



SaapFood Cam: แอปพลิเคชันการรู้จำอาหารไทยอีสาน โดยภาพถ่ายจากอุปกรณ์พกพา

SaapFood Cam: A Thai-Isan Food Recognition Mobile Application

นางสาววิมนศิณี ยวนใจ

Wimonsinee Yuanjai

นางสาวสลิลทิพย์ นิลประยูร

Salintip Ninprayoon

นางสาวฉัตรวี สังข์ทอง

Chatrawee Sangtong

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2562



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ SaapFood Cam: แอปพลิเคชันการรู้จำอาหารไทยอีสาน โดย
ภาพถ่ายจากอุปกรณ์พกพา

SaapFood Cam: A Thai-Isan Food Recognition Mobile
Application

นิสิต	นางสาววิมลศิริณี	ยวนใจ	59102010286
	นางสาวสลิทธิพิทย์	นิลประยูร	59102010292
	นางสาวฉัตรวี	สังข์ทอง	59102010564

ปริญญา วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ ผู้ช่วยศาสตราจารย์ นุวิทย์ วิวัฒน์วัฒนา

ลงชื่อ.....

(ผู้ช่วยศาสตราจารย์ นุวิทย์ วิวัฒน์วัฒนา)

อาจารย์ที่ปรึกษาโครงการ

บทคัดย่อ

งานวิจัยเรื่องนี้มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันที่มีชื่อว่า SaapFood Cam ที่สามารถใช้ในการรู้จำอาหารไทยอีสานในชีวิตประจำวันจากภาพถ่ายโดยโทรศัพท์มือถือบนระบบปฏิบัติการแอนดรอยด์ (Android) ได้ เนื่องจากในปัจจุบันการรับประทานอาหารนั้นถือได้ว่าเป็นปัจจัยที่ส่งผลโดยตรงต่อสุขภาพของมนุษย์ ทางผู้วิจัยจึงได้นำเทคนิคการทำ Food Recognition โดยเทคนิคการเรียนรู้ของเครื่อง Deep Learning และ Convolutional Neural Network (CNN) มาช่วยในการสร้างแบบจำลองเพื่อทำนายรูปภาพอาหาร ผู้วิจัยใช้ฐานข้อมูลที่ประกอบไปด้วยอาหารอีสานจำนวน 44 อย่าง ชื่อว่า ISANTHFOOD-44 ในการฝึกฝนแบบจำลอง โดยรวบรวมรูปภาพอาหารมาจากโครงการมณี มหาวิทยาลัยขอนแก่นและจาก search engine ผู้ใช้งานสามารถถ่ายรูปอาหารจากโทรศัพท์มือถือผ่านแอปพลิเคชันเพื่อให้แอปพลิเคชันนำรูปภาพไปประมวลผลกับแบบจำลองและระบุว่าเป็นอาหารชนิดใด ผลการฝึกฝนและทดลองของงานวิจัยนี้พบว่าจากการฝึกฝนทั้งหมด 3 แบบจำลอง ประกอบด้วย InceptionV3 ResNet50 และ MobileNetV2 แบบจำลอง ResNet50 ให้ค่าความแม่นยำที่ดีที่สุด (Top1-accuracy) อยู่ที่ 97.25% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) ที่ดีที่สุดอยู่ที่ 99.04% แบบจำลอง InceptionV3 ให้ค่าความแม่นยำ (Top1-accuracy) อยู่ที่ 97.23% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) อยู่ที่ 98.93% และแบบจำลองสุดท้าย MobileNetV2 ให้ค่าความแม่นยำ (Top1-accuracy) อยู่ที่ 96.88% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) อยู่ที่ 98.86% โดยงานวิจัยนี้สามารถนำไปต่อยอดในอนาคตได้ เช่น การนำชื่ออาหารที่ได้ไปแสดงรายละเอียดต่างๆ ให้แก่ผู้ใช้ได้ทราบข้อมูล อาทิ แคลอรี่ของอาหารระดับความเค็มหรือความหวานโดยทั่วไปของอาหาร เป็นต้น และยังสามารถต่อยอดไปถึงการตรวจจับความเค็มโดยดูจากสีอาหารได้ในอนาคตอีกด้วย

Keywords: Convolutional Neural Networks, Deep Learning, Food Recognition, Smartphone, Thai-Isan food

Abstract

Nowadays, one of the important factors that directly affects human health is food consumption. The main purpose of this research is to develop an Android application called “SaapFood Cam” which recognizes Northeastern Thai (Isan) food images in the daily life. Deep Learning and Convolutional Neural Network (CNN) were used for building the food image recognition models. A food database containing 44 kinds of famous Thai-Isan food called “ISANTHFOOD-44” was used for training and testing processes. These images were collected from two sources, Khon Kaen University's Manee project, and the search engine. Users can take a photo of food from the application and then the food image will be processed and the name of that food predicted will be shown on the screen. Three deep convolutional neural network architectures were trained on ISANTHFOOD-44 dataset, including InceptionV3, ResNet50, and MobileNetV2. ResNet50 outperformed the other two with the top-1 accuracy of 97.25 percent and the top-5 accuracy of 99.04 percent on the test dataset. InceptionV3 achieved 97.23 percent top-1 accuracy and 98.93 percent top-5 accuracy, and MobileNetV2 achieved 96.88 percent top-1 accuracy and 98.86 percent top-5 accuracy. For future research, this work can be extended in several ways, for example, showing the amount of calories or degrees of saltiness or sweetness from food images, possibly by colors of the food or proportions of ingredients.

Keywords: Convolutional Neural Networks, Deep Learning, Food Recognition, Smartphone, Thai-Isan food

กิตติกรรมประกาศ

ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ สำหรับเครื่องมือ เครื่องใช้ และอุปกรณ์ต่าง ๆ ที่ใช้ในการดำเนินงานในครั้งนี้ รวมถึงสถานที่ที่ใช้ในการทำงาน ใช้ทดสอบอุปกรณ์ต่าง ๆ และขอขอบพระคุณอาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ทุกท่าน ที่ได้ให้ความรู้ ให้คำปรึกษาและการช่วยเหลือ ให้ความสนับสนุนในการดำเนินโครงการในครั้งนี้จนสามารถจัดทำโครงการได้ดีและมีประสิทธิภาพมากยิ่งขึ้น

ขอขอบคุณมหาวิทยาลัยศรีนครินทรวิโรฒ ที่อำนวยความสะดวกในเรื่องของสถานที่สำหรับดำเนินโครงการ และเป็นแหล่งศึกษาหาความรู้เพิ่มเติมเพื่อนำมาความรู้ที่ได้จากการศึกษามาพัฒนาระบบงาน และปรับปรุงระบบงานให้ดียิ่งขึ้น

สุดท้ายนี้ ขอขอบพระคุณทุกท่านที่มีส่วนเกี่ยวข้องในการดำเนินโครงการในครั้งนี้ ทั้งที่คอยให้คำแนะนำ ให้คำปรึกษา และให้การช่วยเหลือในเรื่องต่างๆ จนโครงการสำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำหวังว่าโครงการพัฒนาแอปพลิเคชัน SaapFood Cam: แอปพลิเคชันการรู้จำอาหารไทยอีสาน โดยภาพถ่ายจากอุปกรณ์พกพา (SaapFood Cam: A Thai-Isan Food Recognition Mobile Application) จะเป็นประโยชน์ต่อการศึกษ เพื่อที่จะสามารถนำไปพัฒนาต่อยอดในอนาคตได้ดียิ่งๆขึ้นไป

สารบัญ

หัวข้อ	หน้า
บทคัดย่อ	ข
Abstract	ค
กิตติกรรมประกาศ	ง
สารบัญ.....	จ
สารบัญรูปภาพ.....	ช
สารบัญตาราง	ฅ
บทที่ 1	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่2.....	5
2.1 การเรียนรู้ของเครื่อง Deep Learning.....	5
2.2 Convolutional Neural Network (CNN).....	6
2.3 Classification	7
2.4 Network Architecture	8
2.4.1 GoogLeNet	8
2.4.2 ResNet.....	9
2.4.3 MobileNet	10
2.5 Data Augmentation	10
2.6 งานวิจัยที่เกี่ยวข้อง.....	12

บทที่ 3.....	17
3.1 วิธีการดำเนินงาน	17
3.2 แผนการดำเนินงานตลอดโครงการ	18
3.3 อุปกรณ์และเครื่องมือที่ใช้	19
3.4 ชุดข้อมูล (Data Set) ที่ใช้งาน	20
3.5 การออกแบบระบบ	24
3.6 ปัญหาและอุปสรรค	25
บทที่ 4	26
บทที่ 5	38
5.1 สรุปผลการศึกษาค้นคว้า	38
5.2 อภิปรายผล	39
5.3 ข้อเสนอแนะ	39
5.4 ข้อจำกัดในการทำงาน (Limitation).....	40
5.5 การต่อยอดโครงการวิจัย.....	41
บรรณานุกรม.....	42
ภาคผนวก	44
ภาคผนวก ก การติดตั้งโปรแกรม.....	45
ภาคผนวก ข ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam.....	51
ภาคผนวก ค Code ขั้นตอนในการฝึกฝนโมเดล (Model Training).....	56
ภาคผนวก ง Code ขั้นตอนในการทดสอบโมเดล (Model Testing).....	61

สารบัญรูปภาพ

รูปภาพที่ 1 : ตัวอย่างรูปภาพอาหารไทยอีสานในฐานข้อมูล ISANTHFOOD-44.....	3
รูปภาพที่ 2 : สูตรการหาค่า accuracy [8].....	4
รูปภาพที่ 3 : ภาพแสดง layer ของ digit-classification model [6].....	6
รูปภาพที่ 4 : Classification model [13].....	7
รูปภาพที่ 5 : โมเดล GoogLeNet รูปแบบหนึ่งของ Inception architecture [14]	8
รูปภาพที่ 6 : GoogLeNet network [14].....	9
รูปภาพที่ 7 : การ train บน ImageNet [15]	9
รูปภาพที่ 8 : ตารางเปรียบเทียบ accuracy ของ model ที่มีใน Keras.....	10
รูปภาพที่ 9 : ตัวอย่างรูปที่ผ่านกระบวนการ Data Augmentation ในฐานข้อมูล ISANTHFOOD-44	11
รูปภาพที่ 10 : งานวิจัยที่เกี่ยวข้อง.....	16
รูปภาพที่ 11 : ฐานข้อมูล ISANTHFOOD-44 : อาหาร 44 classes.....	21
รูปภาพที่ 12 : Workflow การพัฒนาโมเดล.....	24
รูปภาพที่ 13 : Application Workflow.....	24
รูปภาพที่ 14 : กราฟ Training accuracy และ Training loss ของ InceptionV3_v10.....	26
รูปภาพที่ 15 : Confusion matrix ของ InceptionV3_v10.....	28
รูปภาพที่ 16 : กราฟ model accuracy และ model loss ของ resnet_v2.....	29
รูปภาพที่ 17 : Confusion matrix ของ resnet_v2.....	31
รูปภาพที่ 18 : กราฟ model accuracy และ model loss ของ mobilenet_v2.....	32
รูปภาพที่ 19 : Confusion matrix ของ mobilenet_v2.....	34
รูปภาพที่ 20 : Training : Top1-accuracy , Validation accuracy และ Model size.....	35

รูปภาพที่ 21 : Testing : Top1-accuracy และ Top5-accuracy.....	35
รูปภาพที่ 22 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง InceptionV3.....	36
รูปภาพที่ 23 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง ResNet50.....	37
รูปภาพที่ 24 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง MobileNetV2.....	37
รูปภาพที่ 25 : รูปการทำนายผลจากโมเดล.....	40
รูปภาพที่ 26 : รูปการทำนายผลจากแอปพลิเคชัน.....	40
รูปภาพที่ 27 : รูปการทำนายผลจากโมเดล.....	41
รูปภาพที่ 28 : รูปการทำนายผลจากแอปพลิเคชัน.....	41

สารบัญตาราง

ตารางที่ 1 : แผนการดำเนินงานโครงการวิจัย	19
ตารางที่ 2 : ตารางแสดงจำนวนรูปภาพในฐานข้อมูล ISANTHFOOD-44	20
ตารางที่ 3 : ตารางแสดงจำนวนรูปภาพในแต่ละคลาสของฐานข้อมูล ISANTHFOOD-44.....	23
ตารางที่ 4 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ InceptionV3_v10.....	27
ตารางที่ 5 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ resnet_v2.....	30
ตารางที่ 6 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ mobilenet_v2.....	33
ตารางที่ 7 : ตารางสรุปผลลัพธ์ในแต่ละแบบจำลอง.....	39

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันการรับประทานอาหารนั้นถือได้ว่าเป็นปัจจัยสำคัญอย่างหนึ่งที่ส่งผลโดยตรงต่อสุขภาพของมนุษย์ กระบวนการของการแปรรูปอาหารในชนิดต่าง ๆ โดยทั่วไปแล้ว ผู้ที่ปรุงอาหารมักจะมุ่งเน้นและให้ความสำคัญไปที่รสชาติของอาหาร โดยอาจลืมไปว่าเครื่องปรุงที่นำมาปรุงแต่งเติมในอาหารนั้นอาจส่งผลร้ายต่อสุขภาพ ทางนิสิตมหาวิทยาลัยขอนแก่นได้จัดทำโครงการวิจัยที่ชื่อว่า โครงการมณี ขึ้น โดยเป็นการทำฐานข้อมูลของอาหาร จากกลุ่มตัวอย่างอาสาสมัครในจังหวัดขอนแก่น เนื่องจากจังหวัดขอนแก่นนั้นถือได้ว่าเป็นเมืองอัจฉริยะ (Smart City) จึงเป็นปัจจัยที่ทำให้และมีการดึงดูดให้มีประชากรเพิ่มขึ้นทุกปี และทำให้เป็นเมืองที่มีการเจริญเติบโตอย่างรวดเร็ว (Rapidly growing city) ดังนั้นจึงทำให้เกิดปัญหาภายในเมืองตามมา เช่น ปัญหาทางด้านสุขภาพของคนในจังหวัด [1]

โดยจากเหตุผลข้างต้นจึงทำให้เกิดงานวิจัยนี้ขึ้น โดยจะมุ่งเน้นเกี่ยวกับการวิเคราะห์อาหารไทยในส่วนของอาหารอีสานจากภาพถ่ายซึ่งยังไม่มีเคยมีผู้วิจัยมาก่อน และเป็นอาหารที่ชาวขอนแก่นนิยมรับประทาน โดยผู้วิจัยได้ทำร่วมกับทางมหาวิทยาลัยขอนแก่น เพื่อที่จะนำฐานข้อมูลอาหารจากโครงการมณีมาใช้ประโยชน์ และสนับสนุนการเป็นเมืองอัจฉริยะของจังหวัดขอนแก่น ซึ่งงานวิจัยนี้จะสามารถเข้ามาช่วยแก้ไขปัญหาได้ ในด้านการวิเคราะห์อาหารจากภาพถ่าย (Food Image Recognition) ว่าในแต่ละมื้อของวันนั้น ประชากรของจังหวัดได้รับประทานอาหารชนิดใดบ้าง ซึ่งหลังจากรู้ได้ว่าเป็นอาหารชนิดใด ในขั้นถัดไปจะทำให้สามารถวิเคราะห์ได้อีกว่าอาหารที่ได้รับประทานไปนั้นมีส่วนผสมต่างๆ ประมาณเท่าไรและจะส่งผลต่อสุขภาพผู้รับประทานมากหรือน้อยเพียงใดได้ เป็นต้น โดยทางผู้วิจัยจะทำการพัฒนาการวิเคราะห์อาหารไทยอีสานจากภาพถ่ายนี้ให้อยู่ในรูปแบบของแอปพลิเคชัน (Application) ที่มีชื่อว่า SaapFood Cam สำหรับระบบปฏิบัติการแอนดรอยด์ (Android) ที่ผู้ใช้งานสามารถถ่ายรูปอาหารจากโทรศัพท์มือถือผ่านแอปพลิเคชันเพื่อให้แอปพลิเคชันนำรูปภาพไปประมวลผลกับโมเดลและระบุว่าอาหารนั้นคืออาหารอะไร โดยงานวิจัยนี้สามารถนำไปต่อยอดในอนาคตได้ เช่น การนำชื่ออาหารที่ได้ไปแสดงรายละเอียดต่างๆ ให้แก่ผู้ใช้ได้ทราบข้อมูลอาทิ แคลอรีของอาหารนั้น หรือระดับความเค็ม

โดยทั่วไปของอาหาร เป็นต้น และยังสามารถต่อยอดไปถึงการตรวจจับความเค็มโดยดูจากสีอาหารได้ในอนาคตอีกด้วย

เทคโนโลยี Deep Learning ในปัจจุบันเป็นเทคโนโลยีที่ได้รับความนิยมเป็นอย่างมาก โดยเฉพาะงานทางด้าน image recognition ที่ผู้วิจัยให้ความสนใจ ก็ใช้เทคโนโลยี Deep Learning แบบ Convolutional Neural Network (CNN) ซึ่งเป็นหนึ่งใน Deep Learning ที่มี input เป็นรูปภาพที่ได้รับความนิยมมากเช่นกัน ดังนั้นในงานวิจัยนี้ ผู้วิจัยจะใช้หลักการและเทคโนโลยี Deep Learning ในการศึกษาและดำเนินการวิจัย

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษา Deep Learning Models ต่างๆ โดยนำมาพัฒนาเพื่อใช้จำแนกอาหารไทยอีสานจากภาพถ่ายและสามารถระบุชื่ออาหารเหล่านั้นได้

1.2.2 เพื่อพัฒนาแอปพลิเคชันที่สามารถใช้ในการระบุอาหารไทยอีสานในชีวิตประจำวันจากภาพถ่ายโดยโทรศัพท์มือถือได้

1.3 ขอบเขตของการวิจัย

งานวิจัยนี้ได้นำภาพอาหารไทยอีสานที่ได้รับการเก็บข้อมูลจากโครงการวิจัยที่ชื่อว่า การทำฐานข้อมูลของอาหาร จากกลุ่มตัวอย่างอาสาสมัคร หมู่ที่ 22 บ้านหนองหิน อำเภอเมือง จังหวัดขอนแก่น ในการประมาณปริมาณแคลอรีที่ได้รับจากการรับประทานอาหาร (โครงการมณี) โดยนิสิตมหาวิทยาลัยขอนแก่น มาใช้ในการวิจัย ข้อมูลที่ใช้ในการวิจัยครั้งนี้จะประกอบด้วย ภาพถ่ายอาหารอีสานของอาสาสมัครจำนวน 30 คน ที่เป็นผู้สูงอายุ (อายุ 60-79 ปี) ทั้งหญิงและชายจากบ้านหนองหิน หมู่ที่ 22 ตำบลศิลาจังหวัดขอนแก่น มีทั้งผู้ที่มีโรคประจำตัว และไม่มีโรคประจำตัว และซึ่งไม่เป็นโรคไต โดยจัดเก็บข้อมูลเป็นภาพถ่ายทั้ง 3 มื้ออาหาร เช้า กลางวัน และเย็น ทั้งหมด 3 เดือน (มกราคม, กุมภาพันธ์, และมีนาคม) และมีการเก็บรวบรวมรูปภาพจาก search engine เช่น Google มาไว้ในฐานข้อมูลด้วย ซึ่งฐานข้อมูลรูปภาพที่ใช้ในงานวิจัย ผู้วิจัยได้ตั้งชื่อให้กับฐานข้อมูลนี้ว่า ISANTHFOOD-44



รูปภาพที่ 1 : ตัวอย่างรูปภาพอาหารไทยอีสานในฐานข้อมูล ISANTHFOOD-44

ผู้วิจัยจะนำข้อมูลเหล่านี้มาใช้ในการจำแนกว่าอาหารอีสานที่ปรากฏในภาพถ่ายนั้นคืออาหารอะไร โดยใช้เทคนิคการเรียนรู้ Deep Learning ในด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) ในการทำ Object Recognition คือการตรวจจับวัตถุในภาพต่างๆแล้วระบุว่าวัตถุนั้นคือวัตถุอะไร และ Object Detection ซึ่งเป็นปัญหาทางด้าน computer vision ที่เป็นปัญหาพื้นฐานและมีความท้าทายเป็นอย่างมาก ซึ่งเป้าหมายของ Object Detection คือการหาวัตถุที่สนใจในภาพต่างๆว่าอยู่ตรงจุดไหนบ้างในภาพนั้นๆ ซึ่งอาจจะมีวัตถุหลายวัตถุในภาพเดียวหรือมีวัตถุเดียวแต่หลายชิ้น เป็นต้น [7]

