การรู้จำอาหารและการประเมินปริมาณอาหารเหลือด้วยเทคนิคการเรียนรู้เชิงลึก

พลเทพ มีไผ่ขอ

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

การรู้จำอาหารและการประเมินปริมาณอาหารเหลือด้วยเทคนิคการเรียนรู้เชิงลึก

พลเทพ มีไผ่ขอ

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

วิทยานิพนธ์ เรื่อง "การรู้จำอาหารและการประเมินปริมาณอาหารเหลือด้วยเทคนิคการเรียนรู้เชิงลึก"
ของ พลเทพ มีไผ่ขอ
ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบวิทยานิพนธ์
(รองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง)

..... ประธานที่ปรึกษาวิทยานิพนธ์
(รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.สัญญา เครือหงษ์)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.ดวงเดือน อัศวสุธีรกุล)

อนุมัติ

.....
(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)
คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง การรู้จำอาหารและการประเมินปริมาณอาหารเหลือด้วยเทคนิคการเรียนรู้เชิงลึก

ผู้วิจัย พลเทพ มีไผ่ขอ

ประธานที่ปรึกษา รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต

ประเภทสารนิพนธ์ วิทยานิพนธ์ วท.ม. วิทยาการคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2565

คำสำคัญ การเรียนรู้เชิงลึก, การรู้จำอาหาร, อาหารเหลือ, การวิเคราะห์ข้อมูล

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึก ซึ่งผู้วิจัยสร้างแบบจำลองด้วยเทคนิค Mask R-CNN ที่ได้พัฒนาเพิ่มเติมวิธีการคำนวณพื้นที่ภายในภาพ เพื่อให้สามารถประเมินปริมาณอาหารเหลือได้ ในการฝึกสอนแบบจำลองผู้วิจัยใช้ชุดข้อมูลภาพถ่ายภายในจานที่เกลี่ยอาหารเหลือรวมกันไว้เป็นสัดส่วน จำนวน 141 ภาพเป็นชุดข้อมูลภาพอาหารต้นฉบับ และแบ่งข้อมูลเป็นชุดฝึกสอนร้อยละ 80 และชุดทดสอบร้อยละ 20 กระบวนการเสริมข้อมูลถูกเพิ่มเข้าไปในขณะฝึกสอน เพื่อลดการเกิดการแก้ไขปัญหาเกินพอดีของแบบจำลอง และกำหนดให้แบบจำลองเรียนรู้จำนวน 500 รอบ จากนั้นทดสอบประสิทธิภาพแบบจำลองด้วยการวัดค่าความผิดพลาดสัมบูรณ์ และค่าความผิดพลาดสัมพันธ์ เพื่อนำไปคำนวณหาค่าความถูกต้องของการประเมินปริมาณอาหารเหลือ

ผลลัพธ์ของการพัฒนาการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึกพบว่าที่จำนวนการฝึกสอน 82 รอบ มีประสิทธิภาพในการประเมินปริมาณอาหารเหลือสูงสุดด้วยค่าความผิดพลาดสัมบูรณ์เฉลี่ยอยู่ที่ร้อยละ 4.65 ค่าความผิดพลาดสัมพันธ์เฉลี่ยอยู่ที่ร้อยละ 12.46 และค่าความถูกต้องเฉลี่ยอยู่ที่ร้อยละ 87.54 ตามลำดับ

Title FOOD RECOGNITION AND LEFTOVERS FOOD ESTIMATION USING DEEP LEARNING TECHNIQUES

Author Phonthep Meephaikhor

Advisor Associate Professor Dr. Chakkrit Snae Namahoot

Academic Paper M.S. Thesis in Computer Science, Naresuan University, 2022

Keywords Deep Learning, Food Recognition, Leftovers Food, Data Analysis

ABSTRACT

The goal of this study was to use a deep learning technique to estimate food leftovers after eating. The model was created using the Mask R-CNN technique, which has also developed a method for calculating the area within the image to estimate the amount of leftover food. To train the model, the researcher used a dataset of 141 original food images of food leftovers in a plate that were proportionally equalized and divided the data into 80% training and 20% test. To reduce overfitting, the data augmentation process was added during training and the model was assigned to learn 500 rounds, after which the model performance was tested by measuring the absolute error and the relative error value to calculate the accuracy of the leftover food estimation

The results of developing the food leftovers estimation after eating using deep learning techniques revealed that at 82 training rounds, the efficiency in estimating maximum amount of leftovers food was found to be effective, with a mean absolute error of 4.65%, a mean correlation error of 12.46 percent, and a mean accuracy of 87.54 percent, respectively

ประกาศคุณูปการ

ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงในความกรุณาของ รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต ประธานที่ปรึกษาวิทยานิพนธ์ ที่ได้อุตส่าห์สละเวลาอันมีค่ามาเป็นที่ปรึกษา พร้อมทั้งให้คำแนะนำตลอดระยะเวลาในการทำวิทยานิพนธ์ฉบับนี้ และขอกราบขอบพระคุณคณะกรรมการวิทยานิพนธ์อันประกอบไปด้วย ผู้ช่วยศาสตราจารย์ ดร.สัญญา เครือหงษ์ ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย ผู้ช่วยศาสตราจารย์ ดร.ดวงเดือน อัศวสุธีรกุล และรองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง กรรมการผู้ทรงคุณวุฒิ ที่ได้กรุณาให้คำแนะนำตลอดจนแก้ไขข้อบกพร่องของวิทยานิพนธ์ด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า

ขอกราบขอบพระคุณอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ รวมทั้งเจ้าหน้าที่ที่เกี่ยวข้องทุกท่าน ที่กรุณาให้คำแนะนำ ให้ความรู้ และให้คำปรึกษาในด้านเอกสารการวิจัย ขั้นตอนการวิจัย รวมถึงการเดินเอกสาร ซึ่งทำให้สำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ขอน้อมรำลึกถึงพระคุณของบิดา มารดา ตลอดครูบาอาจารย์ ที่ได้สั่งสอนและอบรมตั้งแต่อดีตจนถึงปัจจุบัน ขอบคุณพี่ ๆ เพื่อน ๆ น้อง ๆ บุคคลรอบข้าง และทุกท่านที่ไม่ได้เอ่ยนาม ที่ให้กำลังใจ ให้คำปรึกษา และความช่วยเหลือที่ดีตลอดมา

ผู้วิจัยขอระลึกถึงคุณงามความดีที่ทุกท่านได้ช่วยเหลือด้วยดีตลอดมา และขอให้คุณค่าและคุณประโยชน์อันพึงจะมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบ และอุทิศแด่ผู้มีพระคุณทุก ๆ ท่าน ผู้วิจัยหวังเป็นอย่างยิ่งว่า งานวิจัยนี้จะเป็นประโยชน์ต่อการช่วยลดประชากรโรคอ้วนได้บ้างไม่มากก็น้อย

พลเทพ มีไผ่ขอ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

ประกาศคุณูปการ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ญ

บทที่ 1 1

บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 จุดมุ่งหมายของการศึกษา 2

1.3 ขอบเขตของงานวิจัย 2

1.4 ข้อตกลงเบื้องต้น 2

1.5 นิยามศัพท์เฉพาะ 2

1.6 สมมติฐานของการวิจัย 3

บทที่ 2 4

เอกสารและงานวิจัยที่เกี่ยวข้อง 4

2.1 ทฤษฎีที่เกี่ยวข้อง 4

2.1.1 การเรียนรู้เชิงลึก (Deep Learning) 4

2.1.2 เครือข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network, CNN) 5

2.1.3 การแบ่งส่วนภาพ (Image Segmentation) 8
2.1.4 การทำ Augmentation 8
2.1.5 ResNet 9
2.1.6 Mask R-CNN 10
2.1.7 ระบบสี (Color model) 10
2.2 งานวิจัยที่เกี่ยวข้อง 11
บทที่ 3 15
วิธีดำเนินงานวิจัย 15
3.1 เครื่องมือที่ใช้ในการวิจัย 16
3.1.1 ซอฟต์แวร์ (Software) 16
3.1.2 ฮาร์ดแวร์ (Hardware) 16
3.2 ขั้นตอนการดำเนินการวิจัย 16
3.2.1 การเก็บรวบรวมข้อมูล 17
3.2.2 การเตรียมชุดข้อมูลและสร้างแบบจำลอง 17
3.2.3 การประเมินปริมาณอาหารเหลือ 20
3.3 การวิเคราะห์ข้อมูล 23
3.3.1 ค่าความผิดพลาดสัมบูรณ์ (Absolute Error : e) 24
3.3.2 ค่าความผิดพลาดสัมพันธ์ (Relative Error : %Error) 24
3.3.3 ค่าความแม่นยำ (Accuracy : %Acc) 24
บทที่ 4 25
ผลการวิจัย 25

4.1 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1 26
4.2 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2 27
4.3 การวัดประสิทธิภาพแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน 29
บทที่ 5 31
บทสรุป 31
5.1 สรุปผลการวิจัย 31
5.2 ข้อเสนอแนะ 32
บรรณานุกรม 34
ภาคผนวก 36
ประวัติผู้วิจัย 50

สารบัญตาราง

หน้า

ตาราง 1 ผลการวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1 27

ตาราง 2 ผลการวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2 28

ตาราง 3 ผลการวัดประสิทธิภาพแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน 29

ตาราง 4 การเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วน และแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน 30

สารบัญภาพ

หน้า

ภาพ 1 การเปรียบเทียบประสิทธิภาพอัลกอริทึมการเรียนรู้ของเครื่อง 4

ภาพ 2 การเปรียบเทียบส่วนประกอบเซลล์ประสาทสมองมนุษย์กับเซลล์ประสาทเทียม ... 5

ภาพ 3 สถาปัตยกรรมของเครือข่ายประสาทแบบคอนโวลูชัน 6

ภาพ 4 ความแตกต่างระหว่าง Machine learning กับ Deep learning 6

ภาพ 5 ตัวอย่างการเปรียบเทียบรูปภาพอาหารต้นฉบับกับรูปภาพอาหารที่ทำการแบ่งส่วนภาพ 8

ภาพ 6 ตัวอย่างภาพแสดงการทำ Augmentation 9

ภาพ 7 ตัวอย่างโครงสร้างของ ResNet 9

ภาพ 8 ลูกบาศกระบบสี RGB 11

ภาพ 9 Single-hexcone HSV color model 11

ภาพ 10 แสดงขนาดภาพ และสัดส่วนอาหารต่อขนาดภาพที่สามารถตรวจจับวัตถุดิบได้ดี 12

ภาพ 11 ภาพตัวอย่างการกำหนดขอบเขตตำแหน่งอาหารด้วยเทคนิค Mask R-CNN 12

ภาพ 12 สถาปัตยกรรมกระบวนการทำงานระบบ RES-DIP 13

ภาพ 13 การแบ่งกลุ่มของไขมันใต้ผิวหนังจากเนื้อซี่โครงหมู 14

ภาพ 14 ตัวอย่างปริมาณอาหารภายในจานที่ถูกแบ่งออกเป็น 9 ส่วน 15

ภาพ 15 ตัวอย่างชุดข้อมูลภาพสำหรับสร้างแบบจำลองแบบเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน 17

ภาพ 16 ตัวอย่างชุดข้อมูลภาพสำหรับสร้างแบบจำลองแบบไม่เกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน 17

ภาพ 17 ตัวอย่างการนำชุดข้อมูลรูปภาพเก็บไว้ในโฟลเดอร์.....18
ภาพ 18 การกำหนดขอบเขตตำแหน่งอาหารเหลือภายในจานด้วยโปรแกรม Via 1.0.6...18
ภาพ 19 ตัวอย่างการเก็บไฟล์ JSON จากการกำหนดขอบเขตตำแหน่งอาหารเหลือ.....18
ภาพ 20 ตัวอย่างภาพการทำงานการสร้างแบบจำลอง.....19
ภาพ 21 ตัวอย่างภาพไฟล์แบบจำลองนามสกุล h5.....20
ภาพ 22 ตัวอย่างภาพการแก้ไขการกำหนดขอบเขตตำแหน่งเป็นสีแดง.....20
ภาพ 23 การกำหนดขอบเขตตำแหน่งอาหารเหลือหลังจากแก้ไขไลบรารี่ Mask R-CNN...20
ภาพ 24 ตัวอย่างภาพอาหารก่อนการรับประทาน และหลังรับประทานเสร็จ.....21
ภาพ 25 ตัวอย่างภาพที่กำหนดขอบเขตตำแหน่งอาหารเหลือด้วยเทคนิค Mask R-CNN ..21
ภาพ 26 ตัวอย่างรูปภาพการปรับภาพเป็นโหมด HSV 21
ภาพ 27 ตัวอย่างรูปภาพขั้นตอนการลบวัตถุที่ไม่ใช่อาหารเหลือ 22
ภาพ 28 ตัวอย่างรูปภาพการแปลงภาพเป็นระบบสี RGB 22
ภาพ 29 ขั้นตอนการแปลงภาพให้อยู่ในรูปแบบของอาเรย์.....22
ภาพ 30 แสดงตัวอย่างภาพเม็ดพิกเซล.....23
ภาพ 31 กระบวนการทำงานโดยรวมของการประเมินปริมาณอาหารเหลือ 23
ภาพ 32 ตัวอย่างการทำงานการประเมินปริมาณอาหารเหลือจากรูปภาพ.....25
ภาพ 33 ตัวอย่างการจัดเก็บข้อมูลการประเมินปริมาณอาหารเหลือรูปแบบ Log File.....25
ภาพ 34 ตัวอย่างข้อมูลภาพทดสอบแบบจำลองแบบเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน
.....26
ภาพ 35 ตัวอย่างข้อมูลภาพทดสอบแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็น
สัดส่วน.....26
ภาพ 36 ตัวอย่างการพัฒนาการถ่ายรูปภาพอาหารภายในจาน 33

ฎ

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันประชากรในประเทศไทยเป็นโรคอ้วน (Obesity) เพิ่มมากขึ้น โดยโรคอ้วนนำมาซึ่งการเกิดโรคไม่ติดต่อเรื้อรัง (Non-communicable diseases-NCDs) โดยเฉพาะ 5 กลุ่มโรคหลัก คือ โรคหัวใจและหลอดเลือด โรคเบาหวาน โรคมะเร็ง โรคไต และโรคทางเดินหายใจเรื้อรัง (กรุงเทพธุรกิจ, 2565) อันเป็นผลมาจากการมีพฤติกรรมการบริโภคที่ไม่เหมาะสม เช่น การทานอาหารที่มีรสชาติหวานจัด เค็มจัด และอาหารที่มีไขมันสูง เป็นต้น ฉะนั้นวิธีการรักษาหรือการป้องกันการเกิดโรคที่ดีที่สุด คือ การปรับเปลี่ยนด้านโภชนาการรวมถึง ปรับเปลี่ยนพฤติกรรมการกินให้ถูกวิธี โดยพิจารณาจากการรับประทานอาหารให้ได้แคลอรี่ที่สมดุล และเหมาะสมต่อร่างกายที่ควรจะได้รับในแต่ละวัน การควบคุมแคลอรี่ก็เป็นอีกวิธีหนึ่งที่เป็นส่วนช่วยให้สามารถลดความอ้วนได้ หากสามารถลดปริมาณอาหารที่รับประทานลงจากพลังงานที่ควรจะได้รับต่อวัน ให้ได้วันละ 500 แคลอรี่ (คลีนิค BDMS Wellness, 2565) ทำให้ได้อย่างต่อเนื่อง และมีวินัยในการควบคุมการรับประทานอย่างเคร่งครัด ปริมาณไขมันในร่างกายจะลดน้อยลง ดัชนีมวลกาย (BMI) ก็จะค่อย ๆ กลับเข้าสู่เกณฑ์ปกติ ภาวะเสี่ยงต่อการเกิดโรคจะลดน้อยลง

สำหรับผู้มีภาวะโรคอ้วน หรือน้ำหนักเกินเกณฑ์มาตรฐาน การมีสิ่งอำนวยความสะดวกที่ให้ข้อมูลโภชนาการทางอาหารที่รับประทานในแต่ละมื้อ อย่างเช่น เว็บไซต์ประเมินคุณค่าทางโภชนาอาหาร และแอปพลิเคชันคำนวณแคลอรี่ ก็เป็นอีกสิ่งหนึ่งที่มีบทบาทในการช่วยอำนวยความสะดวกให้การควบคุมแคลอรี่เป็นไปอย่างถูกต้อง เหมาะสมกับร่างกายที่ควรจะได้รับ แต่เว็บไซต์หรือแอปพลิเคชันประเมินคุณค่าทางโภชนาอาหารในปัจจุบันส่วนใหญ่คำนวณแคลอรี่ หรือประเมินคุณค่าทางโภชนาการโดยใช้ปริมาณอาหารที่รับประทานหมดครบ 100 เปอร์เซ็นต์ ซึ่งในบางครั้งผู้บริโภคก็ไม่ได้รับประทานอาหารจนหมดครบ 100 เปอร์เซ็นต์ เป็นสาเหตุให้การประเมินคุณค่าทางโภชนามีความคลาดเคลื่อนไม่เป็นไปตามปริมาณที่ได้รับประทานจริง ทั้งที่ในปัจจุบันความก้าวหน้าทางเทคโนโลยีด้านการประมวลผลภาพ (Image Processing) มีประสิทธิภาพสูงที่สามารถนำมาใช้ในการแก้ไขปัญหาในเรื่องการประเมินคุณค่าทางโภชนาการที่มีความคลาดเคลื่อนได้

