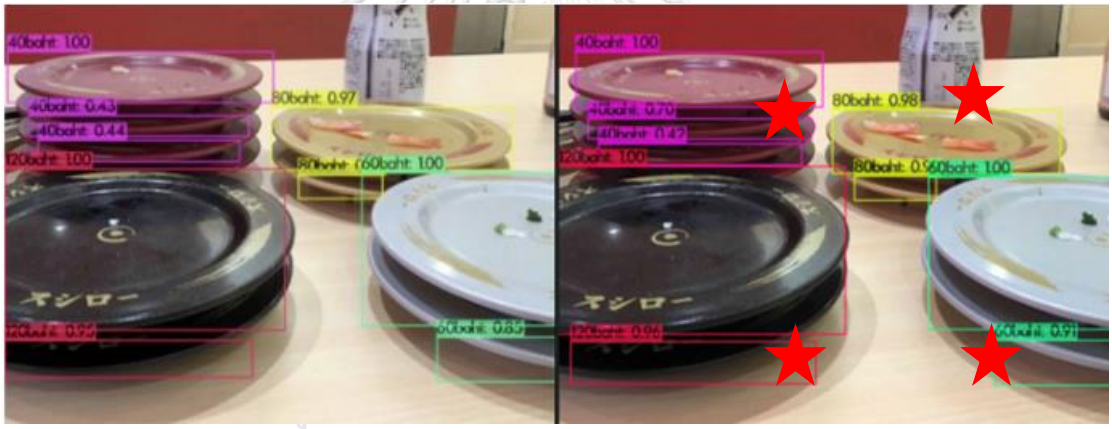
ภาพที่ 64 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 4

ภาพที่ 65 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 1 กรณีที่ 5



ภาพที่ 66 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 1

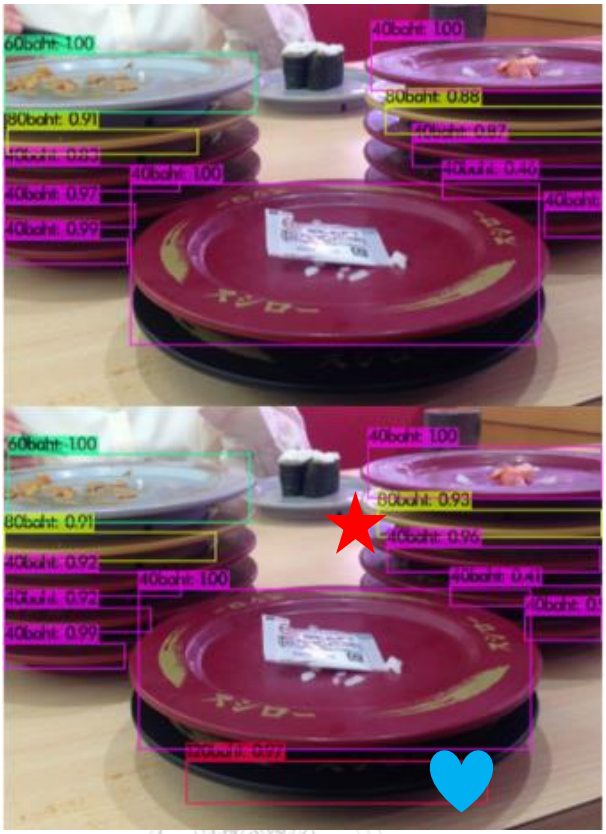
ภาพที่ 67 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 2