โดยสถาปัตยกรรม Convolutional Neural Network (CNN) ที่ใช้คือ

1. GoogLeNet/InceptionV3
2. ResNet50
3. MobileNetV2

ทำการเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึมที่นำมาใช้ในการจำแนกอาหารด้วย ค่า Accuracy แบบ Top-1 accuracy และ Top-5 accuracy และทำการเปรียบเทียบขนาดของแบบจำลอง (Model size)

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Number Of Examples}}$$

รูปภาพที่ 2 : สูตรการหาค่า accuracy [8]

โดยที่ Correct Predictions คือจำนวน sample ใน Test Set ที่ทำนายถูกต้อง และ Total Number Of Examples คือจำนวน Sample ทั้งหมดใน Test Set

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถระบุซื้ออาหารอีสานจากรูปถ่ายบนโทรศัพท์มือถือได้

1.4.2 สามารถใช้แอปพลิเคชัน SaapFood Cam ในการวิเคราะห์รายละเอียดอาหารอีสานในชีวิตประจำวันและเป็นส่วนช่วยให้สุขภาพของชาวขอนแก่นพัฒนาดีขึ้น

1.4.3 สามารถนำงานวิจัยนี้ไปพัฒนาต่อยอดในอนาคตได้ เช่นการนำซื้ออาหารที่ได้ไปแสดงรายละเอียดต่างๆให้แก่ผู้ใช้ได้ทราบข้อมูลเช่น แคลอรีของอาหารนั้นหรือระดับความเค็มโดยทั่วไปของอาหาร และสามารถทำการตรวจจับความเค็มโดยดูจากสีอาหารในภาพถ่ายได้ในอนาคต

บทที่ 2

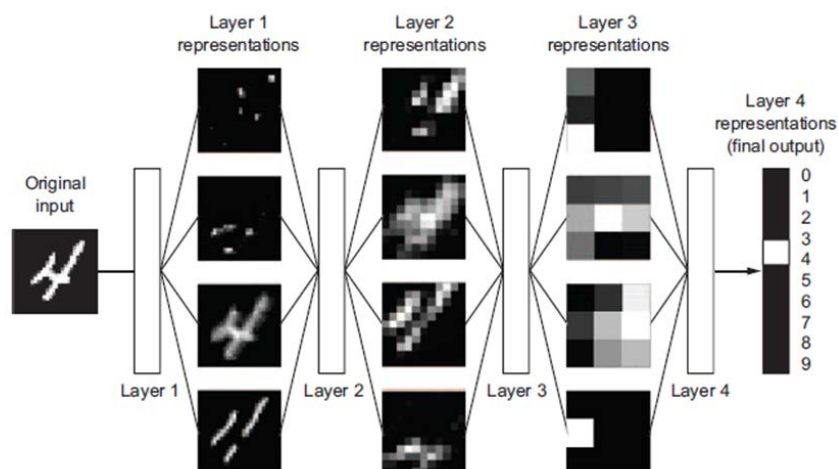
องค์ความรู้ที่เกี่ยวข้อง

เนื่องจากในปัจจุบันการรับประทานอาหารนั้นถือได้ว่าเป็นตัวการสำคัญที่ส่งผลโดยตรงต่อสุขภาพของมนุษย์ ทางผู้วิจัยจึงได้นำเทคนิคการทำ Food Recognition โดยเทคนิคการเรียนรู้ของเครื่อง Deep Learning และ Convolutional Neural Network (CNN) มาช่วยในการวิเคราะห์รูปภาพอาหารจากการถ่ายภาพจากอุปกรณ์พกพา ซึ่งจะช่วยในการวิเคราะห์ข้อมูลได้ว่าในแต่ละมื้อนั้นทางผู้ใช้งานได้มีการรับประทานอาหารชนิดใดบ้าง ซึ่งจะเป็นส่วนช่วยอีกช่องทางหนึ่งที่จะสามารถวิเคราะห์ทางด้านสุขภาพของผู้ใช้เองได้ โดยมีเนื้อหาและงานวิจัยที่เกี่ยวข้องดังนี้

2.1 การเรียนรู้ของเครื่อง Deep Learning

Deep learning เป็นซับเซตของ Machine learning ซึ่งเป็นตัวอย่างการเรียนรู้แบบใหม่จาก data ที่ให้ความสำคัญกับการเรียนรู้เกี่ยวกับ layers ส่วนคำว่า deep ใน deep learning นั้นไม่ได้หมายความว่า deep learning จะเข้าใจได้ยากแต่อย่างใด แต่ deep นั้นหมายถึงแนวคิดของ successive layers จำนวนของชั้น layers ที่โมเดลนั้นมีเรียกว่า depth หรือความลึก ชื่ออื่นๆที่ใช้เรียก deep learning คือ layered representations learning และ hierarchical representations learning โมเดล deep learning มักจะมีจำนวน successive layers อยู่ที่ 10 หรือ 100 layers และ โมเดลของ deep learning นั้นจะถูกเรียนรู้โดยอัตโนมัติจาก training data ในขณะที่ machine learning มีแนวโน้มที่จะเรียนรู้เพียง 1 หรือ 2 layers ทำให้ในบางครั้ง machine learning จึงถูกเรียกว่า shallow learning

ใน deep learning มีการเรียนรู้ผ่านรูปแบบโมเดลที่เรียกว่า neural networks โดยมีการสร้างโครงสร้างตามระบบประสาทมนุษย์ โดยการอ้างอิงประสาทชีววิทยา แต่ deep learning นั้น ไม่ใช่ model ของสมองจริงๆ เนื่องจากยังไม่มีหลักฐานโดยแน่ชัดว่าสมองนั้นมีการเรียนรู้เหมือนกับ modern deep-learning models ตัวอย่างการเรียนรู้โดย deep learning algorithm ที่ได้มีการ ทดสอบ เช่น การดูภาพตัวเลขตั้งแต่ 0 ถึง 9 และ ให้โมเดลทำการวิเคราะห์ออกมาว่าเป็นตัวเลขอะไรจากภาพนั้นๆ ซึ่งเป็นกระบวนการเพื่อให้เกิดการจำตัวเลขแล้วแสดงผลลัพธ์ของการจดจำออกมา [6]



รูปภาพที่ 3 : ภาพแสดง layer ของ digit-classification model [6]

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) มีรูปแบบของ Neural Network แบบ Feed-Forward artificial neural network (ANN) โดย CNN จะประกอบด้วยชั้น (layer) หลายๆชั้นแตกต่างกัน ได้แก่ input layer, convolutional layer, pooling layer และ fully-connected layer ชั้นเหล่านี้จะวางเป็นชั้นๆซ้อนกัน ขึ้นอยู่กับโครงสร้าง (Architecture) ของ CNN เพื่อที่จะทำงาน recognition ต่างๆ โดยในแต่ละชั้นจะมีคำอธิบายการทำงานอย่างสั้นๆได้ดังนี้

Input layer คือชั้นที่ประกอบด้วยข้อมูลที่ใช้ในการ train หรือ test ซึ่งตัวอย่างในที่นี้คือรูปภาพต่างๆ รูปเหล่านี้จะมีรูปแบบหรือมี format แบบ RGB ขนาดของรูปภาพขึ้นอยู่กับ model ที่จะใช้ใน network ยกตัวอย่างเช่น รูปภาพขนาด 256×256 พิกเซล ข้อมูลจะทำการจัดเก็บรูปภาพเท่ากับ $[256 \times 256 \times 3]$ โดยเลข 3 ในที่นี้หมายถึงรูปภาพนี้มี 3 channels หรือ RGB

Convolutional layer ชั้นนี้คือชั้นที่มีความสำคัญสำหรับ CNN โดยจะมี dot product ระหว่าง filter และ particular volume ของ input data โดยจะเริ่มที่ position (0,0) ของ input data และขยับไป 1 พิกเซล (stride 1) จากซ้ายไปขวาและจากบนลงล่างของรูปภาพ ผลลัพธ์หรือ output ที่ได้ของ product เรียกว่า activation map ยกตัวอย่างเช่น หากมีข้อมูลขนาด $[224 \times 224 \times 3]$ และ filter ขนาด 2 พิกเซล (stride 2) ขนาดของ activation map จะเป็น $[111 \times 111 \times 96]$

Pooling layer คือชั้นที่อยู่ถัดจาก convolutional layer เพื่อที่จะลดหรือย่อขนาดรูปภาพ function ที่ใช้ในชั้นนี้สามารถเป็น non-linear function ได้ ยกตัวอย่างเช่น max pooling, average pooling และ L2

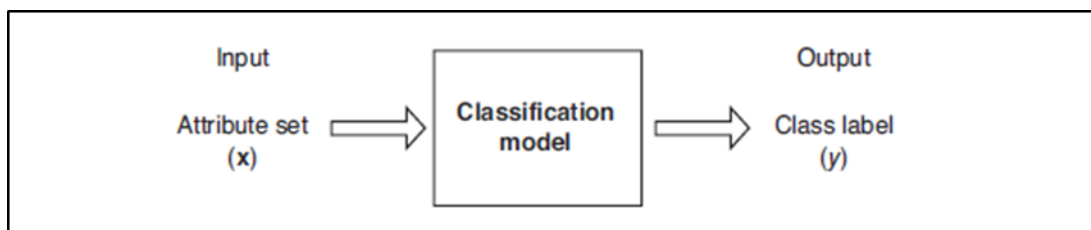
pooling เป็นต้น โดย max pooling เป็น pooling ที่มีการใช้โดยทั่วไปเป็นมาตรฐานและให้ performance ที่ดีกว่า การทำงานของ max pooling คือ max pooling จะเลือกค่าสูงสุดในบริเวณที่ตัวกรองหรือ filter ทาบอยู่ parameter ของ network ทั้งหมด จะถูกลดลงขึ้นอยู่กับขนาดของ filter และ striding step ยกตัวอย่างเช่น หากมี input ขนาด $[111 \times 111 \times 96]$ เมื่อใช้ filter ขนาด 3×3 และ striding step ไป 2 จะทำการลดขนาดข้อมูล และผลลัพธ์ที่ได้คือ $[55 \times 55 \times 96]$

Fully-connected layer คือชั้นที่ถูกล้างไว้มากที่สุดท้ายของ network การ activation ทั้งหมดในชั้นก่อนหน้านี้จะมีการเชื่อมต่อกับชั้นนี้ โดย Fully-connected layer จะทำการลดขนาดของข้อมูลให้เหลือ 1 dimension

CNN ได้รับการพัฒนาเพื่อที่จะจดจำข้อมูลใน deep level โดยการเพิ่ม hidden layer เข้าไป CNN สามารถจดจำรูปภาพต่างๆได้ใน 3 dimensions นั่นคือ ความกว้าง ความยาว และความลึก ตัว network จะทำการแบ่งรูปภาพเป็นส่วนๆ และทำการวิเคราะห์แต่ละส่วนเพื่อที่จะสกัดคุณสมบัติหรือ feature ที่สำคัญของรูปภาพ ยกตัวอย่างเช่น สีของรูปภาพ, รูปทรง และพื้นผิว เป็นต้น [5]

2.3 Classification

การจำแนกข้อมูล (Classification) เป็นกระบวนการของการจัดกลุ่มของข้อมูลที่มีความคล้ายกันให้อยู่ด้วยกัน โดยการที่สร้างโมเดลขึ้นมาเพื่อที่จะทำการจำแนกข้อมูล (Classifier) หรือทำการทำนายประเภทข้อมูล โดยในกระบวนการสร้างโมเดลจำแนกข้อมูลหรือการ prediction ข้อมูลนั้น จะทำได้โดยการ training ของข้อมูล โดยจะเรียกว่าการ training data ซึ่งกระบวนการ Classification นั้นเป็นกระบวนการที่จะช่วยให้ผู้ใช้จัดการและจัดระเบียบ แยกแยะข้อมูลรวมไปถึงความสมเหตุสมผลของข้อมูล ซึ่ง Classification model นั้นจะสามารถที่จะใช้เป็นเครื่องมือเพื่อที่จะแยกความแตกต่างระหว่าง object ที่แตกต่างกันได้



รูปภาพที่ 4 : Classification model [13]

การจำแนกข้อมูล (Classification) จะเกี่ยวกับการจัดข้อมูลที่มีความคล้ายและคุณสมบัติเดียวกันไว้ด้วยกันใน class และแยกข้อมูลที่มีความแตกต่างกันออกไปอีก class หนึ่ง [12] [13]

2.4 Network Architecture

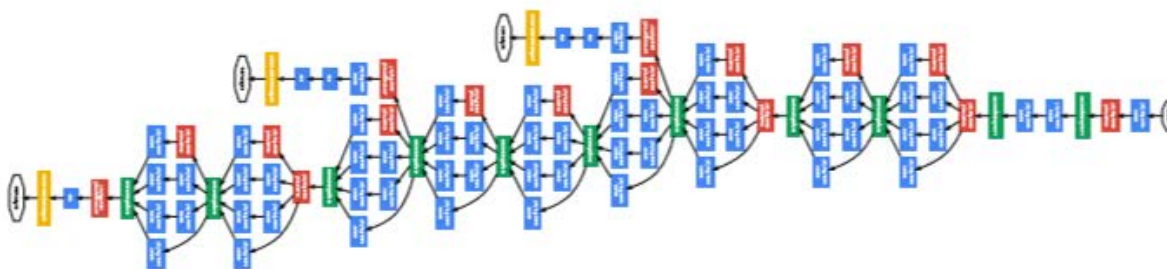
2.4.1 GoogLeNet

GoogLeNet เป็น Inception architecture ที่ทีมวิจัย Google ใช้ในการแข่งขัน ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC) ซึ่งเป็นการแข่งขัน image classification และสามารถเอาชนะ ZFNet ที่เป็นผู้ชนะในปี 2013 และ AlexNet ที่เป็นผู้ชนะในปี 2012 ไปได้ อีกทั้งยังมี error rate ที่น้อยกว่าเมื่อเปรียบเทียบกับ VGGNet ที่ป็นรองชนะเลิศอันดับ 1 ในปี 2014 อีกด้วย ซึ่งมี top-5 error rate เท่ากับ 6.67% GoogLeNet ยังมีอีกชื่อที่เป็นที่รู้จัก นั่นคือ Inception v1 และต่อมายังมี v2, v3 และ v4 เป็นต้น GoogLeNet ใช้ CNN และได้มีการพัฒนามาจาก LeNet ภายใน architecture ประกอบด้วย 22 layers deep CNN แต่ลดจำนวนของ parameter จาก 60 ล้าน (AlexNet) ไปเป็น 4 ล้าน [14]

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture.

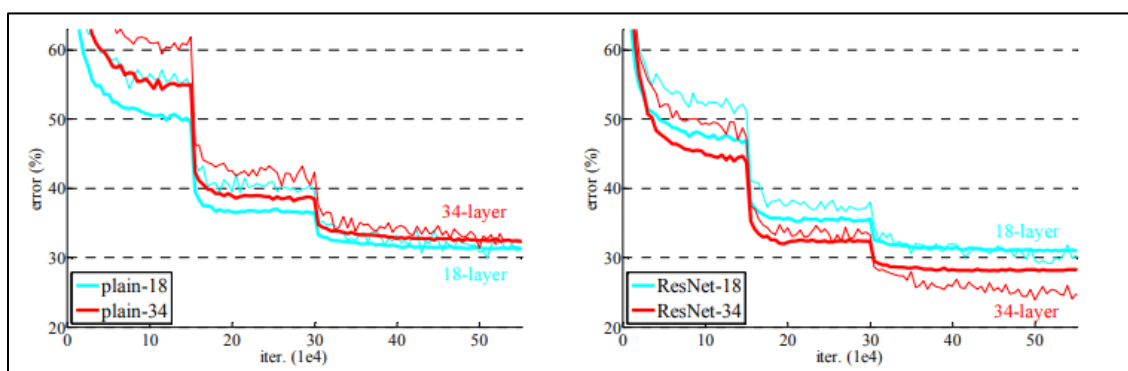
รูปภาพที่ 5 : โมเดล GoogLeNet รูปแบบหนึ่งของ Inception architecture [14]



รูปภาพที่ 6 : GoogLeNet network [14]

2.4.2 ResNet

ResNet หรือ Residual Networks เป็น neural network ที่ใช้ในงานประเภท computer vision หลายๆงาน ง่ายต่อการปรับให้เหมาะสมและมีความแม่นยำสูง โมเดลนี้เป็นโมเดลที่ชนะเลิศอันดับ 1 ในการแข่งขัน ImageNet challenge ปี 2015 (ILSVRC 2015) ResNet-152 มีทั้งหมด 152 layers ซึ่งมีความลึกกว่า VGGNet และยังมีความซับซ้อนที่น้อยกว่า ซึ่งมี top-5 error rate คือ 3.57% จากการทดสอบจาก ImageNet และยังมี version ที่เล็กกว่านั้นคือ ResNet-50 ResNet สามารถที่จะ train ได้ง่ายและยังมี error น้อยจากการ train อีกด้วย สามารถรวมความแม่นยำได้จากความลึกก็มากขึ้นมาและให้ผลลัพธ์ที่ดีกว่า networks อื่นก่อน [15]



รูปภาพที่ 7 : การ train บน ImageNet [15]

2.4.3 MobileNet

MobileNet เป็นโมเดลที่ถูกออกแบบมาสำหรับมือถือโดย Google มาพร้อมกับ Weight ของโมเดลด้วยข้อมูลจากฐานข้อมูล ImageNet ซึ่งมีขนาดที่เล็กเหมาะสมในการนำมาใช้กับอุปกรณ์พกพาที่มีทรัพยากรที่จำกัด แต่ประสิทธิภาพอาจไม่ดีเท่ากับ model จาก ImageNet ตัวอื่นๆ แต่ก็ให้ accuracy ที่ดีในระดับหนึ่ง

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

รูปภาพที่ 8 : ตารางเปรียบเทียบ accuracy ของ model ที่มีใน Keras

(ที่มา : <https://keras.io/applications/>)

2.5 Data Augmentation

เนื่องจากภาพในฐานข้อมูลที่ได้เก็บรวบรวมมาในเบื้องต้นมีจำนวนที่น้อย ดังนั้นผู้วิจัยจึงใช้เทคนิค Data Augmentation เพื่อเพิ่มจำนวนรูปภาพด้วยวิธีต่างๆ โดยใช้ ImageDataGenerator จาก Keras library

Data Augmentation คือวิธีการในการเพิ่มรูปภาพโดยใช้วิธีต่างๆ เช่น การพลิกรูปภาพ (Flip) การขยายรูปภาพด้วยสเกลต่างๆ (Zoom) และการหมุนรูปภาพ (Rotation) เป็นต้น [16] โดยวิธีที่ใช้ในการเพิ่มข้อมูลในฐานข้อมูล ISANTHFOOD-44 มีดังนี้

1. horizontal_flip และ vertical_flip
2. zoom_range
3. width_shift_range และ height_shift_range
4. rotation_range



รูปภาพที่ 9 : ตัวอย่างรูปที่ผ่านกระบวนการ Data Augmentation ในฐานข้อมูล ISANTHFOOD-44

2.6 งานวิจัยที่เกี่ยวข้อง

การทบทวนวรรณกรรมของงานวิจัยนี้ ได้ทำการศึกษางานวิจัยที่เกี่ยวข้องกับการประมวลผลภาพถ่ายอาหาร โดยเน้นในเรื่อง Food Image Recognition มีรายละเอียดดังต่อไปนี้

2.6.1 C. Termritthikun, P. Muneesawang, and S. Kanprachar [5] ทำการศึกษา Image Recognition อาหารไทยจากรูปภาพบน smartphone โดยใช้ Deep Learning, Convolutional Neural Networks (CNN) และได้มีการเสนอ network ที่ชื่อว่า NU-InNet (Naresuan University Inception Network) โดย network นี้ได้นำ inception concept ใน GoogLeNet มาปรับใช้ในการ train และ test model จะใช้ database ที่ชื่อว่า THFOOD-50 ประกอบด้วยรูปภาพอาหารไทยยอดนิยมทั้งหมด 50 ประเภท นำภาพมาจาก search engine คือ Google, Bing และ Flickr ในอาหารแต่ละชนิดจะมีรูปภาพประมาณ 200-700 รูป และมีการปรับขนาดภาพให้มีขนาด 256x256 pixels เพื่อการใช้ใน CNN model และมีการแบ่งข้อมูลเป็น 2 กลุ่ม โดย 90% ใช้สำหรับ train และ 10% สำหรับ test ในกระบวนการ train จะใช้ model ทั้งหมด 5 ตัวดังนี้ (Network Architectures) 1) AlexNet 2) GoogLeNet 3) SqueezeNet 4) NU-InNet 1.0 และ 5) NU-InNet 1.1 โดยมี benchmark คือ processing time กับ model size ต้องน้อยกว่าจาก GoogLeNet แต่ accuracy จะต้องไม่น้อยกว่าจาก GoogLeNet ซึ่งพบว่าค่า accuracy ของ NU-InNet 1.0 และ 1.1 นั้นดีกว่าของ GoogLeNet เล็กน้อย และเมื่อพิจารณาจาก top-1 และ top-5 accuracies จะเห็นได้ว่า NU-InNet 1.0 และ 1.1 จะดีกว่า the winning models จากการแข่งขัน ILSVRC ซึ่ง top-1 ที่ดีที่สุดคือ NU-InNet 1.0 ได้ 69.8% และ top-5 คือ NU-InNet 1.0 และ 1.1 ได้ 92.3%