ดังนั้นงานวิจัยนี้จึงมุ่งเน้นการเพิ่มประสิทธิภาพการประเมินคุณค่าทางโภชนาการ ด้วยการพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิง

ลึก ซึ่งเป็นการประมวลผลภาพถ่ายอาหารก่อนการรับประทาน และหลังการรับประทาน เพื่อการประเมินคุณค่าทางโภชนาการที่ถูกต้องเป็นไปตามปริมาณการบริโภคที่แท้จริง

1.2 จุดมุ่งหมายของการศึกษา

เพื่อพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึก

1.3 ขอบเขตของงานวิจัย

ขอบเขตด้านข้อมูล

- รูปภาพถ่ายอาหารเหลือหลังจากการรับประทานภายในจาน แบ่งสัดส่วนร้อยละของปริมาณอาหารภายในจานออกเป็น 9 ส่วน คือ 0, 12.5, 25, 37.5, 50, 62.5, 75, 87.5 และ 100

ขอบเขตด้านระบบ

- พัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือภายในจาน โดยใช้เทคนิคการเรียนรู้เชิงลึก

1.4 ข้อตกลงเบื้องต้น

1. ในงานวิจัยนี้จะใช้จาน เป็นภาชนะที่ใช้ในสำหรับการประเมินปริมาณอาหารเหลือ
2. ความแม่นยำของการประเมินปริมาณอาหารเหลือ ขึ้นอยู่กับการจัดสัดส่วนปริมาณอาหารภายในจาน

1.5 นิยามศัพท์เฉพาะ

โรคอ้วน (Obesity) หมายถึง ภาวะที่ร่างกายมีการสะสมไขมันมากเกินกว่าปกติหรือมากเกินกว่าที่ร่างกายจะเผาผลาญ จึงสะสมพลังงานที่เหลือเอาไว้ในรูปของไขมันตามอวัยวะต่าง ๆ ซึ่งมีความเสี่ยงต่อการเกิดปัญหาสุขภาพ และเป็นสาเหตุของการเกิดโรคเรื้อรังต่าง ๆ ตามมา

กลุ่มโรคไม่ติดต่อ (NCDs) หมายถึง กลุ่มโรคที่ไม่ได้เกิดจากเชื้อโรค แต่เกิดจากนิสัยหรือพฤติกรรมการดำเนินชีวิต โดยจะมีการสะสมอาการอย่างช้า ๆ และต่อเนื่อง หากไม่ได้รับการรักษาที่ถูกต้อง จะยิ่งทำให้เกิดการเรื้อรังของโรคตามมา ซึ่งมีอัตราผู้ป่วยและการเสียชีวิตสูงสุด7 โรค ได้แก่ โรคเบาหวาน โรคหลอดเลือดสมองและหัวใจ โรคถุงลมโป่งพอง โรคมะเร็ง โรคความดันโลหิตสูง โรคไขมันในเลือดสูง และโรคอ้วนลงพุง

ดัชนีมวลกาย (BMI) หมายถึง ค่าสากลที่ใช้เพื่อคำนวณเพื่อหาน้ำหนักตัวที่ควรจะเป็น และประมาณระดับไขมันในร่างกายโดยใช้น้ำหนักตัว และส่วนสูง

การรู้จำอาหาร (Food Recognition) หมายถึง กระบวนการในการจำแนกประเภท (Classification) อาหารจากรูปภาพ ยกตัวอย่าง เช่น จำแนกได้ว่าภาพใดเป็นภาพอาหารหรือภาพใดไม่ใช่ภาพอาหาร ภาพอาหารนี้มีชื่ออาหารว่าอะไร เป็นต้น

การรู้จำภาพ (Image Recognition) หมายถึง กลุ่มตัวเลขที่แสดงถึงคุณลักษณะของภาพ ในการรู้จำภาพจะต้องรู้จำรูปแบบ หรือคุณลักษณะเด่นของแต่ละภาพ เพื่อที่จะสามารถแยกแยะภาพที่ต่างกันออกจากกันได้ แบบภาพที่ดีจะบ่งบอกถึงลักษณะของภาพ ซึ่งอาจได้จากการวัด เช่น อัตราส่วนความกว้างต่อความยาว เป็นต้น

การประเมินคุณค่าทางโภชนาการ หมายถึง การประเมินปริมาณของสารอาหารต่างๆ ที่อยู่ในอาหาร ซึ่งร่างกายสามารถย่อย ดูดซึม และนำไปใช้ประโยชน์ได้

1.6 สมมติฐานของการวิจัย

ภาพถ่ายอาหารที่นำมาใช้ต้องสามารถมองเห็นอาหารได้มากกว่าร้อยละ 70 ของภาพทั้งหมด และต้องมีอาหารในภาพจำนวน 1 เมนู โดยประกอบด้วยภาชนะ และพื้นหลัง

บทที่ 2

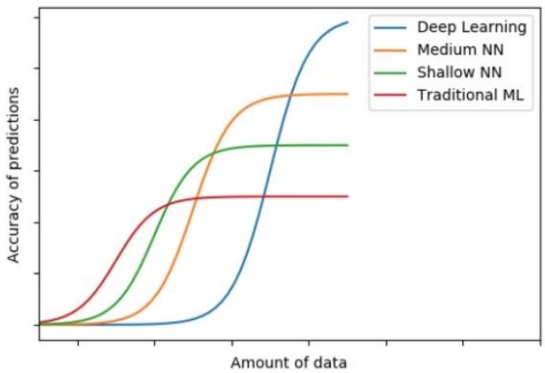
เอกสารและงานวิจัยที่เกี่ยวข้อง

การวิจัยในครั้งนี้ผู้วิจัยได้ทำการศึกษาค้นคว้าความรู้จากเอกสารต่าง ๆ ตลอดจนได้มีการทบทวนวรรณกรรมที่เกี่ยวข้องกับการวิจัยในครั้งนี้ดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

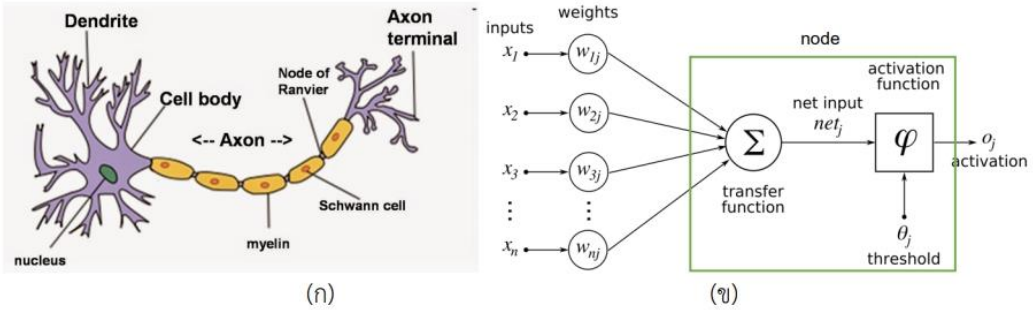
2.1.1 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก เป็นสาขาย่อยของการเรียนรู้ของเครื่อง (machine learning) การเรียนรู้ของเครื่องเป็นสาขาย่อยของปัญญาประดิษฐ์ (artificial intelligence) (Kelleher, 2019) ความแตกต่างที่สำคัญที่สุดประการหนึ่งของการเรียนรู้ของเครื่องกับการเรียนรู้เชิงลึก คือ ความสามารถในการปรับขนาดของการเรียนรู้ เมื่อข้อมูลมีขนาดเล็กการเรียนรู้เชิงลึกนั้นทำงานได้ไม่ดี แต่เมื่อปริมาณข้อมูลเพิ่มขึ้น ประสิทธิภาพในการเรียนรู้ทำความเข้าใจ และดำเนินการกับข้อมูลเชิงลึกจะเพิ่มสูงขึ้นแบบก้าวกระโดด ในขณะที่การเรียนรู้ของเครื่องแบบดั้งเดิมไม่ได้ขึ้นอยู่กับปริมาณข้อมูลมากนัก ดังแสดงในภาพ 1 ที่แสดงให้เห็นอย่างชัดเจนว่าอัลกอริทึมการเรียนรู้ของเครื่องอย่างเช่น การเรียนรู้เชิงลึก จะเจริญเติบโตได้ดีมีความถูกต้องในการทำนายสูงเมื่อได้รับข้อมูลจำนวนมหาศาล อย่างไรก็ตาม การเรียนรู้ของเครื่องแบบดั้งเดิม (traditional ML) และโครงข่ายประสาทเทียมขนาดเล็ก (shallow neural networks) ที่มีเซลล์ประสาทต่อการเชื่อมต่อจำนวนน้อย สามารถให้ผลลัพธ์ที่ดีเมื่อจำนวนข้อมูลที่จำกัด ดังนั้นการเลือกอัลกอริทึมการเรียนรู้ของเครื่องที่เหมาะสมสำหรับปัญหา และปริมาณข้อมูลที่มีอยู่เป็นวิธีที่ง่ายที่สุดในการออกจากกับดักเรื่องปริมาณข้อมูลที่ใช้ในการเรียนรู้ (Vento & Fanfarillo, 2019)



ภาพ 1 การเปรียบเทียบประสิทธิภาพอัลกอริทึมการเรียนรู้ของเครื่อง

การเรียนรู้เชิงลึก เป็นการนำแนวคิดของประสาทวิทยา (neuroscience) ที่ศึกษาการทำงานของสมองมนุษย์มาเลียนแบบ เพื่อสอนให้คอมพิวเตอร์สามารถเรียนรู้ได้อย่างที่มนุษย์เรียนรู้ การเรียนแบบนั้นเป็นการเลียนแบบการทำงานของสมอง โดยการจำลองโครงสร้างทางสมองของมนุษย์ขึ้นมา เรียกว่า โครงข่ายใยประสาทเทียม (artificial neural networks : ANN) โดยที่เซลล์ประสาท (neuron) จะถูกเรียกว่า โหนด (nodes) และมีใยประสาท (nerve fiber) ที่ทำหน้าที่รับสัญญาณประสาท (input) เรียกว่า เดนไดรต์ (dendrite) และส่งสัญญาณประสาท (output) เรียกว่า แอกซอน (axon) การประมวลผลในเซลล์ประสาทเทียมจะรับข้อมูลหรือสัญญาณที่ส่งมาจากเซลล์ประสาทที่อยู่ในชั้นก่อนหน้า และมีการประมวลผลเกิดขึ้นเกิดขึ้นภายในกรอบสีเขียวโดยเรียกส่วนนี้ว่าโหนด และส่งกระแสสัญญาณที่มีการถ่วงน้ำหนักแล้ว ให้เซลล์ประสาทที่อยู่ในชั้นถัดไป ดังภาพ 2

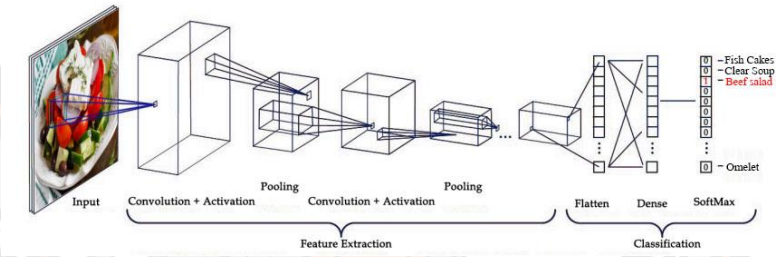


ภาพ 2 การเปรียบเทียบส่วนประกอบเซลล์ประสาทสมองมนุษย์กับเซลล์ประสาทเทียม
(ก) เซลล์ประสาทสมองมนุษย์ (ข) เซลล์ประสาทเทียม

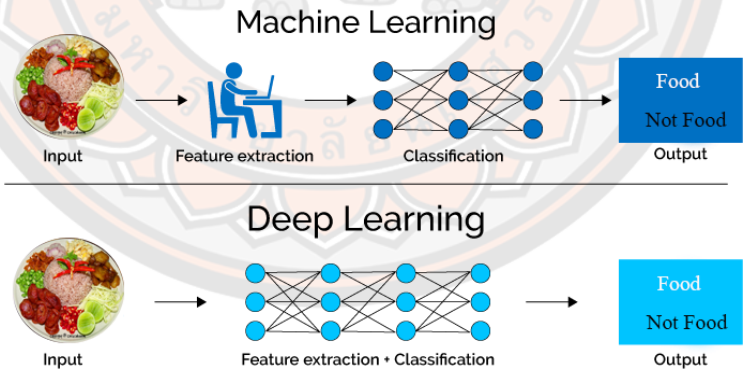
2.1.2 เครือข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network, CNN)

เครือข่ายประสาทแบบคอนโวลูชัน จัดเป็นเครือข่ายประสาทเทียมที่อยู่ในกลุ่มวิธีการค้นหาคำตอบที่เหมาะสมที่สุด (optimizations) ที่ได้รับแรงบันดาลใจจากสิ่งมีชีวิตจากธรรมชาติ (bio-inspired) โดยที่ CNN จะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นส่วนที่ย่อย ๆ จากนั้นนำกลุ่มของพื้นที่ย่อย ๆ เหล่านั้นมาผสานกันเพื่อตรวจสอบดูว่าสิ่งที่กำลังมองอยู่ในพื้นที่หรือภาพดังกล่าวหรือไม่ หรืออาจกล่าวได้ว่าเป็นการตรวจจับหรือการสกัดลักษณะเด่น (Feature extraction) ของภาพ และเรียกว่าเป็น “Local feature” ของภาพดิจิทัลนั้น ๆ ค่า Local feature ของภาพที่ตรวจจับได้นั้นถือเป็นส่วนหนึ่งของเครือข่ายประสาทเทียมที่จะถูกปรับไปพร้อม ๆ กับตัวที่ใช้ในการแบ่งแยกประเภท (classifier) ดังนั้นเมื่อทำการสอนให้เครือข่ายทำการเรียนรู้ (training) เสร็จสิ้นแล้วจะได้ลักษณะของเครือข่ายประสาทเทียมที่สามารถสกัดและจำแนกลักษณะเด่น (feature extractor) ที่เหมาะสมกับงาน ซึ่งถือว่าเป็นจุดเด่นของเครือข่าย CNN

เครือข่ายประสาทแบบคอนโวลูชันจัดว่าเป็นการเรียนรู้เชิงลึก ซึ่งมีความแตกต่างไปจากการเรียนรู้ของเครื่องจักรทั่ว ๆ ไป ที่ผู้ใช้งานจะต้องทำการสกัดลักษณะเด่นด้วยตนเองก่อนจะป้อนเป็นอินพุตให้กับเครือข่ายประสาทเทียมใช้ในการเรียนรู้ ส่วนการเรียนรู้เชิงลึกนั้นมีการใช้เครือข่าย ANN ที่มีชั้นซ่อนเร้น (hidden layers) หลายชั้น ทั้งนี้ เพื่อเพิ่มความสามารถในการคิดที่มากกว่าปกติทำให้สามารถทำการคำนวณโจทย์ปัญหาซับซ้อนได้สามารถใช้เทคนิคต่าง ๆ ได้มากขึ้น และที่สำคัญที่สุดคือ CNN สามารถทำการคิดอย่างเป็นขั้นเป็นตอนได้ซึ่งสามารถลอกเลียนแบบการทำงานของสมองมนุษย์ได้ดีขึ้น แผนภาพตัวอย่างสถาปัตยกรรมของเครือข่ายประสาทแบบคอนโวลูชันและแผนภาพแสดงความแตกต่างระหว่าง Machine learning กับ Deep learning แสดงในภาพ 3 และภาพ 4 ตามลำดับ



ภาพ 3 สถาปัตยกรรมของเครือข่ายประสาทแบบคอนโวลูชัน
ดัดแปลงมาจาก : https://arxiv.org/ftp/arxiv/papers/2004/2004.03357.pdf



ภาพ 4 ความแตกต่างระหว่าง Machine learning กับ Deep learning

สถาปัตยกรรมของเครือข่ายประสาทแบบคอนโวลูชันแสดงในภาพ 3 แสดงให้เห็นว่า CNN คือ Neural networks (NN) ที่มีหลายเลเยอร์ (layers) และโครงสร้างเฉพาะตัวซึ่งถูกออกแบบมาเพื่อการเพิ่มความสามารถในการสกัดเอาลักษณะเด่น (features) ที่มีความซับซ้อนมาก จากข้อมูลในปัจจุบัน CNN มักจะถูกใช้เพื่อการสกัดลักษณะเด่นจากข้อมูลประเภทที่ไม่ค่อยเป็นระเบียบหรือข้อมูลที่ไม่ได้มีโครงสร้างเป็นรูปแบบเฉพาะตัว (unstructured data) เช่น ภาพดิจิทัล (digital image) เป็นต้น