ภาพที่ 68 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 3

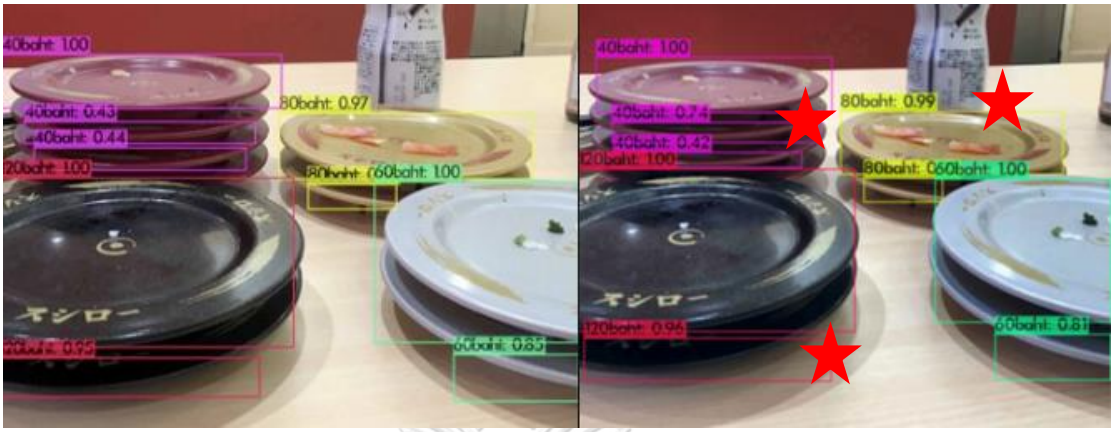
ภาพที่ 69 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 4

ภาพที่ 70 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 2 กรณีที่ 5

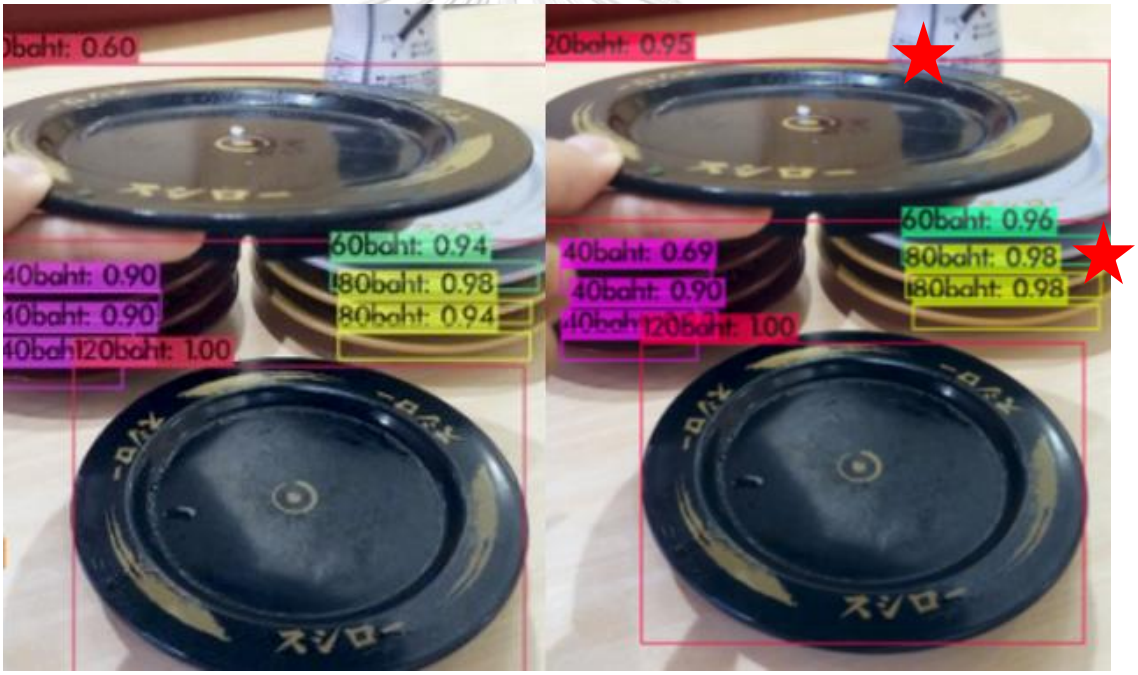


ภาพที่ 71 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 1

ภาพที่ 72 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 2



ภาพที่ 73 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 3



ภาพที่ 74 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 4

ภาพที่ 75 เปรียบเทียบก่อนและหลังการปรับคุณภาพของภาพที่ถ่ายด้วย อุปกรณ์ที่ 3 กรณีที่ 5