2.6.2 D. Sahoo และคณะ [9] ทำการพัฒนา FoodAI (FoodAI Project) โดยใช้ deep learning: food image recognition system สำหรับ smart food logging FoodAI ช่วยลดภาระในการ logging แบบ manual กับ online food journal โดยได้ทำการ train เพื่อระบุ 756 ประเภทอาหารครอบคลุมอาหารที่บริโภคทั่วไปในสิงคโปร์ มี partner ใหญ่คือ Health Promotion Board (HPB) ที่ประเทศสิงคโปร์ และได้ทำการ integrate FoodAI เข้าไปใน Healthy 356 Application ทั้งแบบ IOS และ Android

Application นี้เปิดให้ user สามารถ track แคลอรีที่บริโภคไปในแต่ละวันได้ เมื่อ user ต้องการอัปเดต diet journal ตัวเอง user จะทำการถ่ายรูปมื้ออาหารนั้นๆ เพื่อเป็นการระบุอาหารนั้นๆ โดย FoodAI system หลังจากตัว system ได้ทำการคำนวณแล้ว FoodAI ก็จะทำการ return list ของ top visual food categories ที่ทำการ

sort โดย classification score เมื่อมีการแสดง list นั้นให้แก่ user user ก็จะทำการเลือกว่ารูปไหนที่อธิบายหรือแสดงลักษณะอาหารของ user ได้ดีที่สุด ถ้าคำตอบที่ถูกไม่แสดงขึ้นบน top predicted result user จะสามารถเลือกอาหารที่ถูกต้องด้วยตนเองได้ feedback ต่างๆจาก user จะถูกจัดเก็บเพื่อที่จะนำมาพัฒนา performance ต่างๆ ของ model Dataset รูปภาพที่ใช้จะนำมาจาก Google, Bing, IG, Flickr และ Facebook เป็นต้น ในปัจจุบันมี 756 visual foods ประมาณ 400,000 images มี minimum 174 ภาพ และ maximum 2,312 ภาพ ต่อ visual food จากนั้นทำการ split dataset ให้เป็น train, validation และ test data สำหรับการ train และ ประเมิน model งานวิจัยนี้จะเรียก dataset ว่า FoodAI-756 (756 classes / visual food categories) ในงานวิจัยนี้จะเน้นไปที่ model ที่ใหม่ขึ้น อาทิ ResNet [14], ResNeXt [38] และ SENet [17] ซึ่งพบว่าค่า accuracy สูงสุดที่ได้ top-1 คือ SENet + ResNeXt101 + Focal loss ได้ 83.2% และ top-5 คือ SENet + ResNeXt101 + Focal loss ได้ 95.7%

2.6.3 X. Chen, Y. Zhu, H. Zhou, L. Diao, และ D. Wang [10] ทำการพัฒนา ChineseFoodNet เพื่อจดจำภาพอาหารจีนได้อย่างอัตโนมัติ เนื่องจากอาหารจีนมีการจดจำได้ยากเมื่อเปรียบเทียบกับอาหารญี่ปุ่นหรืออิตาลีที่จะมีจำนวน dataset ที่เยอะกว่าแต่มีลักษณะอาหารบางอย่างที่คล้ายกัน จึงได้สร้าง dataset ขนาดใหญ่ขึ้นมารวมทั้งยังได้แสดงวิธีการเลือกประเภทอาหาร ชุดข้อมูล การ clean data และการระบุ label โดยเฉพาะการใช้ machine learning มาลดการระบุ label แบบ manual ซึ่งใน dataset จะมีภาพจากทั้งในอินเทอร์เน็ตและภาพที่ถ่ายจากของจริง โดยมีจำนวนทั้งหมด 250 ประเภท และมีมากกว่า 180,000 รูป ใน dataset จะใช้ CNNs และ TastyNet ในการรวมข้อมูลและรวมผลของการทำนายต่าง ๆ ด้วย voting method ซึ่งมีความถูกต้องสูงถึง 81.43% จาก validation set และ 81.55% จาก test set โดยใช้ model ของ CNNs 4 ชนิดคือ SqueezeNet, VGG, ResNet และ DenseNet และรวมข้อมูลด้วยการ voting method โดยใช้ CNNs Model ที่แตกต่างกัน คือใช้ ResNet152, DenseNet121, DenseNet169, DenseNet201 และ VGG19-batch และได้ผลลัพธ์เป็น TastyNet ซึ่งมีความถูกต้องสูงมากถึง 81.43% จากการ validation และ 81.55% จาก test set และได้้นำ dataset ไป training, validation, และ test sets โดยการ random selection โดยจะใช้ deep learning network architecture ที่แตกต่างกันและจำนวน label ที่แตกต่างกัน การ Validation ที่ทำได้ที่ดีที่สุดอันดับ1คือ DenseNet201 มีความถูกต้อง 79.05% การ Test set ที่ทำได้ดีที่สุดคือ VGG19 ซึ่งมีความถูกต้อง 79.22% และ Resnet152 ทำได้ 79.00% และผลลัพธ์ที่ดีที่สุดได้จาก Model ResNet152, DenseNet121, DenseNet169, DenseNet201 และ VGG19-BN ซึ่งได้จากการเฉลี่ยผลลัพธ์ของทุกโมเดลด้วย Voting

method ซึ่ง ResNet ที่ layers ที่แตกต่างกันจะมีประสิทธิภาพสูงขึ้น 79.46% ซึ่งดีกว่า single ResNet และจากการรวมของ Resnet152 และ Densenet121, Densenet169, Densenet201, และ VGG19-BN ความถูกต้องคือ 81.43% จาก validation , test set 81.55% ซึ่งสูงกว่า single network ใน validation และ test set ซึ่ง top-1 ที่ได้คือ 81.43% โดย ResNet152 + DenseNet121, 169, 201+ VGG19-BN และ top-5 ที่ได้คือ 96.79% โดย ResNet18, 34, 50, 152+DenseNet121, 169, 201+VGG19-BN

2.6.4 Z. Heng, S. Roy, K. Yap, A. Kot และ L. Duan [11] ได้ทำการพัฒนาแอปพลิเคชัน e-health ซึ่งเป็นแอปพลิเคชันการออกกำลังกายบนแพลตฟอร์มมือถือ ซึ่งจะเป็นตัวช่วยให้ผู้ใช้มีนิสัยเกี่ยวกับการบริโภคอาหารที่ดีขึ้น แอปพลิเคชันนี้จะให้ความสำคัญเกี่ยวกับการบันทึกข้อมูลทางสรีรวิทยาโดยแอปพลิเคชันนั้นคือ MyFitness-Pal และ Loselt โดยใช้ HOG และ color patches กับ Fisher Vector และทำการทดสอบบน food dataset 100 ประเภท และใช้คุณสมบัติของภาพที่ผ่านการตรวจสอบและการแบ่ง ใช้ random forest เพื่อจำแนก super pixel-grouped ในรูปภาพอาหาร ซึ่งการจดจำอาหารที่มีอยู่ส่วนใหญ่จะมีพื้นฐานมาจาก CNN ผู้วิจัยได้ Train generic heavyweight network (GHN) กับ dataset โดย compact knowledge-distilled network (CKDN) ถูกปรับให้ใช้ได้กับผู้ใช้หลายคนโดยในโทรศัพท์มือถือจะการใช้ CKDN model ซึ่งจะเก็บประวัติเกี่ยวกับการรับประทานอาหารซึ่งเรียกว่า Food Diary ซึ่งภาพจะถูกอัปโหลดไปยังเซิร์ฟเวอร์ของผู้ใช้ โดย CKDN จะถูกปรับและคำนวณเป็น personalized knowledge-distilled network (PKD-Net) โดยจะมีการรวบรวมข้อมูลจาก Google, Yahoo and Baidu ซึ่ง Food Diary เป็น food databases คล้ายกับ Food-101 , Vireo Food-172 หรือ UEC Food-256 และทางผู้วิจัยได้แนะนำชุดข้อมูลใหม่โดยอ้างอิงจากอาหารอินเดีย เรียกว่า NTUIndianFood50 โดยรวบรวมภาพจาก Google และ Yahoo image search หากภาพมีความละเอียดต่ำกว่า 256x256 จะโดนลบโดยอัตโนมัติ ส่วน Food Diary จะเลือกอาหารที่ได้รับความนิยมทั้งหมด 24 ชนิดจาก NTUIndianFood50 โดย Food Diary มีรูปภาพทั้งหมด 2,073 ภาพและเมื่อเทียบกับ NTUIndianFood50 แล้ว Food Diary จะมีคุณภาพของภาพที่ต่ำกว่า และใช้ VGG16 ในการแยกประเภทอาหารจาก NTUIndianFood50 และพบว่า VGG16 ให้ความแม่นยำที่สูง จึงได้เสนอ personalized knowledge distillation based network (PKD-Net) เพื่อการจดจำอาหารบนโทรศัพท์มือถือเพราะมีการบันทึกภาพจากการถ่ายบนโทรศัพท์และขึ้นอยู่กับ MobileNet PKD-Net มีประสิทธิภาพในการแยกประเภทบน neural network และแนะนำชุดข้อมูล NTUIndianFood50 เมื่อเปรียบเทียบความแม่นยำในการแยกประเภทระหว่าง CKDN และ PKD-Net ได้ Top-1 CKDN 74.3% และ Top-5 PKD-Net 95.6% จากการทดลองในการสร้าง dataset ชุดใหม่แสดงให้เห็นว่า PKD-Net มีประสิทธิภาพที่สูง

2.6.5 I. Shota Horiguchi , Member, IEEE, Sosuke Amano, Makoto Ogawa, และ Kiyoharu Aizawa , Fellow [3] ในงานวิจัยนี้ มีการนำเสนอ personalization framework ที่มีประสิทธิภาพ มีการใช้ dataset ใหม่ ๆ ของรูปภาพอาหารรายวัน และ dataset เหล่านี้ถูกเก็บรวบรวมไว้ในแอปพลิเคชัน food-logging ผลลัพธ์ของการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอไปให้ผลลัพธ์ที่ดีกว่าวิธีการเดิม ในงานวิจัยนี้ มีการนำเสนอ personalization framework ที่เรียกว่า sequential personalized classifier (SPC) ซึ่งเป็นวิธีที่มีพื้นฐานมาจากวิธี weighted nearest neighbor ในงานวิจัยนี้ มีการสร้าง dataset ของรูปภาพอาหาร และเก็บรูปภาพอาหารที่ถูกถ่ายรายวันไว้ในแอปพลิเคชัน ที่ชื่อว่า FoodLog ซึ่งมีการเก็บรวบรวมรูปภาพนี้ไว้ประมาณ 2 ปี วิธีการที่นำเสนอสามารถ classify classes ที่เกิดขึ้นบ่อยๆ เพื่อนำไปเปรียบเทียบกับความแม่นยำกับ CNN classifier วิธีการที่ถูกนำเสนอในงานวิจัยนี้ คือ personalization framework ที่เรียกว่า sequential personalized classifier (SPC) ในขั้นตอนแรก มีการ train convolutional neural network (CNN) เป็น fixed-class classifier โดยทำการหา dataset ที่มีขนาดที่หลากหลาย มีการดึง features จาก CNN และคำนวณค่าเฉลี่ยของแต่ละ class โดยใช้ตัวอย่างของ training ในการทดลองนี้พบว่า NCM classifiers มีการใช้ mean vectors สำหรับแต่ละ class ทำได้ดีเทียบเท่ากับวิธี CNN softmax classifier มีการแสดงถึงการทำ deep features ได้ดีมากใน datasets อื่นๆ คุณลักษณะทั่วไปที่สูงๆ เป็นประโยชน์ต่อการเรียนรู้ class ใหม่ๆ ในการทดลองมีการใช้ 4 Model ได้แก่ pool5, GoogLeNet [54], ImageNet [1] และ Batch normalization [51] มี top-1 = 38.8 (Pool5 [51]) และ top-5 = 53.3 (Pool5 [51]) ในงานวิจัยนี้มีการแนะนำถึงวิธีการแก้ปัญหา personalization problem ใน food image recognition และเสนอวิธีการสำหรับ classifier personalization ในส่วนของวิธี Personalization food recognition มีการเรียนรู้เพิ่มเติม ได้แก่ fixed-class NCM classifier และ CNN ส่วนผลลัพธ์ในงานวิจัยนี้ แสดงให้เห็นว่าวิธีการ SPC ที่นำเสนอไปข้างต้น ให้ผลลัพธ์ที่ดีกว่าวิธีการเดิม

2.6.6 E. Aguilar, M. Bolanos, และ P. Radeva [4] มีการใช้วิธีการ Food Recognition ที่มีพื้นฐานมาจาก CNNs การทดลองแบ่งเป็น 2 datasets ชื่อว่า Food-101 และ Food-11 : Food-11 เป็น dataset บรรจุ 16,643 รูป จัดกลุ่มเป็น 11 ประเภทอาหาร (พาสต้า,เนื้อ,ของหวาน เป็นต้น) และยังนำรูปภาพมาจากใน social network เช่น Flickr และ Instagram โดยแบ่งเป็น 60% สำหรับ training , 20% validation และ 20% สำหรับ testing และ Food-101 เป็น dataset รูปอาหารจริง โหลดจาก foodspotting.com แบ่งเป็น 75% สำหรับ training และ 25% สำหรับ testing โดยมีการใช้ Model 7 Model ได้แก่ GoogLeNet, ResNet50 [13], ILSVRC, Keras, InceptionV3 [18], WISer [7] และ WRN [19] มีค่า top-1= 90.27 (WISer [7]) และ

top-5 = 88 (InceptionV3 + ResNet200 + WRN [19]) ผลการทดลองบน Food-11 : มีการเทียบผลของ CNN กับ CNN Fusion มีการแบ่งเป็น 2 ส่วน คือ center crop (1-Crop) และ 10 different crops มีผลดีกว่าเมื่อใช้ 10-Crops โดย Accuracy ของ CNN Fusion ดีกว่า CNN ถึง 0.75% และผลการทดลองบน Food-101 : มี overall accuracy = 90.27% โดย WiSeR[7] มีวิธีการ deep learning ด้วย CNN 3 model มี accuracy มากกว่า 88% (InceptionV3 + ResNet200 + WRN [19]) ดังนั้นผลลัพธ์ที่ดีที่สุด คือ InceptionV3 และ ResNet50 models จาก CNNs Fusion

ชื่องานวิจัย	ฐานข้อมูลที่ใช้	Architecture	
		Top1-accuracy	Top5-accuracy
NU-InNet- Thai Food Image Recognition Using Convolutional Neural Networks on Smartphone [8]	THFOOD-50	NU-InNet 1.0 (69.8%)	NU-InNet 1.0 & 1.1 (92.3%)
ChineseFoodNet: A Large-scale Image Dataset for Chinese Food Recognition [9]	ChineseFoodNet	ResNet152+DenseNet121,169,201+VGG19-BN (81.43%)	ResNet18,34,50,152+DenseNet121,169,201+VGG19-BN (96.79%)
Personalized Classifier for Food Image Recognition [10]	NUS-WIDE และ FLD	Pool5 [51] (38.8%)	Pool5 [51] (53.3%)
FoodAI: Food Image Recognition via Deep Learning for Smart Food Logging [11]	FoodAI-756	SENet + ResNeXt101 + Focal loss (83.2%)	SENet + ResNeXt101 + Focal loss (95.7%)
Food Recognition using Fusion of Classifiers based on CNNs [12]	Food-101 และ Food-11	WiSeR [7] (90.27%)	InceptionV3 + ResNet200 + WRN [19] (88%)
Personalized Knowledge Distillation-based Mobile Food Recognition [13]	NTUIndianFood50	CKDN (74.3%)	PKD-Net (95.6 %)

รูปภาพที่ 10 : งานวิจัยที่เกี่ยวข้อง

บทที่3

วิธีการดำเนินโครงการ

3.1 วิธีการดำเนินงาน

3.1.1 เริ่มต้นและวางแผนโครงการ

3.1.1.1 ประชุมเพื่อเลือกหัวข้อโครงการ

3.1.1.2 หาข้อมูลศึกษาแนวคิดและทฤษฎี

3.1.1.3 ศึกษาเทคโนโลยีที่จำเป็นมาใช้

3.1.2 การออกแบบระบบ

3.1.2.1 กำหนดวัตถุประสงค์และขอบเขตของระบบ

3.1.3 การเตรียมข้อมูลรูปภาพ

3.1.3.1 นำรูปภาพอาหารอีสานที่ได้มาจากโครงการมณี และรูปภาพอาหารอีสานเพิ่มเติม

จาก search engine ต่างๆ มาทำการศึกษาและประมวลผลภาพ

3.1.3.2 ทำ Data Augmentation และปรับขนาดรูปภาพ กับฐานข้อมูลรูปภาพที่มี

3.1.3.3 ตรวจสอบความพร้อมของฐานข้อมูล โดยฐานข้อมูลที่สร้างเสร็จจะมีชื่อว่า

ISANTHFOOD-44

3.1.4 การพัฒนาโมเดลสำหรับการรู้จำภาพอาหารอีสาน

3.1.4.1 ทำการเทรนและพัฒนาโมเดลทั้ง 3 โมเดล กับฐานข้อมูล ISANTHFOOD-44

3.1.4.2 ทำการทดสอบและปรับปรุงโมเดล

3.1.5 การพัฒนาแอปพลิเคชันสำหรับการรู้จำภาพอาหารอีสาน

3.1.5.1 ออกแบบแอปพลิเคชันเพื่อใช้กับโมเดลที่ดีที่สุดจากขั้นตอนการพัฒนาโมเดล

3.1.5.2 ทำการทดสอบและปรับปรุง

3.1.6 สรุปและเผยแพร่งานวิจัย

3.1.6.1 ทำการปรับปรุงระบบต่าง ๆ

3.2 แผนการดำเนินงานตลอดโครงการ

แผนการดำเนินงานโครงการวิจัย							
ขั้นตอนการดำเนินงาน	ก.ย. 62	ต.ค. 62	พ.ย. 62	ธ.ค. 62	ม.ค. 63	ก.พ. 63	มี.ค. 63
1. เริ่มต้นและวางแผนโครงการ							
1.1 ประชุมเพื่อเลือกหัวข้อโครงการ							
1.2 หาข้อมูลศึกษาแนวคิดและ ทฤษฎี							
1.3 ศึกษาเทคโนโลยีที่จำเป็นมาใช้							
2. การออกแบบระบบ							
2.1 กำหนดวัตถุประสงค์และขอบเขต ของระบบ							
3. การเตรียมข้อมูลรูปภาพ							
3.1 นำรูปภาพอาหารอีสานที่ได้มา จากโครงการมณี และรูปภาพอาหาร อีสานเพิ่มเติมจาก search engine ต่างๆ มาทำการศึกษาและประมวลผล ภาพ							
3.2 ทำ Data Augmentation และ ปรับขนาดรูปภาพ กับฐานข้อมูล รูปภาพที่มี							
3.3 ตรวจสอบความพร้อมของ ฐานข้อมูล โดยฐานข้อมูลที่สำเร็จ จะมีชื่อว่า ISANTHFOOD-44							

4. การพัฒนาโมเดลสำหรับการรู้จำภาพอาหารอีสาน							
4.1 ทำการเทรนและพัฒนาโมเดล							
4.2 ทำการทดสอบและปรับปรุงโมเดล							
5. การพัฒนาแอปพลิเคชัน							
5.1 ออกแบบแอปพลิเคชันเพื่อใช้กับโมเดลที่ดีที่สุดจากขั้นตอนการพัฒนาโมเดล							
5.2 ทำการทดสอบและปรับปรุง							
6 สรุปและเผยแพร่งานวิจัย							
6.1 ทำการปรับปรุงระบบต่างๆ							

ตารางที่ 1 : แผนการดำเนินงานโครงงานวิจัย

3.3 อุปกรณ์และเครื่องมือที่ใช้

โมเดลที่ผู้วิจัยสนใจและได้เสนอไปนั้นประกอบด้วย 3 โมเดล ดังนี้ 1. InceptionV3 2. ResNet50 และ 3. MobileNetV2 ผู้วิจัยจะใช้วิธีการ Transfer Learning หรือการนำโมเดลเหล่านี้ที่เคยมีการพัฒนามาแล้วมาทำการ train กับฐานข้อมูล ISANTHFOOD-44 และทำการปรับตัวโมเดลให้เหมาะสมกับข้อมูลและจุดประสงค์ของงานวิจัยนี้ หลังจากที่ทำ การ train และ test ข้อมูลพร้อมกับเปรียบเทียบและวัดประสิทธิภาพจนได้โมเดลที่เป็นที่พอใจแล้ว ผู้วิจัยจะนำโมเดลมาดัดแปลงให้อยู่ในรูปแบบที่พร้อมใช้งานกับอุปกรณ์พกพา หรือ โทรศัพท์มือถือ พร้อมสร้าง Application ภายใต้อัปเดตว่า SaapFood Cam ที่มีรูปแบบพร้อมใช้งานกับตัวโมเดลที่ได้ในระบบการ Android และทำการเรียกใช้โมเดลภายในแอปพลิเคชันดังกล่าว โดยมีตัวอย่างของเครื่องมือที่จะใช้ในงานวิจัยดังต่อไปนี้

- Tensorflow : Deep Learning library ของ Google
- Tensorflow Lite : ทำให้สามารถทำการ convert โมเดลเพื่อไปใช้ในระบบปฏิบัติการ Android ได้
- Keras : Python library สำหรับงาน Deep Learning ต่างๆ
- Scikit-learn : Python library รวมเครื่องมือเพื่อใช้ในการวัดประสิทธิภาพโมเดล

- Google Colab หรือ Colaboratory โดย Google
- Android studio : IDE สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android และภาษาโปรแกรมที่จะใช้ในงานวิจัยนี้ คือ Python

3.4 ชุดข้อมูล (Data Set) ที่ใช้งาน

งานวิจัยนี้จะใช้ฐานข้อมูลที่มีชื่อว่า ISANTHFOOD-44 ที่ทางผู้วิจัยได้สร้างขึ้น ซึ่งเกิดจากการรวบรวมรูปภาพจาก

3.4.1 โครงการวิจัยที่มีชื่อว่า การทำฐานข้อมูลของอาหาร จากกลุ่มตัวอย่างอาสาสมัคร หมู่ที่ 22 บ้านหนองหิน อำเภอเมือง จังหวัดขอนแก่น ในการประมาณปริมาณแคลอรีที่ได้รับจากการรับประทานอาหาร (โครงการมณี) จัดทำโดยนิสิตมหาวิทยาลัยขอนแก่น

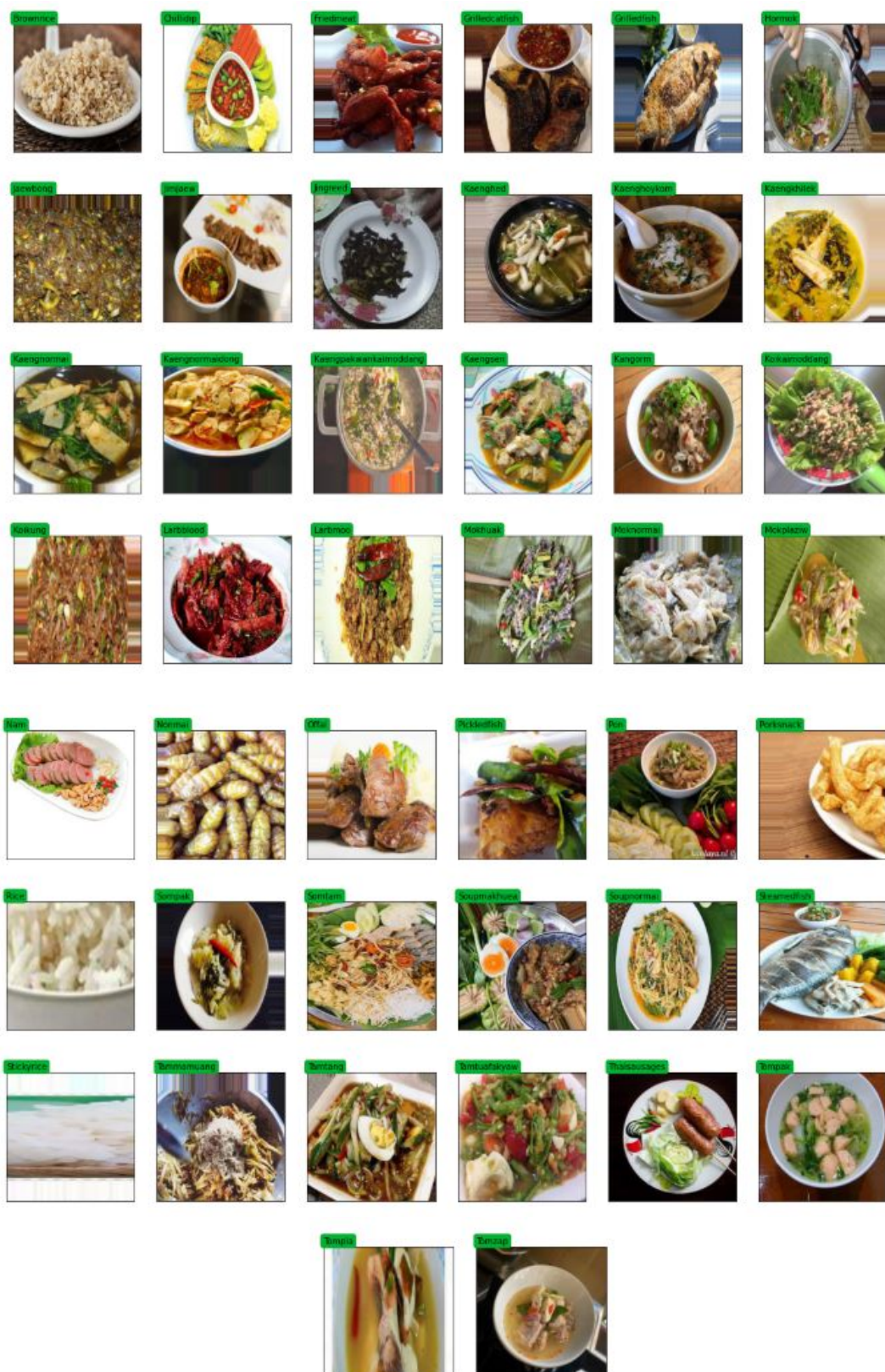
3.4.2 Search engine เช่น Google

โดยรูปภาพในฐานข้อมูล ISANTHFOOD-44 มี class ทั้งหมด 44 classes ซึ่งจะมีจำนวนรูปภาพโดยรวมกับภาพที่ทำ Augmentation แล้วดังนี้

ISANTHFOOD-44			
	Train (69 %)	Validation (4 %)	Test (27 %)
จำนวนรูปภาพ	44,000	2,200	17,500
รวม	63,700 รูป		

ตารางที่ 2 : ตารางแสดงจำนวนรูปภาพในฐานข้อมูล ISANTHFOOD-44

และมีรายชื่ออาหารพร้อมตัวอย่างรูปในฐานข้อมูล ISANTHFOOD-44 ทั้ง 44 classes ดังนี้



รูปภาพที่ 11 :ฐานข้อมูล ISANTHFOOD-44 : อาหาร 44 classes

ฐานข้อมูล ISANTHFOOD-44 จะประกอบด้วยรูปภาพสำหรับการ Train, Validation และ Test

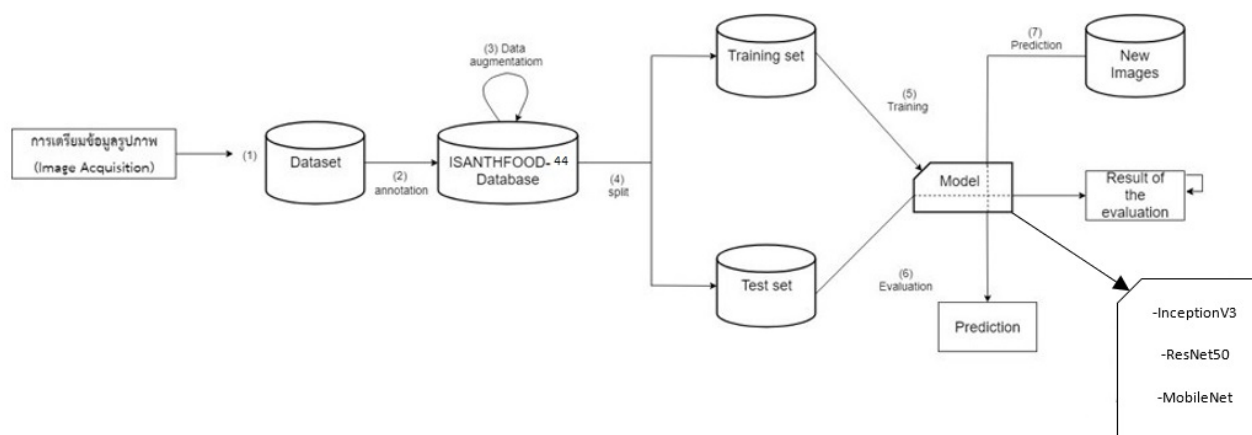
จำนวนรูปภาพในแต่ละคลาส			
ชื่อคลาส	Train	Validation	Test
ข้าวกล้อง	1000	50	400
น้ำพริก	1000	50	400
เนื้อทอด หมูทอด	1000	50	400
ปลาดุก	1000	50	400
ปลาอย่าง	1000	50	400
ห่อหมก	1000	50	400
แจ่วบอง	1000	50	400
น้ำจิ้มแจ่ว	1000	50	400
จิ้งหรีด	1000	50	400
แกงเห็ด	1000	50	400
แกงหอยขม	1000	50	400
แกงขี้เหล็ก	1000	50	400
แกงหน่อไม้	1000	50	400
แกงหน่อไม้ดอง	1000	50	400
แกงผักหวานไข่มดแดง	1000	50	300
แกงเส้น	1000	50	400
แกงอ่อม	1000	50	400
ก้อยไข่มดแดง	1000	50	400
ก้อยกุ้ง	1000	50	400
ลาบเลือด	1000	50	400
ลาบหมู	1000	50	400
หมกฮวก	1000	50	400
หมกหน่อไม้	1000	50	400
หมกปลาชิว	1000	50	400
แหนม	1000	50	400

จำนวนรูปภาพในแต่ละคลาส			
ชื่อคลาส	Train	Validation	Test
ดักแด้ หนอนไหม	1000	50	400
เครื่องใน	1000	50	400
ปลาส้ม	1000	50	400
ป่น	1000	50	400
แคบหมู	1000	50	400
ข้าวเจ้า	1000	50	400
ส้มผัก	1000	50	400
ส้มตำ	1000	50	400
ซูปมะเขือ	1000	50	400
ซูปหน่อไม้	1000	50	400
ปลานึ่ง	1000	50	400
ข้าวเหนียว	1000	50	400
ตำมะม่วง	1000	50	400
ตำแตง	1000	50	400
ตำถั่วฝักยาว	1000	50	400
ไส้กรอกอีสาน	1000	50	400
ต้มผัก	1000	50	400
ต้มปลา	1000	50	400
ต้มแซ่บ	1000	50	400

ตารางที่ 3 : ตารางแสดงจำนวนรูปภาพในแต่ละคลาสของฐานข้อมูล ISANTHFOOD-44

3.5 การออกแบบระบบ

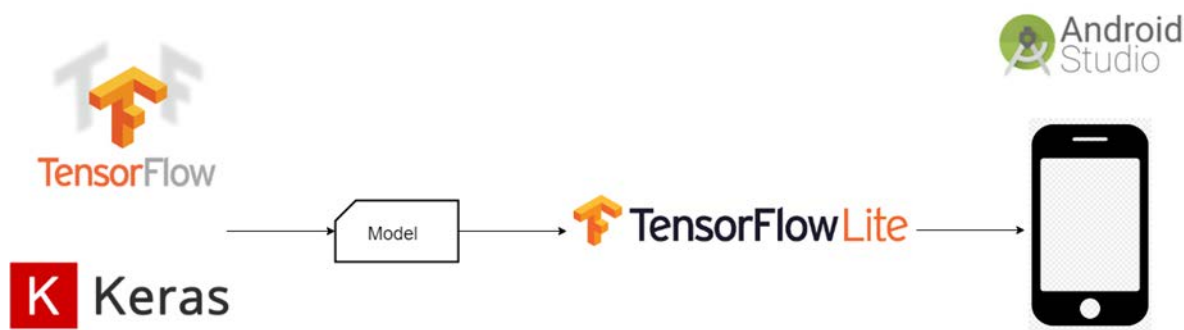
3.5.1 Workflow การพัฒนาโมเดล



รูปภาพที่ 12 : Workflow การพัฒนาโมเดล

โดยในขั้นตอนนี้จะเริ่มจากการเตรียมรูปภาพมาสร้างเป็นฐานข้อมูลโดยใช้ชื่อว่า ISANTHFOOD-44 และหลังจากนั้นจะนำข้อมูลที่ได้มาทำการแบ่งเป็น Training dataset และ Testing dataset และทำการ Train Model และ Test Model จนได้โมเดลที่ดีที่สุดออกมาทั้ง 3 โมเดล คือ InceptionV3 ResNet50 และ MobileNetV2 และนำไปใช้ในแอปพลิเคชันต่อไป

3.5.2 Application Workflow



รูปภาพที่ 13 : Application Workflow

ในขั้นแรก มีการใช้ TensorFlow และ Karas ในการสร้าง Model หลังจากนั้นนำ Model ที่ได้ไป convert model ใน TensorFlow Lite เพื่อให้ Model เหมาะสำหรับนำมาใช้ในอุปกรณ์พกพา และ มีการใช้ Android Studio สร้างแอปพลิเคชันเพื่อนำมาเชื่อมกับ Model ต่อไป

3.6 ปัญหาและอุปสรรค

3.6.1 จำนวนของรูปภาพจากโครงการมณี (ขอนแก่น) ไม่เพียงพอ ทำให้ผู้วิจัยต้องทำการเพิ่มจำนวนรูปภาพจาก search engine และใช้เทคนิค Data Augmentation เพิ่มเติม

3.6.2 เนื่องจากโมเดลที่นำมาใช้เป็นโมเดลขนาดใหญ่ ทำให้ใช้เวลาในการประมวลผลนาน ผู้วิจัยจึงต้องลดขนาดของฐานข้อมูลลง เพื่อที่จะลดเวลาที่ใช้ในการเทรนโมเดล แต่โมเดลที่ได้นั้นจะต้องมีประสิทธิภาพตามวัตถุประสงค์ของโครงการด้วย

บทที่ 4

ผลการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ผู้จัดทำได้เสนอผลการดำเนินโครงการ โดยแสดงผลดังต่อไปนี้

4.1 ผู้วิจัยได้ทำการเทรนโมเดลทั้ง 3 โมเดลโดยใช้วิธีที่ชื่อว่า Transfer Learning โดยใช้โมเดลที่เป็น built in model จาก Keras และใช้ weights จาก imageNet

4.2 มีการปรับเปลี่ยนขนาดของ input เป็น 256x256 pixels ตามขนาดรูปภาพในฐานข้อมูล และเพิ่ม layer เข้าไปอีก 3 layer คือ AveragePooling2D, Dropout layer และ Flatten layer

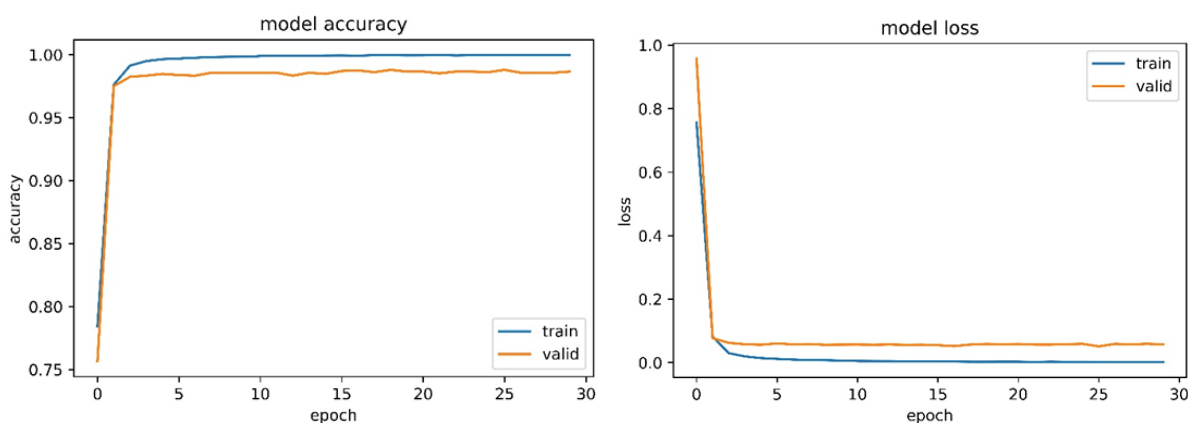
4.3 ทำการเทรนแต่ละโมเดลโดยกำหนด input size, epoch, learning rate, optimizer และ batch size ที่เหมือนกัน

4.5 กำหนด optimizer แบบ Stochastic gradient descent หรือ SGD และกำหนด momentum = 0.9 learning rate = 0.01 และ epoch = 30

จนได้โมเดลที่มี Top1-accuracy และ Top5-accuracy ที่ดีที่สุดในแต่ละโมเดลที่มีรายละเอียดดังต่อไปนี้

- InceptionV3

ผู้วิจัยได้โมเดล InceptionV3 ที่ดีที่สุดโดยมีชื่อว่า InceptionV3_v10 ซึ่งมีจำนวน parameter ทั้งหมด 21,892,940 พารามิเตอร์ ขนาดโมเดล 167MB และได้ผลลัพธ์ accuracy ดังนี้



รูปภาพที่ 14 : กราฟ Training accuracy และ Training loss ของ InceptionV3_v10

จากกราฟในรูปภาพที่ 12 แสดงถึง Training accuracy และ Training loss ของ InceptionV3_v10 ซึ่งเห็นได้ว่า โมเดล InceptionV3_v10 มี Training accuracy อยู่ในระดับที่สูง และเมื่อทำการเทรนโมเดลใน epoch ที่มากขึ้นเรื่อยๆ Training accuracy ก็ยังคงไม่ต่ำลง เช่นเดียวกัน กับ Training loss ที่พบว่า InceptionV3_v10 มีค่า loss ที่น้อยลงใน epoch มากขึ้น

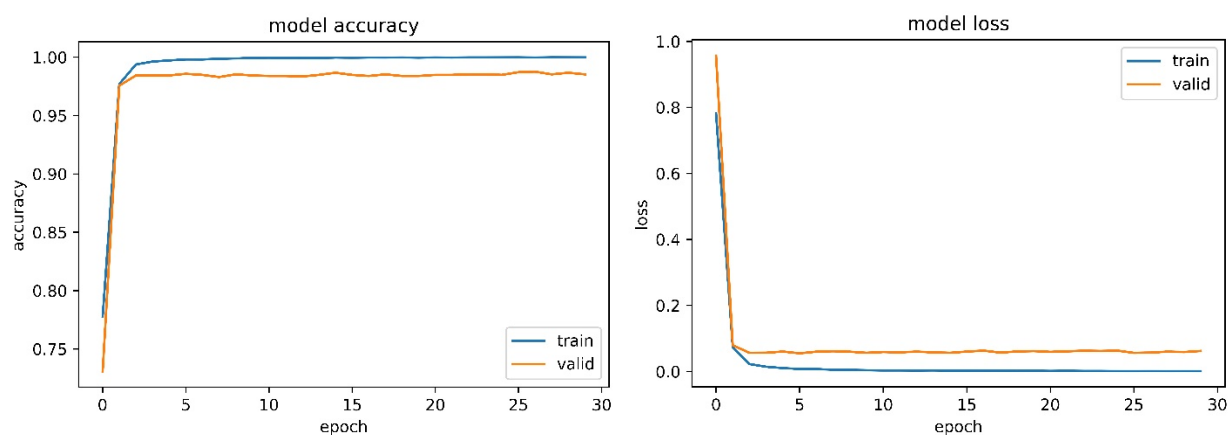
Model	Top1-accuracy	Top5-accuracy
InceptionV3_v10	97.23%	98.93%

ตารางที่ 4 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ InceptionV3_v10

จากตารางที่ 3 แสดงถึง Test Top1-accuracy และ Top5-accuracy ของ InceptionV3_v10 ซึ่งเห็นได้ว่า โมเดล InceptionV3_v10 มี Test Top1-accuracy และ Top5-accuracy อยู่ในระดับที่สูงถึง 97.23% และ 98.93% ซึ่งสอดคล้องกับ Training accuracy ที่มี accuracy ในระดับที่สูงเช่นกัน และ จากรูปภาพที่ 13 แสดงถึง Confusion matrix ของ InceptionV3_v10 ซึ่งก็พบว่าการทำนายรูปภาพอาหารอีสานของโมเดล InceptionV3_v10 นั้นมีการทำนายได้ถูกต้องแม่นยำเมื่อพิจารณาในการทำนายในแต่ละคลาสอีกด้วย

- ResNet50

ผู้วิจัยได้โมเดล ResNet50 ที่ดีที่สุดโดยมีชื่อว่า resnet_v2 ซึ่งมีจำนวน parameter ทั้งหมด 23,677,868 พารามิเตอร์ ขนาดแบบจำลอง 181MB และได้ผลลัพธ์ accuracy ดังนี้



รูปภาพที่ 16 : กราฟ model accuracy และ model loss ของ resnet_v2

จากกราฟในรูปภาพที่14 แสดงถึง Training accuracy และ Training loss ของ resnet_v2 ซึ่งเห็นได้ว่า โมเดล resnet_v2 มี Training accuracy อยู่ในระดับที่สูงและเมื่อทำการเทรน โมเดลใน epoch ที่มากขึ้นเรื่อยๆ Training accuracy ก็ยังคงไม่ต่ำลง เช่นเดียวกันกับ Training loss ที่พบว่า resnet_v2 มีค่า loss ที่น้อยลงใน epoch มากขึ้น

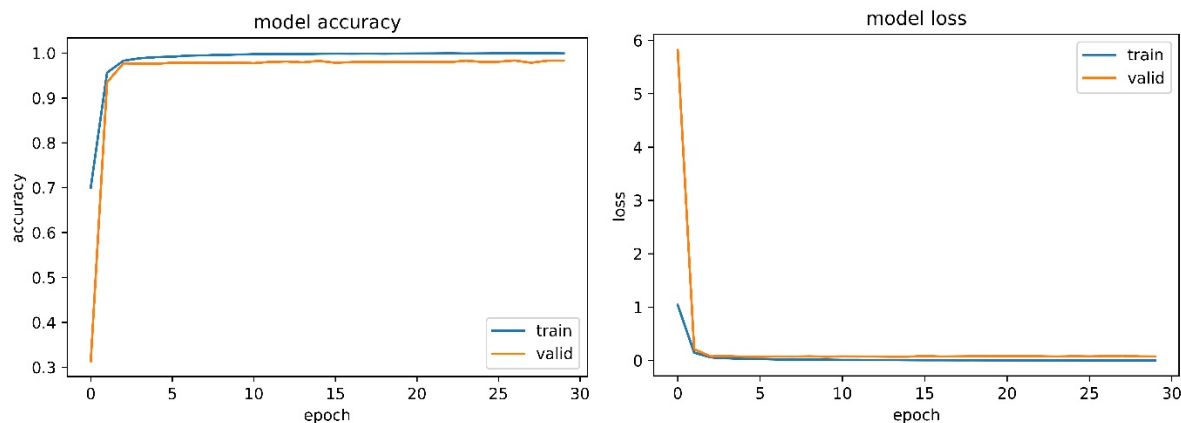
Model	Top1-accuracy	Top5-accuracy
ResNet50	97.25%	99.04%

ตารางที่ 5 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ resnet_v2

จากตารางที่ 4 แสดงถึง Test Top1-accuracy และ Top5-accuracy ของ resnet_v2 ซึ่งเห็นได้ว่า โมเดล resnet_v2 มี Test Top1-accuracy และ Top5-accuracy อยู่ในระดับที่สูงถึง 97.25% และ 99.04% ซึ่งสอดคล้องกับ Training accuracy ที่มี accuracy ในระดับที่สูงเช่นกัน และ จากรูปภาพที่ 15 แสดงถึง Confusion matrix ของ resnet_v2 ซึ่งก็พบว่าการทำนายรูปภาพอาหารอีสานของโมเดล resnet_v2 นั้นมีการทำนายได้ถูกต้องแม่นยำเมื่อพิจารณาในการทำนายในแต่ละคลาสอีกด้วย

- MobileNetV2

ผู้วิจัยได้โมเดล MobileNetV2 ที่ดีที่สุดโดยมีชื่อว่า mobilenet_v2 ซึ่งมีจำนวน parameter ทั้งหมด 2,314,348 พารามิเตอร์ ขนาดแบบจำลอง 18MB และได้ผลลัพธ์ accuracy ดังนี้



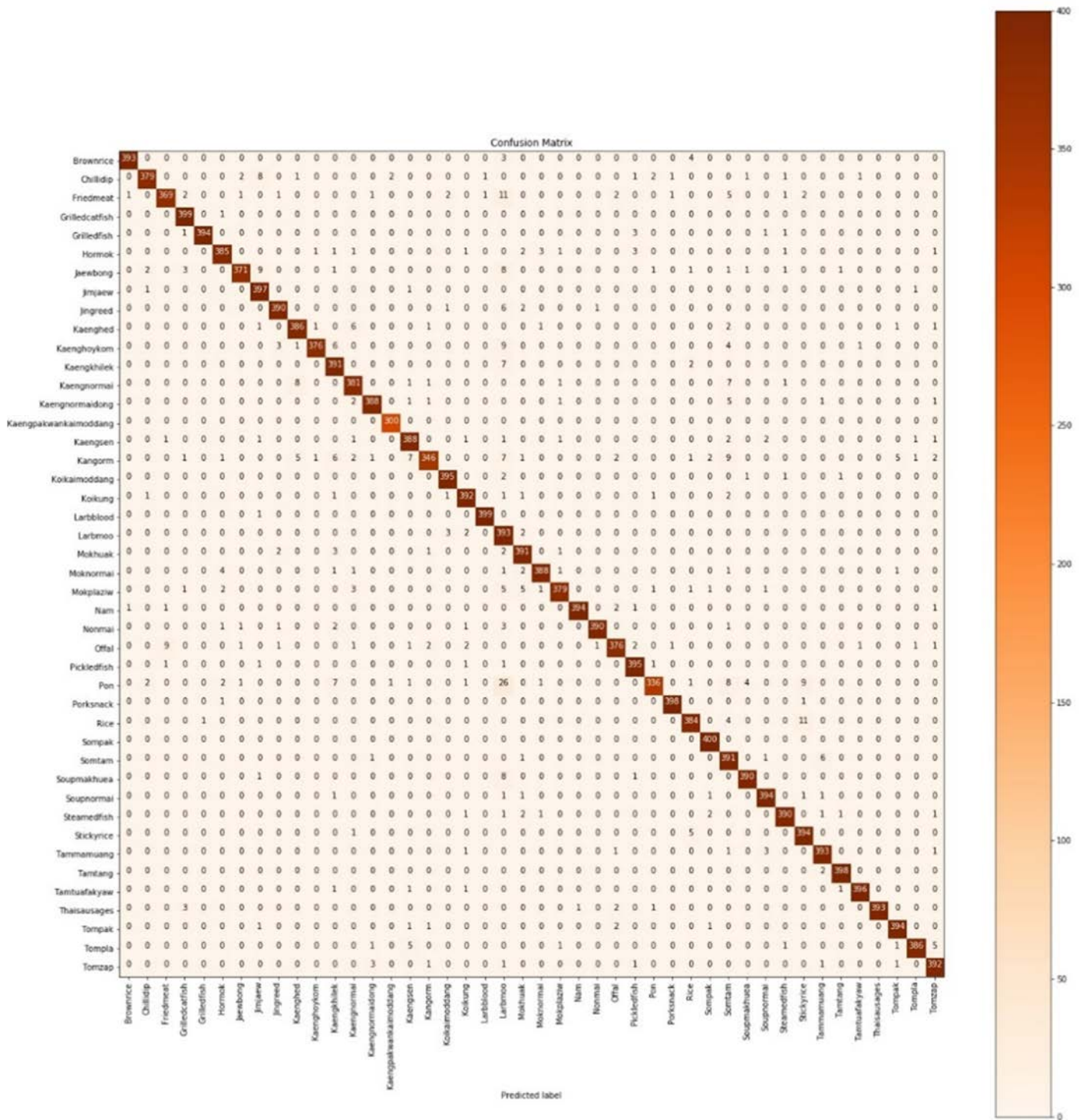
รูปภาพที่ 18 : กราฟ model accuracy และ model loss ของ mobilenet_v2

จากกราฟในรูปภาพที่16 แสดงถึง Training accuracy และ Training loss ของ mobilenet_v2 ซึ่งเห็นได้ว่า โมเดล mobilenet_v2 มี Training accuracy อยู่ในระดับที่สูงและเมื่อทำการเทรนโมเดลใน epoch ที่มากขึ้นเรื่อยๆ Training accuracy ก็ยังคงไม่ต่ำลง เช่นเดียวกันกับ Training loss ที่พบว่า mobilenet_v2 มีค่า loss ที่น้อยลงใน epoch มากขึ้น

Model	Top1-accuracy	Top5-accuracy
MobileNetV2	96.88%	98.86%

ตารางที่ 6 : ตารางแสดง Test Top1-accuracy และ Top5-accuracy ของ mobilenet_v2

และได้ Confusion matrix ดังนี้



รูปภาพที่ 19 : Confusion matrix ของ mobilenet_v2

จากตารางที่ 5 แสดงถึง Test Top1-accuracy และ Top5-accuracy ของ mobilenet_v2 ซึ่งเห็นว่า โมเดล mobilenet_v2 มี Test Top1-accuracy และ Top5-accuracy อยู่ในระดับที่สูงถึง 96.88% และ 98.86% ซึ่งสอดคล้องกับ Training accuracy ที่มี accuracy ในระดับที่สูงเช่นกัน และจากรูปภาพที่ 17 แสดงถึง Confusion matrix ของ mobilenet_v2 ซึ่งก็พบว่าการทำนายรูปภาพอาหารอีสานของโมเดล mobilenet_v2 นั้นมีการทำนายได้ถูกต้องแม่นยำเมื่อพิจารณาในการทำนายในแต่ละคลาสอีกด้วย

4.6 สรุปผลค่าความแม่นยำของทั้ง 3 แบบจำลอง

- Training : Top1-accuracy , Validation accuracy และ Model size

Model	Top1-accuracy	Validation	Model size
InceptionV3	100%	98.68%	167MB
MobileNetV2	100%	98.32%	18MB
ResNet50	100%	98.55%	181MB

รูปภาพที่ 20 : Training : Top1-accuracy , Validation accuracy และ Model size

- Testing : Top1-accuracy และ Top5-accuracy

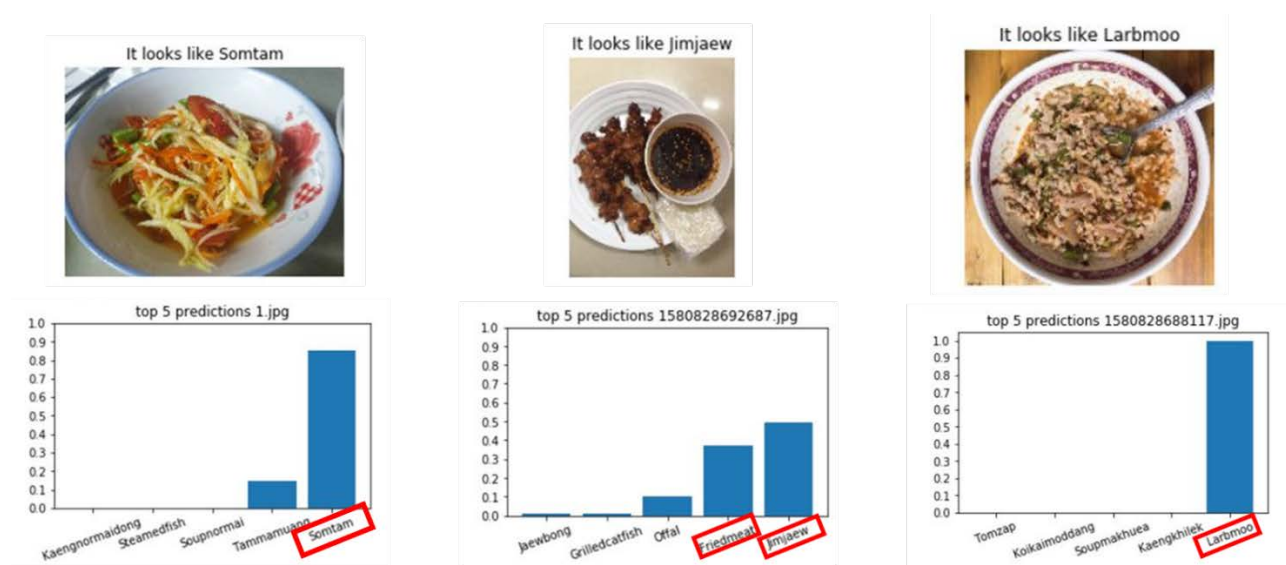
Model	Top1-accuracy	Top5-accuracy
InceptionV3	97.23%	98.93%
MobileNetV2	96.88%	98.86%
ResNet50	97.25%	99.04%

รูปภาพที่ 21 : Testing : Top1-accuracy และ Top5-accuracy

4.7 ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย

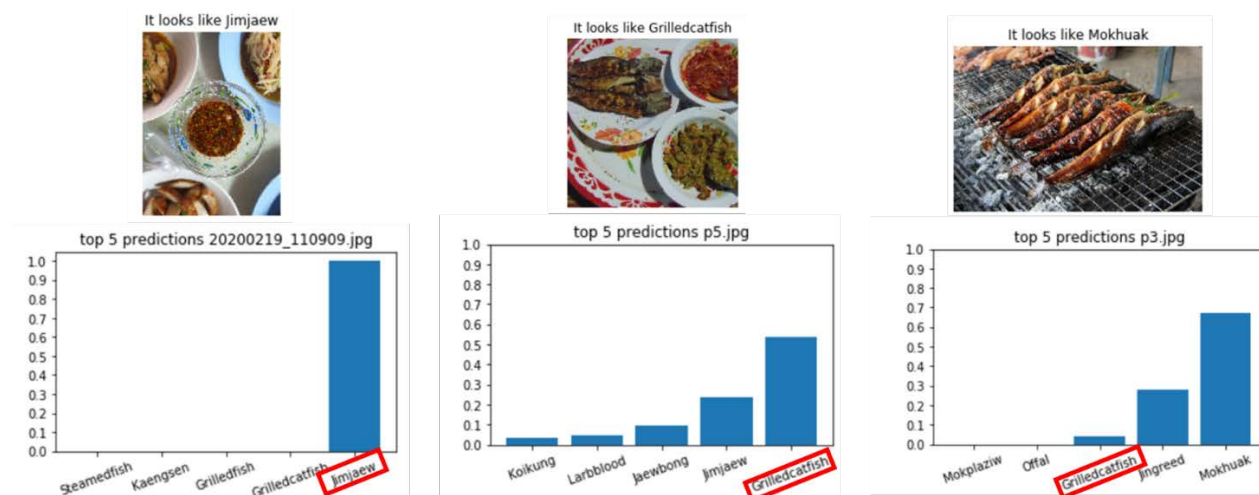
จากผลการทำนายรูปถ่ายอาหารอีสานด้วยภาพอาหารที่ผู้วิจัยได้ทำการถ่ายภาพเอง พบว่าทั้ง 3 แบบจำลองสามารถทำนายรูปภาพได้ถูกต้องเป็นส่วนใหญ่ บางรูปภาพสามารถทำนายได้ถูกต้องได้ตั้งแต่ Top1 ส่วนรูปภาพที่ทำนาย Top1 ได้ไม่ถูกต้อง ก็ยังคงพบผลลัพธ์ที่ถูกต้องแสดงอยู่ใน Top5 ดังภาพต่อไปนี้ โดยตารางด้านล่างคือการทำนายค่าความถูกต้องของโมเดลใน 5 ลำดับแรก (Top5-accuracy) และ label ในกรอบสีแดงคือชื่อของอาหารที่ถูกต้องจากการทำนายของโมเดล

- บนแบบจำลอง InceptionV3 : อาหารในภาพ คือ 1.ส้มตำ (Somtam) 2.จิ้มแจ่ว (JimJaew) 3.ลาบหมู (Larbmoos)



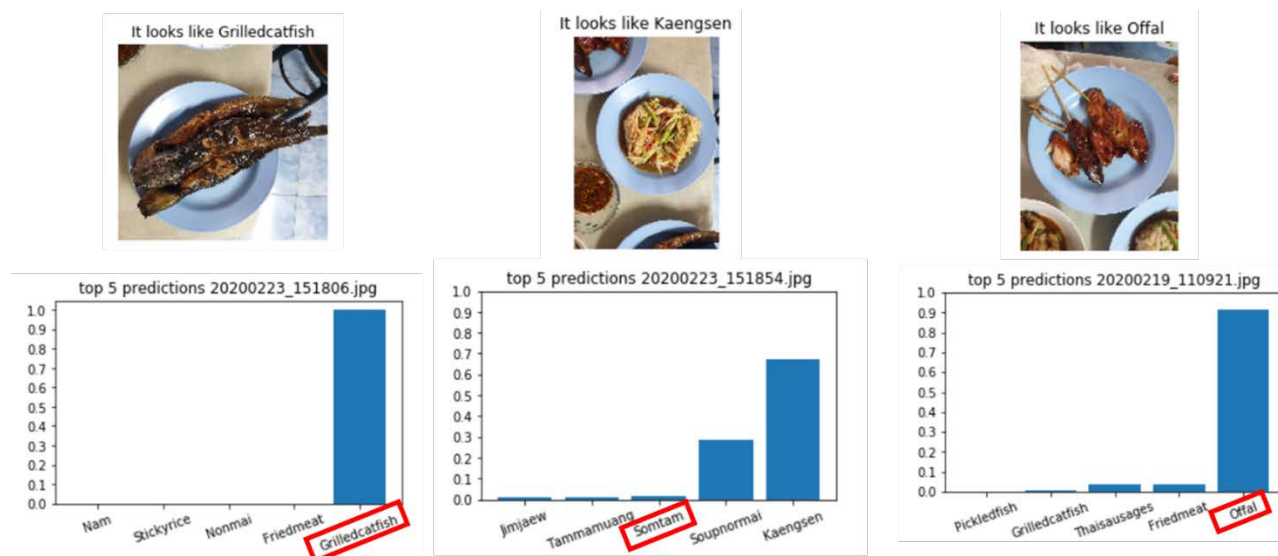
รูปภาพที่ 22 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง InceptionV3

- บนแบบจำลอง ResNet50 : อาหารในภาพ คือ 1.จิ้มแจ่ว (JimJaew) 2.ปลาตุ๋กและแจ่ว (Grilledcatfish, Jaewbong) 3.ปลาตุ๋ก (Grilledcatfish)



รูปภาพที่ 23 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง ResNet50

- บนแบบจำลอง MobileNetV2 : อาหารในภาพ คือ 1.ปลาตุ๋ก (Grilledcatfish) 2.ส้มตำ (Somtam) 3.เครื่องในย่าง (Offal)



รูปภาพที่ 24 : ตัวอย่างการทำนายโดยรูปถ่ายของผู้วิจัย บนแบบจำลอง MobileNetV2

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษาค้นคว้า

จากการศึกษาค้นคว้า และการพัฒนา SaapFood Cam: แอปพลิเคชันการรู้จำอาหารไทยอีสาน โดยภาพถ่ายจากอุปกรณ์พกพา (SaapFood Cam: A Thai-Isan Food Recognition Mobile Application) ผู้วิจัย ได้มีการสร้างฐานข้อมูล ISANTHFOOD-44 ที่มีการรวบรวมรูปภาพที่เป็นภาพถ่ายอาหารอีสานจากโครงการมณี และ search engine มาใช้ในการฝึกฝนและประเมินแบบจำลอง ซึ่งในฐานข้อมูลยังมีการใช้เทคนิค Data Augmentation มาใช้ในการเพิ่มจำนวนรูปภาพในฐานข้อมูลอีกด้วย และได้ทำการฝึกฝนทั้งหมด 3 แบบจำลอง ประกอบด้วย InceptionV3 ResNet50 และ MobileNetV2 ซึ่งใช้เทคนิคที่เรียกว่า Transfer learning มาใช้ในการฝึกฝน จากนั้นได้นำแบบจำลองที่ดีที่สุดของทั้ง 3 แบบจำลอง มาแปลงเป็นแบบจำลองในรูปแบบ .tflite เพื่อนำมาใช้ในแอปพลิเคชัน SaapFood Cam ทำให้แอปพลิเคชันนี้สามารถทำนายภาพถ่ายอาหารอีสานผ่านตัวแอปพลิเคชันและโชว์รายละเอียด ประกอบไปด้วย ชื่ออาหาร เปอร์เซ็นต์การทำนายแสดงผล 5 ลำดับแรกที่สูงที่สุด (Top5-accuracy) และ แคลอรี่ของอาหารในภาพถ่ายนั้นได้

โดยปกติแล้วงานวิจัยทางด้านการรู้จำอาหาร (Food recognition) โดยส่วนมากนั้น จะมีการรู้จำเฉพาะรูปภาพอาหารที่เป็นอาหารชื่อดังหรืออาหารเมนูธรรมดาโดยทั่วไป งานวิจัยนี้จึงได้มีการรู้จำอาหารในรูปแบบใหม่นั้นคืออาหารอีสาน ซึ่งแบบจำลองที่จัดทำขึ้นนั้นสามารถทำการรู้จำอาหารอีสานได้เป็นอย่างดี แต่ข้อยกเว้นในการทำนายรูปภาพอาหารที่พบจากงานวิจัยนั้น คือ แบบจำลองจะสามารถทำนายอาหารที่มีลักษณะที่สามารถเห็นได้ชัด แต่ถ้าสัดส่วนของอาหารในงานมีน้อยเกินไป เป็นอาหารมีลักษณะรูปลักษณ์ที่คล้ายกับเมนูอื่น หรือส่วนผสมต่างๆอยู่ภายใต้เนื้อน้ำแกง ก็สามารถทำให้แบบจำลองทำนายผิดพลาดได้เช่นกัน

5.2 อภิปรายผล

จากการดำเนินโครงการและทำการฝึกฝนแบบจำลองทั้ง 3 แบบจำลอง พบว่าแบบจำลอง ResNet50 ให้ค่าความแม่นยำที่ดีที่สุด (Top1-accuracy) อยู่ที่ 97.25% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) ที่ดีที่สุดอยู่ที่ 99.04% แบบจำลอง InceptionV3 ให้ค่าความแม่นยำ (Top1-accuracy) อยู่ที่ 97.23% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) อยู่ที่ 98.93% และแบบจำลองสุดท้าย MobileNetV2 ให้ค่าความแม่นยำ (Top1-accuracy) อยู่ที่ 96.88% และให้ค่าความแม่นยำของการทำนายที่ปรากฏอยู่ในการทำนาย 5 อันดับแรก (Top5-accuracy) อยู่ที่ 98.86% ทั้ง 3 โมเดลเมื่อแปลงให้อยู่ใน รูปแบบ .tflite สามารถนำมาใช้ในตัวแอปพลิเคชันได้โดยไม่ใช้เวลาในการประมวลผลที่นานเกินไป ดังนั้นแบบจำลองที่ถูกเลือกนำมาใช้ในแอปพลิเคชัน SaapFood Cam คือแบบจำลอง MobileNetV2 เนื่องจากเป็นแบบจำลองที่มีขนาดเล็กที่สุด อยู่ที่ 18MB และยังให้ค่าความแม่นยำในการทำนายที่ดีไม่น้อยกว่า InceptionV3 และ ResNet50 ซึ่งหลังจากทำการติดตั้งแบบจำลอง เข้าไปในแอปพลิเคชันพบว่าแอปพลิเคชันมีขนาดเพียง 15.8 MB และมีการประมวลผลที่รวดเร็วอีกด้วย

Model	Average Accuracy		Parameters	Model size (MB)
	Top1 (%)	Top5 (%)		
InceptionV3	97.23%	98.93%	21,892,940	167
ResNet50	97.25%	99.04%	23,677,868	181
MobileNetV2	96.88%	98.86%	2,314,348	18

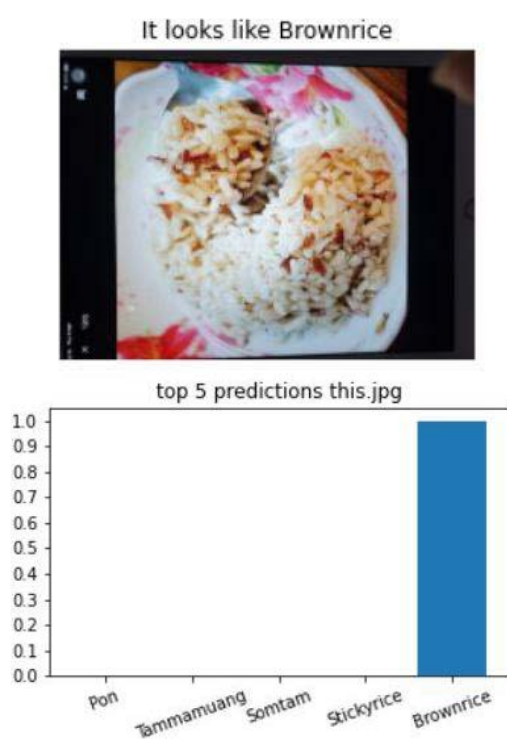
ตารางที่ 7 : ตารางสรุปผลลัพธ์ในแต่ละแบบจำลอง

5.3 ข้อเสนอแนะ

แบบจำลองทั้ง 3 แบบจำลองนี้จะทำนายได้ถูกต้องเมื่ออาหารนั้นเป็นอาหารอีสานเท่านั้น อีกทั้งรูปภาพที่ใช้ในการทำนายหากมีสภาพแวดล้อมแตกต่างกัน เช่น ภาพถ่ายในบริเวณที่มีแสงน้อย ภาพถ่ายที่มีสิ่งของวางอยู่ ภาพถ่ายที่มีปริมาณของอาหารภายในรูปที่น้อยเกินไป เป็นอาหารมีลักษณะรูปลักษณะที่คล้ายกับเมนูอื่น หรือส่วนผสมต่างๆอยู่ภายใต้้น้ำแกง จะสามารถทำให้การทำนายของตัวโมเดลคลาดเคลื่อนหรือผิดพลาดได้ รวมทั้งการทำนายรูปภาพอาหารอีสานผ่านแอปพลิเคชันยังพบ ข้อจำกัดในการทำงาน (Limitation) ดังข้อที่ 5.4

5.4 ข้อจำกัดในการทำงาน (Limitation)

แอปพลิเคชัน SaapFood Cam นี้ยังมีข้อจำกัดในการทำงาน คือ การทำนายรูปภาพอาหารอีสานโดยใช้แบบจำลองผ่านแอปพลิเคชันยังทำนายผลชื่ออาหารได้ไม่ถูกต้องเท่าที่ควร บางรูปภาพสามารถทำนายได้อย่างถูกต้อง แต่บางรูปภาพยังมีการทำนายที่ผิด เมื่อเทียบกับการทำนายรูปภาพอาหารอีสานผ่านแบบจำลองโดยตรงใน Google Colab ที่มีการทำนายที่ถูกต้องและแม่นยำกว่ามาก



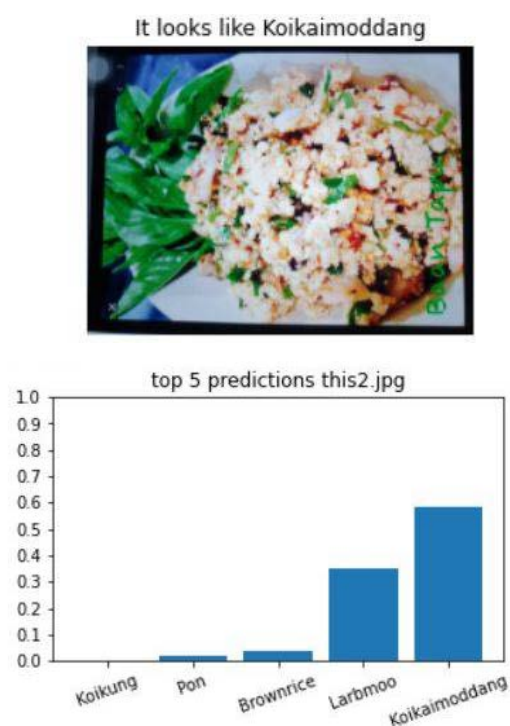
รูปภาพที่ 25 : รูปการทำนายผลจากโมเดล

อาหารอีสานในภาพนี้คือ

The name of food in this image is

ชื่ออาหาร	เปอร์เซ็นต์	แคลอรี
ข้าวเจ้า	(16.68%)	80 / 1กิโลกรัม
หอมกขวก	(13.75%)	100 / 1จาน
แกงผักหวานไข	(10.61%)	100 / 1ถ้วย
แกงหอยขม	(9.11%)	44 / 1ถ้วย
จังหวัด	(7.60%)	133 / 100กรัม

รูปภาพที่ 26 : รูปการทำนายผลจากแอปพลิเคชัน



รูปภาพที่ 27 : รูปการทำนายผลจากโมเดล

อาหารอีสานในภาพนี้คือ

The name of food in this image is

ชื่ออาหาร	เปอร์เซ็นต์	แคลอรี
ทอดหมกฮวก	(48.98%)	100 / 1จาน
แกงผักหวานไข	(18.55%)	100 / 1ถ้วย
ป่น	(7.03%)	35 / 1ถ้วย
จิ้งหรีด	(6.51%)	133 / 100กรัม
น้ำพริก	(3.08%)	35 / 1ถ้วย

รูปภาพที่ 28 : รูปการทำนายผลจากแอปพลิเคชัน

จะเห็นได้ว่าการทำนายรูปภาพอาหารโดยผ่านโมเดลโดยตรง ในรูปภาพที่ 18 และ 20 จะให้ผลลัพธ์ที่ถูกต้องกว่าการทำนายผ่านโมเดลที่เชื่อมในตัวแอปพลิเคชัน ในรูปภาพที่ 19 และ 21

5.5 การต่อยอดโครงการวิจัย

งานวิจัยนี้สามารถนำไปต่อยอดในอนาคตได้ เช่น การนำชื่ออาหารที่ได้ไปแสดงรายละเอียดต่างๆ ให้แก่ผู้ใช้ได้ทราบข้อมูล อาทิ ส่วนผสมของอาหารนั้น หรือระดับความเค็มและความหวานโดยทั่วไปของอาหาร เป็นต้น เพิ่มการตรวจจับตำแหน่งของอาหารภายในจานหรือตรวจจับปริมาณของอาหารเพื่อการทำนายที่ดีขึ้น หรือเพิ่มการตรวจจับความเค็มโดยดูจากสีอาหารภายในภาพเพื่อแสดงให้กับผู้ใช้ได้ทราบ

บรรณานุกรม

- [1] P. C. Pongskul, “Khon kaen smart living in khon kaen smart city.”
- [2] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 7132–7141, 2018.
- [3] I. Shota Horiguchi , Member, IEEE, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa , Fellow, “Personalized Classifier for Food Image Recognition,” pp. 1-13, 2018.
- [4] E. Aguilar, M. Bolaños, and P. Radeva, “Food Recognition Using Fusion of Classifiers Based on CNNs,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10485 LNCS. pp. 213–224, 2017.
- [5] C. Termritthikun, P. Muneesawang, and S. Kanprachar, “NU-InNet: Thai food image recognition using Convolutional Neural Networks on smartphone,” J. Telecommun. Electron. Comput. Eng., vol. 9, no. 2–6, pp. 63–67, 2017.
- [6] F. Chollet, “Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek.” p. 447, 2018.
- [7] L. Liu et al., “Deep Learning for Generic Object Detection: A Survey,” Sep. 2018.
- [8] “Machine Learning Glossary | Google Developers.” [Online]. Available: <https://developers.google.com/machine-learning/glossary/#accuracy>. [Accessed: 13-Sep-2019].
- [9] D. Sahoo et al., “FoodAI,” Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '19, no. May, pp. 2260–2268, 2019.
- [10] X. Chen, Y. Zhu, H. Zhou, L. Diao, and D. Wang, “ChineseFoodNet: A large-scale Image Dataset for Chinese Food Recognition,” pp. 1–8, 2017.

- [11] Z. Heng, S. Roy, K. Yap, A. Kot, and L. Duan, “Personalized Knowledge Distillation-based Mobile Food Recognition,” pp. 22–28.
- [12] R. Sullivan and R. Sullivan, “Classification and Prediction,” *Introd. to Data Min. Life Sci.*, pp. 455–500, 2012.
- [13] P.-N. Tan, M. Steinbach, and V. Kumar, “Classification : Basic Concepts , Decision Trees , an model evaluation,” *Introd. to Data Min.*, vol. 67, no. 17, pp. 145–205, 2006.
- [14] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” *Cvpr*, vol. 91, no. 8, pp. 2322–2330, 2016.
- [15] G. A. Hembury, V. V. Borovkov, J. M. Lintuluoto, and Y. Inoue, “Deep Residual Learning for Image Recognition Kaiming,” *Chem. Lett.*, vol. 32, no. 5, pp. 428–429, 2003.
- [16] Y. Lu, “Food Image Recognition by Using Convolutional Neural Networks (CNNs).”
- [17] โครงการวิจัย การทำฐานข้อมูลของอาหาร จากกลุ่มตัวอย่างอาสาสมัคร หมู่ที่ 22 บ้านหนองหิน อำเภอเมือง จังหวัดขอนแก่น ในการประมาณปริมาณแคลอรีที่ได้รับจากการรับประทานอาหาร (โครงการมณี) โดย นาย ศราวุฒิ ชุ่มบัวทอง มหาวิทยาลัยขอนแก่น

ภาคผนวก

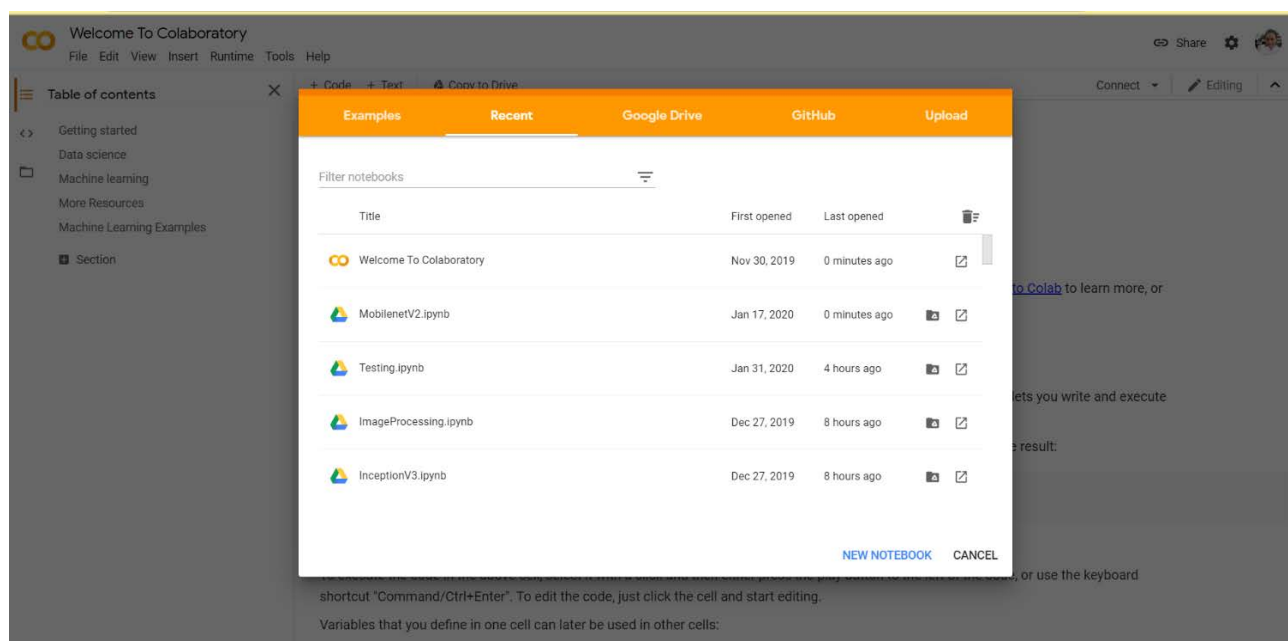
ภาคผนวก ก

การติดตั้งโปรแกรม

การใช้งานโปรแกรม Google Colab และ Android Studio

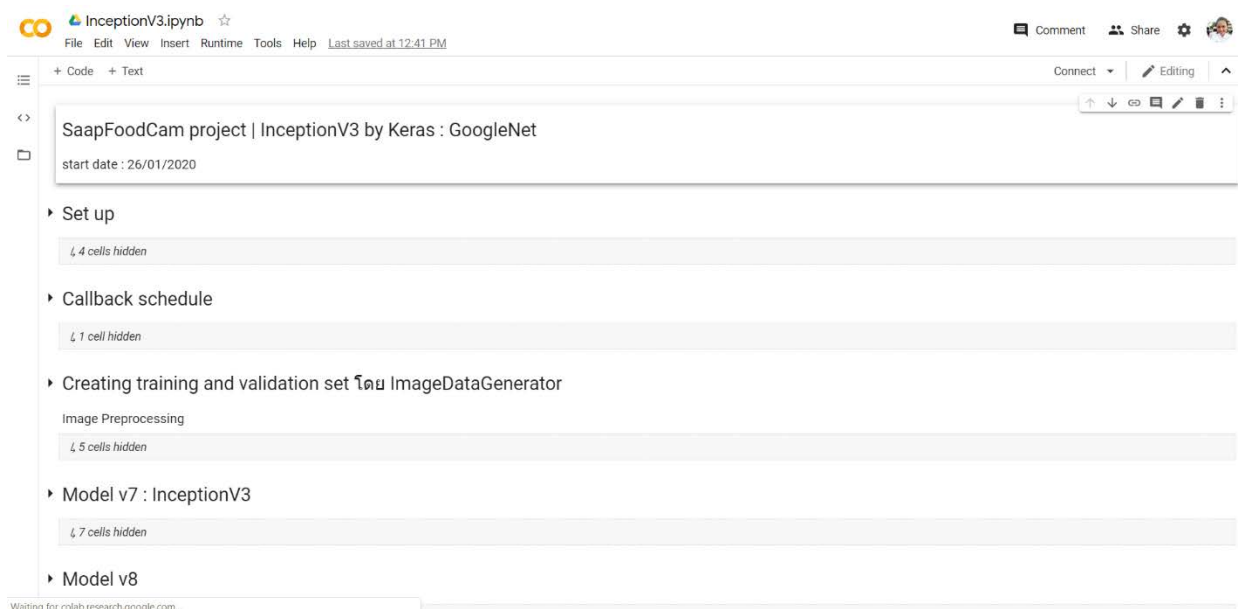
การใช้งานโปรแกรม Google Colab

1. ทำการ Login เข้าสู่ระบบ Gmail เพื่อเข้าใช้งาน Google Colab จากนั้น เข้าไปที่เว็บไซต์ <https://colab.research.google.com/notebooks/intro.ipynb#recent=true> และทำการสร้างพื้นที่ในการสร้างแบบจำลอง หรือ กด New Notebook



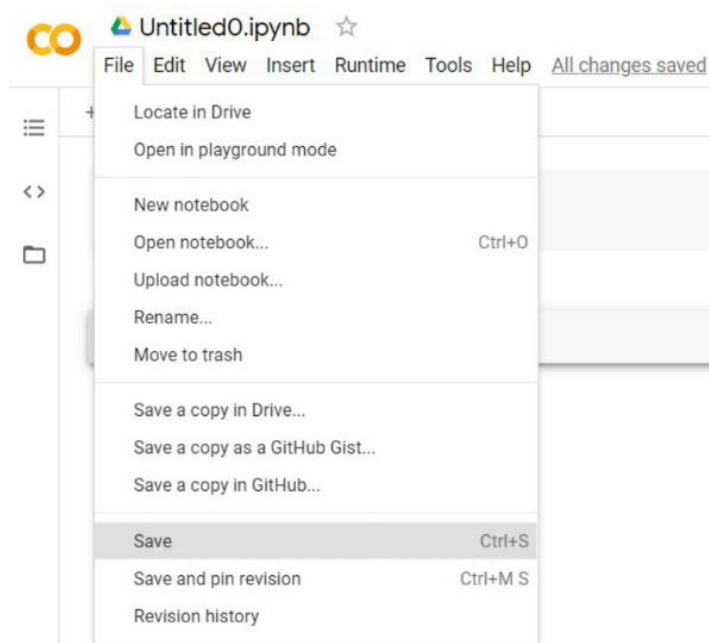
รูปที่ ก-1 การเข้าใช้งาน Google Colab

2. ทำการสร้างแบบจำลองต่างๆ (coding) และทำการรันโค้ดเพื่อดูผลลัพธ์



ก-2 การใช้งาน Google Colab

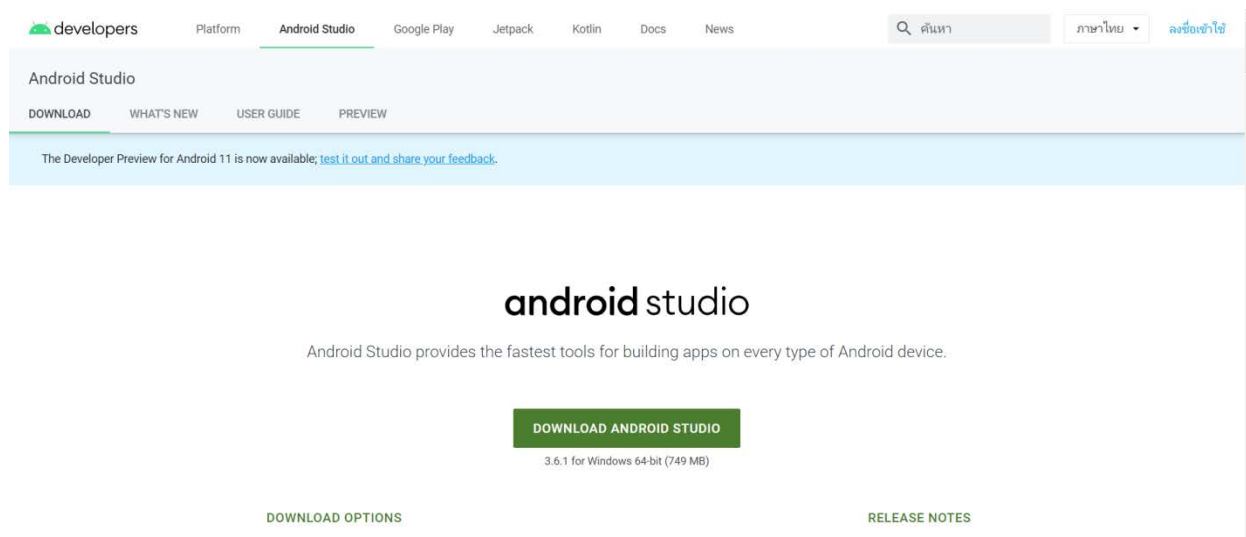
3. เมื่อพิมพ์โค้ดทั้งหมดเสร็จเรียบร้อยแล้ว ต้องการบันทึกไฟล์ ให้ไปที่คำสั่ง File -> Save



รูปที่ ก-3 การใช้งาน Google Colab

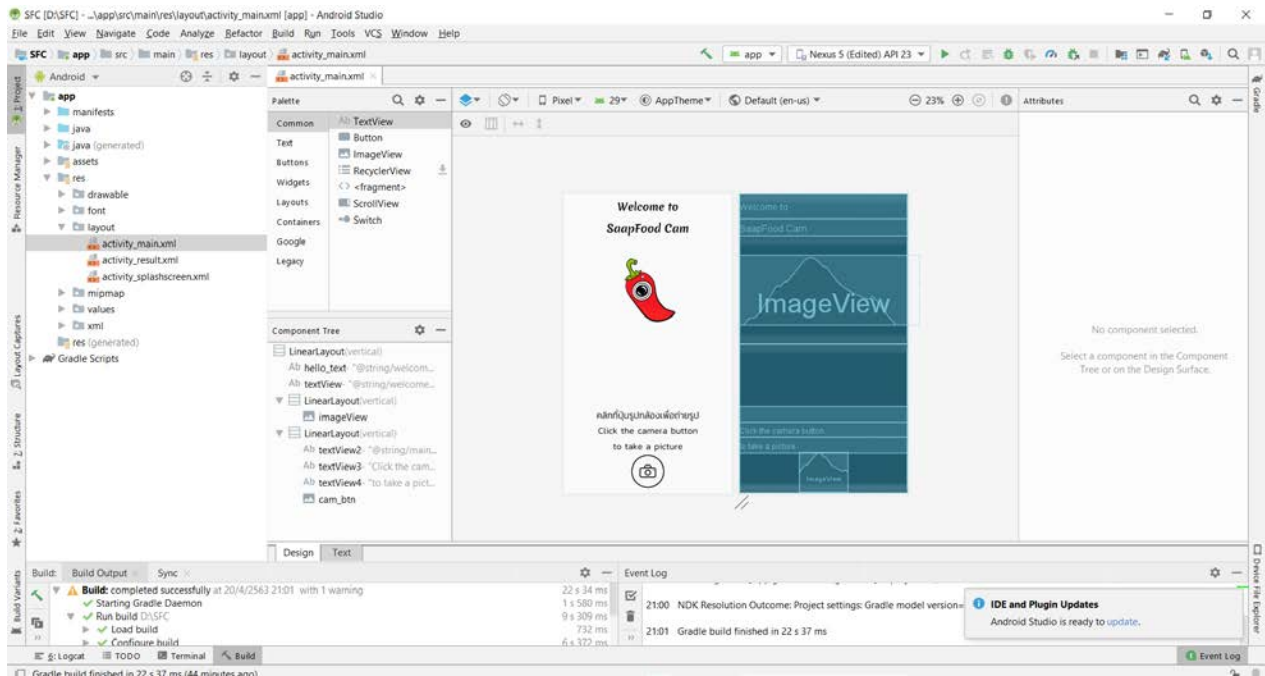
การใช้งานโปรแกรม Android Studio

1. ดาวน์โหลด โปรแกรม Android Studio จากเว็บไซต์ <https://developer.android.com/studio> โดยคลิกที่ปุ่ม ซึ่งหน้าตาของเว็บไซต์ เป็นดังนี้



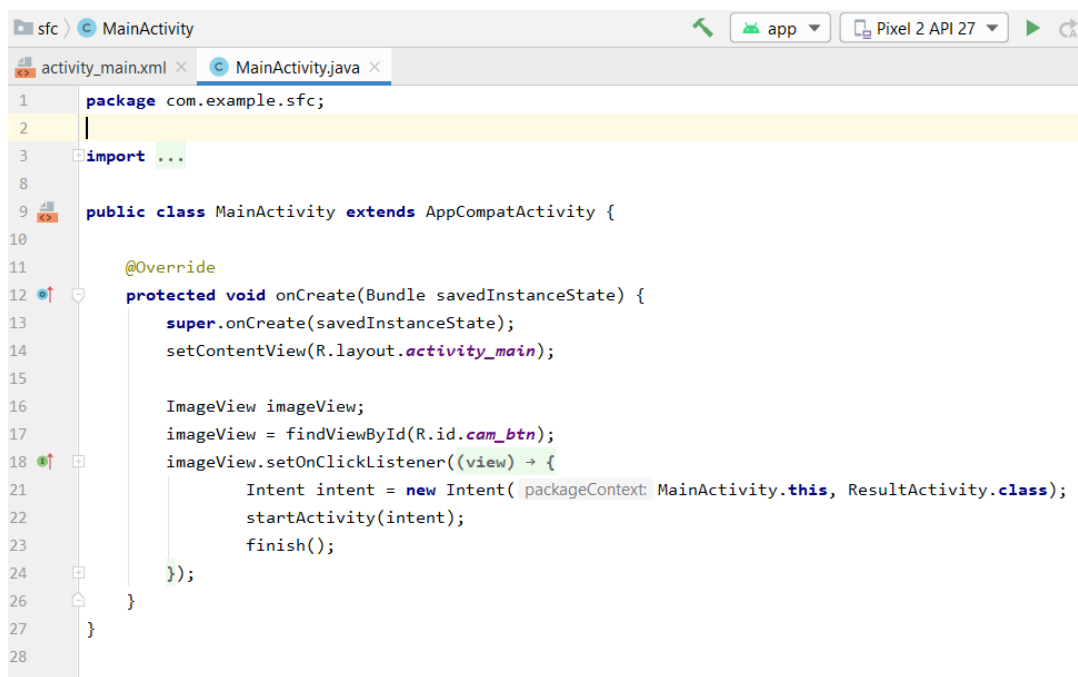
รูปที่ ก-4 การใช้งาน Android Studio

2. เมื่อทำการดาวน์โหลดโปรแกรมเสร็จเรียบร้อยแล้ว จากนั้นคลิกเปิดโปรแกรม Android Studio จะปรากฏโครงสร้างพื้นฐานของโปรแกรม อาทิเช่น แบบจำลอง Layout ซึ่งจะปรากฏรูปร่างหน้าตาของแอปพลิเคชันในโทรศัพท์มือถือ ก่อนนำไปใช้งานจริง โดยตัวอย่าง Layout จะมีลักษณะดังนี้



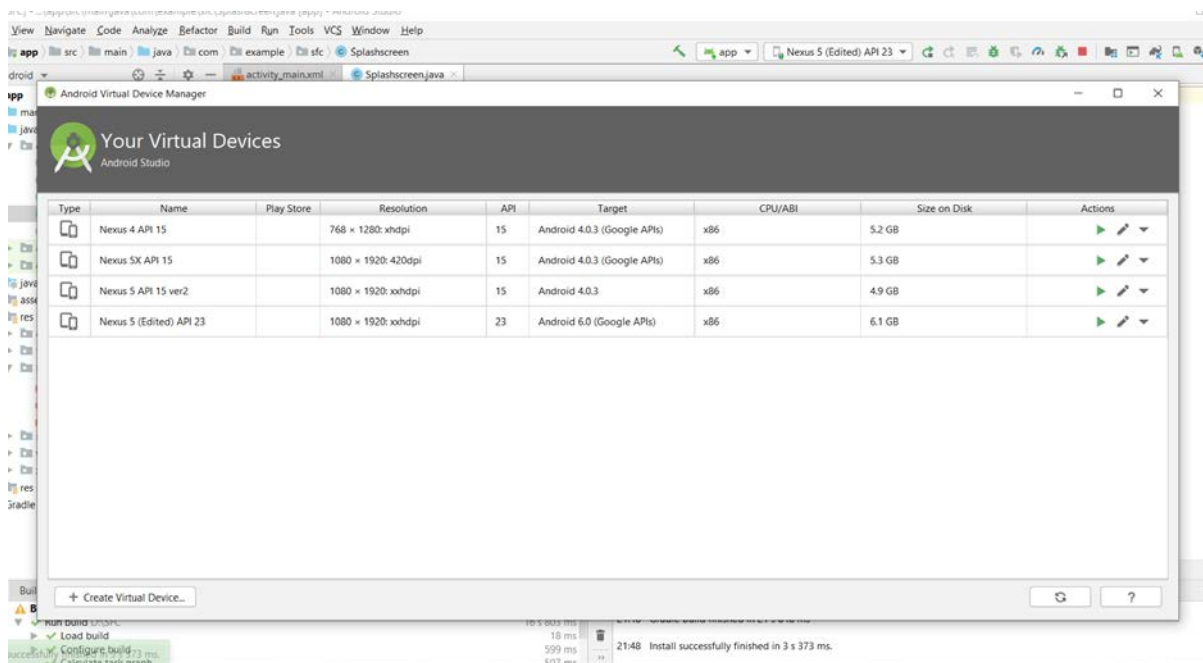
รูปที่ ก-5 การใช้งาน Android Studio

3. การเขียนโค้ดสามารถเขียนได้หลายภาษา อาทิ เช่น ภาษา XML และ Java



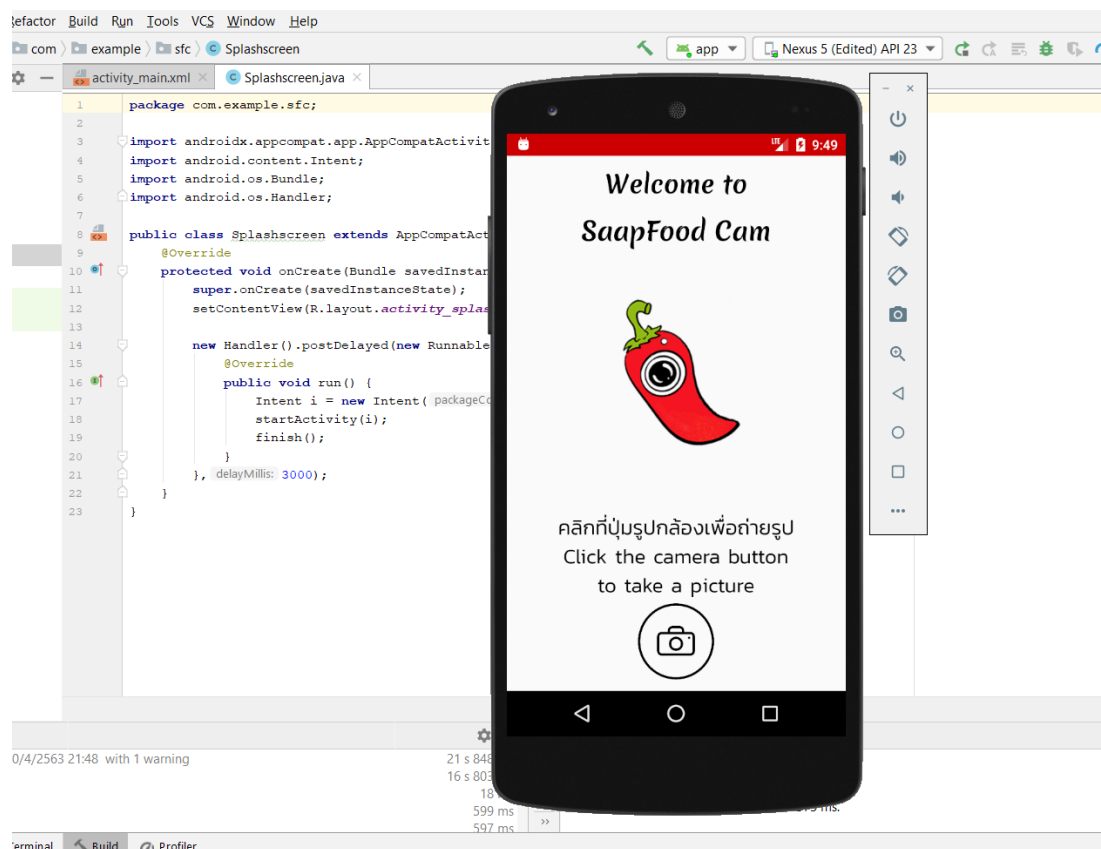
รูปที่ ก-6 การใช้งาน Android Studio

4. ก่อนรันโค้ดให้ทำการเลือกรุ่นของแบบจำลองหรือโทรศัพท์ที่ต้องการจะติดตั้งและทดลองแอปพลิเคชัน



รูปที่ ก-7 การใช้งาน Android Studio

5. เมื่อเขียนโค้ดทั้งหมดเรียบร้อยแล้ว สามารถทดสอบแอปพลิเคชันที่ปุ่ม  จะปรากฏผลลัพธ์เป็นดังนี้



รูปที่ ก-8 การใช้งาน Android Studio

ภาคผนวก ข

ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

1. ดาวน์โหลดแอปพลิเคชัน SaapFood Cam ลงบนโทรศัพท์มือถือ



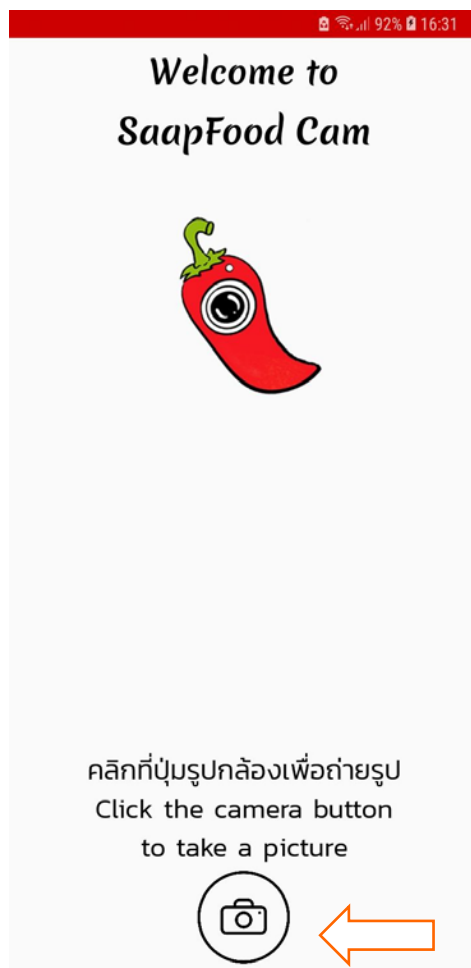
รูปที่ ข-1 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

2. กดเข้าใช้งาน แอปพลิเคชัน SaapFood Cam จะเจอหน้า Splash Screen ขึ้นมา ซึ่งเป็นหน้าแสดงโลโก้ของแอปพลิเคชัน



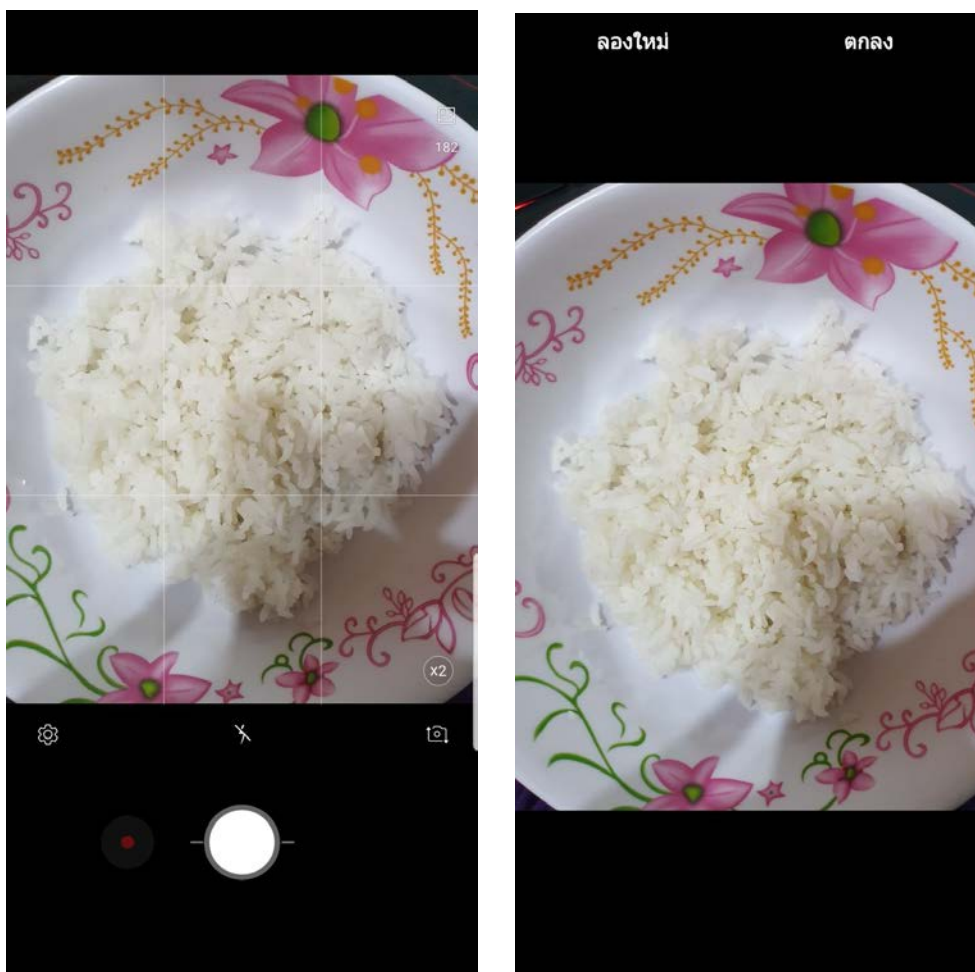
รูปที่ ข-2 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

3. หลังจากนั้นจะมีการขออนุญาตเข้าถึงแหล่งจัดเก็บข้อมูลของโทรศัพท์และกล้องถ่ายรูป ให้ผู้ใช้งานทำการกดอนุญาตการเข้าถึงและกดที่ปุ่มรูปกล้องถ่ายรูปเพื่อทำการถ่ายรูปอาหารอีสานที่ท่านต้องการให้โมเดลทำนาย



รูปที่ ข-3 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

4. ถ่ายรูปอาหารที่ต้องการ



รูปที่ ข-4 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

5. แอปพลิเคชันจะทำการประมวลผลสักครู่ หลังจากนั้นจะแสดงรายละเอียดของอาหารในรูปนั้น ประกอบไปด้วย ชื่ออาหาร เปอร์เซ็นต์การทำนายแสดงผล 5 ลำดับแรกที่สูงที่สุด (Top5-accuracy) และ แคลอรี่ ของอาหารนั้น



 93%
  16:48



อาหารอีสานในภาพนี้คือ

The name of food in this image is

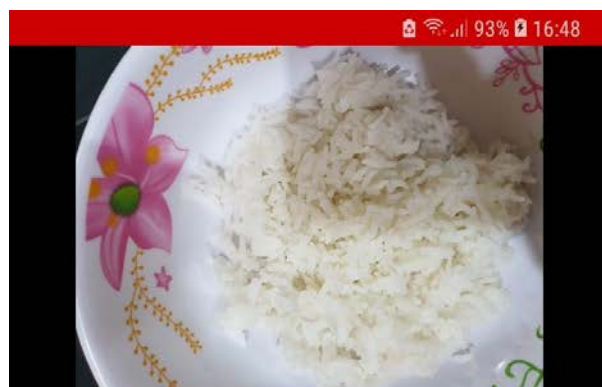
ชื่ออาหาร	เปอร์เซ็นต์	แคลอรี่
ข้าวเจ้า	(15.39%)	80 / 1กัฟพี
ข้าวเหนียว	(10.10%)	160 / 1กัฟพี
จังหวัด	(4.51%)	133 / 100กรัม
ส้มตำ	(4.29%)	80 / 1จาน
ลาบหมู	(4.06%)	119 / 1จาน

กลับไปยังหน้าหลัก

BACK TO HOME

รูปที่ ข-5 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

6. เมื่อผู้ใช้งานต้องการถ่ายรูปอีกครั้ง สามารถกดกลับไปเมนูหลัก เพื่อทำการถ่ายรูปอาหารอีสานใหม่อีกครั้ง



อาหารอีสานในภาพนี้คือ

The name of food in this image is

ชื่ออาหาร	เปอร์เซ็นต์	แคลอรี
ข้าวเจ้า	(15.39%)	80 / 1กัฟพี
ข้าวเหนียว	(10.10%)	160 / 1กัฟพี
จังหวัด	(4.51%)	133 / 100กรัม
ส้มตำ	(4.29%)	80 / 1จาน
ลาบหมู	(4.06%)	119 / 1จาน

กลับไปยังหน้าหลัก

BACK TO HOME

รูปที่ ข-6 ขั้นตอนการใช้งานแอปพลิเคชัน SaapFood Cam

ภาคผนวก ค

Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

1. ส่วนของการ import library และ directory ที่จำเป็น

SaapFoodCam project | MobilenetV2 by Keras

▼ Set up

```
[ ] # Connect to Drive
from google.colab import drive
drive.mount('/content/drive')

[ ] %%time
zip_path1 = 'drive/Shared drives/SaapFoodCamSeniorProject/ISANTHFOOD44/train.zip'
!cp '{zip_path1}' .
!unzip -q train.zip
!rm train.zip

zip_path2 = 'drive/Shared drives/SaapFoodCamSeniorProject/ISANTHFOOD44/valid.zip'
!cp '{zip_path2}' .
!unzip -q valid.zip
!rm valid.zip

[ ] # Import
import matplotlib.pyplot as plt
import matplotlib
import cv2
import pandas as pd
from keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img

import tensorflow.keras
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import save_model, load_model
import keras
from keras.models import Sequential, load_model
from keras.layers import Convolution2D, MaxPooling2D, Conv2D, BatchNormalization, GlobalAveragePooling2D, Input, concatenate, AveragePooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.optimizers import SGD, Adam
from keras.callbacks import EarlyStopping, ModelCheckpoint, History, LearningRateScheduler
from keras import backend as K
from keras.models import Model
from keras.metrics import top_k_categorical_accuracy
import math
from keras.applications.inception_v3 import preprocess_input
from keras.applications import MobileNetV2

tf.test.gpu_device_name()

[ ] tf.test.gpu_device_name()

[ ] # Directory
trainDir = 'train'
valDir = 'valid'
saveModel = '/content/drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/model'
saveModelW = '/content/drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/modelWeight'
savePic = '/content/drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/pic'
```

รูปที่ ค-1 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

2. ส่วนของการ save weight ที่ดีที่สุดในช่วงของการ train model

▼ Callback schedule ****

```
[ ] # Callback and Save Model Weights when Fit model
class decaylr_loss(keras.callbacks.Callback):
    def __init__(self):
        super(decaylr_loss, self).__init__()
    def on_epoch_end(self, epoch, logs={}):
        loss=logs.get('loss')
        print("loss: ",loss)
        old_lr = 0.001 #needs some adjustments
        new_lr= old_lr*np.exp(loss) #lr*exp(loss)
        print("New learning rate: ", new_lr)
        K.set_value(self.model.optimizer.lr, new_lr)

lrate = decaylr_loss()

#early stopping
#patience = 30
#earlystopper = EarlyStopping(monitor='val_acc', patience=patience, verbose=1, mode='max')

#check point
wdir = saveModelW
filepath = os.path.join(wdir, 'mobilenet_v2_weights.best.hdf5')
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')
```

รูปที่ ค-2 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

3. ส่วนของการสร้าง training batch และ validation batch เตรียมข้อมูลเพื่อใช้ในการ train

▼ Creating training and validation set โดย ImageDataGenerator

Image Preprocessing

```
[ ] # Initail #จะแบ่งเทรน epoch หรือครั้งละ batch_size รูป เช่น batch=32 แสดงว่า จะเทรนครั้งละ 32 เอามาครั้งละ 32 รูป
img_size = 256
batch_size = 32
```

```
[ ] # Training

imgDataGen = ImageDataGenerator()

train_data = imgDataGen.flow_from_directory( trainDir, target_size = (img_size, img_size), color_mode='rgb', batch_size = batch_size, class_mode='categorical')
valid_data = imgDataGen.flow_from_directory( valDir, target_size = (img_size, img_size), color_mode='rgb', batch_size = batch_size, class_mode='categorical', shuffle=False)
```

```
Found 44000 images belonging to 44 classes.
Found 2200 images belonging to 44 classes.
```

```
[ ] print(list(valid_data.class_indices.keys()))
```

```
['Brownrice', 'Chillidip', 'Friedmeat', 'Grilledcatfish', 'Grilledfish', 'Hormok', 'Jaewbong', 'Jimjaew', 'Jingreed', 'Kaenghed', 'Kaenghoykom', 'Kaengkhilek', 'Kaengnormai', 'Kaengnormm']
```

```
[ ] # train is 400 per class / valid is 50 per class
print(train_data.batch_size, 'batches train = ', train_data.n, 'imgs')
print(valid_data.batch_size, 'batches valid = ', valid_data.n, 'imgs')
```

```
32 batches train = 44000 imgs
32 batches valid = 2200 imgs
```

รูปที่ ค-3 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

4. ส่วนของการสร้าง model architecture

▼ Model

```
[ ] base_model = MobileNetV2(weights='imagenet',include_top=False)
x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x)
x=Dense(1024,activation='relu')(x)
x=Dense(512,activation='relu')(x)
preds=Dense(44,activation='softmax')(x)

mobilenet=Model(inputs=base_model.input,outputs=preds)
mobilenet.summary()
```

```
[ ] n_classes = 44

# base model is inception_v3 weights pre-trained on ImageNet
base_model = MobileNetV2(
    weights='imagenet',
    include_top=False,
    input_shape=(256,256,3)
)

x = base_model.output

# added layers to the base model
x = AveragePooling2D(pool_size=(6, 6))(x)
x = Dropout(.4)(x)
x = Flatten()(x)

# add softmax activation
predictions = Dense(n_classes, activation='softmax')(x)

mobilenet = Model(inputs=base_model.input, outputs=predictions)
```

```
[ ] mobilenet.summary()
```

รูปที่ ค-4 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

5. ส่วนของการ compile , fit model และ plot กราฟ accuracy และ loss

▼ Train

```
[ ] %%time
# optimizer
opt = SGD(lr=.01, momentum=.9)

# add LearningRateScheduler to update it
lr_scheduler = LearningRateScheduler(step_decay)

# calculate top_5_accuracy to evaluate the model
#def top_5_accuracy(y_true, y_pred):
#    return top_k_categorical_accuracy(y_true, y_pred, k=5)

# putting them together and compile the model
mobilenet.compile(
    optimizer=opt,
    loss='categorical_crossentropy',
    metrics=['accuracy', 'top_k_categorical_accuracy'] #top_5_accuracy
)

[ ] # Train
step_size_train = train_data.n//train_data.batch_size # steps_per_epoch
step_size_val = valid_data.n//valid_data.batch_size # validation_steps
print('each epoch is ',step_size_train,step_size_val)

[ ] # Fit
%%time
epochs = 30
hist = History()
mobilenet.fit_generator(train_data,
                        validation_data = valid_data,
                        steps_per_epoch = step_size_train,
                        validation_steps = step_size_val,
                        epochs = epochs,
                        verbose=1,
                        callbacks = [lr_scheduler, checkpoint, hist]) #earlystopper

# Save weight หรือยัง
# Save model ด้วย

[ ] # plot acc and loss vs epochs
print(hist.history.keys())

# accuracy
plt.plot(hist.history['acc'])
plt.plot(hist.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'valid']) #loc='best'
plt.savefig(os.path.join(savePic,'mobilenet_v2_accuracy.jpeg'), dpi=1000, bbox_inches='tight') #change
plt.show()

# loss
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'valid']) #loc='best'
plt.savefig(os.path.join(savePic,'mobilenet_v2_loss.jpeg'), dpi=1000, bbox_inches='tight') #change
plt.show()
```

รูปที่ ค-5 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

6. ส่วนของการ save model

▼ Save Model *

```
[ ] #save model
mobilenet.save_weights(os.path.join(saveModel, 'mobilenet_v2.h5')) # save weights after training or during training
mobilenet.save(os.path.join(saveModel, 'mobilenet_v2.h5')) #save complied model
```

```
[ ] ### saving the trained model
# serialize model to JSON
model_json = mobilenet.to_json()
with open("json/food44_mobilenet_v2_model.json", "w") as json_file:
    json_file.write(model_json)
```

```
[ ] %%time
scores_train = mobilenet.evaluate_generator(train_data, verbose=1) #steps=step_size_train
scores_test_valid = mobilenet.evaluate_generator(valid_data , verbose=1) #steps=step_size_val
```

```
[ ] scores_train[1]
```

```
[ ] scores_test_valid[1]
```

รูปที่ ค-6 Code ขั้นตอนในการฝึกฝนโมเดล (Model Training)

ภาคผนวก ง

Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

1. ส่วนของการ import library และ directory ที่จำเป็น

SaapFoodCam project | Testing

▼ Setting

```
[ ] # Connect to Drive
from google.colab import drive
drive.mount('/content/drive')

[ ] %%time
zip_path = 'drive/Shared drives/SaapFoodCamSeniorProject/ISANTHFOOD44/test.zip'
!cp '{zip_path}' .
!unzip -q test.zip
!rm test.zip

[ ] # Import
import matplotlib.pyplot as plt
import matplotlib
import cv2
import pandas as pd
from sklearn.metrics import confusion_matrix , accuracy_score
from keras.preprocessing.image import ImageDataGenerator,img_to_array, load_img
import itertools
from sklearn import preprocessing
from keras.preprocessing import image

import tensorflow.keras
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import save_model, load_model
import keras
from keras.models import Sequential, load_model
from keras.layers import Convolution2D, MaxPooling2D, Conv2D, BatchNormalization ,GlobalAveragePooling2D, Input, concatenate, AveragePooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.optimizers import SGD, Adam
from keras.callbacks import EarlyStopping, ModelCheckpoint, History, LearningRateScheduler
from keras import backend as K
from keras.models import Model
from keras.metrics import top_k_categorical_accuracy
import math
from keras.applications.inception_v3 import preprocess_input
from keras.applications import InceptionV3, MobileNetV2, ResNet50

tf.test.gpu_device_name()

[ ] # Directory
testDir = 'test'
savePredict = '/content/drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/predict/' #เปลี่ยนชื่อโฟลเดอร์เซฟ
```

รูปที่ ง-1 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

2. ส่วนของการสร้าง testing batch เตรียมข้อมูลเพื่อใช้ในการ test

▼ Test Batches

```
[ ] # Initail #จะแบ่งเทรน epoch หรือครั้งละ batch_size รูป เช่น batch=32 แสดงว่า จะเทรนครั้งละ 32 เอามาครั้งละ 32 รูป #32รูป ต่อ 1 epoch
img_size = 256
batch_size = 1

# Training
imgDataGen = ImageDataGenerator()
test_batches = imgDataGen.flow_from_directory(testDir, target_size = (img_size, img_size), color_mode='rgb', batch_size = batch_size, class_mode='categorical',shuffle=False)

Found 17500 images belonging to 44 classes.

[ ] food44 = list(test_batches.class_indices.keys())
print(food44)

['Brownrice', 'Chillidip', 'Friedmeat', 'Grilledcatfish', 'Grilledfish', 'Hormok', 'Jaewbong', 'Jimjaew', 'Jingreed', 'Kaenghed', 'Kaenghoykom', 'Kaengkhilek', 'Kaengnormai', '

```

รูปที่ ง-2 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

3. ส่วนของการโหลดโมเดลและ weight ที่ใช้ในการ test

▼ Load model & weight

```
[ ] #pip uninstall keras

[ ] #pip install keras

[ ] # enter model weight file
model_file = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/model/mobilenet_v2.h5'
model_weight = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/modelWeight/mobilenet_v2_weights.best.hdf5'

#model_file = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/inceptionv3/model/Inceptionv3_v10.h5'
#model_weight = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/inceptionv3/modelWeight/Inceptionv3_v10_weights.best.hdf5'

#model_file = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/resnet50/model/resnet_v2.h5'
#model_weight = 'drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/resnet50/modelWeight/resnet50_v2_weights.best.hdf5'

[ ] %%time
model = load_model(model_file)

CPU times: user 14 s, sys: 376 ms, total: 14.4 s
Wall time: 16.2 s

[ ] model.load_weights(model_weight)

[ ] model.summary()

[ ] scores_test = model.evaluate_generator(test_batches , test_batches.n // test_batches.batch_size, verbose=1, workers=1) # + 1
print("Test Accuracy = %.2f%%" % (scores_test[1]*100))

17500/17500 [=====] - 464s 27ms/step
Test Accuracy = 97.23%
```

รูปที่ ง-3 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

4. ส่วนของการทำนายรูปภาพที่ละภาพและแสดงผลลัพธ์ Top5

▼ Predict ที่ละภาพ

```
[ ] def path_to_tensor(img_path):
    # loads RGB image as PIL.Image.Image type
    img = image.load_img(img_path, target_size=(256, 256))
    # convert PIL.Image.Image type to 3D tensor with shape (256, 256, 3)
    x = image.img_to_array(img)
    # convert 3D tensor to 4D tensor with shape (1, 256, 256, 3) and return 4D tensor
    return np.expand_dims(x, axis=0)

[ ] savepath = '/content/s2/'

[ ] def img_analysis(img_path, namefood, plot=False):
    # process image
    img = path_to_tensor(img_path)

    # make prediction
    predicted_vec = model.predict(img)
    predicted_label = food44[np.argmax(predicted_vec)]

    # show predicted image
    img = cv2.imread(img_path)
    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.figure(figsize=(4,3))
    plt.imshow(rgb)
    plt.axis('off')
    plt.title("It looks like {}".format(predicted_label))
    #plt.savefig(os.path.join(savepath, '1'+namefood ))
    plt.show()

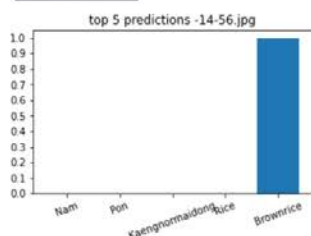
    # show top 5 predictions with probability
    if plot:
        # take top 5 probable pics
        top5_probs = np.sort(predicted_vec)[0][-5:]
        top5_labels = np.argsort(predicted_vec)[0][-5:]

        # plot bar graph
        x_pos = np.arange(len(top5_labels))
        plt.figure(figsize=(5,3))
        plt.bar(x_pos, top5_probs)
        plt.title("top 5 predictions " + namefood)
        plt.xticks(x_pos, [food44[int(idx)] for idx in top5_labels], rotation=20)
        plt.yticks(np.arange(0, 1.1, 0.1))
        #plt.savefig(os.path.join(savepath, namefood))
        plt.show()

[ ] #inceptionv3
foodimgDir = '/content/drive/Shared drives/SaapFoodCamSeniorProject/Inceptionv3pics/'
inceptfoodlist = os.listdir(foodimgDir)
print(inceptfoodlist)

[ ] for namefood in inceptfoodlist:
    img_analysis(os.path.join(foodimgDir, namefood), namefood, plot=True)
```

It looks like Brownrice



รูปที่ ง-4 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

5. ส่วนของการแสดงผล Top5-accuracy ของโมเดลที่ทำการ test

▼ Top1 and Top5 : show Top5 and save in csv

```
[ ] IMG_PATH = 'test'
    num_class = 17500 #number of all test imgs

[ ] def top5_accuracy(CLASSES):
    top5_df = pd.DataFrame(columns=['class','predicted_image','top5_accuracy(%)'])
    top5all = 0
    for c in CLASSES:
        cnt = 0
        images = os.listdir(os.path.join(IMG_PATH,c))
        for img in images:

            # process image
            imagepath = os.path.join(IMG_PATH,c)
            # loads RGB image as PIL.Image.Image type
            img = image.load_img(os.path.join(imagepath,img), target_size=(256, 256))
            # convert PIL.Image.Image type to 3D tensor with shape (256, 256, 3)
            x = image.img_to_array(img)
            # convert 3D tensor to 4D tensor with shape (1, 256, 256, 3) and return 4D tensor
            img = np.expand_dims(x, axis=0)

            # make prediction
            predicted_vec = model.predict(img)
            predicted_label = food44[np.argmax(predicted_vec)]

            # take top 5 probable pics
            top5_probs = np.sort(predicted_vec)[0][-5:]
            top5_labels = np.argsort(predicted_vec)[0][-5:]

            # check top5_label with class name
            for idx in top5_labels:
                if food44[int(idx)] == c:
                    cnt = cnt + 1

            if c == 'Kaengpakwankaimoddang':
                top5one = cnt/(300)
            else:
                top5one = cnt/(400)

            #print(c, 'in Top5 prediction = ', cnt , '/ top5 accuracy ',c, top5one*100, '%')
            top5_df = top5_df.append({'class': c, 'predicted_image': cnt, 'top5_accuracy(%)': top5one*100}, ignore_index=True)
            top5all = top5all + cnt
    top5_df.to_csv('/content/df/resnes_top5.csv')
    print('Summary top5 accuracy = ', (top5all/num_class)*100, 'from top5-predicted image ',top5all, ' images')

[ ] %%time
    top5_accuracy(food44)
```

รูปที่ ง-5 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

6. ส่วนของการแสดง confusion matrix

▼ Predict : show confusion matrix

```
[ ] def plot_confusion_matrix(cm, classes,
                             normalize=False,
                             title='Confusion matrix',
                             cmap=plt.cm.Oranges):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.figure(figsize=(20,20)) #20,20
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=90)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.savefig('/content/drive/Shared drives/SaapFoodCamSeniorProject/output/isanfood_44/mobilenet/predict/Normalize_confution_v2_01.jpg')
```

```
[ ] test_imgs, test_labels = next(test_batches)
```

```
[ ] test_labels = test_labels[:,0]
test_labels
```

```
array([1.], dtype=float32)
```

รูปที่ ง-6 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)

```
[ ] test_batches.reset() # resetting generator
    predictions = model.predict_generator(test_batches, test_batches.n // test_batches.batch_size, verbose=1) #+1
```

```
17500/17500 [=====] - 120s 7ms/step
```

```
[ ] y_pred = np.argmax(predictions, axis=1)
```

```
[ ] y_true = test_batches.classes
```

```
[ ] labels = (test_batches.class_indices)
```

```
[ ] labels2 = dict((v,k) for k,v in labels.items())
```

```
[ ] preds = [ labels2[k] for k in y_pred ]
```

```
[ ] print(y_pred) #ที่ทำนายมาแบบเลข
    print(y_true) #แต่ละคลาสคือ index อะไร 0-43
    print(labels) #นอกว่าแต่ละ index คืออะไร
    print(preds) #ที่ทำนายมาแบบ string
```

```
[ ] print(y_true.shape)
    print(y_pred.shape)
```

```
(17500,)
(17500,)
```

```
[ ] cm = confusion_matrix(y_true, y_pred)
```

```
[ ] cm_plot_labels = food44
    plot_confusion_matrix(cm, cm_plot_labels, title='Confusion Matrix')
```

```
accuracy_score(y_true, y_pred)
```

↑ ↓ ↺ 🗨 ⚙ 🗑 ⋮

รูปที่ ง-7 Code ขั้นตอนในการทดสอบโมเดล (Model Testing)