การคำนวณตามสถาปัตยกรรมของ CNN มีขั้นตอนการคำนวณที่แบ่งออกได้ 3 ขั้นตอน คือ ขั้นตอนการคอนโวลูชัน (convolution stage) ขั้นตอนการตรวจจับ (detector stage) และขั้นตอนการพูลลิ่ง (pooling stage)

1) ขั้นตอนการคอนโวลูชัน การคำนวณในขั้นตอนนี้จะใช้หลักการเดียวกันกับการคำนวณคอนโวลูชันเชิงพื้นที่ (spatial convolution) ที่ใช้ในงานในด้านการประมวลผลภาพดิจิทัล (digital image processing) จุดมุ่งหมายของการคำนวณคอนโวลูชันเชิงพื้นที่กับรูปภาพดิจิทัลก็คือ การสกัดลักษณะเด่นจากรูปภาพอินพุตแบบดิจิทัล โดยการคำนวณคอนโวลูชันทำให้เกิดการแปลงเชิงเส้น (linear transformation) ของรูปภาพอินพุตที่สอดคล้องกับในข้อมูลเชิงพื้นที่จากตัวกรอง (filters) โดยมีค่าถ่วงน้ำหนัก (weight) ของแต่ละ layer จะเป็นตัวกำหนดรายละเอียดของคอนโวลูชันเคอร์นอล (convolution kernel) ดังนั้น Convolution kernel สามารถทำการเทรน หรือทำการฝึกสอนได้ และขึ้นอยู่กับอินพุตของเครือข่ายประสาทเทียบแบบ CNN ขั้นตอนการคอนโวลูชันจะเริ่มจากการกำหนดจำนวนของตัวกรอง สำหรับใช้ในการแยกลักษณะเด่นหรือ Feature ของรูปภาพโดยปกติตัวกรอง 1 ตัวกรองจะสามารถทำการคัดแยกได้ 1 Feature โดยต้องมีการกำหนดขนาดของหน้าต่าง Sliding window หรือขนาดของเคอร์นอล (kernel size) ที่ใช้ของตัวกรองนั้น ๆ ไว้ด้วยในขั้นตอนนี้จะมีการคำนวณคอนโวลูชันเชิงพื้นที่ ระหว่างตัวกรองกับรูปภาพอินพุตโดยใช้การเทคนิค Sliding window หรือการสแกนรูปอินพุตเพื่อสร้างผังคุณลักษณะ (feature map) โดยมีการกำหนดค่า Strides เพื่อให้ Sliding window ทำการเลื่อนตำแหน่งไปครั้งละ กี่พิกเซลภาพ (pixels) ในขั้นตอนของการสแกนภาพอินพุตตามแกนแนวตั้งและแนวแกนนอนให้ครอบคลุมตลอดทั้งรูปภาพอินพุต

2) ขั้นตอนการตรวจจับ ในขั้นตอนนี้จะทำหน้าที่รับข้อมูลที่ได้จากขั้นตอน Convolution stage มาแปลงให้อยู่ในรูปแบบที่ไม่เป็นเชิงเส้น (nonlinear) โดยใช้ฟังก์ชันการกระตุ้น (activation function) เช่น Rectified Linear Units (ReLU) โดยผลลัพธ์ที่ได้จากการทำคอนโวลูชันในแต่ละตำแหน่งจะผ่านการแปลงค่าด้วยฟังก์ชัน ReLU ที่เป็นการแปลงแบบไม่เป็นเชิงเส้น เพื่อความง่ายในการคำนวณและประสิทธิภาพของผลลัพธ์

3) ขั้นตอนการพูลลิ่ง การคำนวณ Pooling เป็นการประมวลผลที่ทำให้เกิดการลดขนาดข้อมูลหรือการสุ่มต่ำ (subsampling) ข้อมูลโดยทำให้ข้อมูลที่ได้ทางด้านเอาต์พุตมีขนาดเล็กลงโดยที่รายละเอียดของข้อมูลที่ป้อนเข้ามายังคงครบถ้วนเหมือนเดิม การพูลลิ่งแบบค่าสูงสุด (max pooling) เป็นตัวกรองแบบหนึ่งที่ค้นหาค่าสูงสุด (maximum) ในบริเวณที่ตัวกรองทาบอยู่แล้วนำมาเป็นผลลัพธ์โดยจะเตรียมตัวกรองในลักษณะเดียวกับขั้นตอนการทำ Feature extraction ของ CNN มาทาบบนข้อมูลแล้วเลือกค่าสูงสุดบนตัวกรองนั้นมาเป็นผลลัพธ์ใหม่ จากนั้นทำการเลื่อนตัวกรองไปตาม Stride ที่กำหนดไว้ การ Pooling มีประโยชน์ในเรื่องของการเพิ่มความไวในการคำนวณ และยังช่วยในการแก้ปัญหาการเกิด Overfitting ในขั้นตอนของการเรียนรู้วิธีการ Pooling ที่นิยมใช้ทั่วไปคือ

วิธีการ MAX Pooling กับวิธีการ L2 โดยในสถาปัตยกรรมของ CNN โดยทั่วไปแล้ว การคำนวณ Pooling จะใช้ขนาดของหน้าต่างในการคำนวณเท่ากับ 2x2 และใช้ขนาดของ Stride เท่ากับ 2 โดยที่ไม่ต้องมีการเพิ่มพิกเซลภาพ (padding) ที่บริเวณขอบของภาพ ซึ่งเป็นวิธีการคำนวณที่แตกต่างไปจากการคำนวณ Convolution โดยทั่วไป

2.1.3 การแบ่งส่วนภาพ (Image Segmentation)

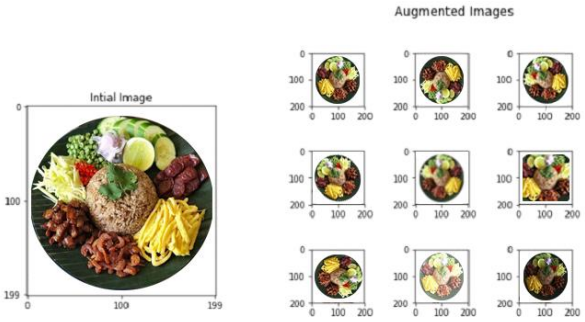
การแบ่งส่วนภาพ คือ การแบ่งข้อมูลภาพออกเป็นส่วนย่อย ๆ โดยใช้การแยกองค์ประกอบต่าง ๆ ของรูปภาพออกจากกันตามลักษณะสำคัญที่เราพิจารณา ขั้นตอนนี้เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพ ประโยชน์ของการแบ่งข้อมูลภาพออกเป็นส่วน ๆ คือ จะทำให้สามารถลดจำนวนข้อมูลในรูปภาพ โดยการคัดแยกส่วนประกอบภายในภาพที่ไม่ต้องการ ซึ่งเป็นการแยกแยะระหว่างส่วนที่เราสนใจ เช่น วัตถุในภาพ กับส่วนที่ไม่ต้องการ เช่น ฉากหลัง และเมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไปจำนวนข้อมูลที่เหลือที่จำเป็นในการวิเคราะห์จริงจะลดลงอย่างมาก และจัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ดีขึ้น ข้อมูลภาพที่ผ่านการแบ่งแยกแล้ว จะมีโครงสร้างที่ชัดเจนขึ้น และนำไปใช้งานได้สะดวกขึ้น ดังตัวอย่างภาพ 5

(ก) (ข)

ภาพ 5 ตัวอย่างการเปรียบเทียบรูปภาพอาหารต้นฉบับกับรูปภาพอาหารที่ทำการแบ่งส่วนภาพ
(ก) รูปภาพอาหารต้นฉบับ (ข) รูปภาพอาหารที่ทำการแบ่งส่วนภาพ

2.1.4 การทำ Augmentation

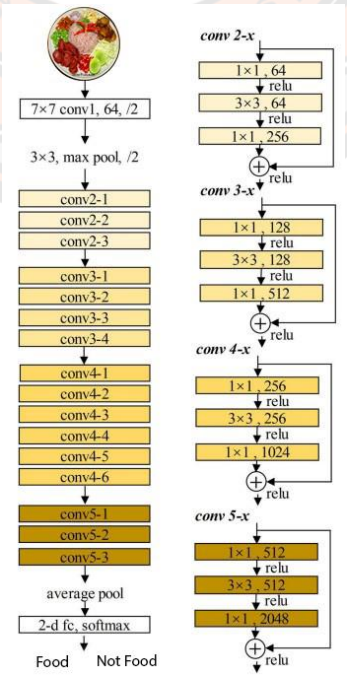
การทำ Augmentation (Pawara et al., 2017) ให้กับชุดข้อมูลที่นำมาเรียนรู้ทำเพื่อเพิ่มความหลากหลายของชุดข้อมูลให้มีความหลากหลาย เพื่อที่ Model จะได้เรียนรู้ได้ประสิทธิภาพดียิ่งขึ้น โดยการทำ Augment ได้แก่ การหมุนภาพ (Rotation) ในมุมต่าง ๆ การเลื่อนภาพ (Shift) การบิดภาพ (Shearing) การซูม (Zooming) การปรับแสงเงา (Shading) เป็นต้น เพื่อให้ได้ภาพในลักษณะต่าง ๆ กัน สำหรับตัวอย่างการทำ Augment สามารถแสดงได้ดังภาพ 6



ภาพ 6 ตัวอย่างภาพแสดงการทำ Augmentation

2.1.5 ResNet

ResNet คือ Deep Residual Network ถูกนำเสนอในงานวิจัย Deep residual learning for image recognition ซึ่งแก้ปัญหาเรื่อง vanishing gradient ซึ่งเกิดขึ้นกับโครงข่ายที่มีความลึกค่อนข้างมาก ซึ่งมีจำนวนชั้นของ network ถึง 152 เลเยอร์ (8 เท่าของโมเดล VGG16) โดยใช้เทคนิคการออกแบบ module ที่มีลักษณะทางลัดลงใน network ตัวโครงข่ายนี้ประกอบด้วยกัน 4 บล็อก โดยจำนวนที่มีพารามิเตอร์สำหรับฝึกทั้งหมด คือ ชั้นที่เราใช้เรียกชื่อ เช่น ResNet50 จะหมายถึงมีจำนวน 50 เลเยอร์ ซึ่งจะอธิบายขนาดว่า [3, 4, 6, 3] ซึ่งคือ (3 + 4 + 6 + 3) x 3 = 48 ชั้น + 2 ชั้น เท่ากับ 50 เป็นต้น ซึ่ง ResNet ที่นิยมใช้จะเป็น ResNet18, ResNet34, ResNet50, ResNet101 และ ResNet152 (Chayatummagoon & Chongstitvatana, 2021) ตัวอย่างโครงสร้าง ResNet ดังภาพ 7



ภาพ 7 ตัวอย่างโครงสร้างของ ResNet

2.1.6 Mask R-CNN

Mask R-CNN เป็นแนวคิดที่เรียบง่ายในการตรวจจับ และแบ่งส่วนวัตถุ มีการพัฒนาต่อยอดมาจาก R-CNN, Faster R-CNN ส่วนมากจะเป็นการทำ Object Detection ซึ่งมีข้อจำกัดหลายอย่าง ต่อมามีการพัฒนาเพื่อแก้ไขข้อจำกัด Mask R-CNN จะเป็น Instance Segmentation ที่เป็นแบบ multiple object

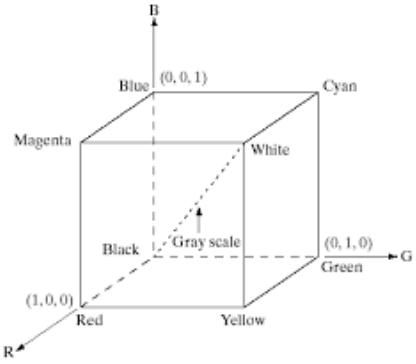
โครงสร้างของ Mask R-CNN ประกอบไปด้วย

- Convolutional ResNet backbone: ResNet เป็น CNN ชนิดหนึ่ง ซึ่งเป็นวิธีการนำเอา residual modules มาต่อกัน และใช้ stochastic descent gradient (SGD) มาฝึกสอน ข้อมูลที่นำเข้ามาจะถูกคัดแยกคุณลักษณะเด่น โดยการแบ่งเป็น patch เล็ก ๆ ก่อนถูกนำเข้ามาในแบบจำลอง
- Region proposal Network (RPN) เป็น layer หนึ่งของ Mask R-CNN ทำหน้าที่สกัดคุณสมบัติที่คาดว่าจะเป็นวัตถุออกจาก feature map จากนั้นนำส่วนที่คาดว่าจะเป็นวัตถุเข้า CNN เพื่อคำนวณหา feature ที่สามารถใช้แยกแยะได้ว่าพื้นที่ใดเป็นวัตถุ โดยต้องทำหลังจาก Selective Search
- ROI Pooling เป็นขั้นตอนการใช้พิกัดตำแหน่งบน Activation Map เพื่อหาเวกเตอร์ของคุณลักษณะ (feature vector) ให้มีขนาดคงที่แล้วส่งไปยัง Classification Layer กับฟังก์ชัน Softmax เพื่อทำการจัดหมวดหมู่ว่า proposed region feature จะตอบเป็นคลาสอะไร

2.1.7 ระบบสี (Color model)

2.1.7.1 ระบบสี RGB (Red/Green/Blue)

RGB คือ ระบบสีพื้นฐานที่ได้มาจากสีที่คนทั่วไปรู้จักนั้น คือ สีแดง สีเขียว และสีน้ำเงิน ในระบบสีของคอมพิวเตอร์นั้น RGB มีค่าสีที่อัตราส่วนของสีทั้ง 3 สีอยู่ในช่วง 0 ถึง 255 ซึ่งค่า 0 คือ สีดำ และ 255 คือ สีขาว โดยแต่ละค่าสีมีค่าเท่ากับ 256 สีกลางคือ ทั้ง 3 สีมีค่าตั้งแต่ 0 ถึง 255 จะมองเห็นแต่ละสีเป็นภาพระดับสีเทาเมื่อเราจะใช้การผสมค่าสีของทั้ง 3 สีทำให้ได้สีที่ต้องการออกมา สีที่ได้นั้นสามารถแสดงสีได้มากถึง 16 ล้านสี เพราะ(R× B×G) = 256×256×256 = 16,777,216 แต่ในความเป็นจริงคอมพิวเตอร์สามารถแสดงค่าสีออกมาได้สูงสุด 256 สีเทานั้น ดังนั้นจึงต้องมีการแปลงค่าสีเพื่อให้ได้อัตราส่วนของสีที่ไม่เกิน 256 สีเมื่อเป็นภาพระดับสีเทา

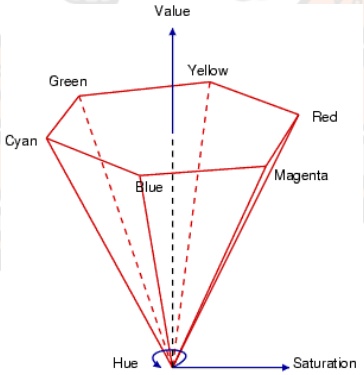


ภาพ 8 ลูกบาศกระบบสี RGB

ที่มา : http://www.wbi.msu.ac.th/file/1209/doc_41086.425196759318.pdf

2.1.7.2 ระบบสี HSV

สีโหมด HSV ยอมาจาก Hue คือ คาสีบริสุทธิ์ Saturation คือ คาแสงผสมกับค่าสีบริสุทธิ์ และValue คือ ค่าความสวางของภาพ ตามลำดับ HSV Model ตางจากสีในโหมด RGB CMY หรือ YIQ ซึ่ง HSV มีลักษณะเปนพีระมิดฐานหกเหลี่ยมดังภาพ 9 โดยพีระมิดนี้แปลงมาจากลูกบาศกระบบสี RGB แบบไมเชิงเสน สีโหมด HSV จะใชคาในพิกัดเชิงมุม โดยคา H เปนคามุมรอบแกนตั้ง จะระบุเปนองศามีคาระหวาง 0 ถึง 360 องศาคา S เปนคาอัตราสวนมีคาตั้งแต 0 ถึง 1 วัดจากแกนตั้ง (V-axis) ไปยังพื้นผิวของพีรามิด และคา V เปนคาความสูงของพีระมิด



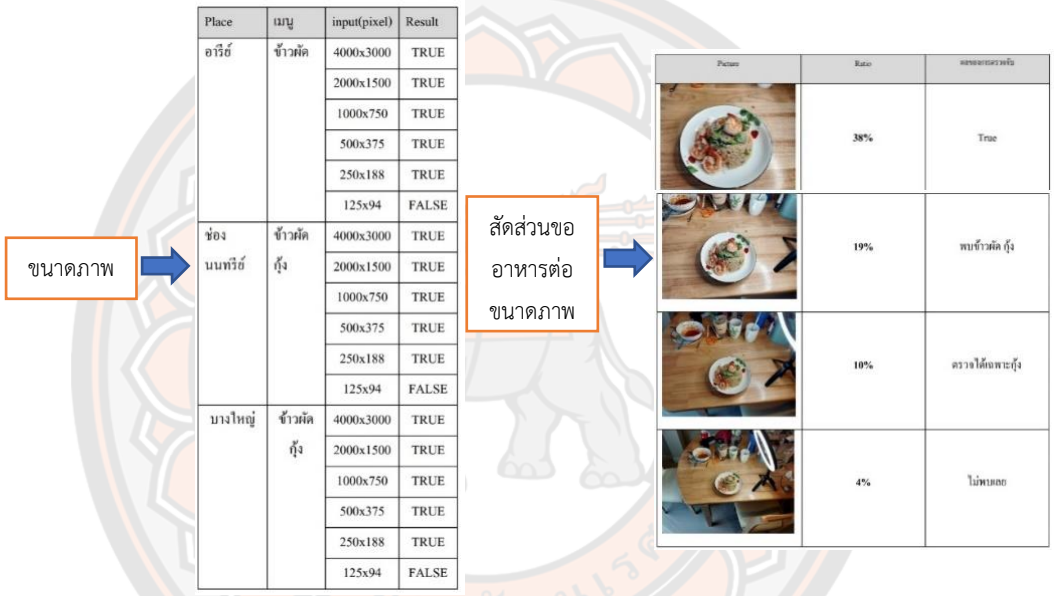
ภาพ 9 Single-hexcone HSV color model

ที่มา : Widjaja et al. (2003)

2.2 งานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยได้มีการนำเอาความรู้ทางด้านการประมวลผลภาษาธรรมชาติ กระบวนการจัดเตรียมข้อมูล ตลอดจนเทคนิคการเรียนรู้ของเครื่องมาประยุกต์ใช้อย่างหลากหลาย โดยงานวิจัยที่ผู้วิจัยได้เลือกมาสำหรับทบทวนวรรณกรรมมีดังนี้

รังสรรค์ มนัสมานะชัย และธนภัทร ฆังคะจิตร (2564) ได้นำเสนอวิธีการจำแนกและตรวจจับวัตถุดิบในเมนูอาหาร โดยใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งงานวิจัยนี้เลือกใช้เทคนิค YOLO v4 (Darkness framework) และแบ่งชุดข้อมูลออกเป็น 2 ส่วน ได้แก่ ชุดฝึกสอน (training) ร้อยละ 80 และชุดทดสอบ (Test) ร้อยละ 20 นำข้อมูลรูปภาพอาหารทั้งหมดมาระบุตำแหน่งของวัตถุดิบในอาหาร และนำไปสร้างแบบจำลอง ผลการวิจัยพบว่ารูปภาพที่มีความละเอียดต่ำกว่า 250x188 พิกเซล หรือภาพอาหารที่มีสัดส่วนน้อยกว่าร้อยละ 20 จะไม่สามารถตรวจจับรายการอาหารและวัตถุดิบในอาหารได้ และหากเมนูอาหารมีจำนวนวัตถุดิบน้อยจะมีความแม่นยำสูง ในทางกลับกันหากเมนูอาหารมีจำนวนวัตถุดิบมากความแม่นยำจะลดลง



Place	เมนู	input(pixel)	Result
อารีย์	ข้าวผัด	4000x3000	TRUE
		2000x1500	TRUE
		1000x750	TRUE
		500x375	TRUE
		250x188	TRUE
		125x94	FALSE
ซ่อง นนทรีย์	ข้าวผัดกุ้ง	4000x3000	TRUE
		2000x1500	TRUE
		1000x750	TRUE
		500x375	TRUE
		250x188	TRUE
		125x94	FALSE
บางใหญ่	ข้าวผัดกุ้ง	4000x3000	TRUE
		2000x1500	TRUE
		1000x750	TRUE
		500x375	TRUE
		250x188	TRUE
		125x94	FALSE

Picture	Ratio	ผลของการตรวจจับ
	38%	True
	19%	พบข้าวผัดกุ้ง
	10%	ตรวจได้เฉพาะกุ้ง
	4%	ไม่พบเลย

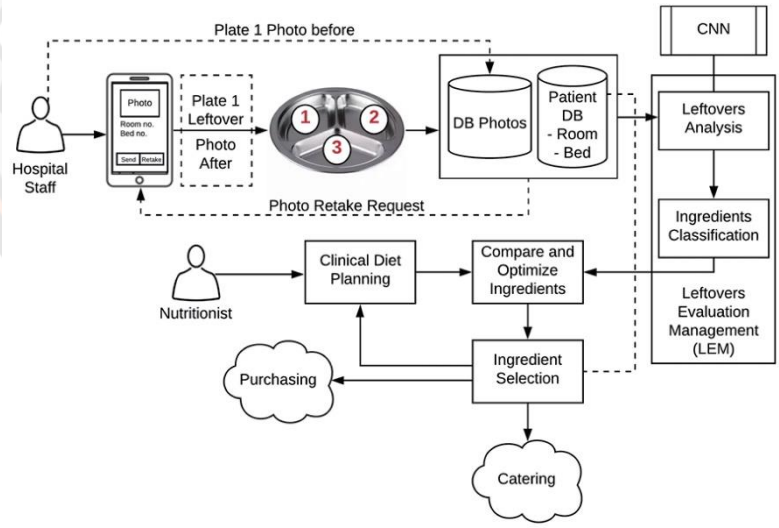
ภาพ 10 แสดงขนาดภาพ และสัดส่วนอาหารต่อขนาดภาพที่สามารถตรวจจับวัตถุดิบได้ดี

Darapaneni et al. (2021) ได้พัฒนาแบบจำลองการรู้จำอาหาร และทำนายแคลอรี่โดยรวมของอาหารที่อยู่ภายในภาพถ่าย โดยใช้เทคนิค Mask R-CNN ในการกำหนดขอบเขตตำแหน่ง (Mask) อาหาร เพื่อนำขนาดของพื้นที่ที่กำหนดขอบเขตตำแหน่งได้ไปใช้ในการคำนวณหาปริมาณแคลอรี่

ภาพ 11 ภาพตัวอย่างการกำหนดขอบเขตตำแหน่งอาหารด้วยเทคนิค Mask R-CNN

Mikolajczyk & Grochowski (2018) ได้นำเสนอวิธีการแก้ไขปัญหาการขาดข้อมูลในการฝึกอบรม หรือความสมดุลของคลาสที่ไม่สม่ำเสมอภายในชุดข้อมูล และวิธีการลดปัญหาการ Overfitting ของแบบจำลอง ด้วยวิธีที่เรียกว่า การเสริมข้อมูล (Data Augmentation) โดยการนำรูปเดิมที่ใช้ในการฝึกอบรมแบบจำลองมาสร้างภาพใหม่ด้วยการปรับเปลี่ยนคุณลักษณะของภาพ เช่น การหมุน การครอบตัด การซูม การทำฮิสโตแกรม ฯลฯ เป็นต้น ผลการวิจัยพบว่าการเสริมข้อมูลช่วยให้ประสิทธิภาพของแบบจำลองมีความแม่นยำเพิ่มมากขึ้น

Namahoot et al. (2020) งานวิจัยนี้ได้พัฒนาต้นแบบระบบสำหรับการสนับสนุนการวางแผนการรับประทานอาหารในโรงพยาบาล (RES-DIP) โดยการนำข้อมูลสถิติการบริโภคอาหารแต่ละมื้อของผู้ป่วยมาวิเคราะห์ว่าส่วนผสมใดที่ผู้ป่วยรับประทานเหลือน้อยที่สุด และคัดเอาเฉพาะส่วนผสมที่ผู้ป่วยทานเหลือน้อยกว่าร้อยละ 50 ในแต่ละมื้อ มาจัดเรียงลำดับวัตถุดิบในรายการอาหารตามร้อยละที่ทานเหลือน้อยที่สุด และมีคุณค่าทางโภชนาการสูงที่สุด จากนั้นนำส่วนผสมที่ได้มาค้นหาเมนูอาหารจากฐานข้อมูลสูตรอาหาร เพื่อใช้แนะนำเมนูอาหารที่เหมาะสมแก่นักโภชนาการ งานวิจัยเลือกใช้ Convolutional Neural Network (CNN) ในการคัดแยกวัตถุดิบ ในส่วนการประเมินปริมาณอาหารเหลือหลังจากการรับประทานได้ใช้สมการอย่างง่ายในการวิเคราะห์



ภาพ 12 สถาปัตยกรรมกระบวนการทำงานระบบ RES-DIP

Sun et al. (2016) ได้วิจัยการทำนายปริมาณกรดไขมันในเนื้อหมูจากภาพถ่าย โดยใช้ภาพถ่ายเนื้อหมูจากโรงงานแปรรูปที่มีพื้นหลังสีดำ ขั้นตอนเริ่มต้นจากการนำภาพถ่ายมาเข้าสู่กระบวนการลบพื้นหลัง แยกภาพสีที่ถูกแบ่งออกเป็นพื้นหลัง และพื้นที่ตัวอย่างเนื้อหมู หลังจากลบพื้นหลังเรียบร้อยแล้ว ขั้นตอนต่อมา คือ การแยกส่วนเนื้อหมูออกจากส่วนที่มีไขมัน ด้วยการวัดระยะความหยาบพื้นผิวภาพ (Roughness) หาความแตกต่างของสี (Contrast) หาเส้นขอบเขตวัตถุภายใน

ภาพ (Directionality) และคำนวณความเข้มของสีแต่ละพิกเซล (Heterogeneity) การทำนายกรดไขมันใช้วิธีการถดถอยเชิงเส้น (Linear Regression) และวิธีการเพิ่มตัวแปรอิสระแบบขั้นตอน (Stepwise Regression)

(ก) (ข) (ค)

ภาพ 13 การแบ่งกลุ่มของไขมันใต้ผิวหนังจากเนื้อซี่โครงหมู
(ก) ภาพต้นฉบับ (ข) การแบ่งสัดส่วนไขมันและพื้นหลัง และ (ค) บริเวณไขมัน

บทที่ 3

วิธีดำเนินงานวิจัย

ในงานวิจัยได้พัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือ 2 รูปแบบดังนี้

1. แบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วน
2. แบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

สำหรับการทดลองได้เก็บรวมรวมข้อมูลภาพถ่ายอาหารภายในจานทั้งก่อนรับการประทานและหลังการรับประทานเสร็จที่อาหารเหลือภายในจานถูกเกลี่ยรวมกันเป็นสัดส่วนไว้แล้ว และแบบที่ไม่เกลี่ยอาหารเหลือรวมกัน ซึ่งแบ่งสัดส่วนร้อยละของปริมาณอาหารเหลือภายในจานออกเป็น 9 ส่วน คือ 0, 12.5, 25, 37.5, 50, 62.5, 75, 87.5 และ100 ดังภาพ 14 ในงานวิจัยนี้เป็นการวิเคราะห์

ภาพ 14 ตัวอย่างปริมาณอาหารภายในจานที่ถูกแบ่งออกเป็น 9 ส่วน

และประเมินปริมาณอาหารเหลือหลังจากการรับประทานว่าเหลือร้อยละเท่าไร โดยใช้เทคนิค Mask R-CNN ที่มีโครงสร้างแบบ ResNet101 ในการสร้างแบบจำลองสำหรับกำหนดขอบเขตตำแหน่งอาหารเหลือภายในจาน และใช้วิธีการคำนวณพื้นที่ภายในภาพในการประเมินปริมาณอาหารเหลือ โดยใช้จำนวนภาพถ่ายอาหารเหลือ 141 ภาพสำหรับสร้างแบบจำลองที่เกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน และภาพถ่ายอาหารเหลือ 107 ภาพสำหรับสร้างแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน ที่เป็นชุดข้อมูลภาพอาหารต้นฉบับในการฝึกแบบจำลอง และแบ่งข้อมูลเป็นชุดฝึกสอน และชุดทดสอบ (80:20) ขนาดของภาพถ่ายมีความละเอียดไม่ต่ำกว่า 250x188 พิกเซล ที่มีสัดส่วนของอาหารต่อขนาดภาพไม่ต่ำกว่าร้อยละ 38 และเพิ่มกระบวนการเสริมข้อมูลเข้าไปในขณะฝึกสอน เพื่อลดการเกิดการแก้ไขปัญหาเกินพอดีของแบบจำลอง และเพิ่มประสิทธิภาพความแม่นยำ

ในการประเมินปริมาณอาหารเหลือมากยิ่งขึ้น โดยสามารถแบ่งขั้นตอนการดำเนินการวิจัยออกเป็น 3 ส่วน ดังนี้

3.1 เครื่องมือที่ใช้ในการวิจัย

3.2 ขั้นตอนการดำเนินการวิจัย

3.3 การวิเคราะห์ข้อมูล

3.1 เครื่องมือที่ใช้ในการวิจัย

ในกระบวนการพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึกผู้วิจัยได้ใช้เครื่องมือต่าง ๆ ดังต่อไปนี้

3.1.1 ซอฟต์แวร์ (Software)

- Windows 11 22h2
- Visual Studio Code เวอร์ชั่น 1.75.1
- Jupyter notebook เวอร์ชั่น 6.4.12
- ไลบรารี TensorFlow เวอร์ชั่น 2.11.0
- ไลบรารี Mask R-CNN
- ภาษา Python เวอร์ชั่น 3.9.13
- โปรแกรม Via 1.0.6

3.1.2 ฮาร์ดแวร์ (Hardware)

- โทรศัพท์มือถือ 1 เครื่อง
- เครื่องคอมพิวเตอร์ 1 เครื่อง
  - CPU INTEL CORE I7-11700KF
  - RAM DDR4(3600) ขนาด 32 GB
  - Graphics Card NVIDIA RTX3090
  - Hard Disk SSD M.2 PCIe 4.0 500 GB

3.2 ขั้นตอนการดำเนินการวิจัย

การพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึกได้ถูกออกแบบโดยมีกระบวนการการทำงานดังภาพ 13 และประกอบด้วย 3 ขั้นตอนหลักดังต่อไปนี้

3.2.1 การเก็บรวบรวมข้อมูล

เก็บรวบรวมข้อมูลภาพถ่ายอาหารภายในจานทั้งก่อนการรับประทาน และหลังการรับประทานเสร็จด้วยกล้องจากโทรศัพท์มือถือ

ภาพ 15 ตัวอย่างชุดข้อมูลภาพสำหรับสร้างแบบจำลองแบบเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

ภาพ 16 ตัวอย่างชุดข้อมูลภาพสำหรับสร้างแบบจำลองแบบไม่เกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

3.2.2 การเตรียมชุดข้อมูลและสร้างแบบจำลอง

ขั้นตอนที่ 1 แบ่งชุดข้อมูลรูปภาพออกเป็น 2 ส่วน ได้แก่ ชุดฝึกสอน 80 เปอร์เซ็นต์ และชุดทดสอบ 20 เปอร์เซ็นต์ โดยนำไฟล์ภาพทั้ง 2 ส่วนเก็บไว้ในโฟลเดอร์ ดังภาพ 17

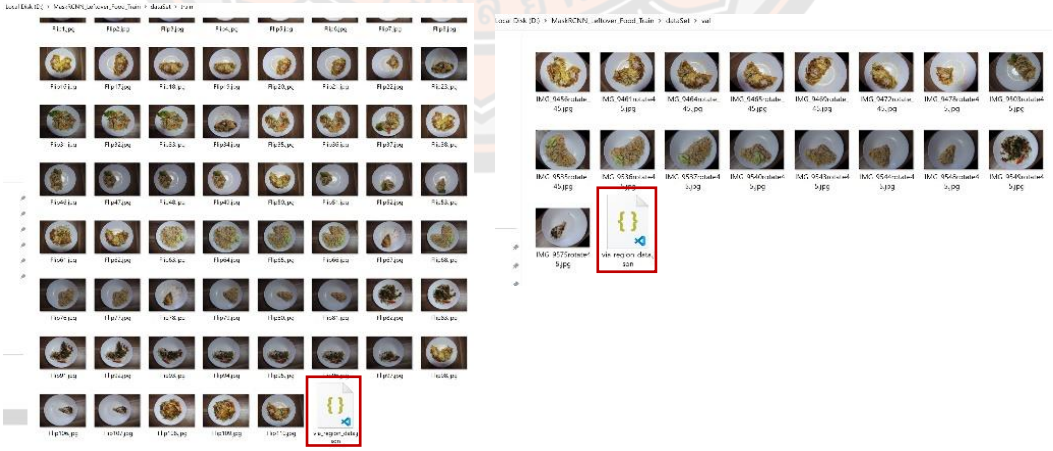
(ก) (ข)

ภาพ 17 ตัวอย่างการนำชุดข้อมูลรูปภาพเก็บไว้ในโฟลเดอร์
(ก) ชุดข้อมูลรูปภาพฝึกสอน (ข) ชุดข้อมูลรูปภาพทดสอบ

ขั้นตอนที่ 2 นำชุดข้อมูลรูปภาพทั้ง 2 ส่วนมากำหนดขอบเขตตำแหน่งอาหารเหลือภายในจานด้วยโปรแกรม Via 1.0.6 และ export ชนิดไฟล์เป็นแบบ JSON

ภาพ 18 การกำหนดขอบเขตตำแหน่งอาหารเหลือภายในจานด้วยโปรแกรม Via 1.0.6

ขั้นตอนที่ 3 นำไฟล์ JSON ที่ export จากโปรแกรม Via 1.0.6 เก็บไว้ในโฟลเดอร์เดียวกับชุดข้อมูลภาพฝึกสอน และชุดข้อมูลภาพทดสอบ



(ก) (ข)

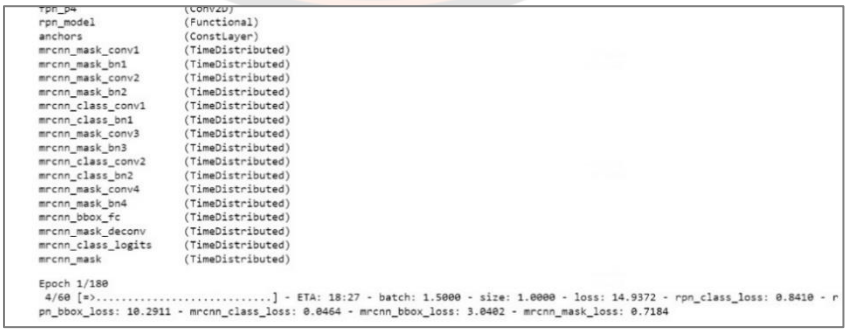
ภาพ 19 ตัวอย่างการเก็บไฟล์ JSON จากการกำหนดขอบเขตตำแหน่งอาหารเหลือ
(ก) เก็บไฟล์ JSON ไว้ในโฟลเดอร์ชุดข้อมูลภาพฝึกสอน
(ข) เก็บไฟล์ JSON ไว้ในโฟลเดอร์ชุดข้อมูลภาพทดสอบ

ขั้นตอนที่ 4 สร้างแบบจำลองการประเมินปริมาณอาหารเหลือ โดยใช้เทคนิค Mask R-CNN ที่ถูก pre-trained บนชุดข้อมูล COCO และเลือกใช้ backbone ใน Mask R-CNN ที่มีโครงสร้างแบบ ResNet101 ซึ่งประกอบด้วยเครื่องมือวิจัยดังนี้

- Jupyter Notebook เวอร์ชั่น 6.4.12
- ไลบรารี TensorFlow เวอร์ชั่น 2.11.0
- ไลบรารี Mask R-CNN for Train
- ภาษา Python เวอร์ชั่น 3.9.13

ในส่วนของการตั้งค่าการฝึกสอนแบบจำลองผู้วิจัยกำหนดให้ batch size=2, epoch=50 ถึง 500 learning rate=0.001 และเพิ่มกระบวนการเสริมข้อมูล ได้แก่ การกลับภาพตามแนวนอนและแนวตั้ง การหมุนภาพ 90° 180° และ180° การเอียงภาพ การแปลงค่าแสงภาพ และการเบลอภาพ

(ก) (ข)



(ค)

ภาพ 20 ตัวอย่างภาพการทำงานการสร้างแบบจำลอง

(ก) แสดงการตั้งค่าการฝึกสอนแบบจำลอง (ข) ทดสอบการกำหนดขอบเขตตำแหน่งอาหารเหลือ

(ค) เริ่มการฝึกสอนแบบจำลอง

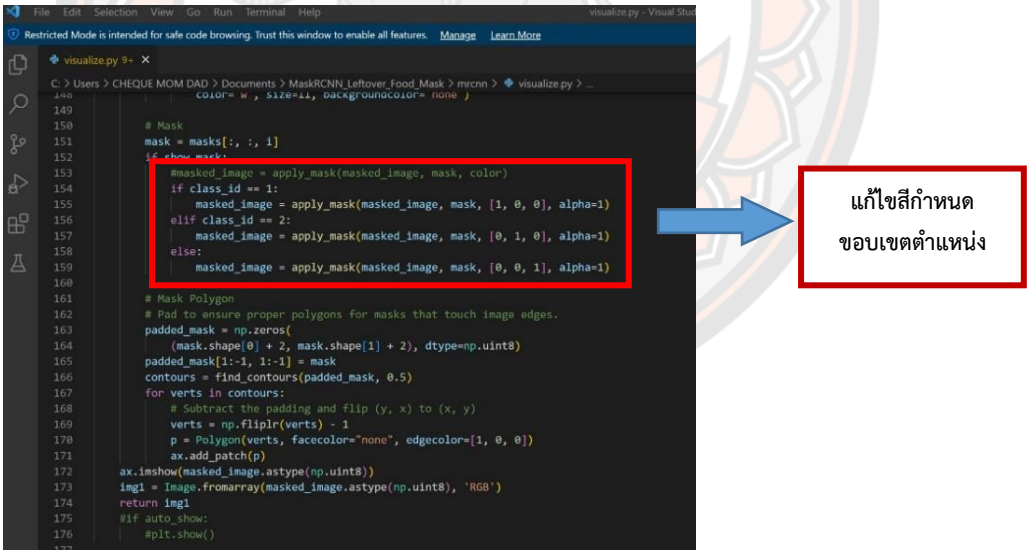
หลังจากสร้างแบบจำลองเสร็จเรียบร้อยแล้วจะได้ไฟล์แบบจำลองนามสกุล h5 จำนวนตามรอบการเรียนรู้ที่กำหนด ตัวอย่างเช่น กำหนดรอบการเรียนรู้จำนวน 50 รอบ ก็จะได้ไฟล์แบบจำลองจำนวน 50 ไฟล์ ซึ่งไฟล์แบบจำลองที่ได้จะนำไปใช้ในการประเมินปริมาณอาหารเหลือและทดสอบประสิทธิภาพการประเมินปริมาณอาหารเหลือภายในจานในขั้นตอนต่อไป

plugins	19/4/2566 11:42	File folder	
events.out.tfevents.1681879301.DESKTOP-L5L...	21/4/2566 3:26	DESKTOP-L5LM0IC Fi...	6,355 KB
events.out.tfevents.1681879342.DESKTOP-L5L...	19/4/2566 11:42	PROFILE-EMPTY File	1 KB
mask_rcnn_leftoverfood_0001.h5	19/4/2566 11:56	H5 File	250,940 KB
mask_rcnn_leftoverfood_0002.h5	19/4/2566 12:12	H5 File	250,940 KB
mask_rcnn_leftoverfood_0003.h5	19/4/2566 12:26	H5 File	250,940 KB
mask_rcnn_leftoverfood_0004.h5	19/4/2566 12:40	H5 File	250,940 KB
mask_rcnn_leftoverfood_0005.h5	19/4/2566 12:54	H5 File	250,940 KB

ภาพ 21 ตัวอย่างภาพไฟล์แบบจำลองนามสกุล h5

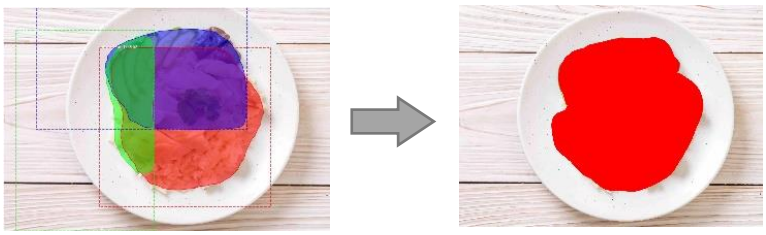
3.2.3 การประเมินปริมาณอาหารเหลือ

ขั้นตอนที่ 1 แก้ไขไลบรารี่ Mask R-CNN ให้สีที่ใช้กำหนดขอบเขตตำแหน่งเป็นสีแดงเท่านั้น และปรับค่าความโปร่งใส (transparent) ของสีให้มีค่าเป็น 0




```
# Mask
mask = masks[:, :, i]
#masked_image = apply_mask(masked_image, mask, color)
if class_id == 1:
    masked_image = apply_mask(masked_image, mask, [1, 0, 0], alpha=1)
elif class_id == 2:
    masked_image = apply_mask(masked_image, mask, [0, 1, 0], alpha=1)
else:
    masked_image = apply_mask(masked_image, mask, [0, 0, 1], alpha=1)
# Mask Polygon
# Pad to ensure proper polygons for masks that touch image edges.
padded_mask = np.zeros(
    (mask.shape[0] + 2, mask.shape[1] + 2), dtype=np.uint8)
padded_mask[1:-1, 1:-1] = mask
contours = find_contours(padded_mask, 0.5)
for verts in contours:
    # Subtract the padding and flip (y, x) to (x, y)
    verts = np.fliplr(verts) - 1
    p = Polygon(verts, facecolor="none", edgecolor=[1, 0, 0])
    ax.add_patch(p)
ax.imshow(masked_image.astype(np.uint8))
img1 = Image.fromarray(masked_image.astype(np.uint8), 'RGB')
return img1
#if auto_show:
    #plt.show()
```

ภาพ 22 ตัวอย่างภาพการแก้ไขการกำหนดขอบเขตตำแหน่งเป็นสีแดง



ภาพ 23 การกำหนดขอบเขตตำแหน่งอาหารเหลือหลังจากแก้ไขไลบรารี่ Mask R-CNN

ขั้นตอนที่ 2 นำภาพถ่ายอาหารทั้งก่อนการรับประทาน และหลังรับประทานเสร็จเข้าสู่กระบวนการกำหนดขอบเขตตำแหน่งอาหารเหลือด้วยเทคนิค Mask R-CNN ซึ่งประกอบด้วยเครื่องมือวิจัยดังนี้

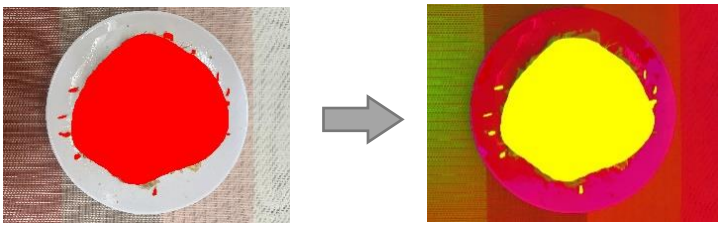
- Visual Studio Code เวอร์ชั่น 1.75.1
- ไลบรารี TensorFlow เวอร์ชั่น 2.11.0
- ไลบรารี Mask R-CNN for Mask
- ภาษา Python เวอร์ชั่น 3.9.13

ก่อน หลัง

ภาพ 24 ตัวอย่างภาพอาหารก่อนการรับประทาน และหลังรับประทานเสร็จ

ภาพ 25 ตัวอย่างภาพที่กำหนดขอบเขตตำแหน่งอาหารเหลือด้วยเทคนิค Mask R-CNN

ขั้นตอนที่ 3 นำภาพที่กำหนดขอบเขตตำแหน่งอาหารเหลือเรียบร้อยแล้ว มาปรับภาพให้อยู่ในระบบสี HSV

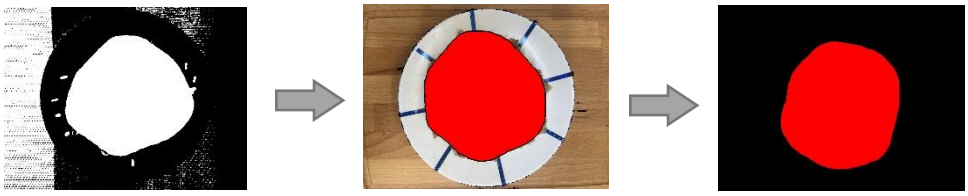


ภาพ 26 ตัวอย่างรูปภาพการปรับภาพเป็นโหมด HSV

ขั้นตอนที่ 4 ค้นหาจุดภาพหรือพิกเซล (pixel) ที่เป็นสีแดง

ขั้นตอนที่ 5 วาดเส้นเค้าโครงกำหนดขอบเขตครอบพื้นที่สีแดงทั้งหมดไว้

ขั้นตอนที่ 6 ลบพื้นหลัง หรือลบพื้นที่วัตถุที่อยู่นอกขอบเขตเส้นเค้าโครงที่วาดไว้



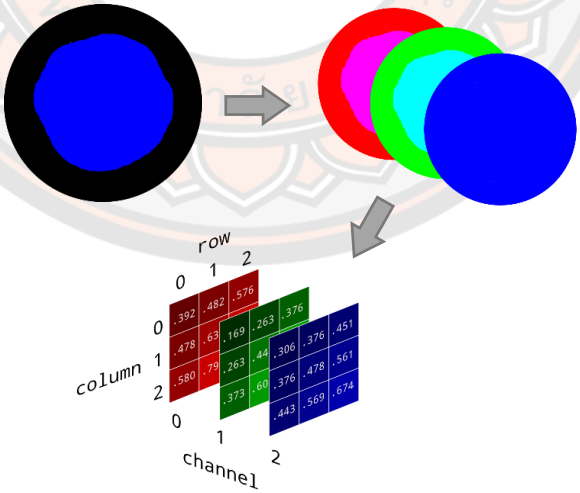
(ก) (ข) (ค)

ภาพ 27 ตัวอย่างรูปภาพขั้นตอนการลบวัตถุที่ไม่ใช่อาหารเหลือ
(ก) ค้นหาพิกเซลสีแดง (ข) วาดเส้นเค้าโครงครอบพื้นที่สีแดง (ค) ลบพื้นที่ที่ไม่ใช่พื้นที่สีแดง

ขั้นตอนที่ 7 แปลงภาพจากระบบสี BGR เป็นระบบสี RGB เนื่องจากการอ่านข้อมูลไฟล์ภาพเข้าระบบจะถูกแปลงจากระบบสี RGB เป็นระบบสี BRG

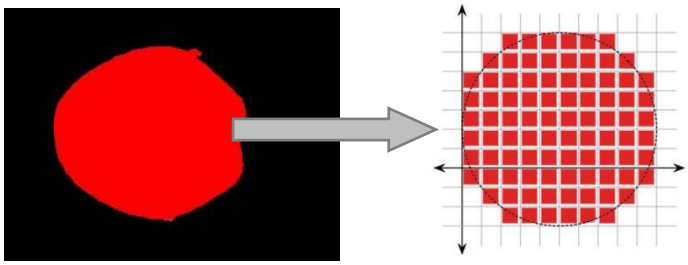
ภาพ 28 ตัวอย่างรูปภาพการแปลงภาพเป็นระบบสี RGB

ขั้นตอนที่ 8 แปลงภาพให้อยู่ในรูปแบบของอาเรย์



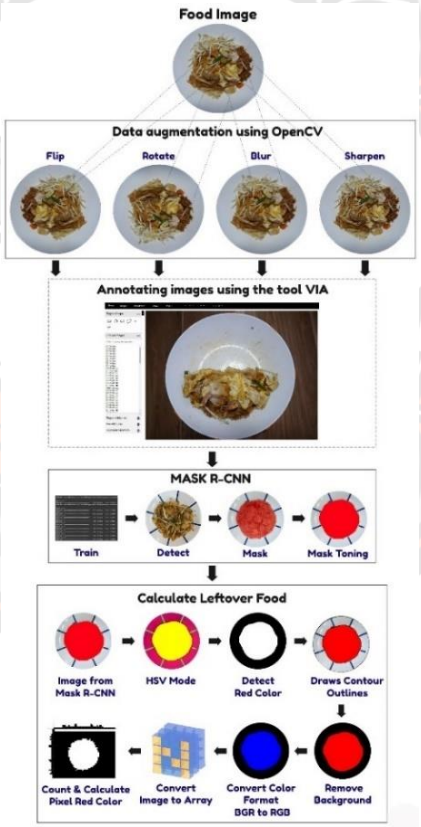
ภาพ 29 ขั้นตอนการแปลงภาพให้อยู่ในรูปแบบของอาเรย์

ขั้นตอนที่ 9 คำนวณหาจำนวนพื้นที่สีแดงทั้งหมด โดยการนับจำนวนพิกเซลที่เป็นสีแดงจากการกำหนดขอบเขตตำแหน่งอาหารเหลือ



ภาพ 30 แสดงตัวอย่างภาพเม็ดพิกเซล

ขั้นตอนที่ 10 นำข้อมูลพื้นที่สีแดงที่นับได้ ของทั้งภาพถ่ายอาหารก่อนรับประทาน และหลังรับประทานเสร็จ มาเปรียบเทียบ และคำนวณหาเปอร์เซ็นต์พื้นที่ที่หายไป เพื่อให้ทราบถึงปริมาณอาหารที่ได้รับประทานเข้าไป และปริมาณอาหารที่เหลือภายในจาน



ภาพ 31 กระบวนการทำงานโดยรวมของการประเมินปริมาณอาหารเหลือ

3.3 การวิเคราะห์ข้อมูล

การประเมินประสิทธิภาพการประเมินปริมาณอาหารเหลือ ในงานวิจัยนี้ใช้การวัดค่าความผิดพลาดสัมบูรณ์ ค่าความผิดพลาดสัมพันธ์ และค่าความแม่นยำ ซึ่งใช้สูตรคำนวณดังสมการที่ (1) (2) และ (3) ตามลำดับ

3.3.1 ค่าความผิดพลาดสัมบูรณ์ (Absolute Error : e)

ค่าความผิดพลาดสัมบูรณ์ คือ ค่าปริมาณความแตกต่างระหว่างค่าจริงกับค่าที่วัดได้

$$e = x_t - x_m \quad (1)$$

เมื่อ e คือ ค่าความผิดสัมบูรณ์ (Absolute Error)

x_t คือ ค่าจริง (True Value)

x_m คือ ค่าที่วัดได้

3.3.2 ค่าความผิดพลาดสัมพันธ์ (Relative Error : %Error)

ค่าความผิดพลาดสัมพันธ์ คือ ค่าเปอร์เซ็นต์ความผิดพลาด ได้จากค่าความผิดพลาดสัมบูรณ์เปรียบเทียบกับค่าจริง

$$\%Error = \left|\frac{x_t - x_m}{x_t}\right| \times 100\% \quad (2)$$

3.3.3 ค่าความแม่นยำ (Accuracy : %Acc)

ค่าความแม่นยำ คือ ค่าเปอร์เซ็นต์ความถูกต้อง ที่ได้จากค่าที่อ่านได้จากการวัดได้ใกล้เคียงกับค่าจริงมากน้อยเพียงใด

$$\%Acc = 100\% - \%Error \quad (3)$$

บทที่ 4

ผลการวิจัย

ผลการทดลองของงานวิจัยการประเมินปริมาณอาหารเหลือภายในจาน ผู้วิจัยได้ทำการวัดประสิทธิภาพแบบจำลองด้วยการวัดค่าความผิดพลาดสัมบูรณ์ ค่าความผิดพลาดสัมพันธ์ และค่าความแม่นยำ โดยได้พัฒนาการวัดประสิทธิภาพแบบจำลองเพิ่มเติมร่วมกับการประเมินปริมาณอาหารเหลือและจัดเก็บข้อมูลการประเมินปริมาณอาหารเหลือที่วัดได้ในรูปแบบข้อมูลจราจรทางคอมพิวเตอร์ (Log File) ดังตัวอย่างภาพ 33 ซึ่งใช้เป็นสถิติ เพื่อหาจุดบกพร่องของแบบจำลอง และพัฒนาแก้ไขเพิ่มเติมให้แบบจำลองมีความถูกต้องสูงสุด

```
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
Processing 1 images
image                  shape: (3024, 4032, 3)       min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
Processing 1 images
image                  shape: (3024, 4032, 3)       min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
Processing 1 images
image                  shape: (3024, 4032, 3)       min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
Processing 1 images
image                  shape: (3024, 4032, 3)       min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
Processing 1 images
image                  shape: (3024, 4032, 3)       min:    0.00000  max:  255.00000  uint8
molded_images          shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas            shape: (1, 14)               min:    0.00000  max: 4032.00000  float64
anchors                shape: (1, 65472, 4)         min:   -0.35425  max:    1.22900  float32
```

ภาพ 32 ตัวอย่างการทำงานการประเมินปริมาณอาหารเหลือจากรูปภาพ

ภาพ 33 ตัวอย่างการจัดเก็บข้อมูลการประเมินปริมาณอาหารเหลือรูปแบบ Log File

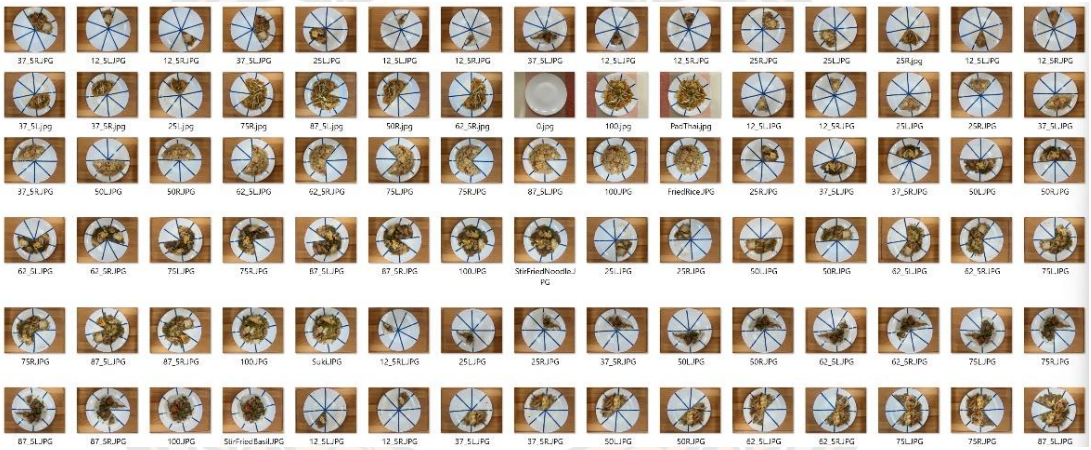
ในงานวิจัยได้พัฒนาแบบจำลอง สำหรับการประเมินปริมาณอาหารเหลือ แบ่งออกเป็น 3 รูปแบบดังนี้

4.1 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1

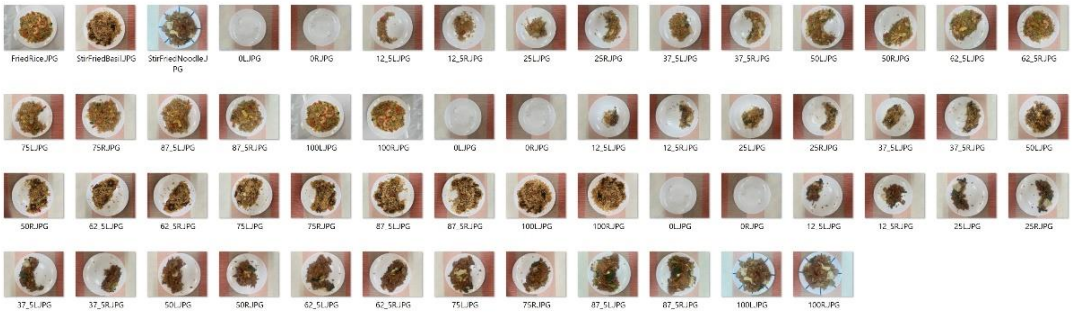
4.2 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2

4.3 การวัดประสิทธิภาพแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

โดยใช้ข้อมูลภาพถ่ายสำหรับการทดสอบแบ่งออกเป็น 2 ชุดข้อมูล ได้แก่ ชุดข้อมูลภาพถ่ายสำหรับทดสอบแบบจำลองแบบเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน ตัวอย่างดังภาพ 34 และชุดข้อมูลภาพถ่ายสำหรับทดสอบแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน จำนวน 54 ภาพ ตัวอย่างดังภาพ 35



ภาพ 34 ตัวอย่างข้อมูลภาพทดสอบแบบจำลองแบบเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน



ภาพ 35 ตัวอย่างข้อมูลภาพทดสอบแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

4.1 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1

การทดสอบประสิทธิภาพการประเมินปริมาณอาหารเหลือ จะใช้ภาพถ่ายอาหารก่อนการรับประทาน และหลังการรับประทานที่เกลี่ยอาหารรวมกันไว้แล้วจากเมนูอาหาร 6 รายการ จำนวน

97 ภาพ ดังภาพ 34 ซึ่งเป็นการทดสอบหาแบบจำลองที่มีความถูกต้องสูงสุดจากการฝึกสอนจำนวน 50 รอบ ผลการทดสอบดังตาราง 1

ตาราง 1 ผลการวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1

Epoch	ร้อยละ		
	Absolute Error	Relative Error	Accuracy
5	33.37	131.06	-31.06
10	33.89	133.35	-33.35
15	30.02	123.28	-23.28
20	27.57	110.75	-10.75
25	17.86	65.73	34.27
30	9.14	26.19	73.81
35	12.05	42.96	57.04
40	7.63	25.73	74.27
45	**4.70**	**12.50**	**87.50**
46	5.61	15.51	84.49
47	5.69	17.42	82.58
48	7.91	13.18	86.82
49	5.44	18.93	81.07
50	6.62	20.08	79.92

จากตารางที่ 1 ผลการประเมินประสิทธิภาพการประเมินปริมาณอาหารเหลือ จากการนำเทคนิค Mask R-CNN มาประยุกต์เพิ่มเติมวิธีการคำนวณพื้นที่ภายในภาพ ให้ประสิทธิภาพสูงสุดด้วยค่าความผิดพลาดสัมบูรณ์เฉลี่ยอยู่ที่ร้อยละ 4.70 ค่าความผิดพลาดสัมพันธ์เฉลี่ยอยู่ที่ร้อยละ 12.50 และค่าความถูกต้องเฉลี่ยอยู่ที่ร้อยละ 87.50 ที่จำนวนการฝึกสอน 45 รอบ

4.2 การวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2

อย่างไรก็ตามจากผลการทดสอบการวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1 ผู้วิจัยพบว่ายังมีข้อบกพร่องที่ส่งผลต่อประสิทธิภาพการประเมินปริมาณอาหารเหลือ คือ ตำแหน่งอาหารเหลือภายในจานภายในภาพถ่ายที่ใช้ในการฝึกสอนแบบจำลองยังมีตำแหน่งมุมองศาของอาหารเหลือภายในจานยังมีความหลากหลายไม่เพียงพอ ดังนั้นผู้วิจัยจึงได้ปรับปรุงรูปแบบการฝึกสอนแบบจำลอง โดยการปรับกระบวนการเสริมข้อมูลที่ถูกเพิ่มเข้าไปในขณะ

ฝึกสอน ในส่วนของขั้นตอนการหมุนภาพให้มีองศาในการหมุนภาพทั้ง 360 องศา เพื่อเพิ่มความหลากหลายของข้อมูลภาพในการฝึกสอนแบบจำลองให้มากขึ้น และกำหนดให้ฝึกสอนจำนวน 500 รอบ ส่วนการทดสอบความถูกต้องของแต่ละแบบจำลองจะใช้ชุดข้อมูลทดสอบเดียวกันกับแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 1 ผลการทดสอบประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2 ดังตาราง 2

ตาราง 2 ผลการวัดประสิทธิภาพแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนแบบที่ 2

Epoch	ร้อยละ		
	Absolute Error	Relative Error	Accuracy
34	7.16	19.18	80.82
39	6.76	18.05	81.95
40	5.95	16.13	83.87
41	8.09	18.95	81.05
63	5.20	14.22	85.78
82	**4.65**	**12.46**	**87.54**
155	5.80	13.64	86.36
169	6.51	14.43	85.57
277	6.13	12.79	87.21
278	5.92	13.43	86.57
279	6.11	13.80	86.20
284	6.30	14.22	85.78
290	6.07	13.75	86.25
291	6.34	14.18	85.82
292	6.57	14.31	85.69
293	5.80	12.76	87.24
345	5.83	13.82	86.18
368	5.71	13.52	86.48
370	6.25	14.34	85.66
371	6.28	14.40	85.60
382	5.98	13.82	86.18
396	5.72	13.74	86.26
450	5.97	13.58	86.42
451	5.83	13.60	86.40

จากผลการทดสอบแบบจำลองตาราง 2 พบว่าเมื่อปรับปรุงการฝึกสอนแบบจำลองให้มีความหลากหลายของมุมมองศาอาหารเหลือภายในจานมากขึ้น การเรียนรู้ของแบบจำลองมีค่าความถูกต้องสูงขึ้นในรอบการเรียนรู้ที่น้อยลง และคงที่มากขึ้น แต่ต้องใช้ระยะเวลา หรือรอบการเรียนรู้ที่มากขึ้นในการสร้างแบบจำลองที่มีค่าความถูกต้องสูงสุด กล่าวได้ว่ายิ่งมีความหลากหลายของชุดข้อมูลภาพสำหรับฝึกสอนแบบจำลองมากเท่าไร การสร้างแบบจำลองจะต้องใช้รอบการเรียนรู้ที่สูงมากยิ่งขึ้นเท่านั้นสำหรับการสร้างแบบจำลองที่มีค่าความถูกต้องสูงสุด

4.3 การวัดประสิทธิภาพแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

การทดสอบประสิทธิภาพการประเมินปริมาณอาหารเหลือ จะใช้ภาพถ่ายอาหารก่อนการรับประทาน และหลังการรับประทานที่ไม่ถูกเกลี่ยรวมกันจากเมนูอาหาร 3 รายการ จำนวน 54 ภาพ เป็นการทดสอบหาแบบจำลองที่มีความถูกต้องสูงสุดจากการฝึกสอนจำนวน 100 รอบ ผลการทดสอบดังตาราง 3

ตาราง 3 ผลการวัดประสิทธิภาพแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

Epoch	ร้อยละ		
	Absolute Error	Relative Error	Accuracy
20	**11.78**	**40.49**	**59.51**
33	11.87	42.21	57.79
40	11.99	42.94	57.06
44	12.06	43.43	56.57
45	11.63	42.65	57.35
53	12.16	43.05	56.95
56	12.08	43.31	56.69
60	12.20	43.55	56.45
61	12.17	43.22	56.78
74	12.11	43.12	56.88
81	12.18	43.51	56.49
83	11.78	42.86	57.14
84	11.50	41.06	58.94
86	11.52	41.51	58.49
88	11.74	41.85	58.15
91	11.85	42.57	57.43
96	11.87	42.43	57.57

ผลการประเมินประสิทธิภาพการประเมินปริมาณอาหารเหลือของแบบจำลองสำหรับอาหารเหลือที่ไม่ต้องเกลี่ยรวมกัน มีประสิทธิภาพสูงสุดอยู่ที่ค่าความผิดพลาดสัมบูรณ์เฉลี่ยร้อยละ 11.78 ค่าความผิดพลาดสัมพันธ์เฉลี่ยร้อยละ 40.49 และค่าความถูกต้องเฉลี่ยร้อยละ 59.51 ที่จำนวนการฝึกสอน 20 รอบ สาเหตุที่แบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน มีความผิดพลาดสูง เนื่องจากอาหารเหลือภายในจานที่ไม่เกลี่ยรวมกันเป็นสัดส่วนมีลักษณะที่ใกล้เคียงกันมาก จึงส่งผลให้แบบจำลองประเมินปริมาณอาหารเหลือภายในจานผิดพลาด ซึ่งเมื่อนำแบบจำลองทั้ง 2 รูปแบบเปรียบเทียบกันดังตาราง 4 จะพบว่าประสิทธิภาพของแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วน มีความถูกต้องสูงกว่าแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน ค่อนข้างมาก ซึ่งกล่าวได้ว่าการการประเมินปริมาณอาหารเหลือ โดยใช้เทคนิค Mask R-CNN ในการสร้างแบบจำลองเหมาะกับการประเมินอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วนมากที่สุด

ตาราง 4 การเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองสำหรับอาหารเหลือที่เกลี่ยรวมกันเป็นสัดส่วน และแบบจำลองที่ไม่ต้องเกลี่ยอาหารเหลือรวมกันเป็นสัดส่วน

แบบจำลอง	Epoch	ร้อยละ		
		Absolute Error	Absolute Error	Absolute Error
แบบเกลี่ยแบบที่ 1	**45**	**4.7**	**12.5**	**87.5**
แบบเกลี่ยแบบที่ 1	46	5.61	15.51	84.49
แบบเกลี่ยแบบที่ 1	47	5.69	17.42	82.58
แบบเกลี่ยแบบที่ 1	48	7.91	13.18	86.82
แบบเกลี่ยแบบที่ 1	49	5.44	18.93	81.07
แบบเกลี่ยแบบที่ 2	**82**	**4.65**	**12.46**	**87.54**
แบบเกลี่ยแบบที่ 2	293	5.80	12.76	87.24
แบบเกลี่ยแบบที่ 2	277	6.13	12.79	87.21
แบบเกลี่ยแบบที่ 2	393	5.96	13.27	86.73
แบบเกลี่ยแบบที่ 2	278	5.92	13.43	86.57
แบบไม่เกลี่ย	**20**	**11.78**	**40.49**	**59.51**
แบบไม่เกลี่ย	84	11.50	41.06	58.94
แบบไม่เกลี่ย	86	11.52	41.51	58.49
แบบไม่เกลี่ย	88	11.74	41.85	58.15
แบบไม่เกลี่ย	33	11.87	42.21	57.79

บทที่ 5

บทสรุป

งานวิจัยฉบับนี้มีวัตถุประสงค์เพื่อพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึก ซึ่งทางผู้วิจัยใช้เทคนิค Mask R-CNN ในการสร้างแบบจำลองสำหรับกำหนดขอบเขตตำแหน่งอาหารเหลือ และใช้วิธีการคำนวณพื้นที่ภายในภาพในการประเมินปริมาณอาหารเหลือ

5.1 สรุปผลการวิจัย

การพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึกพบว่าปัจจัยที่ส่งผลให้แบบจำลองการประเมินปริมาณอาหารเหลือทำงานได้อย่างมีประสิทธิภาพ คือ ชุดข้อมูลภาพสำหรับฝึกสอน โดยในการสอนให้แบบจำลองได้เรียนรู้นั้น ยิ่งภาพถ่ายอาหารเหลือภายในจานมีมุมมองตำแหน่งที่หลากหลาย และภาพถ่ายมีความละเอียดสูงจะยิ่งส่งผลต่อประสิทธิภาพความแม่นยำของแบบจำลอง ซึ่งจากผลการทดลองแสดงให้เห็นว่าการเลือกใช้ภาพถ่ายที่มีความละเอียดสูง มีสัดส่วนของอาหารต่อขนาดภาพไม่ต่ำกว่าร้อยละ 38 ช่วยให้การตรวจจับรายการอาหารหรือกำหนดขอบเขตบริเวณอาหารเหลือมีความแม่นยำสูง ซึ่งผู้วิจัยได้นำงานวิจัยของ รังสรรค์ มนัสมานะชัย และธนภัทร ฆังคะจิตร (2564) ที่ได้วิจัยวิธีการจำแนกรายการอาหาร และตรวจจับส่วนประกอบอาหารจากรูปภาพมาปรับใช้ในการวิจัยนี้ และได้นำงานวิจัยของ Mikolajczyk & Grochowski (2018) ที่วิจัยเกี่ยวกับกระบวนการเสริมข้อมูล เข้ามาช่วยเพิ่มความหลากหลายของชุดข้อมูลในการฝึกสอน ซึ่งช่วยให้ประสิทธิภาพของแบบจำลองมีความถูกต้อง และแม่นยำสูงขึ้นถึงแม้จะมีชุดข้อมูลภาพสำหรับการฝึกสอนไม่มากนัก

ผลการประเมินประสิทธิภาพของแบบจำลองการประเมินปริมาณอาหารเหลือหลังจากการรับประทานด้วยเทคนิคการเรียนรู้เชิงลึกพบว่าแบบจำลองที่พัฒนาด้วยเทคนิค Mask R-CNN มีประสิทธิภาพในการกำหนดขอบเขตตำแหน่งอาหารเหลือได้อย่างแม่นยำ ซึ่งสอดคล้องกับงานวิจัยของ Darapaneni et al. (2021) ที่บ่งชี้ว่าเทคนิค Mask R-CNN มีประสิทธิภาพในกำหนดขอบเขตตำแหน่งอาหาร และผู้วิจัยได้พัฒนาเพิ่มเติมให้สามารถประเมินปริมาณอาหารเหลือจากภาพถ่ายได้ โดยผู้วิจัยได้ดัดแปลงวิธีการคำนวณหาพื้นที่ภายในภาพมาจากงานวิจัยของ Sun et al. (2016) ที่วิจัยเกี่ยวกับการทำนายปริมาณกรดไขมันในเนื้อหมูจากภาพถ่าย จากนั้นผู้วิจัยได้นำเทคนิค และวิธีการที่ได้จากการวิจัยครั้งนี้มาช่วยเพิ่มประสิทธิภาพการประเมินปริมาณอาหารเหลือในงานวิจัยของ

Namahoot et al. (2020) ที่พัฒนาต้นแบบระบบสำหรับการสนับสนุนการวางแผนการรับประทานอาหารในโรงพยาบาล (RES-DIP) ที่จากเดิมใช้การประเมินปริมาณอาหารเหลือหลังจากการรับประทานโดยใช้สมการอย่างง่ายในการวิเคราะห์

จุดเด่นของงานวิจัยครั้งนี้ คือ การประเมินปริมาณอาหารเหลือที่ไม่ได้อาศัยคุณลักษณะเด่นของภาพ หรืออาศัยการจดจำภาพ (image recognition) แต่ใช้การแบ่งภาพออกเป็นส่วน ๆ (image segmentation) และการคำนวณพื้นที่ภายในภาพ ส่งผลให้สามารถประเมินปริมาณอาหารเหลือภายในจานได้ทุกเมนูอาหาร แต่อย่างไรก็ตามจากการวิจัยผู้วิจัยพบว่ายังมีข้อบกพร่องที่ส่งผลต่อประสิทธิภาพการประเมินปริมาณอาหารเหลือ คือ ชุดข้อมูลสำหรับการฝึกสอนที่ยังมีความหลากหลายไม่เพียงพอ และจำนวนรอบการเรียนรู้ที่น้อยเกินไป (อ้างอิงจากงานวิจัยของ นพรัตน์ มาน้อย และคณะ (2562) วิจัยเกี่ยวกับการเรียนรู้วัตถุในอาหาร ซึ่งฝึกสอนแบบจำลองรู้จำวัตถุดิบจำนวน 36,000 รอบ แบบจำลองมีค่าสูญเสียเฉลี่ยอยู่ที่ 0.0408 และมีค่า Precision, Recall และ F1-score อยู่ที่ 0.96, 0.98 และ 0.97 ตามลำดับ) จึงส่งผลให้ค่าความแม่นยำในการประเมินอาหารเหลืออยู่ที่ร้อยละ 88 โดยประมาณ

ประโยชน์ของงานวิจัยที่นำพัฒนาต่อยอดเพิ่มเติม ตัวอย่างเช่น 1) สำหรับบุคคลทั่วไป สามารถไปนำช่วยเรื่องการคำนวณแคลอลี่จากอาหารที่รับประทานในแต่ละมื้อที่คำนวณจากปริมาณอาหารที่รับประทานเข้าไปจริง เพื่อแนะนำเมนูอาหารที่เหมาะสมได้อย่างถูกต้อง 2) สำหรับโรงเรียน สามารถนำไปช่วยในการประเมินปริมาณอาหารเหลือจากการรับประทานของเด็กนักเรียนในโครงการอาหารกลางวัน เพื่อเก็บข้อมูลสถิติเมนูอาหารที่เด็กนักเรียนรับประทานเหลือน้อยที่สุด และเหลือมากที่สุด มาใช้สำหรับแนะนำเมนูอาหารให้แก่ผู้จัดทำอาหารในโครงการ ซึ่งช่วยลดต้นทุนการทำอาหารและลดปริมาณขยะจากเศษอาหารภายในโรงเรียน ผลประโยชน์ทางอ้อม คือ ช่วยให้เด็กนักเรียนได้รับปริมาณสารอาหารที่เพียงพอจากการรับประทานอาหารได้เยอะขึ้น 3) สำหรับโรงพยาบาล สามารถนำไปช่วยในการตรวจสอบการรับประทานอาหารของผู้ป่วย ว่าปริมาณการรับประทานของผู้ป่วยในแต่ละมื้อ และโดยรวมต่อวันได้รับปริมาณสารอาหาร และแคลอรี่ เพียงพอหรือไม่ และเก็บข้อมูลเมนูอาหารที่ผู้ป่วยรับประทานหมด หรือรับประทานปริมาณเยอะที่สุด เพื่อใช้สำหรับแนะนำเมนูอาหารที่เหมาะสมในมื้อถัดไปแก่นักโภชนาการ ซึ่งส่งผลให้สามารถลดปริมาณอาหารเหลือหรือมลพิษจากขยะเศษอาหาร และช่วยควบคุมต้นทุนในการจัดหาวัตถุดิบในการประกอบอาหารให้แก่ผู้ป่วย รวมถึงการช่วยลดต้นทุนด้านการกำจัดขยะภายในโรงพยาบาล

5.2 ข้อเสนอแนะ

1. พัฒนาแบบจำลองให้สามารถประเมินปริมาณอาหารเหลือจากภาชนะอื่น ๆ เช่น ถ้วย ชาม กล่องโฟม ถาดอาหารนักเรียน และถาดข้าวโรงพยาบาล เป็นต้น

2. พัฒนาวิธีการประเมินความลึกของอาหารภายในภาชนะ เพื่อให้ได้ผลการประเมินปริมาณอาหารเหลือที่แม่นยำมากยิ่งขึ้น

3. พัฒนาการประเมินคุณค่าทางโภชนาอาหาร เพื่อแนะนำเมนูอาหารที่เหมาะสมต่อผู้บริโภคจากการประเมินปริมาณอาหารเหลือ

4. การพัฒนาแบบจำลองควรใช้ระบบคลาวด์เข้ามาช่วยในการประมวล เพื่อความรวดเร็วต่อการพัฒนาแบบจำลอง เนื่องจากหากใช้เครื่องคอมพิวเตอร์ทั่วไปในการพัฒนาต้องใช้เวลาค่อนข้างมาก จำนวนหลายวันกว่าจะได้แบบจำลองตามที่ต้องการ (สำหรับงานวิจัยนี้ใช้เวลาการเรียนรู้ของแบบจำลอง 100 รอบต่อ 1 วัน)

5. สำหรับการพัฒนาแบบจำลองการประเมินปริมาณอาหารเหลือภายในโรงเรียน และโรงพยาบาล แนะนำควรใช้ภาชนะสำหรับใส่อาหารเป็นแบบถาดหลุม และใช้เครื่องชั่งน้ำหนัก เข้ามาช่วยในการวัดปริมาณอาหาร เพื่อการคำนวณปริมาณอาหารเหลือ และการคำนวณคุณค่าทางโภชนาการอาหารที่แม่นยำมากยิ่งขึ้น

6. พัฒนาแบบจำลองให้สามารถตรวจจับวัตถุดิบภายในอาหาร เพื่อใช้ในการคำนวณคุณค่าทางโภชนาการอาหารที่ถูกต้องตามปริมาณวัตถุดิบภายในอาหารที่ใช้จริง

7. พัฒนาการถ่ายรูปภาพอาหารภายในจานให้มีสัดส่วนจานต่อขนาดภาพทั้งภาพอาหารก่อนการรับประทาน และหลังรับประทานเสร็จ ให้มีขนาดที่เท่ากัน หรือใกล้เคียงกันมากที่สุด เพื่อการประเมินปริมาณอาหารเหลือได้ถูกต้องแม่นยำ ตัวอย่างดังภาพ 36

ภาพ 36 ตัวอย่างการพัฒนาการถ่ายรูปภาพอาหารภายในจาน

บรรณานุกรม

Chayatummagoon, S., & Chongstitvatana, P. (2021). Image classification of sugar crystal with deep learning. 2021 13th International Conference on Knowledge and Smart Technology (KST),

Darapaneni, N., Singh, V., Tarkar, Y. S., Kataria, S., Bansal, N., Kharade, A., & Paduri, A. R. (2021). Food Image Recognition and Calorie Prediction. 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS),

Kelleher, J. D. (2019). *Deep learning*. MIT press.

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW),

Namahoot, C. S., Brückner, M., & Nuntawong, C. (2020). A RECOMMENDER SYSTEM SUPPORTING DIET PLANNING IN HOSPITALS (RES-DIP). *ICIC Express Letters*, *15*(6), 585–594.

Pawara, P., Okafor, E., Surinta, O., Schomaker, L., & Wiering, M. A. (2017). Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition. *ICPRAM*, *479*(2017), 486.

Sun, X., Newman, D., Young, J., Zhang, Y., & Berg, E. (2016). Prediction of Pork Fatty Acid Content using Image Texture Features. *Advance Journal of Food Science and Technology*, *12*(11), 644-647.

Vento, D. D., & Fanfarillo, A. (2019). Traps, pitfalls and misconceptions of machine learning applied to Scientific Disciplines. In *Proceedings of the practice and experience in advanced research computing on rise of the machines (learning)* (pp. 1-8).

Widjaja, I., Leow, W. K., & Wu, F.-C. (2003). Identifying painters from color profiles of skin patches in painting images. Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429),

กรุงเทพธุรกิจ. (2565). โรคอ้วน กระทบเศรษฐกิจ 13.2% ของงบประมาณสาธารณสุขทั่วโลก. Retrieved 4 มีนาคม 2565 from https://www.bangkokbiznews.com/social/991651

คลีนิค BDMS Wellness. (2565). 10 ความเชื่อผิด ๆ เกี่ยวกับการลดความอ้วนที่ควรรู้. Retrieved 25 สิงหาคม 2565 from https://www.pptvhd36.com/health/care/678

นพรัตน์ มาน้อย, อำพล บุญจันดา, ชูพันธุ์ รัตนโภคา. (2562). ระบบแบ่งปันสูตรการทำอาหารและค้นหาสูตรการทำอาหารจากภาพวัตถุดิบด้วยเทคนิคการเรียนรู้เชิงลึก. วารสารวิชาการเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, *15*(2), 97-111.

รังสรรค์ มนัสมานะชัย, ธนภัทร ฆังคะจิตร. (2564). การจำแนกรายการอาหารและตรวจจับส่วนประกอบจากรูปภาพอาหาร ด้วยเทคนิคการเรียนรู้เชิงลึก. วารสารบัณฑิตวิทยาลัย มหาวิทยาลัยธุรกิจบัณฑิตย์, *10*(2), 80-93.

ภาคผนวก

ภาคผนวก ก

ซอร์สโค้ด (source code)

ซอร์สโค้ดการสร้างแบบจำลองกำหนดขอบเขตตำแหน่งอาหารเหลือ

```
import tensorflow as tf
import os
os.environ["CUDA_DEVICE_ORDER"]="0000:01:00.0"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
ROOT_DIR = os.path.abspath("../MaskRCNN_Leftover_Food_Train")

import sys
import json
import numpy as np
import time
from PIL import Image, ImageDraw
import skimage.draw
import imgaug as ia
from imgaug import augmenters as iaa
import matplotlib.pyplot as plt
import pandas as pd
from mrcnn.visualize import display_instances
from mrcnn.utils import extract_bboxes
sys.path.append(ROOT_DIR)
from mrcnn.config import Config
import mrcnn.utils as utils
from mrcnn import visualize
import mrcnn.model as modellib
from mrcnn.model import log
```

ตั้งค่าโฟลเดอร์ที่เก็บไลบรารี่ และนำเข้าไลบรารี่

```
leaf_DIR = os.path.join(ROOT_DIR, "dataSet")
MODEL_DIR = os.path.join(ROOT_DIR, "logs")
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "mask_rcnn_coco.h5")
if not os.path.exists(COCO_MODEL_PATH):
    utils.download_trained_weights(COCO_MODEL_PATH)
ROOT_DIR = os.getcwd()
if ROOT_DIR.endswith("../MaskRCNN_Leftover_Food_Train"):
    ROOT_DIR = os.path.dirname(os.path.dirname(ROOT_DIR))
```

ตั้งค่าโฟลเดอร์ที่เก็บชุดข้อมูลสำหรับฝึกสอนแบบจำลอง และโฟลเดอร์เก็บแบบจำลองที่สร้างเสร็จ

ตั้งค่าโฟลเดอร์และโหลดชุดข้อมูล COCO เพื่อนำมาใช้ในการ pre-trained กับแบบจำลองที่สร้างขึ้น

```
class leftoverFoodConfig(Config): #class เพื่อทำตั้งค่าใช้สำหรับการฝึกสอน
        NAME = "leftoverFood“ #ชื่อแบบจำลอง
        GPU_COUNT = 1
        IMAGES_PER_GPU = 1      } กำหนดจำนวน GPU และจำนวนภาพที่ใช้ต่อ 1 GPU
        NUM_CLASSES = 1 + 1  #กำหนดจำนวน class วัตถุ + พื้นหลัง
        STEPS_PER_EPOCH = 60 #กำหนดจำนวนขั้นตอนการเรียนรู้ต่อ 1 รอบการเรียนรู้
        VALIDATION_STEPS = 10  #กำหนดจำนวนขั้นตอนการตรวจสอบต่อ 1 รอบการเรียนรู้
        BACKBONE = "resnet101“ #กำหนดโมเดลที่ใช้ในการฝึกสอน
        BATCH_SIZE = 2   #แบ่ง data เป็นกลุ่มเล็กลง
        DETECTION_MIN_CONFIDENCE = 0.7 #ตั้งค่าขนาดวัตถุที่ mask
        DETECTION_NMS_THRESHOLD = 0.3 #ตั้งค่าขนาดของวัตถุที่ค้นหาในภาพ
        IMAGE_MAX_DIM = 512 #ขนาดความกว้างและยาวของรูปสูงสุด
        IMAGE_MIN_DIM = 512  #ขนาดความกว้างและยาวของรูปต่ำสุด
        IMAGE_RESIZE_MODE = “square” #ตั้งค่าภาพที่แสดงผลเป็นแบบสี่เหลี่ยม
        LEARNING_MOMENTUM = 0.9 #กำหนดค่าโมเมนตัม
       LEARNING_RATE = 0.001 #กำหนดอัตราการเรียนรู้ของโมเดล
       #กำหนดค่าพื้นในการกำหนดขอบเขต
        ROI_POSITIVE_RATIO = 0.33
        RPN_ANCHOR_RATIOS =  [0.5, 1, 2]
        RPN_ANCHOR_SCALES=  (16, 32, 64, 128, 256)
        RPN_NMS_THRESHOLD = 0.7
        RPN_TRAIN_ANCHORS_PER_IMAGE = 512
        TRAIN_ROIS_PER_IMAGE = 320
        WEIGHT_DECAY = 0.001 #กำหนดน้ำหนักลดความซับซ้อนของโมเดลและ
                                ป้องกันการโอเวอร์ฟิต
    config = leftoverFoodConfig()
    config.display()

    class leftoverFoodDataset(utils.Dataset):        # class เพื่อโหลดชุดข้อมูลภาพ และข้อมูล
        def load_data(self, dataset_dir, subset):        ตำแหน่งอาหารเหลือจากไฟล์ JSON
            self.add_class("leftoverFood", 1, "leftoverFood")
            assert subset in ["train", "val"]
            dataset_dir = os.path.join(dataset_dir, subset)
            annotations = json.load(open(os.path.join(dataset_dir,
                                "via_region_data.json")))
            annotations = list(annotations.values())
```

```
        annotations = [a for a in annotations if a['regions']]
        for a in annotations:
            if type(a['regions']) is dict:
                polygons = [r['shape_attributes'] for r in a['regions'].values()]
            else:
                polygons = [r['shape_attributes'] for r in a['regions']]

            image_path = os.path.join(dataset_dir, a['filename'])
            image = skimage.io.imread(image_path)
            height, width = image.shape[:2]
            self.add_image(
                "leftoverFood",
                image_id=a['filename'],
                path=image_path,
                width=width, height=height,
                polygons=polygons)

    def load_mask(self, image_id):
        image_info = self.image_info[image_id]
        if image_info["source"] != "leftoverFood":
            return super(self.__class__, self).load_mask(image_id)
        info = self.image_info[image_id]
        mask = np.zeros([info["height"], info["width"], len(info["polygons"])],
                        dtype=np.uint8)
        for i, p in enumerate(info["polygons"]):
            rr, cc = skimage.draw.polygon(p['all_points_y'], p['all_points_x'])
            mask[rr, cc, i] = 1
        return mask.astype(np.bool), np.ones([mask.shape[-1]], dtype=np.int32)
annotations = json.load(open(leaf_DIR + '/val/via_region_data.json'))
dataset_train = leftoverFoodDataset() #โหลดข้อมูลฝึกสอน
dataset_train.load_data(leaf_DIR, 'train')
dataset_train.prepare()

dataset_val = leftoverFoodDataset() #โหลดข้อมูลทดสอบ
dataset_val.load_data(leaf_DIR, 'val')
dataset_val.prepare()
```

```
            print("Image Count: {}".format(len(dataset_train.image_ids)))
            print("Class Count: {}".format(dataset_train.num_classes))
            for i, info in enumerate(dataset_train.class_info):  #แสดงตัวอย่างการกำหนดขอบเขตอาหาร
                print("{:3}. {:50}".format(i, info['name']))            เหลือ
            dataset = dataset_train
            image_ids = np.random.choice(dataset.image_ids, 5)
            for image_id in image_ids:
                image = dataset.load_image(image_id)
                mask, class_ids = dataset.load_mask(image_id)
                visualize.display_top_masks(image, mask, class_ids, dataset.class_names)
            image_id = 1
image = dataset_val.load_image(image_id)
mask, class_ids = dataset_val.load_mask(image_id)
bbox = extract_bboxes(mask)

#โหลดค่า Weight จากชุดข้อมูล COCO เพื่อนำมาใช้ในการสร้างแบบจำลอง
model = modellib.MaskRCNN(mode="training", config=config,model_dir=MODEL_DIR)
    model.load_weights(COCO_MODEL_PATH, by_name=True, exclude=["mrcnn_class_logits",
"mrcnn_bbox_fc","mrcnn_bbox", "mrcnn_mask"])

#เพิ่มการเสริมข้อมูล (Data augmentation) เข้าไปในกระบวนการฝึกสอน
augmentation = iaa.SomeOf((0, 3), [
    iaa.Fliplr(0.5),
    iaa.Flipud(0.5),
    iaa.OneOf([iaa.Affine(rotate=5),
               iaa.Affine(rotate=90),
               iaa.Affine(rotate=180),
               iaa.Affine(rotate=270)],
              ),
    iaa.Affine(scale={"x": (0.8, 1.2), "y": (0.8, 1.2)}),
    iaa.Multiply((0.8, 1.5)),
    iaa.GaussianBlur(sigma=(0.0, 5.0))
])
```

```
import warnings
#แจ้งเตือนให้เพิกเฉย
warnings.filterwarnings('ignore')
DEVICE = "/gpu:0"

#เวลาที่เริ่มการฝึกสอน
start_train = time.time()

print("Training network ALL")

#สร้างแบบจำลอง
model.train(dataset_train, dataset_val,learning_rate=config.LEARNING_RATE,epochs=100,
            augmentation=augmentation,layers='all')
history = model.keras_model.history.history
```

ซอร์สโค้ดการประเมินปริมาณอาหารเหลือ และวัดประสิทธิภาพการประเมินปริมาณอาหารเหลือ

```
import os
import cv2
import sys
import random
import numpy as np
import skimage
import skimage.io
import matplotlib.pyplot as plt
lower_range = np.array([0,50,50])
upper_range = np.array([10,255,255])
text_replace = { "_": ".", "L.jpg": "", "R.jpg": "", "L.JPG": "",
                 "R.JPG": '', ".JPG": "", ".jpg": "", "R": "", "L": ""}
class_names = ['BG', 'leftoverFood']
percent = 0.00
num_food=0
filenames_befor = ""
real_test_dir = ""
result_befor=""
result_after=""
```

กำหนดช่วงสีของสีแดง

```
name_file=""
models=""
ans=""
ROOT_DIR = os.path.abspath("../MaskRCNN_Leftover_Food_Mask")
sys.path.append(ROOT_DIR)
from mrcnn.config import Config
from mrcnn import utils
import mrcnn.model as modellib
from mrcnn import visualize
sys.path.append(os.path.join(ROOT_DIR, "\coco"))
import coco
MODEL_DIR = os.path.join(ROOT_DIR, "logs")
COCO_MODEL_PATH = os.path.join(ROOT_DIR,
                               "mask_rcnn_leftoverfood_00082.h5")
models="82"

class CocoConfig(Config): #class เพื่อทำตั้งค่าใช้สำหรับการ Mask
    NAME = "leftoverFood“ #ชื่อแบบจำลอง
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1      กำหนดจำนวน GPU และจำนวนภาพที่ใช้ต่อ 1 GPU
    NUM_CLASSES = 1 + 1  #กำหนดจำนวน class วัตถุ + พื้นหลัง
    STEPS_PER_EPOCH = 60 #กำหนดจำนวนขั้นตอนการเรียนรู้ต่อ 1 รอบการเรียนรู้
    VALIDATION_STEPS = 10  #กำหนดจำนวนขั้นตอนการตรวจสอบต่อ 1 รอบการเรียนรู้
    BACKBONE = "resnet101“ #กำหนดโมเดลที่ใช้ในการฝึกสอน
    BATCH_SIZE = 2   #แบ่ง data เป็นกลุ่มเล็กลง
    DETECTION_MIN_CONFIDENCE = 0.7 #ตั้งค่าขนาดวัตถุที่ mask
    DETECTION_NMS_THRESHOLD = 0.3 #ตั้งค่าขนาดของวัตถุที่ค้นหาในภาพ
    IMAGE_MAX_DIM = 512 #ขนาดความกว้างและยาวของรูปสูงสุด
    IMAGE_MIN_DIM = 512  #ขนาดความกว้างและยาวของรูปต่ำสุด
    IMAGE_RESIZE_MODE = “square” #ตั้งค่าภาพที่แสดงผลเป็นแบบสี่เหลี่ยม
    LEARNING_MOMENTUM = 0.9 #กำหนดค่าโมเมนตัม
   LEARNING_RATE = 0.001 #กำหนดอัตราการเรียนรู้ของโมเดล
   #กำหนดค่าพื้นในการกำหนดขอบเขต
    ROI_POSITIVE_RATIO = 0.33
    RPN_ANCHOR_RATIOS =  [0.5, 1, 2]
    RPN_ANCHOR_SCALES=  (16, 32, 64, 128, 256)
```

```
    RPN_NMS_THRESHOLD = 0.7
    RPN_TRAIN_ANCHORS_PER_IMAGE = 512
    TRAIN_ROIS_PER_IMAGE = 320
    WEIGHT_DECAY = 0.001 #กำหนดน้ำหนักลดความซับซ้อนของโมเดล และป้องกันการโอเวอร์ฟิต

class InferenceConfig(CocoConfig): # class กำหนดค่าการ mask
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1

def red(r, g, b): #ฟังก์ชันตรวจสอบพิกเซลสีแดง
    threshold = max(r, g, b)
    return (threshold > 8 and r == threshold and g < threshold*0.5 and b <
            threshold*0.5)

def processLog(filename):  #ฟังก์ชันคำนวณพื้นที่สีแดง
    imageRGB = cv2.cvtColor(filename, cv2.COLOR_BGR2RGB)
    imnp = np.array(imageRGB)
    h, w = imnp.shape[:2]
    #print(f"{h}  {w}")
    colours, counts = np.unique(imnp.reshape(-1,3), axis=0, return_counts=1)
    #print("counts: ", counts)
    #print("colours: ", colours)
    for index, colour in enumerate(colours):
        count = counts[index]
        proportion = (100 * count) / (h * w)
        if(red(colour[0], colour[1], colour[2]) and proportion > 0.50):
            return proportion
    return 0

def replace_all(text, dic): #ฟังก์ชันแทนที่ข้อความ
    for i, j in dic.items():
        text = text.replace(i, j)
    return text
```

```
#แสดงการตั้งค่าต่าง ๆ
config = InferenceConfig()
config.display()

#โหลดแบบจำลองที่ใช้ในการ mask
model = modellib.MaskRCNN(mode="inference", model_dir=MODEL_DIR, config=config)
model.load_weights(COCO_MODEL_PATH, by_name=True,exclude=[ "mrcnn_class_logits",
                                "mrcnn_bbox_fc", "mrcnn_bbox", "mrcnn_mask"])

#โหลดรูปที่ใช้ทดสอบประกอบด้วยภาพก่อนรับประทาน และหลังรับประทาน
ans=""
average_error_all = 0.0
absolute_all = 0.0
num_food=6
for selectFood in range(num_food):
    if selectFood==0:
        filenames_befor = "image/befor/FriedRice.JPG"
        real_test_dir = "image/after/FriedRice/"
        name_file="FriedRice"
        ans += f"\n\n ################## START : {name_file} ####################"
    elif selectFood==1:
        filenames_befor = "image/befor/PadThai.JPG"
        real_test_dir = "image/after/PadThai/"
        name_file="PadThai"
        ans += f"\n\n ################## START : {name_file} ####################"
elif selectFood==2:
        filenames_befor = "image/befor/RoastedSeafood.JPG"
        real_test_dir = "image/after/RoastedSeafood/"
        name_file="RoastedSeafood"
        ans += f"\n\n ################## START : {name_file} ####################"
    elif selectFood==3:
        filenames_befor = "image/befor/StirFriedBasil.JPG"
        real_test_dir = "image/after/StirFriedBasil/"
        name_file="StirFriedBasil"
        ans += f"\n\n ################## START : {name_file} ####################"
    elif selectFood==4:
```

```
        filenames_befor = "image/befor/StirFriedNoodle.JPG"
        real_test_dir = "image/after/StirFriedNoodle/"
        name_file="StirFriedNoodle"
        ans += f"\n\n ################## START : {name_file} ####################"
    elif selectFood==5:
        filenames_befor = "image/befor/Suki.JPG"
        real_test_dir = "image/after/Suki/"
        name_file="Suki"
        ans += f"\n\n ################## START : {name_file} ####################"

# mask ตำแหน่งอาหารภายในจานก่อนการรับประทาน
    image = skimage.io.imread(filenames_befor)
    results = model.detect([image], verbose=1)
    r = results[0]
    pil_img = visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
class_names, scores=None, show_bbox=False, show_mask=True, title="Predictions")

#แปลงภาพให้อยู่ในโหมด HSV
    cv2_img = np.array(pil_img)
    cv2_img = cv2.cvtColor(cv2_img, cv2.COLOR_RGB2BGR)
    r = 1024.0 / cv2_img.shape[1]
    dim = (1024, int(cv2_img.shape[0] * r))
    resized = cv2.resize(cv2_img, dim, interpolation=cv2.INTER_AREA)
    hsv = cv2.cvtColor(resized, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, lower_range, upper_range) #ค้นหาวัตถุในภาพที่อยู่ในช่วงสีแดง

#วาดกรอบครอบวัตถุที่เป็นสีแดง
    contours, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
    cv2.drawContours(resized, contours, -1, (0, 0, 0), 3)
    imgout = np.zeros_like(resized)
    imgout[mask == 255] = resized[mask == 255]

#คำนวณหาพื้นที่สีแดงภายในภาพ
    result_befor=processLog(imgout)
    print(result_befor)
```

```
# mask ตำแหน่งอาหารภายในจานหลังการรับประทาน
    filename=""
    image_paths = []
    for filename in os.listdir(real_test_dir):
        if os.path.splitext(filename)[1].lower() in ['.png', '.jpg', '.jpeg']:
            image_paths.append(os.path.join(real_test_dir, filename))
    i=0
    average_error = 0.0
    absolute_error = 0.0
    for image_path in image_paths:
        file_name = os.path.basename(image_path)
        image = skimage.io.imread(image_path)
        results = model.detect([image], verbose=1)
        r = results[0]
        pil_img = visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
class_names, scores=None, show_bbox=False, show_mask=True, title="Predictions")
        cv2_img = np.array(pil_img)
        cv2_img = cv2.cvtColor(cv2_img, cv2.COLOR_RGB2BGR)
        r = 1024.0 / cv2_img.shape[1]
        dim = (1024, int(cv2_img.shape[0] * r))
        resized = cv2.resize(cv2_img, dim, interpolation=cv2.INTER_AREA)
        hsv = cv2.cvtColor(resized, cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(hsv, lower_range, upper_range)
        contours, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL,
                                               cv2.CHAIN_APPROX_NONE)
        cv2.drawContours(resized, contours, -1, (0, 0, 0), 3)
        imgout = np.zeros_like(resized)
        imgout[mask == 255] = resized[mask == 255]
        result_after=processLog(imgout)

#คำนวณค่าร้อยละความถูกต้องเฉลี่ยของแบบจำลอง
        if result_befor > result_after:
            percent = round((result_after / result_befor) * 100, 2)
        else:
            percent = round((result_befor / result_after) * 100, 2)
        text=replace_all(file_name, text_replace)
```

```
        ans += "\n Real Leftover Food " + file_name.replace("_",".") + "%"+ f'\n {i+1}.Leftover
Food : {percent:.2f} % \n'
        if text=="0":
            absolute_error += percent
            average_error += percent
            ans += f" %error rate {percent:.2f} \n"
        else:
            absolute_error += (abs(float(text) - float(percent)))
            average_error += (abs(float(percent) - float(text))/float(text)*100)
            ans += f" %error rate {(abs(float(percent) - float(text))/float(text)*100):.2f} \n"
            ans += f" %e rate {(abs(float(text) - float(percent))):.2f} \n"
        i+=1
    absolute_error = float(absolute_error) / int(i)
    average_error = float(average_error) / int(i)
    absolute_all = float(absolute_all) + float(absolute_error)
    average_error_all = float(average_error_all) + float(average_error)
    ans += f"\n\n %error all {name_file} model {models} : {average_error:.2f}\n"
    ans += f"%e all {name_file} model {models} : {absolute_error:.2f}\n\n"
    ans += f"#################### END : {name_file} ####################"
absolute_all = float(absolute_all) / int(num_food)
average_error_all = float(average_error_all) / int(num_food)
ans += f" \n\n average %error all in Model {models} : {average_error_all:.2f} \n"
ans += f" average e all in Model {models} : {absolute_all:.2f} \n"
name_file = "/mask_rcnn_leftoverfood_00"+ str(models) + '.txt'
with open (MODEL_DIR + name_file,'w') as f:
    f.write(ans)
print(f" \n average %error all in Model {models} : {average_error_all:.2f} \n")
```