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บรรณานุกรม

1. *Flutter - Build aaps for any screen.* 2021 [cited 2021 18]; Available from: https://flutter.devgclid=Cj0KCQiAqvaNBhDLARIsAH1Pq51yO3X5GHkoEDGUhmRr_s1BR4Aeozr3VaGO0bgvQRItSxwim5T6FmwaArfDEALw_wcB&gclsrc=aw.ds.
2. *Dart*. 2022 [cited 2022 9]; Available from: https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1/33-google/6709-dart-programming-language.html.
3. *RGB*. 2022 [cited 2022 10]; Available from: https://www.uprintershop.com/rgb-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/.
4. *LAB* 2022 [cited 2022 10]; Available from: https://www.dozzdiy.com/%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B9%89%E0%B8%B2%E0%B9%83%E0%B8%88%E0%B9%83%E0%B8%99%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%B5-lab-%E0%B9%80%E0%B8%9E%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B9%81%E0%B8%81%E0%B9%89%E0%B9%84%E0%B8%82%E0%B8%A0%E0%B8%B2%E0%B8%9E%E0%B8%96%E0%B9%88%E0%B8%B2%E0%B8%A2/.
5. *Color Transfer*. 2022 [cited 2022 10]; Available from: http://raspberrypi-thailand.blogspot.com/2018/07/color-transfer.html.
6. *CLAHE*. 2022 [cited 2022 10]; Available from: https://rtt.kku.ac.th/ejournal/pa_upload_pdf/624308.pdf.
7. *SRCNN (Super Resolution)*. 2022 [cited 2022 10]; Available from: https://medium.com/coinmonks/review-srcnn-super-resolution-3cb3a4f67a7c.
8. Setiawan, A., et al., *Color retinal image enhancement using CLAHE*. 2013. 1-3.
9. Bhairannawar, S., et al. *Color image enhancement using Laplacian filter and contrast limited adaptive histogram equalization*. in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. 2017.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

10. Wong, S., et al. *Comparative analysis of underwater image enhancement methods in different color spaces*. in *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. 2014.

11. Maitriboriruk, R. and Y. Limpiyakorn. *Object Detection for Classifying Sushi Dishes in Conveyor Belt Sushi Business*. in *8th International Conference on Computer Technology Applications (ICCTA)*. 2022.

12. *HDF5 dataset (.h5 file)*. 2022 [cited 2022 10]; Available from: https://pakorn-srs.medium.com/%E0%B8%AA%E0%B8%A3%E0%B9%89%E0%B8%B2%E0%B8%87-hdf5-dataset-h5-file-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B9%82%E0%B8%84%E0%B9%89%E0%B8%94-python-d888968df485.

13. Reinhard, E., et al., *Color Transfer between Images.* IEEE Computer Graphics and Applications, 2001. **21**: p. 34-41.

14. Maitriboriruk, R., *Object Detection for Classifying Sushi Dishes in Conveyor Belt Sushi Business*. 2022, Chulalongkorn university.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียน

ชื่อ-สกุล	พัชริยา ปิยะอารมณ์รัตน์
วัน เดือน ปี เกิด	17 กันยายน 2539
สถานที่เกิด	ไทย
วุฒิการศึกษา	B.Sc. (Computer Science) Thammasat University
	M.Sc. (Computer Engineer) Chulalongkorn University
ที่อยู่ปัจจุบัน	14 ซ.รามคำแหง 118 แยก 14 แขวงสะพานสูง เขตสะพานสูง กทม. 10240

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY