

การพัฒนาแอปพลิเคชันจำแนกความสุขผลทุเรียนจากเสียงเคาะโดยใช้โครงข่ายประสาทคอนโวลูชัน



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

APPLICATION DEVELOPMENT FOR DURIAN RIPENESS CLASSIFICATION FROM THE
KNOCKING SOUNDS USING CONVOLUTIONAL NEURAL NETWORK



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2019

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาแอปพลิเคชันจำแนกความสุกผลทุเรียนจากเสียง
	เคาะโดยใช้โครงข่ายประสาทคอนโวลูชัน
โดย	นายเวียงชัย คาระมาตย์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.มานพ วงศ์สายสุวรรณ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	
.....	ประธานกรรมการ
(รองศาสตราจารย์ ดร.วันเฉลิม โปธา)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.มานพ วงศ์สายสุวรรณ)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.เอกชัย ลีลารัมย์)	

CHULALONGKORN UNIVERSITY

เวียงชัย ภาระมาตย์ : การพัฒนาแอปพลิเคชันจำแนกความสุกผลทุเรียนจากเสียงเคาะ โดยใช้โครงข่ายประสาทคอนโวลูชัน. (APPLICATION DEVELOPMENT FOR DURIAN RIPENESS CLASSIFICATION FROM THE KNOCKING SOUNDS USING CONVOLUTIONAL NEURAL NETWORK) อ.ที่ปรึกษาหลัก : ผศ. ดร.มานพ วงศ์สายสุวรรณ

ทุเรียนเป็นหนึ่งในผลไม้ที่ได้รับความนิยมในการบริโภคภายในประเทศไทย การตรวจสอบความสุกของทุเรียนเป็นสิ่งสำคัญสำหรับการซื้อขายในตลาดผลไม้และการส่งออก การตรวจสอบความสุกทุเรียนสามารถตรวจสอบได้หลายวิธี ในวิทยานิพนธ์เล่มนี้ใช้วิธีเคาะและฟังเสียงเพื่อบอกความสุก ปกติการฟังเสียงจำเป็นต้องอาศัยผู้เชี่ยวชาญในการฟัง ดังนั้นจึงได้ออกแบบแอปพลิเคชันจำแนกความสุกผลทุเรียนจากเสียงเคาะโดยใช้โครงข่ายประสาทคอนโวลูชัน การเก็บข้อมูลเสียงเคาะทุเรียนได้รับคำแนะนำจากพ่อค้าคนกลางในจังหวัดจันทบุรีและบอกว่าทุเรียนที่เคาะมีความสุขในระดับใด การเก็บข้อมูลเสียงใช้ตัวอย่างทุเรียนพันธุ์หมอนทองจำนวน 30 ลูก แบ่งออกเป็น 3 กลุ่มตามระดับความสุก ได้แก่ ทุเรียนสุก ทุเรียนสุกปานกลาง และทุเรียนไม่สุก กลุ่มละ 10 ลูก จากนั้นนำเสียงเคาะมาเพื่อแยกชุดข้อมูลเสียงสำหรับการเรียนรู้ ความยาวเสียงประมาณ 0.3 วินาที หรือความยาวที่ครอบคลุมเสียงเคาะหนึ่งครั้ง กระบวนการเรียนรู้ใช้การกระบวนการเรียนรู้เชิงลึกแบบคอนโวลูชัน (CNN) และกระบวนการการสกัดคุณลักษณะของเสียงด้วยสัมประสิทธิ์เมลฟรีเควินซีเซปสตรอล (MFCC) จากการทดลองพบว่าค่าความถูกต้องของแบบจำลองมีความถูกต้อง 90.78% สำหรับชุดข้อมูลตรวจสอบความถูกต้อง และความถูกต้อง 89.47% สำหรับชุดข้อมูลทดสอบ เมื่อได้แบบจำลองแล้วนำไปใช้ในแอปพลิเคชันบนสมาร์ตโฟนเพื่อสะดวกต่อการใช้งานของผู้บริโภคและชาวสวนทุเรียน การออกแบบแอปพลิเคชันเป็นการส่งข้อมูลผ่านระบบ HTTP Protocol ที่อยู่ในรูปแบบเอพีไอ

สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2562

ลายมือชื่อนิสิต
ลายมือชื่อ อ.ที่ปรึกษาหลัก

6170278421 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: ripeness of durian, convolutional neural network (CNN), knocking sounds of durian, API, MFCC

Weangchai Kharamat : APPLICATION DEVELOPMENT FOR DURIAN RIPENESS CLASSIFICATION FROM THE KNOCKING SOUNDS USING CONVOLUTIONAL NEURAL NETWORK. Advisor: Asst. Prof. MANOP WONGSAISUWAN, Ph.D.

Durian is one of the most popular fruits in Thailand. Checking the ripeness of durian is important for trading in the fruit market and exporting. Durian ripeness examination can be checked in many ways. In Thailand, a common method is knocking a few times using a rubber-tipped stick. In this thesis, the method of knocking and listening is used to tell the ripeness. Normally, listening to sound requires an expert listening. Therefore, an application was made to identify the ripeness of durian from knocking sound by using the convolution neural network. Durian knocking data was suggested by a middleman in Chanthaburi province and told how ripe the durian was ripened. The sound data was collected by using 30 Mon-Tong durian, and divided into 3 groups: ripe, mid-ripe, and unripe. bring the knocking sound to separate the sound data set (0.3 seconds of each durian knocking sound interval) for learning. Our method applies the convolutional neural network (CNN) and Mel-frequency cepstral coefficient spectrogram (MFCC) to classify the durian sound. The experimental results of our model shows an accuracy around 90.78% of validation data and 89.47% of testing data. And we apply the model to use on smartphone application. The application design communication using HTTP protocol and create API system for user.

Field of Study: Electrical Engineering

Student's Signature

Academic Year: 2019

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จสมบูรณ์ได้ด้วยดีเนื่องจากได้รับความกรุณาชี้แนะและช่วยเหลืออย่างดียิ่งจาก รองศาสตราจารย์ ดร.เอกชัย ลีลาธรรม อาจารย์ผู้ทรงคุณวุฒิสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.วันเฉลิม โปธา อาจารย์ประธานกรรมการสอบวิทยานิพนธ์ และ ผู้ช่วยศาสตราจารย์ ดร.มานพ วงศ์สายสุวรรณ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำแนะนำและตรวจสอบแก้ไขข้อบกพร่องมาโดยตลอด การทำวิทยานิพนธ์ ตั้งแต่เริ่มต้นจนสำเร็จเรียบร้อย ผู้วิจัยขอกราบขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ โอกาสนี้ พ่อค้าคนกลางทรัพย์สินนิต์ คนขายทุเรียน ที่ให้คำแนะนำในการฟังเสียงเคาะและข้อมูลเสียงเพื่อนำมาวิเคราะห์ ศึกษา ซึ่งเป็นประโยชน์ต่อการทำวิทยานิพนธ์เล่มนี้

ขอขอบคุณห้องปฏิบัติการ Embedded System and IC Design Research Laboratory หรือ ESID Lab ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และเพื่อนพี่น้องในห้องปฏิบัติการ ที่สนับสนุนและช่วยเหลือให้คำปรึกษา แนะนำ ตลอดจนร่วมดำเนินการเก็บรวบรวมข้อมูล จนทำให้วิทยานิพนธ์ครั้งนี้สำเร็จลุล่วงด้วยดี

กราบขอบพระคุณบิดามารดา ที่สนับสนุนและให้กำลังใจจนงานวิจัยสำเร็จด้วยดี คุณค่าและประโยชน์อันพึงมีจากการศึกษาวิจัยนี้ ผู้วิจัยขอน้อมบูชาพระคุณบิดามารดาและบูรพาจารย์ทุกท่านที่ได้อบรมสั่งสอน และให้ความรู้มาโดยตลอดทำให้การศึกษาวิจัยเล่มนี้สำเร็จลงได้



เวียงชัย คาระมาตย์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ฅ
สารบัญรูปภาพ.....	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตวิทยานิพนธ์	2
1.4 ขั้นตอนและวิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากวิทยานิพนธ์.....	3
1.6 เนื้อหาของวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีและงานที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.1.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNNs)	4
2.1.1.1 กระบวนการสกัดคุณลักษณะ (Feature Extraction)	5
2.1.1.2 ลักษณะของตัวกรอง	5
2.1.1.3 การเลื่อนตัวกรอง Stride และการเพิ่มพื้นที่ Padding	6
2.1.1.4 กระบวนการรวม (Pooling).....	8

2.1.1.5 ไฮเปอร์พารามิเตอร์ (Hyperparameters).....	9
2.1.1.6 การเชื่อมต่ออย่างสมบูรณ์ (Fully connected).....	9
2.1.1.7 ฟังก์ชันการกระตุ้น (Activation Function)	10
2.1.2 เครื่องมือสำหรับใช้พัฒนาการเรียนรู้เชิงลึก (Deep Learning Tools).....	12
2.1.3 การสกัดคุณลักษณะของเสียงด้วยสัมประสิทธิ์เมลฟรีควีนซีเซปสตรอล (Mel-Frequency Cepstral Coefficients (MFCCs) Feature Extraction)	13
2.1.3.1 การบล็อกเฟรม (Frame blocking).....	14
2.1.3.2 หน้าต่างแฮมมิง (Hamming window).....	15
2.1.3.3 การแปลงฟูริเยร์อย่างรวดเร็ว (Fast Fourier Transform: FFT).....	15
2.1.3.4 เมลฟิลเตอร์แบงก์ (Mel Filter Bank).....	17
2.1.3.5 การแปลงโคไซน์ไม่ต่อเนื่อง (Discrete Cosine Transform: DCT).....	18
2.1.4 การพัฒนาแอปพลิเคชัน	19
2.1.4.1 Android Studio	19
2.1.4.2 ภาษาสำหรับการพัฒนาแอปพลิเคชัน.....	21
2.1.4.3 เอพีไอ (Application Program Interface: API).....	21
2.1.4.4 REST และ RESTful API.....	22
2.1.4.5 ไฟร์เบส (Firebase).....	24
2.1.4.6 ดอคเกอร์ Docker	25
2.1.4.7 คลาวด์รัน Cloud Run.....	26
2.1.4.8 ฟลัสก์ Flask.....	26
2.1.5 การทดสอบประสิทธิภาพของแบบจำลอง.....	26
2.2 งานวิจัยที่เกี่ยวข้อง.....	28
บทที่ 3 วิธีการดำเนินการวิจัย	30
3.1 การเตรียมชุดข้อมูลเสียงเคาะทุเรียนและกระบวนการเรียนรู้เสียงทุเรียน	30

3.1.1 การเตรียมชุดเสียงเคาะทุเรียน.....	30
3.1.2 กระบวนการสกัดคุณลักษณะของเสียงเคาะ (Feature Extraction).....	31
3.1.3 กระบวนการเรียนรู้ของเสียงเคาะทุเรียน.....	34
3.2 กระบวนการออกแบบแอปพลิเคชันและระบบการจำแนกเสียงทุเรียน	35
3.2.1 การออกแบบการทำงานแอปพลิเคชัน	37
3.2.2 การออกแบบระบบ RESTful APIs	38
3.2.3 โครงสร้างการทำงานของระบบเบื้องหลัง Backend	39
บทที่ 4 การทดลองและประเมินผล	46
4.1 การทดลองหาแบบจำลองที่ดีที่สุด.....	46
4.2 การทดสอบประสิทธิภาพของแอปพลิเคชันจำแนกความสุกทุเรียน	47
บทที่ 5 บทสรุปและอภิปราย.....	51
5.1 สรุปผลการสร้างแบบจำลอง.....	51
5.2 สรุปผลการใช้งานและปัญหาของแอปพลิเคชัน	51
5.3 แนวทางวิจัยในขั้นถัดไป.....	52
บรรณานุกรม.....	53
ภาคผนวก.....	58
ภาคผนวก ก หน้าต่างของแอปพลิเคชัน และการลองใช้งานแอปพลิเคชันในพื้นที่จริง	59
ภาคผนวก ข โค้ดสำหรับการสร้างและทดสอบแบบจำลองการเรียนรู้เสียงเคาะทุเรียน	64
ประวัติผู้เขียน.....	71

สารบัญตาราง

	หน้า
ตารางที่ 1 การใส่การคัดแยกลงใน Confusion Matrix	27
ตารางที่ 2 การแปลงข้อมูลการคัดแยกเป็นแบบ Binary บน Confusion Matrix	27
ตารางที่ 3 โครงสร้างภายในของการเรียนรู้เชิงลึกแบบคอนโวลูชัน	35
ตารางที่ 4 การทดสอบเพิ่มรอบการเรียนรู้ของแบบจำลอง	46
ตารางที่ 5 การทดลองการแยกกลุ่มทุเรียนในแต่ละกลุ่มละ 1 ลูก จำนวน 50 ครั้ง	48
ตารางที่ 6 การวัดผลการคัดแยกความสุขของแอปพลิเคชันผลลัพธ์จากตารางที่ 5	48
ตารางที่ 7 การทดลองการแยกกลุ่มทุเรียนในร้านค้าจำนวน 6 ลูก จำนวนลูกละ 10 ครั้ง	48
ตารางที่ 8 การวัดผลการคัดแยกความสุขของแอปพลิเคชันผลลัพธ์จากตารางที่ 7	49

สารบัญรูปภาพ

หน้า

รูปที่ 1 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน CNN [3]	4
รูปที่ 2 การมองวัตถุแบบแบ่งเป็นพื้นที่ย่อยของมนุษย์ (ซ้าย) ภาพต้นฉบับ (ขวา) กรอบสี่เหลี่ยมเป็นพื้นที่ที่มองในช่วงเวลาหนึ่ง [4]	5
รูปที่ 3 (ซ้าย) เส้นทแยงสีขาว (ขวา) ตัวกรองขนาด 3x3 pixel สำหรับหาเส้นตรงทแยงสีขาว [5]	6
รูปที่ 4 การทำงานของตัวกรองขนาด 3x3 พิกเซล (กลาง) กับรูปภาพขาเข้าขนาด 6x6 พิกเซล (ซ้าย) ซึ่งได้ฟังก์ชันลักษณะขนาด 4x4 พิกเซล (ขวา) [6]	6
รูปที่ 5 การเลื่อนทีละหนึ่งพิกเซล (stride = 1) จากซ้ายไปขวา โดยมีขนาดตัวกรอง 3x3 [7]	7
รูปที่ 6 การเลื่อนทีละสองพิกเซล (stride = 2) จากซ้ายไปขวา โดยมีขนาดตัวกรอง 3x3 [7].....	7
รูปที่ 7 การเพิ่มพื้นที่สี่เทา (padding) เพื่อให้ฟังก์ชันลักษณะมีขนาดเท่ากับภาพต้นฉบับ [7]	8
รูปที่ 8 กระบวนการ Pooling โดยมีขนาดของรูปภาพขาเข้าเป็น 32x32x10 พิกเซล [7].....	8
รูปที่ 9 การทำ Max-Pooling, (ซ้าย) รูปภาพอินพุต, (ขวา) ผลลัพธ์ [7].....	9
รูปที่ 10 การเชื่อมต่ออย่างสมบูรณ์ (Fully connected) ในโครงข่ายประสาทเทียมเชิงลึก [8]	10
รูปที่ 11 การจำลองฟังก์ชันการทำงานของระบบประสาทของมนุษย์ (a) โครงสร้างระบบประสาท (b) ฟังก์ชันคณิตศาสตร์ที่จำลองการทำงานของระบบประสาท (ภาพจาก http://cs231n.github.io/neural-networks-1/)	11
รูปที่ 12 Rectified Linear Unit (ReLU) activation ซึ่งเป็น 0 เมื่อ $x \leq 0$ และเป็น 1 เมื่อ $x > 0$	11
รูปที่ 13 สัญลักษณ์สากลของภาษาไพธอน [11].....	12
รูปที่ 14 สัญลักษณ์ของเทนเซอร์โฟลว์ (ภาพจาก www.tensorflow.org)	13
รูปที่ 15 สัญลักษณ์ของเคราส [15]	13
รูปที่ 16 การทำบล็อกเฟรมที่มีขนาดหน้าต่าง 20ms และเลื่อนหน้าต่างทุก ๆ 10ms	14
รูปที่ 17 ความสัมพันธ์ระหว่าง α กับการเปลี่ยนแปลงของเส้นโค้งของหน้าต่างแอมมิง [18]	15

รูปที่ 18 การแปลงฟูริเยร์แบบเร็ว (ซ้าย) สัญญาณต้นฉบับ (ขวา) สัญญาณที่ผ่านการทำหน้าต่างแฮมมิง [18]	16
รูปที่ 19 ความสัมพันธ์ของความถี่เชิงเส้นที่เป็นสัดส่วนกันกับความถี่เมล [18]	17
รูปที่ 20 การแปลงความสัมพันธ์ของความถี่เชิงเส้นเป็นช่วงสัญญาณความถี่รูปสามเหลี่ยม	18
รูปที่ 21 การแปลงโคไซน์ไม่ต่อเนื่อง (กราฟด้านล่าง)	18
รูปที่ 22 หน้าต่างการทำงานหลักของ Android Studio	19
รูปที่ 23 การเลือก Android SDK สำหรับการพัฒนาแอปพลิเคชัน	20
รูปที่ 24 ระบบการ compile ระหว่างภาษาจาวา (*.java) กับภาษา Kotlin (*.kt) [27]	20
รูปที่ 25 รูปแบบการจัดเก็บข้อมูลแบบไฟล์เบส [33]	24
รูปที่ 26 บริการของไฟล์เบส [33]	25
รูปที่ 27 การจำลองสภาพแวดล้อมด้วย docker container (ซ้าย) การจำลองสภาพแวดล้อมด้วย virtual machine (ขวา) [35]	25
รูปที่ 28 การจัดเตรียมการขนส่งทุเรียนให้ห้างสรรพสินค้าภายในประเทศ.....	30
รูปที่ 29 กระบวนการสกัดคุณสมบัติเด่นของเสียงด้วยกระบวนการ MFCC.....	31
รูปที่ 30 การทำ frame blocking ของสัญญาณเสียงเคาะทุเรียนที่มีความยาวคลิป 300ms	32
รูปที่ 31 การทำหน้าต่างแฮมมิง (a) สัญญาณเสียงเคาะต้นฉบับ (b) หน้าต่างแฮมมิงเมื่อ (c) สัญญาณเสียงหลังจากเข้ากระบวนการแฮมมิง 1 เฟรมที่มาจากรอบสีแดงจากสัญญาณเสียงต้นฉบับ	33
รูปที่ 32 สัญญาณที่แปลงจากโดเมนเวลาเป็นโดเมนความถี่, (a) Fast-Fourier Transform, (b) Power Spectrum Density ของกลุ่มตัวอย่างทุเรียนสุก (ripe)	33
รูปที่ 33 สัญญาณหลังผ่านกระบวนการเมลฟิเตอร์แบงก์	34
รูปที่ 34 สัญญาณหลังจากผ่านกระบวนการแปลงโคไซน์แบบไม่ต่อเนื่อง.....	34
รูปที่ 35 โครงสร้างการเรียนรู้เสียงเคาะด้วยการเรียนรู้เชิงลึกแบบคอนโวลูชัน	34
รูปที่ 36 ภาพรวมแสดงการทำงานของระบบการจำแนกเสียงทุเรียน.....	36
รูปที่ 37 รูปแบบการจำแนกความสุกทุเรียน (a) ผ่านเว็บ APIs และ (b) ผ่านแอปพลิเคชันมือถือ ...	36

รูปที่ 38 Flow Chart แสดงการทำงานของแอปพลิเคชัน	37
รูปที่ 39 การสื่อสารผ่านระบบ RESTful APIs.....	38
รูปที่ 40 โครงสร้างระบบเบื้องหลัง Backend	39
รูปที่ 41 โครงสร้างโฟลเดอร์สำหรับการสร้างระบบเบื้องหลัง	40
รูปที่ 42 Flow chart สำหรับการสร้าง Docker container	41
รูปที่ 43 การทำงานของไฟล์ pred.py ที่อยู่ในส่วนเบื้องหลัง	42
รูปที่ 44 สัญญาณเสียงเคาะที่ได้รับจากผู้ให้บริการระยะเวลา 5 วินาที	43
รูปที่ 45 การเลือกเสียงเคาะสำหรับไปทำนาย	43
รูปที่ 46 การระบุกลุ่มความสุขหลังจากเลือกเสียงเคาะ	44
รูปที่ 47 ประสิทธิภาพในการทำนายของแบบจำลอง (a) ทดสอบกับชุดข้อมูลทดสอบ (Test) ทั้งหมด 114 ตัวอย่างเสียง (b) ทดสอบกับชุดข้อมูลทั้งหมด 900 ตัวอย่างเสียง.....	47
รูปที่ 48 ประสิทธิภาพในการคัดแยกความสุขของทุเรียนในแต่ละกลุ่มในกรณีที่มีเสียงรบกวนน้อย .	49
รูปที่ 49 ประสิทธิภาพในการคัดแยกความสุขของทุเรียนในแต่ละกลุ่มในกรณีที่มีเสียงรบกวนมาก..	50

บทที่ 1

บทนำ

เนื้อหาของบทนี้จะนำเสนอที่มาและความสำคัญของวิทยานิพนธ์ วัตถุประสงค์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับจากวิทยานิพนธ์ งานวิจัยในอดีตที่เกี่ยวข้องกับวิทยานิพนธ์และเนื้อหาของวิทยานิพนธ์

1.1 ที่มาและความสำคัญของปัญหา

กระทรวงเกษตรและสหกรณ์ได้เปิดเผยผลผลิตทุเรียนที่มีคุณภาพส่วนใหญ่จะจำหน่ายแบบเหมาสวนให้กับพ่อค้าคนกลาง ซึ่งชาวสวนจะจำหน่ายให้กับพ่อค้าคนกลางที่ให้ราคาดีกว่า จากนั้นทางพ่อค้าคนกลางจะคัดเกรดเพื่อการส่งออก ซึ่งในปัจจุบันยังมีความต้องการอย่างต่อเนื่อง สำหรับคุณภาพเกรด A และเกรด B จะทำการส่งออกไปยังตลาดจีน และเกรดพรีเมียมจะทำการส่งไปยังห้างสรรพสินค้าภายในประเทศ [1] ส่วนราคาสำหรับเกรดพรีเมียมจะตกอยู่ประมาณ 115 ถึง 130 บาทต่อกิโลกรัม ราคาจะลดหลั่งลงไปตามคุณภาพของทุเรียน และการความผันผวนของราคาตลาด

ตามพระราชบัญญัติมาตรฐานสินค้าเกษตร พ.ศ. 2551 ได้ประกาศข้อกำหนดมาตรฐานทุเรียนเพื่อส่งเสริมสินค้าการเกษตรให้ได้คุณภาพ มาตรฐาน และปลอดภัย ซึ่งต้องมีคุณสมบัติขั้นต่ำ ดังนี้ (1) เป็นทุเรียนทั้งผล (มีขั้วผล) (2) ตรงตามพันธุ์ (3) สด (4) สะอาดปราศจากสิ่งแปลกปลอมที่มองเห็นได้ (5) ไม่มีรอยแตกที่เปลือก (6) ไม่มีศัตรูพืชที่มีผลกระทบต่อลักษณะภายนอกของทุเรียน (7) ไม่มีร่องรอยความเสียหายเนื่องจากศัตรูพืชที่มีผลกระทบต่อคุณภาพของเนื้อทุเรียน (8) ไม่มีความเสียหายเนื่องจากอุณหภูมิต่ำหรืออุณหภูมิสูง (9) ไม่มีกลิ่นแปลกปลอมหรือรสชาติผิดปกติ (10) เมื่อผลทุเรียนสุก ไม่มีความผิดปกติของเนื้อ ได้แก่ แขนง เต่าเผา ไส้ซึม ถ้ามีอย่างใดอย่างหนึ่งหรือรวมตัวกันไม่เกิน 5 เปอร์เซ็นต์ของส่วนที่บริโภคได้ [2] ทำให้เห็นว่าการควบคุมและการตรวจสอบทุเรียนก่อนการส่งออกมีความสำคัญเพื่อให้มูลค่าของทุเรียนที่ส่งผลต่อการขนส่งแต่ละครั้งไม่ลดลงอีกทั้งยังเป็นชื่อเสียงในการส่งทุเรียนคุณภาพให้กับประเทศ

ในบทความวิจัยนี้เสนอการพัฒนาแอปพลิเคชันสำหรับการจำแนกความสุก-ดิบของผลทุเรียน โดยวิเคราะห์จากเสียงที่ได้จากการเคาะเปลือกของทุเรียน ซึ่งปกติชาวสวนทุเรียนหรือพ่อค้าคนกลางจะใช้วิธีการเคาะและฟังเสียงที่ได้ จากนั้นวิเคราะห์ว่าทุเรียนลูกนั้นสุกหรือไม่สุก นี่ก็เป็นวิธีที่ง่ายแต่ต้องอาศัยความเคยชินและประสบการณ์ของผู้ฟัง โดยหลักการทำงานของแอปพลิเคชันออกแบบให้รับเสียงที่เคาะผ่านไมโครโฟน จากนั้นตัวแอปพลิเคชันจะทำการวิเคราะห์ระดับความสุก-ดิบของทุเรียน ซึ่งมีอยู่ 3 ระดับ คือ ไม่สุก (unripe), สุกกลาง (mid-ripe), และสุก (ripe) การวิเคราะห์จะอาศัยการเรียนรู้จำแบบโครงข่ายประสาทเทียมคอนโวลูชัน ประกอบกับในปัจจุบันคนส่วนใหญ่มี

สมาร์ทโฟนเป็นของตัวเอง แอปพลิเคชันนี้จะทำให้คนที่ชื่นชอบบริโภคทุเรียนสามารถใช้งานได้อย่างสะดวกในการเลือกซื้อและบริโภคทุเรียนได้ทันที และลดความกังวลที่จะเจอกับคนขายที่โก่งว่าทุเรียนที่นำมาขายสุกหรือไม่สุก ซึ่งแอปพลิเคชันนี้จะเป็ประโยชน์สำหรับคนที่ดูความสุกทุเรียนไม่เป็น หรือกระทั่งชาวสวนสามารถนำไปจำแนกความสุก-ดิบของทุเรียนก่อนการเก็บเกี่ยวได้

1.2 วัตถุประสงค์

1.2.1 เพื่อทำความเข้าใจเกี่ยวกับเสียงที่ได้จากการเคาะผลทุเรียนเพื่อนำมาจัดกลุ่มระดับความสุก

1.2.2 เพื่อสร้างต้นแบบแบบจำลองการเรียนรู้เกี่ยวกับการจำแนกความสุกทุเรียนจากเสียงเคาะเปลือกด้วยกระบวนการการเรียนรู้เชิงลึกแบบคอนโวลูชัน

1.2.3 เพื่อสร้างต้นแบบแอปพลิเคชันสำหรับการจำแนกความสุกของทุเรียนด้วยเสียงเคาะจากเปลือก

1.3 ขอบเขตวิทยานิพนธ์

1.3.1 เพื่อพัฒนาต่อยอดวิธีชาวบ้านในการตรวจสอบความสุกของทุเรียนโดยการฟังเสียงที่ได้จากการเคาะและนำมาประยุกต์ใช้ในการเรียนรู้เชิงลึกแบบคอนโวลูชัน

1.3.2 ใช้ข้อมูลเสียงเคาะที่เก็บจากพ่อค้าคนกลางในจังหวัดจันทบุรีเป็นข้อมูลสำหรับการเรียนรู้

1.3.3 การเก็บข้อมูลเสียงได้สุ่มตัวอย่างทุเรียนพันธุ์หมอนทอง จำนวน 30 ลูก แล้วให้ชาวสวนที่มีประสบการณ์ระบุว่าผลทุเรียนสุก สุกปานกลาง หรือไม่สุก

1.3.4 การระบุความของทุเรียนมี 3 ระดับ ได้แก่ สุก สุกปานกลาง และไม่สุก

1.3.5 แอปพลิเคชันที่ถูกพัฒนาสามารถจำแนกความสุกของทุเรียนได้ 3 ระดับตามที่ได้กล่าวไว้ในหัวข้อ 1.3.4

1.4 ขั้นตอนและวิธีการดำเนินงาน

1.4.1 ศึกษาบทความทางวิชาการและงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ ดังนี้

1.4.1.1 ศึกษาการทำงานของกรเรียนรู้เชิงลึกแบบคอนโวลูชัน (CNN)

1.4.1.2 ศึกษาการสกัดคุณสมบัติของเสียง (Audio Feature Extraction)

1.4.2 สร้างแบบจำลองการเรียนรู้ด้วยการเรียนรู้เชิงลึกแบบคอนโวลูชัน

1.4.3 พัฒนารูปแบบปรับปรุงประสิทธิภาพของแบบจำลอง

1.4.4 ทดสอบประสิทธิภาพของแบบจำลอง

1.4.5 พัฒนาแบบจำลองให้ประยุกต์ใช้กับสมาร์ทโฟน

1.4.5 วิเคราะห์และสรุปประสิทธิภาพของแบบจำลองและการทำงานของแอปพลิเคชัน

1.5 ประโยชน์ที่คาดว่าจะได้รับจากวิทยานิพนธ์

1.5.1 ผู้ใช้งานทั่วไปสามารถใช้งานแอปพลิเคชันสำหรับการจำแนกความสุขของทุเรียนได้

1.5.2 แบบจำลองการเรียนรู้ความสุขของทุเรียนสามารถนำไปประยุกต์ใช้กับงานวิจัยอื่น ที่ต้องการจำแนกข้อมูลออกเป็นกลุ่มได้

1.5.3 ความรู้ที่ได้สามารถนำไปประยุกต์ใช้ในระบบงานอื่น ๆ ในเชิงการพัฒนาระบบได้

1.6 เนื้อหาของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งออกเป็น 5 บท โดยแบ่งเป็นเนื้อหา ดังนี้

บทที่ 1 บทนำ แบ่งเป็น 5 หัวข้อ ได้แก่ (1) ที่มาและความสำคัญของปัญหา (2) วัตถุประสงค์ (3) ขอบเขตวิทยานิพนธ์ (4) ขั้นตอนการศึกษาและวิธีการดำเนินงาน (5) ประโยชน์ที่คาดว่าจะได้รับจากวิทยานิพนธ์

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง แบ่งเป็น 2 หัวข้อ ได้แก่ (1) งานวิจัยที่เกี่ยวข้อง (2) ความรู้พื้นฐานที่เกี่ยวข้อง

บทที่ 3 วิธีดำเนินการวิจัย แบ่งออกเป็น 2 หัวข้อ ได้แก่ (1) การเตรียมชุดข้อมูลเสียงเคาะทุเรียนและกระบวนการเรียนรู้เสียงเคาะทุเรียน (2) เป็นส่วนการพัฒนาแอปพลิเคชันในการจำแนกเสียงเคาะทุเรียน

บทที่ 4 การทดลองและประเมินผล แบ่งออกเป็น 2 หัวข้อ ได้แก่ (1) การทดลองหาแบบจำลองที่ดีที่สุด (2) การทดสอบประสิทธิภาพของแอปพลิเคชันจำแนกความสุขทุเรียน

บทที่ 5 บทสรุป อภิปรายและแนวทางวิจัยในขั้นถัดไป

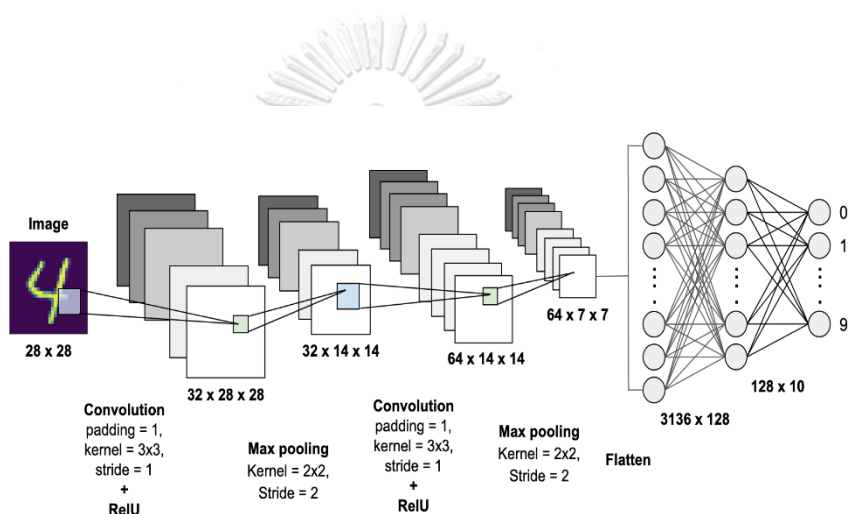
บทที่ 2

ทฤษฎีและงานที่เกี่ยวข้อง

เนื้อหาในบทนี้แบ่งออกเป็น 2 ส่วน ได้แก่ ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์

2.1 ทฤษฎีที่เกี่ยวข้อง

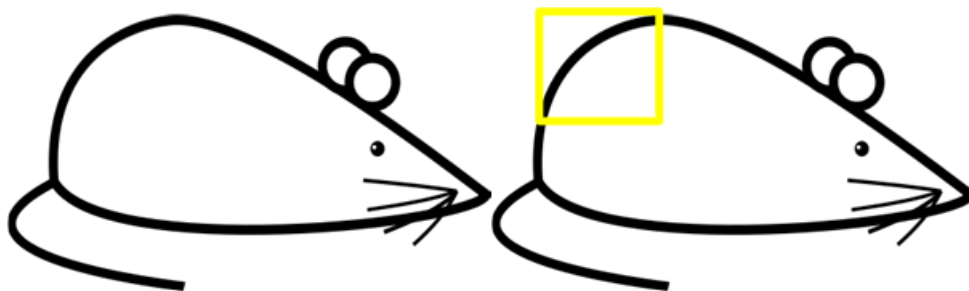
ทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์นี้แบ่งออกเป็นส่วนต่าง ๆ ได้แก่ โครงข่ายประสาทเทียมแบบคอนโวลูชัน เครื่องมือสำหรับใช้พัฒนาการเรียนรู้เชิงลึก การสกัดคุณลักษณะของเสียงด้วยสัมประสิทธิ์เมลฟรีควีนซีเซปสตรอล การพัฒนาแอปพลิเคชัน และการทดสอบประสิทธิภาพของแบบจำลอง



รูปที่ 1 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน CNN [3]

2.1.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNNs)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน มีสถาปัตยกรรมดังรูปที่ 1 เป็นโครงข่ายประสาทเทียมที่จำลองการมองเห็นของมนุษย์ที่สามารถมองเห็นวัตถุและสมองของมนุษย์จะแบ่งสิ่งที่เห็นแบ่งออกเป็นพื้นที่ย่อย ๆ จากนั้นนำพื้นที่เหล่านี้มารวมเป็นภาพเดียวและมองว่าสิ่งที้นั้นเป็นอะไร การมองพื้นที่ย่อย ๆ นั้นจะมีคุณลักษณะที่แตกต่างกันไป (feature) เช่น ลายเส้น การตัดกันของสี เป็นต้น ยกตัวอย่างการมองพื้นที่ย่อยของรูปหนูในรูปที่ 2 โดยจำลองกรอบสี่เหลี่ยมเป็นพื้นที่ที่มนุษย์มองในช่วงเวลาหนึ่งและค่อย ๆ เลื่อนกรอบนี้ไปเรื่อย ๆ จนครบตัวหนู จากนั้นสมองของมนุษย์จะนำพื้นที่ทุกส่วนมาประกอบกันและวิเคราะห์ว่าสิ่งที่มองเห็นเป็นหนู คอมพิวเตอร์ก็ใช้หลักการเดียวกันแต่จะแปลงบริเวณกรอบสี่เหลี่ยมเป็นตัวเลขของค่าความเข้มสีซึ่งเรียกว่าการสกัดคุณลักษณะ



รูปที่ 2 การมองวัตถุแบบแบ่งเป็นพื้นที่ย่อยของมนุษย์ (ซ้าย) ภาพต้นฉบับ (ขวา) กรอบสี่เหลี่ยมเป็นพื้นที่ที่มองในช่วงเวลาหนึ่ง [4]

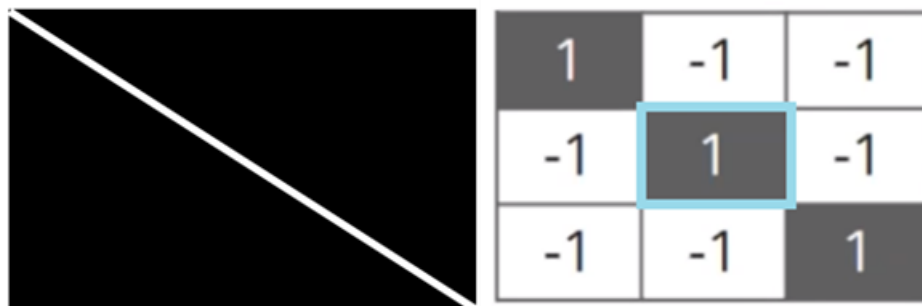
2.1.1.1 กระบวนการสกัดคุณลักษณะ (Feature Extraction)

กระบวนการสกัดคุณลักษณะของซีเอ็นเอ็นเริ่มจากการกำหนดค่าสำหรับตัวกรอง (filter) หรือ เคอร์เนล (kernel) สำหรับการดึงคุณลักษณะที่ใช้ในการเรียนรู้หรือการจำวัตถุออกมา โดยปกติตัวกรองหนึ่งตัวกรองสามารถดึงคุณลักษณะที่สนใจออกมาได้หนึ่งอย่าง ดังนั้นจำเป็นต้องมีตัวกรองหลายอันเพื่อหาคุณลักษณะของพื้นที่ย่อยหลาย ๆ ส่วนมาประกอบกัน

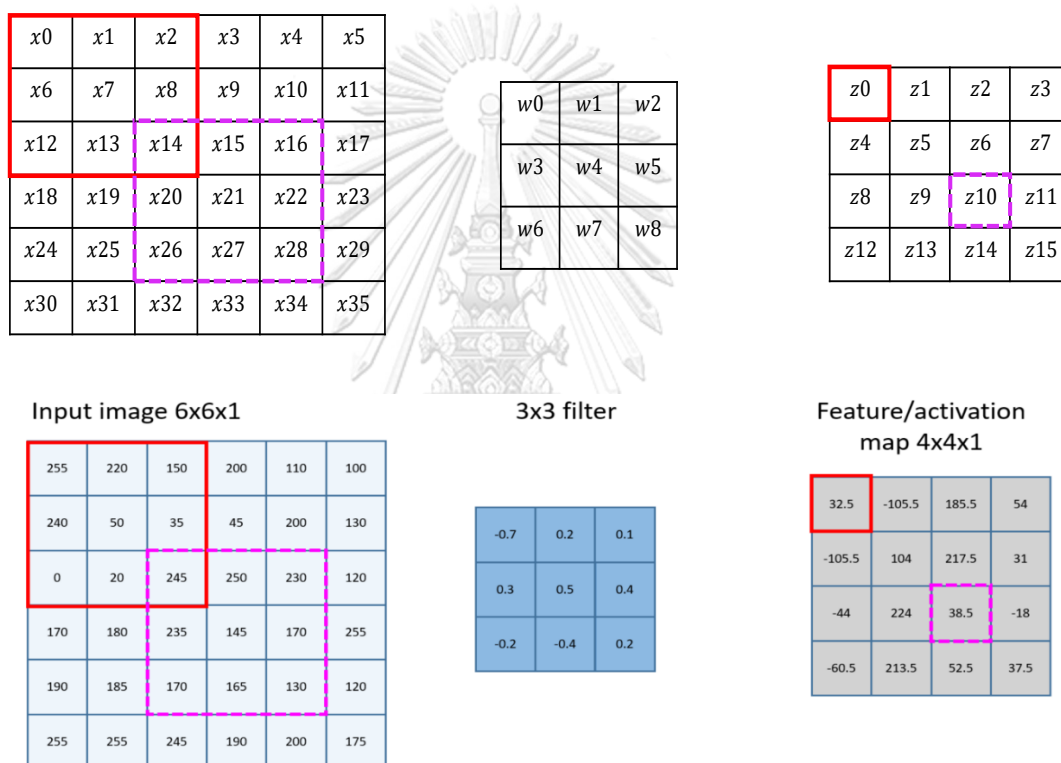
2.1.1.2 ลักษณะของตัวกรอง

สำหรับตัวกรองรูปภาพแบบดิจิทัลนั้น เป็นตัวกรองรูปภาพแบบตารางสองมิติที่มีขนาดตามพื้นที่ย่อย ๆ ที่จะพิจารณา ยกตัวอย่างถ้าต้องการหาเส้นตรงสีขาวดังรูปที่ 3 ตำแหน่งตรงกลางที่มีกรอบสี่ฟ้าคือตัวกรองที่เอามาทาบบนพิกเซล (pixel) ของภาพข้อมูลขาเข้า โดยตัวกรองจะถูกทาบบลงบนพิกเซลแรกๆ ของข้อมูลขาเข้าจากนั้นตัวกรองจะถูกเลื่อนไปทาบบนพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซล เมื่อเลื่อนไปจนครบทุกพิกเซลที่ตัวกรองสามารถเลื่อนได้ จะได้ข้อมูลที่เป็นผลจากการทาบบตัวกรองมาเรียกว่า ผังคุณลักษณะ (feature map) ดังแสดงดังรูปที่ 4 ซึ่งเป็นการแปลงรูปภาพสีเทาขนาด 6x6 พิกเซล ที่มีตัวกรองขนาด 3x3 พิกเซล และทำการเลื่อน (stride) ครั้งละ 1 พิกเซล ในแต่ละตำแหน่งของการเลื่อนนั้นจะคูณด้วยค่าน้ำหนักที่ได้กำหนดขึ้นมาในตัวกรองที่ทาบบลงไปในรูปภาพและนำค่าน้ำหนักที่คูณแล้วมารวมกันจะได้ค่าผังคุณลักษณะมาหนึ่งตำแหน่ง (กรอบสีแดงและกรอบสี่ชมพู) ยกตัวอย่างในรูปที่ 4 กรอบสีแดงคือผลลัพธ์ของผังคุณลักษณะจะได้ตำแหน่งที่ z_0 ซึ่งจะได้

$$\begin{aligned}
 z_0 &= x_0w_0 + x_1w_1 + x_2w_2 + x_6w_3 + x_7w_4 + x_8w_5 + x_{12}w_6 + x_{13}w_7 + x_{14}w_8 \\
 z_0 &= 255(-0.7) + 220(0.2) + 150(0.1) + 240(0.3) + 50(0.5) + 35(0.4) + 0(-0.2) + \\
 &\quad 20(-0.4) + 245(0.2) \\
 z_0 &= 32.5
 \end{aligned}$$



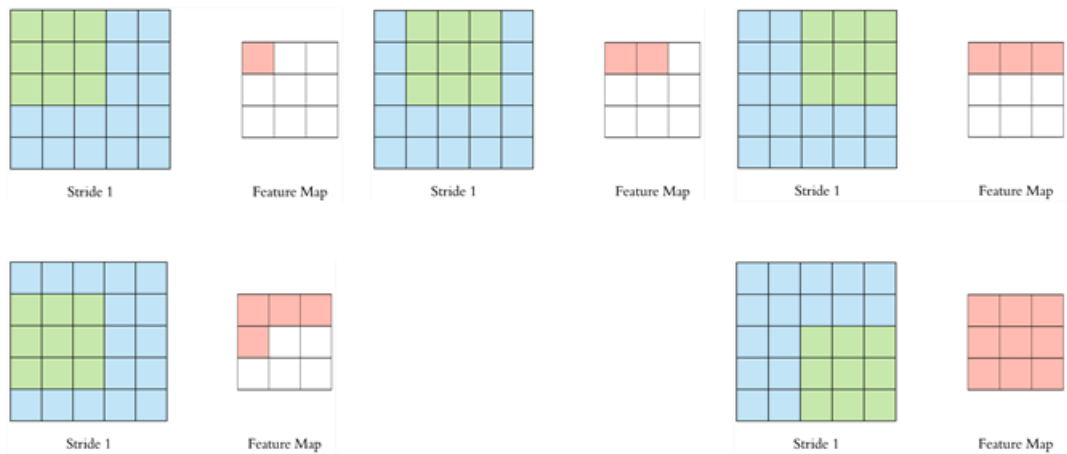
รูปที่ 3 (ซ้าย) เส้นทแยงสีขาว (ขวา) ตัวกรองขนาด 3x3 pixel สำหรับหาเส้นตรงทแยงสีขาว [5]



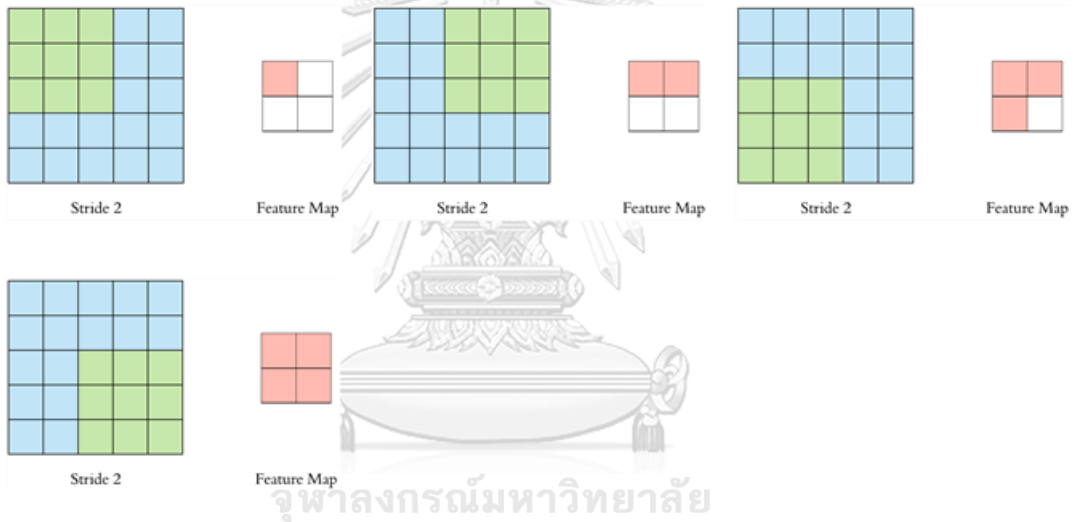
รูปที่ 4 การทำงานของตัวกรองขนาด 3x3 พิกเซล (กลาง) กับรูปภาพขาเข้าขนาด 6x6 พิกเซล (ซ้าย) ซึ่งได้ฟังก์ชันลักษณะขนาด 4x4 พิกเซล (ขวา) [6]

2.1.1.3 การเลื่อนตัวกรอง Stride และการเพิ่มพื้นที่ Padding

Stride เป็นตัวกำหนดว่าตัวกรองสามารถเลื่อนไปที่ละกี่พิกเซลต่อการเลื่อนหนึ่งครั้ง (step) ยกตัวอย่างในรูปที่ 5 กำหนดให้ stride = 1 ทั้งนี้สามารถกำหนดค่าของ stride ให้มากขึ้นได้เพื่อให้พื้นที่ที่ซ้อนทับกันลดลง แต่จะทำให้ขนาดของฟังก์ชันลักษณะเล็กลงดังแสดงในรูปที่ 6 ที่กำหนดให้ stride = 2

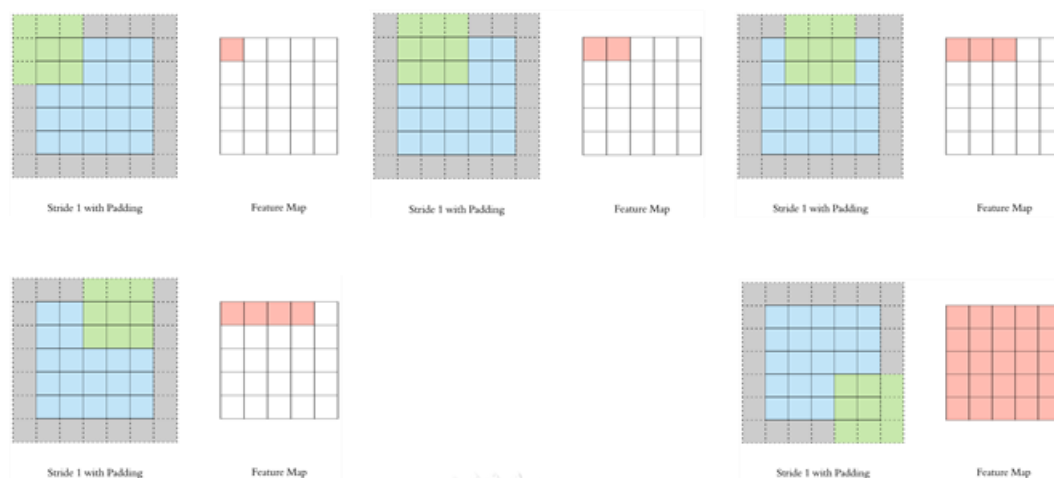


รูปที่ 5 การเลื่อนทีละหนึ่งพิกเซล (stride = 1) จากซ้ายไปขวา โดยมีขนาดตัวกรอง 3x3 [7]



รูปที่ 6 การเลื่อนทีละสองพิกเซล (stride = 2) จากซ้ายไปขวา โดยมีขนาดตัวกรอง 3x3 [7]

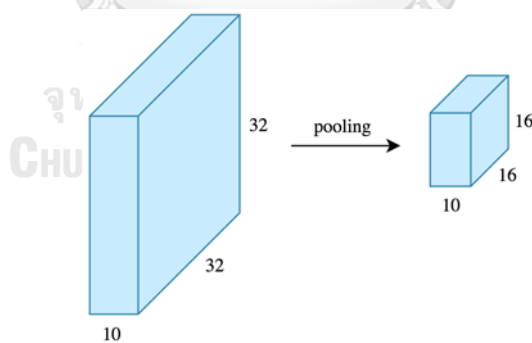
ซึ่งวิธีการแก้ปัญหาเพื่อไม่ให้ฟังก์ชันลดขนาดลงเราจำเป็นต้องทำ Padding หรือการวาดพื้นที่ของรูปภาพเพิ่มขึ้นไปซึ่งจะแสดงในรูปที่ 7 จะเห็นพื้นที่สีเทาขอบ ๆ รูปภาพซึ่งนั่นก็คือพื้นที่ที่เติมเข้าไปซึ่งสามารถเติมค่า 0 หรือค่าตัวเลขอื่น ๆ เข้าไปได้ เพื่อให้เวลาเข้ากระบวนการซีเอ็นเอ็นแล้วฟังก์ชันมีขนาดเท่ากับภาพต้นฉบับ



รูปที่ 7 การเพิ่มพื้นที่สี่เทา (padding) เพื่อให้ฟังก์ชันลักษณะมีขนาดเท่ากับภาพต้นฉบับ [7]

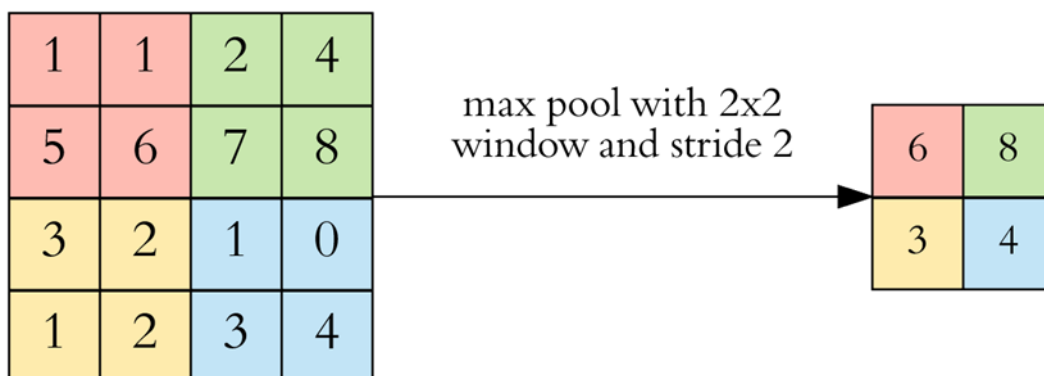
2.1.1.4 กระบวนการรวม (Pooling)

กระบวนการรวมจะช่วยให้ขนาดของฟังก์ชันลักษณะลดลงครึ่งหนึ่งของขนาดภาพต้นฉบับ ยกตัวอย่างเช่น เมื่อมีอินพุตภาพขนาด $32 \times 32 \times 10$ พิกเซล เมื่อรูปภาพได้ผ่านกระบวนการ Pooling ขนาดของรูปภาพเป็น $16 \times 16 \times 10$ พิกเซล โดยที่ลดลงแค่ขนาดของความกว้างกับความสูง แต่ความลึกของรูปภาพยังมีขนาดเท่าเดิม ซึ่งกระบวนการ Pooling นี้จะทำให้พารามิเตอร์ของซีเอ็นเอ็นลดลง แต่ยังคงรักษาคุณลักษณะของภาพไว้เหมือนเดิมแสดงดังรูปที่ 8



รูปที่ 8 กระบวนการ Pooling โดยมีขนาดของรูปภาพขาเข้าเป็น $32 \times 32 \times 10$ พิกเซล [7]

การทำ Pooling ที่นิยมในปัจจุบันคือ Max-Pooling ซึ่งเป็นการเก็บค่าสูงสุดในหน้าต่าง (window) ในการทำ Pooling นั้น ยกตัวอย่างรูปภาพอินพุตขนาด 4×4 พิกเซล ดังรูปที่ 9 (ซ้าย) มีการกำหนดขนาดหน้าต่างตัวกรองของ Pooling เท่ากับ 2×2 พิกเซล แล้วทำการเลื่อนครั้งละ 2 พิกเซล ผลลัพธ์จะออกมาดังรูปที่ 9 (ขวา) จะเห็นว่าขนาดของผลลัพธ์ลดลงแต่ลักษณะเด่นของกลุ่มพิกเซลยังคงอยู่เหมือนเดิม



รูปที่ 9 การทำ Max-Pooling, (ซ้าย) รูปภาพอินพุต, (ขวา) ผลลัพธ์ [7]

การทำ Max-Pooling เพื่อต้องการลดขนาดของข้อมูลขาเข้าและต้องการเก็บคุณลักษณะเด่นที่ตัวกรองในกระบวนการซีเอ็นเอ็นที่ทำได้ก่อนหน้านี้นี้จะไม่ได้ค่าคุณลักษณะเด่นที่เหลืออยู่

2.1.1.5 ไฮเปอร์พารามิเตอร์ (Hyperparameters)

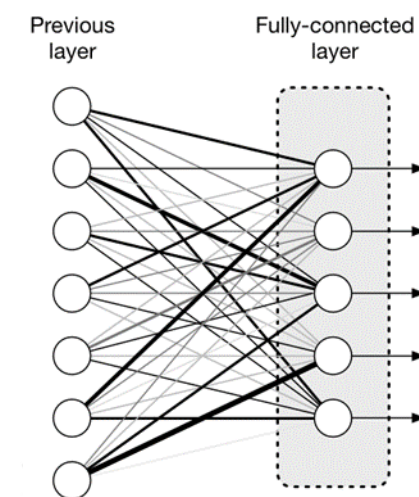
การเลือกพารามิเตอร์นั้นจะเลือกตรงส่วนที่เป็นชั้นของคอนโวลูชัน โดยจะไม่สนใจส่วนของ Pooling โดยพิจารณา 4 ส่วนในการตัดสินใจ

- ขนาดตัวกรอง (Filter size): โดยทั่วไปแล้วจะเลือกใช้ตัวกรองขนาด 3x3 แต่สามารถเลือกใช้ขนาดที่ใหญ่กว่าหรือเล็กกว่าได้ขึ้นอยู่กับแอปพลิเคชัน ตัวกรองเหล่านี้เป็นตัวกรองแบบ 3 มิติคือมีความลึก และความลึกของตัวกรองในแต่ละชั้นจะมีค่าเท่ากับความลึกของอินพุต
- จำนวนของตัวกรอง (Filter count): จำนวนตัวกรองคือส่วนที่มีตัวแปรมากที่สุดซึ่งเป็นการกำลังสองของตัวกรองอยู่ระหว่าง 32 ถึง 1024 การใช้ตัวกรองหลายตัวส่งผลให้แบบจำลองมีประสิทธิภาพที่เพิ่มขึ้น ทั้งนี้หากมากเกินไปมีโอกาสเกิด overfitting ของชุดข้อมูลได้ โดยปกติมักเริ่มต้นด้วยตัวกรองที่น้อยในชั้นเริ่มต้นและเพิ่มจำนวนตามความเหมาะสมในแต่ละชั้นของโครงข่าย
- การเลื่อน (stride): ค่าปกติที่ใช้เริ่มต้นให้มีการเลื่อนครั้งละ 1 พิกเซล
- การเพิ่มพื้นที่ของอินพุต (Padding): ค่าเริ่มต้นมีการกำหนดการ Padding เพื่อให้เอาต์พุตมีขนาดเท่ากับอินพุต

2.1.1.6 การเชื่อมต่ออย่างสมบูรณ์ (Fully connected)

หลังจากการเพิ่มขึ้นของคอนโวลูชันกับการ Padding มาแล้วขั้นสุดท้ายเป็นการเชื่อมต่ออย่างสมบูรณ์ของชั้นคอนโวลูชัน (convolution layer) ซึ่งเป็นชุดข้อมูล 3 มิติ แต่ละชั้นจะมีเวกเตอร์ 1 มิติเชื่อมต่ออยู่ ดังนั้นผลลัพธ์จากชั้นคอนโวลูชันขั้นสุดท้ายส่งค่าไปยังชั้นเชื่อมต่ออย่างสมบูรณ์ โดย

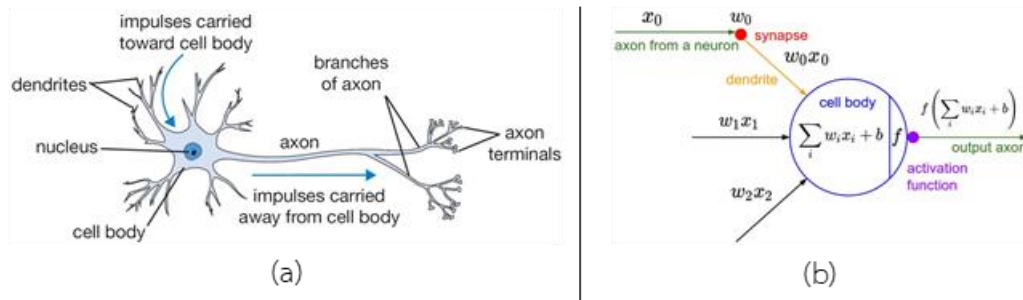
ขั้นนี้จะทำให้ชุดข้อมูลแบนราบ (Flatten) หมายถึงการจัดเรียงปริมาณของตัวเลขเวกเตอร์ 3 มิติให้เป็นเวกเตอร์ 1 มิติ ซึ่งแสดงรูปที่ 10 เพื่อให้ง่ายในการหาค่าความน่าจะเป็นของกลุ่มที่สนใจ



รูปที่ 10 การเชื่อมต่ออย่างสมบูรณ์ (Fully connected) ในโครงข่ายประสาทเทียมเชิงลึก [8]

2.1.1.7 ฟังก์ชันการกระตุ้น (Activation Function)

ฟังก์ชันการกระตุ้นคือฟังก์ชันที่รวมการประมวลผลทั้งหมดจากทุกอินพุตภายใน 1 นิวรอน (Neural) และฟังก์ชันการกระตุ้นนี้จะพิจารณาว่าจะส่งต่อค่าเอาต์พุตเป็นค่าเท่าไร มีแนวคิดมาจากการทำงานของระบบประสาทมนุษย์ แต่ละนิวรอนจะได้รับอินพุตจากหลายๆ สาขาของกิ่งเส้นประสาท Dendrite แล้วนำมาประมวลผลออกมาเป็นหนึ่งเอาต์พุตออกไปที่ Axon เพื่อส่งต่อให้ Dendrite อันต่อไปซึ่งสามารถแสดงการทำงานดังรูปที่ 11 โดยที่สัญญาณที่ผ่าน Axon โดยกำหนดเป็นตัวแปร x จะคูณเข้ากับค่าน้ำหนัก (weight) ที่ได้กำหนดขึ้นมาเป็นตัวแปร w ทำให้ได้ผลลัพธ์เป็น xw ค่าน้ำหนักนี้จะเป็นตัวควบคุมทิศทางของเซลล์ประสาทตัวอื่นว่าจะไปในทางทิศใด โดยที่ฟังก์ชันการกระตุ้นนี้มีฟังก์ชันให้เลือกใช้หลากหลาย ในวิทยานิพนธ์นี้กล่าวถึงฟังก์ชันการกระตุ้นชื่อว่า Rectified Linear Unit: ReLU [9] และ Softmax [10]

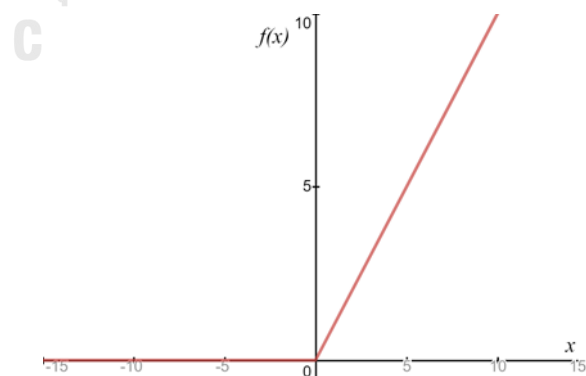


รูปที่ 11 การจำลองฟังก์ชันของการทำงานของระบบประสาทของมนุษย์ (a) โครงสร้างระบบประสาท (b) ฟังก์ชันคณิตศาสตร์ที่จำลองการทำงานของระบบประสาท (ภาพจาก <http://cs231n.github.io/neural-networks-1/>)

- ReLU: เป็นฟังก์ชันการกระตุ้นที่นิยมใช้ในการเรียนรู้โครงข่ายประสาทเทียมมากที่สุดในปัจจุบัน เนื่องจากมีหลักการคำนวณที่ง่ายไม่ซับซ้อน สามารถเขียนเป็นสมการได้ดังสมการที่ (1)

$$f(x) = \max(0, x) = \begin{cases} 0, & \text{for } x \leq 0 \\ x, & \text{for } x > 0 \end{cases} \quad (1)$$

หลักการคือเมื่อเอาค่าอินพุตใด ๆ เข้าฟังก์ชันนี้แล้ว ถ้าค่าที่นำเข้าน้อยกว่าหรือเท่ากับ 0 ผลลัพธ์จะออกมาเป็น 0 แต่หากมีเป็นค่ามากกว่า 0 ผลลัพธ์จะออกมาเป็นค่า x นั้น ๆ และค่าความชันของฟังก์ชันจะเท่ากับ 1 ตลอดแสดงดังรูปที่ 12 ทำให้ฟังก์ชันการกระตุ้นแบบ ReLU นี้สามารถเรียนรู้ได้เร็ว



รูปที่ 12 Rectified Linear Unit (ReLU) activation ซึ่งเป็น 0 เมื่อ $x \leq 0$ และเป็น 1 เมื่อ $x > 0$

- Softmax: เป็นฟังก์ชันที่รับอินพุตเป็นเวกเตอร์ของลอจิกจำนวนจริง แล้วแปลงค่าออกมาเป็นความน่าจะเป็น (probability) ที่มีผลรวมเท่ากับ 1 สามารถเขียนค่า

ความน่าจะเป็นได้ดังสมการที่ (2) Softmax มักถูกนำไปวางไว้เลเยอร์สุดท้ายของโครงข่ายประสาทเทียมเพื่อให้ค่าเอาต์พุตออกมาเป็นค่าความน่าจะเป็น ซึ่งถูกใช้บ่อยในงานการคัดแยก (classification)

$$P(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \text{ for } i = 1, \dots, K \text{ and } z = (z_1, \dots, z_K) \in R^K \quad (2)$$

2.1.2 เครื่องมือสำหรับใช้พัฒนาการเรียนรู้เชิงลึก (Deep Learning Tools)

ในการพัฒนาการเรียนรู้เชิงลึกนั้นมีเครื่องมือที่หลากหลายให้เลือกใช้งาน เครื่องมือที่ใช้งานสำหรับวิทยานิพนธ์นี้ ได้แก่ ไพธอน เทนเซอร์โพล์ และเคราส

ไพธอน (Python)



รูปที่ 13 สัญลักษณ์สากลของภาษาไพธอน [11]

ภาษาโปรแกรม Python [11, 12] คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง สัญลักษณ์สากลของภาษาแสดงดังรูปที่ 13 ภาษาไพธอนถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย เนื่องจากตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เขียนขึ้นให้เป็นภาษาเครื่อง ไพธอนมีการทำงานแบบแปลงชุดคำสั่งทีละบรรทัด (interpreter) เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการ นอกจากนั้นภาษาไพธอนยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ในงานเฉพาะทางใดทางหนึ่งและโครงสร้างภาษาสามารถทำความเข้าใจได้ง่าย และยังมีนิยามไปใช้งานอย่างแพร่หลายในองค์กรใหญ่ๆ หลายองค์กร ปัจจุบันภาษาไพธอนมีเวอร์ชันให้ใช้งาน 2 เวอร์ชันหลัก ๆ ได้แก่ Python 2.x และ Python 3.x สำหรับการพัฒนาโปรแกรมในวิทยานิพนธ์นี้ใช้งาน Python 3.7 ซึ่งสามารถใช้งานกับเครื่องมืออื่น ๆ ได้หลากหลาย ทั้งยังรองรับพีเจอรืใหม่ ๆ ได้ในอนาคต

เทนเซอร์โฟลว์ (TensorFlow)



รูปที่ 14 สัญลักษณ์ของเทนเซอร์โฟลว์ (ภาพจาก www.tensorflow.org)

เทนเซอร์โฟลว์ [13, 14] เป็นไลบรารีการเรียนรู้เชิงลึกของบริษัท Google ที่ได้พัฒนาสำหรับการเรียนรู้เครื่อง หรือ Machine Learning เพื่อเพิ่มประสิทธิภาพให้กับผลิตภัณฑ์ต่าง ๆ ที่ได้ผลดีออกมาสู่ท้องตลาดให้ฉลาดและตอบโต้กับผู้บริโภคมากขึ้น อุปกรณ์ที่รองรับเทนเซอร์โฟลว์ควรมีคุณสมบัติที่เหมาะสมด้วยกัน เช่น อุปกรณ์สำหรับการฝึกฝนข้อมูลให้เทนเซอร์โฟลว์เก่งขึ้นด้วยคอมพิวเตอร์ และอุปกรณ์สำหรับการนำผลลัพธ์จากการฝึกฝนข้อมูลไปใช้ได้ ไม่ว่าจะบนคอมพิวเตอร์หรือมินิคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows, MacOS, หรือ Linux หรือแม้แต่ระบบคลาวด์เซอร์วิส (Cloud Services) และยิ่งรวมถึงมือถือแบบสมาร์ตโฟนบนระบบ IOS หรือ Android ได้

เคราส (Keras)



รูปที่ 15 สัญลักษณ์ของเคราส [15]

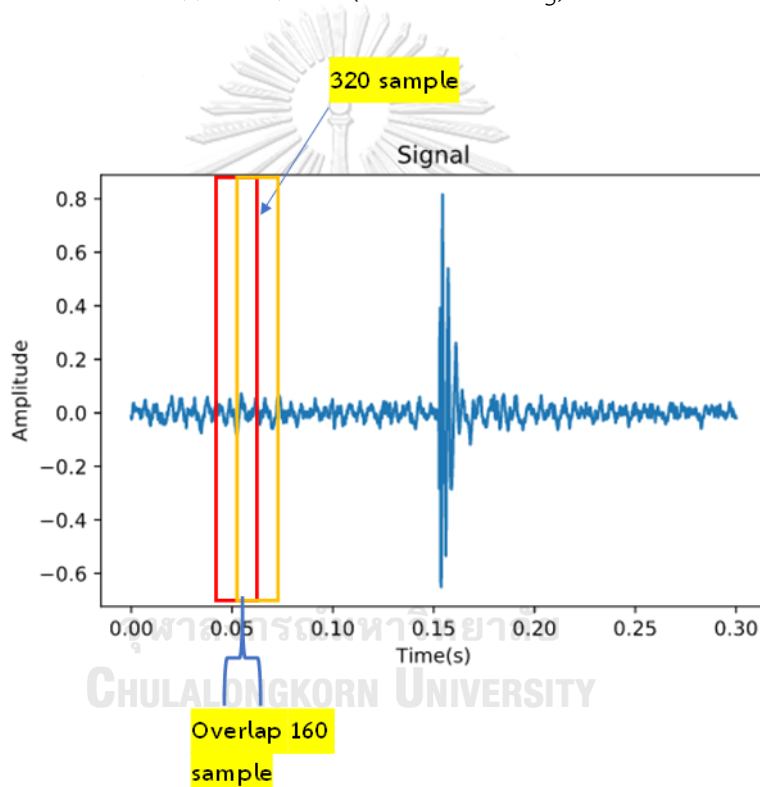
เคราส [15] เป็นไลบรารีการเรียนรู้เชิงลึกที่ได้รับความนิยมมากขึ้นอย่างรวดเร็ว เนื่องจากสามารถใช้งานได้ง่าย สัญลักษณ์ของเคราสแสดงดังรูปที่ 15 การทำงานของเคราสเสมือนเป็นล่ามแปลภาษาให้กับ TensorFlow และ Theano [16] สามารถตั้งค่าได้ว่าจะให้เคราสรันไลบรารีใดได้ โดยการแก้ไขไฟล์ `keras.json` บนไอดีอีที่สามารถแก้ไขได้ สำหรับผู้เขียนใช้โปรแกรมวิม (Vim)

2.1.3 การสกัดคุณลักษณะของเสียงด้วยสัมประสิทธิ์เมลฟรีควีนซีเซปสตรอล (Mel-Frequency Cepstral Coefficients (MFCCs) Feature Extraction)

ขั้นตอนแรกในการใช้ระบบจดจำเสียงพูดหรือเสียงต่าง ๆ ที่สนใจนั้นคือการแยกคุณลักษณะของเสียงนั้น ๆ เช่น การแยกคุณลักษณะองค์ประกอบที่ดีของสัญญาณเสียง การแยกคุณลักษณะความถี่ของสัญญาณเสียง และทำการตัดสัญญาณเสียงที่เป็นพื้นหลังทิ้ง เช่น สัญญาณรบกวน หรือ

สัญญาณที่ไม่ต้องการ โดยการให้สัญญาณผ่านตัวกรองเพื่อกำจัดสัญญาณพื้นหลังทิ้งไป การทำความเข้าใจเกี่ยวกับเสียงสามารถสร้างรูปแบบของเสียงให้อยู่ในรูปแบบของกราฟระหว่างความถี่ที่มีความสัมพันธ์กันกับเวลาที่ผ่านตัวกรองสัญญาณรบกวนมาแล้ว รูปร่างที่ได้นี้จะมีลักษณะที่เป็นเอกลักษณ์ และเมื่อเรียนรู้เอกลักษณ์ของสัญญาณเสียงเหล่านั้นแล้วจะสามารถระบุได้ว่า ลักษณะหรือเอกลักษณ์ของเสียงเช่นนี้คืออะไร จากนั้นนำรูปร่างนี้แสดงในกราฟสเปกตรัมพลังงาน (Power spectrum) จากนั้นเข้าสู่กระบวนการของ Mel frequency cepstral coefficients: MFCCs [17] ที่มีกระบวนการภายใน 5 ขั้นตอน ดังนี้:

2.1.3.1 การบล็อกเฟรม (Frame blocking)



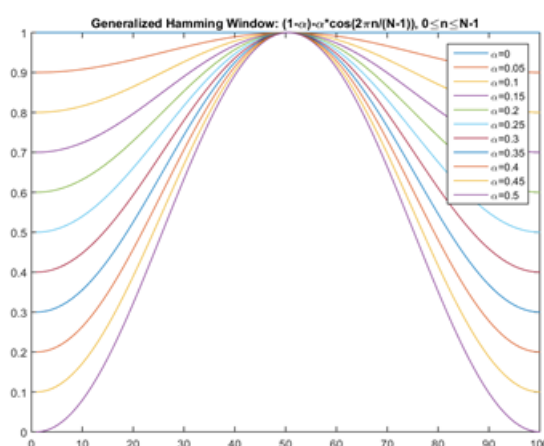
รูปที่ 16 การทำบล็อกเฟรมที่มีขนาดหน้าต่าง 20ms และเลื่อนหน้าต่างทุก ๆ 10ms

สัญญาณเสียงจะถูกเลือกด้วยหน้าต่างขนาด 20ms เพื่อหาคุณลักษณะของเสียงทุก ๆ 20ms โดยหน้าต่างนี้จะเลื่อนไปที่ละ 10ms เพื่อหาคุณลักษณะของสัญญาณไปจนจบสัญญาณเสียงที่รับเข้ามา ยกตัวอย่างเช่น หากตัวอย่างของเสียงมีอัตราการสุ่มตัวอย่าง (sampling rate) 16kHz ขนาดของเฟรมเท่ากับ 320 ตัวอย่าง ดังนั้นระยะเวลาของบล็อกเฟรมจะมีค่าเท่ากับ $320/16000 = 0.02$ วินาที (20ms) และหากมีการซ้อนทับกัน 1/2 หรือ 160 เฟรม อัตราการสุ่มตัวอย่างจะมีค่าเท่ากับ $16000/(320-160) = 100$ เฟรมต่อวินาที (frames/s) ลักษณะการทำบล็อกเฟรมแสดงดังรูปที่ 16

2.1.3.2 หน้าต่างแฮมมิง (Hamming window)

เนื่องจากสัญญาณมีลักษณะไม่คงที่และมีการเปลี่ยนแปลงอย่างช้า ๆ ไปตามเวลา ดังนั้นจึงจำเป็นต้องวิเคราะห์สัญญาณเสียงออกเป็นช่วงระยะเวลาสั้น ๆ โดยใช้การบล็อกเฟรมที่ได้อธิบายไว้ในหัวข้อ 2.1.3.1 การใช้หน้าต่างแฮมมิงจะช่วยเน้นสัญญาณเสียงในบล็อกเฟรมที่พิจารณาที่มีความสำคัญสูงสุดโดยลดความสำคัญของสัญญาณเสียงที่อยู่ในกรอบข้าง

$$w(n, \alpha) = (1 - \alpha) - \alpha \cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \quad (3)$$



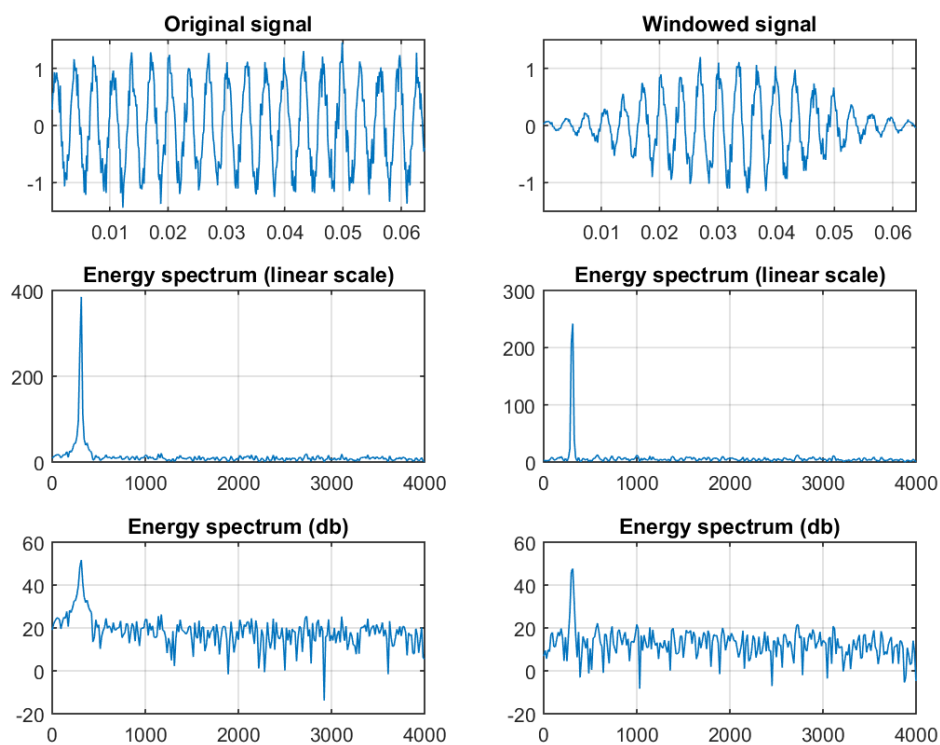
รูปที่ 17 ความสัมพันธ์ระหว่าง α กับการเปลี่ยนแปลงของเส้นโค้งของหน้าต่างแฮมมิง [18]

การแบ่งสัญญาณด้วยเทคนิคหน้าต่างแฮมมิง [19] เพื่อให้สัญญาณย่อยเป็นช่วง ๆ โดยในแต่ละเฟรมที่มาจากการบล็อกเฟรมเพื่อรักษาความต่อเนื่องของสัญญาณเสียงจากจุดแรกไปยังจุดสุดท้าย สัญญาณเสียงในแต่ละเฟรมกำหนดด้วย $s(n), n=0, \dots, N-1$, เมื่อ N คือ ความกว้างของหน้าต่าง ดังนั้นสัญญาณหลังจากการทำแฮมมิงคือ $s(n) * w(n)$ โดยที่ $w(n)$ คือ หน้าต่างแฮมมิงที่คำนวณมาจากสมการที่ (3) และ α คือ ค่าความแตกต่างที่สอดคล้องเส้นโค้งของหน้าต่างแฮมมิง แสดงดังรูปที่ 17 เมื่อค่าของ α เพิ่มขึ้น ความชันของเส้นโค้งก็จะเพิ่มขึ้นตาม โดยค่าจะอยู่ระหว่าง 0 ถึง 1 สำหรับการทดลองได้กำหนดค่าของ α เท่ากับ 0.46 และส่วนที่โค้งลงไปตามขอบของช่องแคบมีผลทำให้เกิดการเลื่อนออกแบบเป็นคาบของเฟรมที่กำลังวิเคราะห์ตามสัญญาณอินพุต แต่ไม่ได้มีผลกระทบต่อพารามิเตอร์ของเสียง

2.1.3.3 การแปลงฟูริเยร์อย่างรวดเร็ว (Fast Fourier Transform: FFT)

เอฟเอฟที [20] เป็นการลดจำนวนครั้งของการคูณจำนวนเชิงซ้อน การวิเคราะห์ทางสเปกตรัมแสดงให้เห็นว่าลักษณะของเสียงที่แตกต่างกันจะสอดคล้องกับการกระจายพลังงานของเสียง

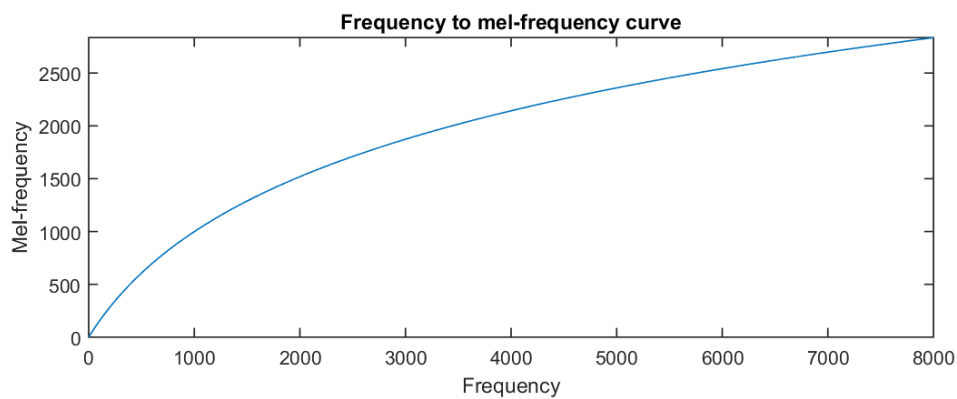
ที่แตกต่างกันมากกว่าความถี่ การใช้ฮอปเฟอที่มากเพื่อมารองรับค่าความถี่ที่เกิดขึ้นในแต่ละเฟรม ในการทำ FFT เฟรมของสัญญาณ สัญญาณที่นำมาแปลงจะต้องเป็นสัญญาณที่ต่อเนื่อง ถ้าหากเป็นสัญญาณไม่ต่อเนื่องจะมีกระบวนการแปลงอยู่ 2 วิธี คือ (1) คูณแต่ละเฟรมของสัญญาณด้วยหน้าต่างแฮมมิงเพื่อเพิ่มความต่อเนื่องของสัญญาณ (2) ใช้ขนาดของเฟรมที่นำมาหาขนาดหน้าต่างกับช่วงเวลาที่สัญญาณไม่ต่อเนื่อง โดยกระบวนการนี้จะมีผลต่อการทำเนื่องจากระยะเวลาที่จะคูณเข้านั้น ไม่ใช่ปัญหาเล็กน้อย นอกจากนี้สัญญาณเสียงที่ออกมาจะก็ไม่มีต่อเนื่องจากสัญญาณเสียงเดิม ดังนั้นจึงนิยมใช้วิธีแรกก่อนที่จะทำการแปลง FFT จากรูปที่ 18 แสดงให้เห็นว่าสัญญาณขาเข้าคือสัญญาณฟังก์ชันไซน์ (sinusoidal) ที่ผสมกับสัญญาณรบกวน (noise) สัญญาณที่ไม่ผ่านกระบวนการทำหน้าต่างแฮมมิงจะมีความไม่ต่อเนื่องกันจากจุดแรกไปยังจุดสุดท้ายของสัญญาณสูงสุด การตอบสนองความถี่จะมีความกว้างขึ้นและไม่ชัดเจน ซึ่งแตกต่างจากสัญญาณผ่านกระบวนการทำหน้าต่างแฮมมิงความถี่ในการตอบสนองของความถี่แคบลงแต่มีความชัดเจนระหว่างจุดเริ่มต้นและจุดสุดท้ายของสัญญาณสูงสุด



รูปที่ 18 การแปลงฟูริเยร์แบบเร็ว (ซ้าย) สัญญาณต้นฉบับ (ขวา) สัญญาณที่ผ่านการทำหน้าต่างแฮมมิง [18]

2.1.3.4 เมลฟิลเตอร์แบงก์ (Mel Filter Bank)

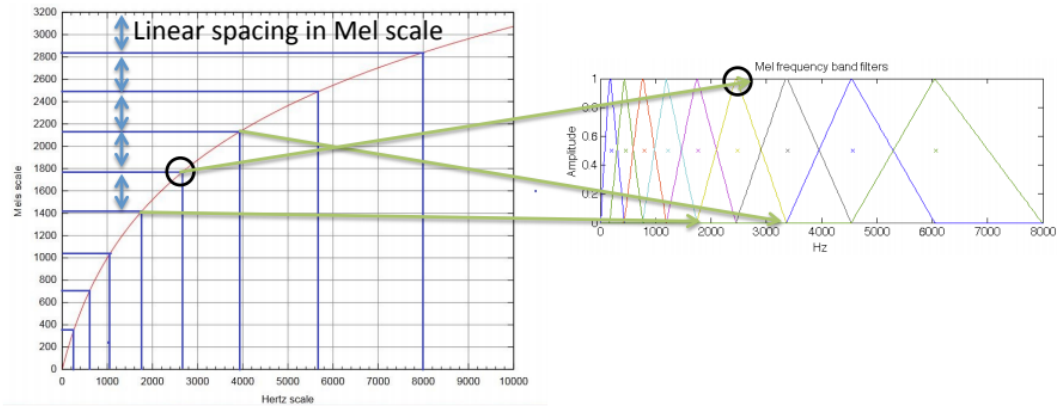
Mel Filter Bank [21] หรือตัวกรองรูปสามเหลี่ยมเป็นการกรองการตอบสนองของความถี่ด้วยตัวกรองแบนพาส (Bandpass) สามารถเก็บได้หลายความถี่ตามที่ต้องการโดยตำแหน่งของตัวกรองเหล่านี้จะมีระยะห่างการกระจายความถี่เป็นไปตามสเกลความถี่ของเมลซึ่งเกี่ยวข้องกับความถี่เชิงเส้นทั่วไป f โดยแสดงดังสมการที่ (4)



รูปที่ 19 ความสัมพันธ์ของความถี่เชิงเส้นที่เป็นสัดส่วนกันกับความถี่เมล [18]

$$\text{Mel}(f) = 1125 \ln\left(1 + \frac{f}{700}\right) \quad (4)$$

ตัวกรองความถี่เมล (Mel-frequency) ได้อาศัยหลักการของการได้ยินเสียงคลื่นสัญญาณความถี่ของมนุษย์ที่ไม่อยู่ในลักษณะเชิงเส้น โดยหลักการนี้ได้ถูกนำไปเปลี่ยนค่าของสัญญาณให้เป็นค่าสเปกตรัมของสัญญาณเสียงเพื่อใช้ในการแสดงค่าลักษณะเด่นจากการใช้ชุดตัวกรองความถี่ (Filter Bank) แสดงดังรูปที่ 19 ซึ่งการแปลงความสัมพันธ์ของความถี่เชิงเส้นมาเป็นสัญญาณรูปสามเหลี่ยมจะแสดงดังรูปที่ 20 ตัวกรองความถี่นี้จะใช้การแปลงฟูรีเยร์แบบรวดเร็วในการออกแบบตัวกรองบนพื้นฐานของสเกลเมลที่ได้อธิบายไว้ข้างต้น การสร้างตัวกรองของสัญญาณเคาะจะได้ขนาด (Magnitude) ของสัญญาณเสียงเคาะนั้น ๆ ออกมาจำนวนมาก ค่าสัมประสิทธิ์ของค่าขนาดจะถูกกรองออกมาผ่านตัวกรองสามเหลี่ยมในแต่ละช่วงความถี่ จากนั้นนำค่าขนาดมาหาค่าเฉลี่ยกลางโดยการนำค่าที่ผ่านตัวกรองมาแปลงฟูรีเยร์แบบรวดเร็วแล้วนำไปคูณกับค่าอัตราการขยายของตัวกรองในแต่ละช่วง สุดท้ายจึงนำค่าที่คำนวณได้มารวมกันแล้วแสดงผลเป็นค่าขนาดของสเปกตรัมของแต่ละช่วงตัวกรอง

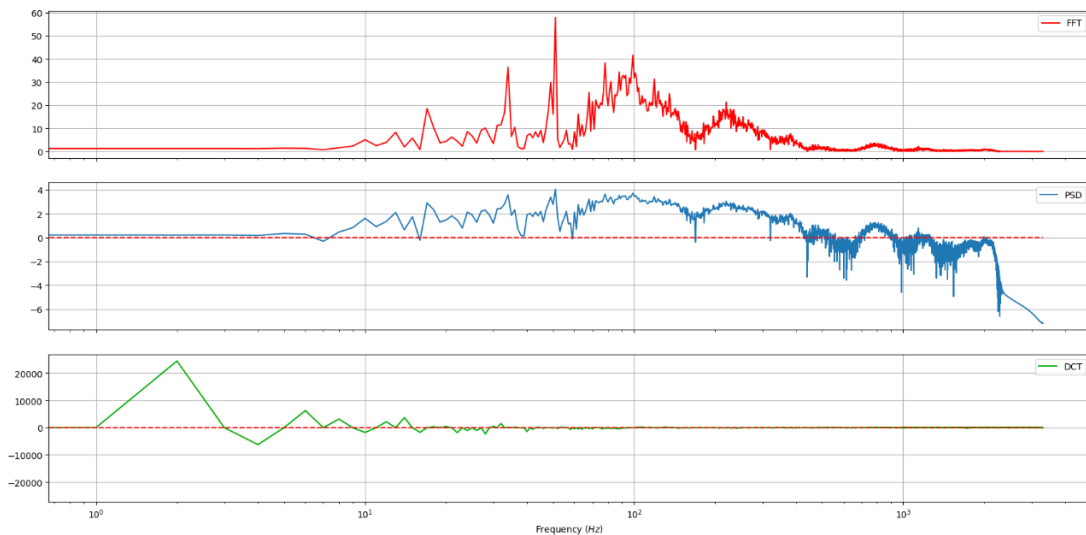


รูปที่ 20 การแปลงความสัมพันธ์ของความถี่เชิงเส้นเป็นช่วงสัญญาณความถี่รูปสามเหลี่ยม

2.1.3.5 การแปลงโคไซน์ไม่ต่อเนื่อง (Discrete Cosine Transform: DCT)

ในขั้นตอนนี้ได้ใช้ DCT [22] กรองสัญญาณที่ผ่านการกรองความแบบเมลฟิลเตอร์แบ่งกั มาแล้วเพื่อหาค่าสัมประสิทธิ์เซปสตรอล (cepstral coefficients) [23] ซึ่งสามารถคำนวณได้ตั้ง สมการที่ (5) โดยที่ E_k คือพลังงานที่ได้จากกระบวนการกรองความถี่และ L คือค่าสัมประสิทธิ์ เซปสตรอลที่ต้องการโดยปกติ ในขณะที่ N คือจำนวนตัวกรองแบนพาสรูป แสดงดังรูปที่ 21

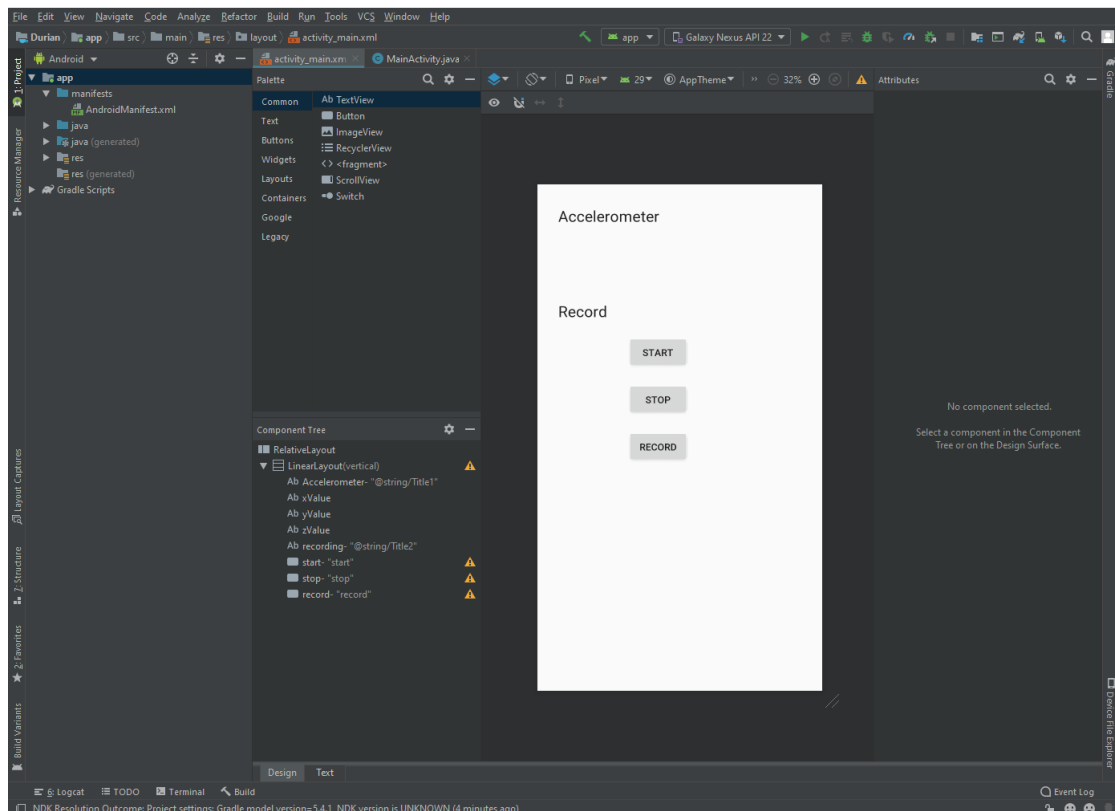
$$C_m = \sum_{k=1}^N \cos\left[m\left(k - 0.5\right)\frac{\pi}{N}\right]E_k, m = 1, 2, \dots, L \quad (5)$$



รูปที่ 21 การแปลงโคไซน์ไม่ต่อเนื่อง (กราฟด้านล่าง)

2.1.4 การพัฒนาแอปพลิเคชัน

ส่วนการพัฒนาแอปพลิเคชันเป็นการอธิบายการประยุกต์ใช้แบบจำลองที่ได้เรียนรู้มาแล้วมาใช้เป็นส่วนหนึ่งในการสร้างแอปพลิเคชันวิเคราะห์ความสุขทุเรียนบนมือถือ

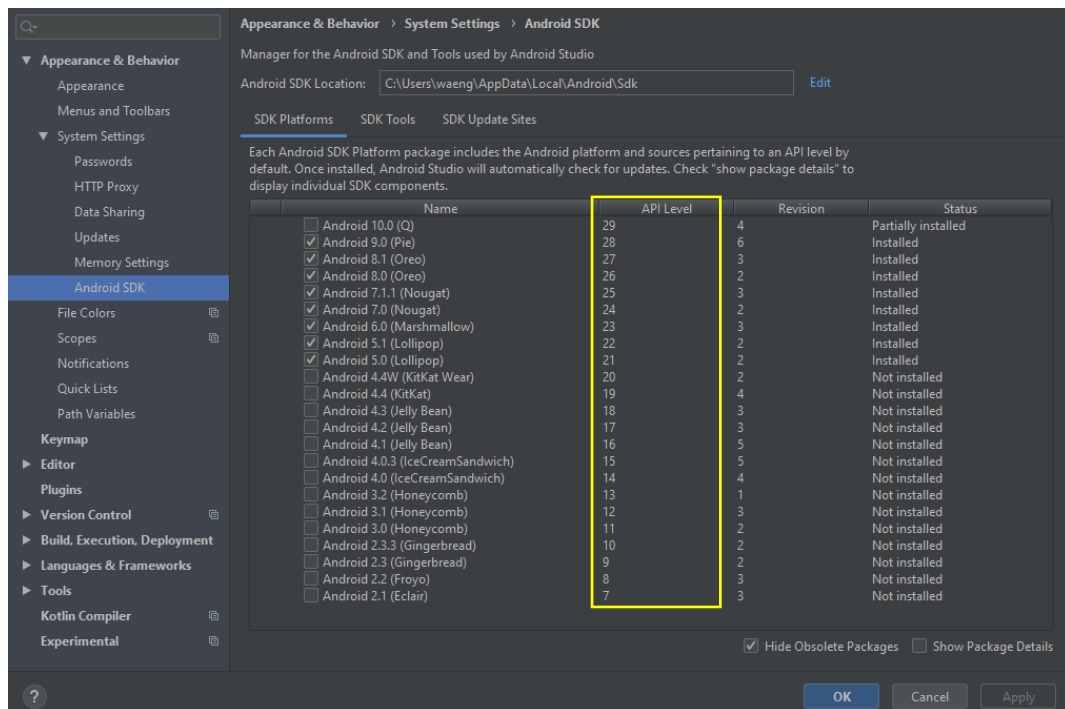


รูปที่ 22 หน้าต่างการทำงานหลักของ Android Studio

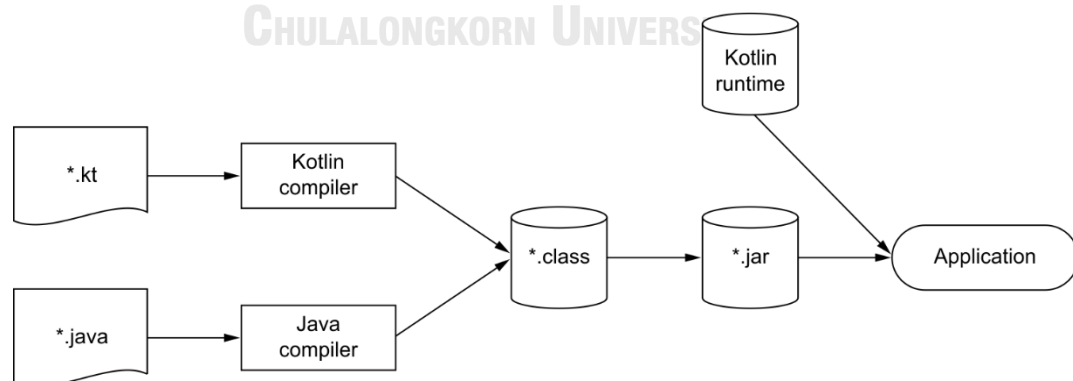
2.1.4.1 Android Studio

เป็นโปรแกรมสำหรับพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการ Android [24] โดยที่ Android Studio [25] ถูกพัฒนาขึ้นโดยบริษัท Google ซึ่งมีพื้นฐานมาจาก Linux [26] ที่เป็นโอเพนซอร์ส (Open source) ให้นักพัฒนาได้ใช้กันอย่างอิสระ ทั้งนี้ต้องอยู่ภายใต้ข้อกำหนดของ Google หน้าต่างการทำงานของ Android Studio แสดงดังรูปที่ 22 ซึ่งพัฒนาจากแนวคิดพื้นฐานของ IntelliJ IDEA การทำงานคล้ายกับโปรแกรม Eclipse วัตถุประสงค์ของ Android Studio คือเป็นเครื่องมือไอทีที่สามารถพัฒนาแอปพลิเคชันบน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ Graphic User Interface: GUI สามารถดูตัวอย่างการออกแบบบนสมาร์โฟนได้ และยังสามารถแสดงผลบางอย่างได้ทันทีโดยไม่ต้องรันแอปพลิเคชันบนโปรแกรมจำลองมือถือ (Emulator) นอกจากนี้ยังมี Android Software Development Kit: Android SDK ที่เปรียบเสมือนไลบรารีสำหรับการพัฒนาแอปพลิเคชันบน Android เนื่องจาก Android มีหลายรุ่น Android SDK จึงมีหลาย

รุ่นสำหรับการเลือกพัฒนาให้ตรงตามความต้องการ ยกตัวอย่าง หากต้องการพัฒนาแอปพลิเคชันบนระดับเอพียู (API) รุ่นเก่าแอปพลิเคชันจะสามารถรันบนระดับเอพียูรุ่นใหม่ได้ ในทางกลับกันหากต้องการพัฒนาได้แอปพลิเคชันบนระดับเอพียูรุ่นใหม่ แอปพลิเคชันจะไม่สามารถรันบนเอพียูรุ่นเก่าได้ การตั้งค่ารุ่นที่ต้องการพัฒนาแอปพลิเคชันแสดงดังรูปที่ 23 ภายในกรอบสี่เหลี่ยมแสดงหมายเลขรุ่นของ API Level



รูปที่ 23 การเลือก Android SDK สำหรับการพัฒนาแอปพลิเคชัน



รูปที่ 24 ระบบการ compile ระหว่างภาษาจาวา (*.java) กับภาษา Kotlin (*.kt) [27]

2.1.4.2 ภาษาสำหรับการพัฒนาแอปพลิเคชัน

ภาษาที่ใช้สำหรับการพัฒนาแอปพลิเคชันได้ใช้ภาษา Kotlin [28] ซึ่งเป็นภาษาใหม่ที่พัฒนา มาเพื่อใช้แทนภาษาจาวา (JAVA) แนวคิดของ Kotlin คือสามารถพัฒนาร่วมกันกับภาษาจาวาได้ ทำให้สามารถใช้ประโยชน์จากไลบรารี (Library), Applications Program Interface: API [29], และ เครื่องมือที่มีอยู่ในภาษาจาวาได้ ไฟล์ของ Kotlin มีนามสกุลของไฟล์คือ .kt และมีตัวคอมไพเลอร์ (compiler) ชื่อว่า kotlinc เช่นเดียวกับภาษาจาวาที่มีตัวคอมไพเลอร์ชื่อว่า javac กระบวนการ คอมไพล์ (compile) แสดงในรูปแบบที่ 24 จากรูปจะเห็นว่าไฟล์ .kt กับ .java ถูกแปลงไปเป็นไฟล์ .class และ .jar ตามลำดับ จากนั้นนำไปรันบน Java Virtual Machine: JVM [29] ที่สามารถทดสอบการใช้งาน ได้บนเว็บไซต์ <https://kotlinlang.org/> เพื่อให้เข้าใจการทำงานของภาษา Kotlin ได้ดียิ่งขึ้น

2.1.4.3 เอพีไอ (Application Program Interface: API)

เอพีไอทำหน้าที่ช่วยในการเข้าถึงข้อมูล นำข้อมูลออกหรือส่งข้อมูลเข้าไปบนเว็บไซต์ได้ เจ้าของเว็บไซต์ที่มีเอพีไอจะกำหนดขอบเขตในการเข้าถึงบริการต่าง ๆ ของทางเว็บไซต์ได้ ปัจจุบันมี การนำเอพีไอไปใช้กับแอปพลิเคชันได้หลากหลายเพื่อให้สามารถสื่อสารกับผู้ใช้บริการได้ง่ายขึ้น โดยที่ ผู้ใช้บริการไม่จำเป็นต้องมีความรู้ในด้านนั้น ๆ เช่น บริษัทใหญ่หลายบริษัทเปิดเอพีไอเพื่อให้ ผู้ใช้บริการสามารถเข้าใช้งานได้ เช่น Facebook, Google, Twitter และอื่น ๆ อีกมากมาย ทำให้ ผู้พัฒนาระบบสามารถนำเอาเอพีไอเหล่านี้ไปต่อยอดได้ การใช้งานเอพีไอมีการใช้งานหลักอยู่ 4 รูปแบบดังนี้

Web APIs นิยมใช้กันอย่างแพร่หลายในปัจจุบัน เนื่องจากอยู่ในกลุ่มการสื่อสารในรูปแบบ Hypertext Transfer Protocol: HTTP และยังสามารถขยายการใช้งานไปสู่รูปแบบ Extensible Markup Language: XML และ JavaScript Object Notation: JSON ได้ ซึ่งรวมอยู่ใน Web service เช่น Simple Object Access Protocol: SOAP ที่ใช้รูปแบบ XML หรือ JSON ในการรับ-ส่งข้อมูลแบบ Representational State Transfer: REST

Operating Systems API สามารถใช้งานในการสื่อสารระหว่างแอปพลิเคชันและ ระบบปฏิบัติการ (Operating System: OS) เช่น Portable Operating System Interface: POSIX หรือ มาตรฐานการสื่อสารของโอเอสเองก็มีเอพีไอเป็นชุดคำสั่งควบคุมหรือ command line เพื่อ ควบคุมการทำงานของโอเอส

Remote APIs คือไลบรารีอย่างหนึ่งใน Java SDK ที่สร้างขึ้นมานักพัฒนาสามารถเข้าไป ควบคุมทรัพยากรผ่านทางโปรโตคอลเพื่อให้มีมาตรฐานการสื่อสารเดียวกัน เช่น สามารถใช้ Remote API เพื่อเข้าถึงที่เก็บข้อมูลการผลิตจากแอปที่ทำงานบนเครื่อง และยังสามารถใช้ API ระยะไกลเพื่อเข้าถึงที่เก็บข้อมูลจากแอปหนึ่งไปยังอีกแอปหนึ่ง

Libraries คือชุดของฟังก์ชันหรือวัตถุ (object) ที่ผู้ใช้งานสร้างขึ้นเพื่อวัตถุประสงค์บางอย่าง โดยสามารถนำไปประยุกต์ใช้ได้กับหลาย ๆ โปรแกรม และเฟรมเวิร์ค (frameworks) คือชุดของรูปแบบ (pattern) หรือไลบรารี เพื่อช่วยในการสร้างโปรแกรมประยุกต์ และเอพีไอคืออินเทอร์เฟซสำหรับโปรแกรมในการโต้ตอบกับโปรแกรมที่ถูกสร้างขึ้นโดยไม่ต้องเข้าถึงโดยตรง สรุปได้ว่า framework API คือโปรแกรมประยุกต์ที่สามารถโต้ตอบกับผู้ใช้งานซึ่งถูกสร้างมาจากไลบรารี

2.1.4.4 REST และ RESTful API

Representational state transfer: REST [30] ถูกพัฒนาโดย Roy Fielding Ph.D. [31] เป็นการนำสถาปัตยกรรมซอฟต์แวร์ที่ใช้ประโยชน์จากเทคโนโลยีการสื่อสารผ่านเว็บโปรโตคอล (Web Protocol) ในการสร้างเว็บเซอร์วิส (Web Service) เพื่อให้ผู้ใช้บริการสามารถเรียกใช้เว็บของเราได้ โครงสร้างของ REST ประกอบด้วย 7 รูปแบบโครงสร้างดังนี้:

Performance: ประสิทธิภาพในการตอบโต้ของระบบ

Scalability: สามารถปรับขยายขนาดโครงสร้างภายในเพื่อให้สามารถรองรับกับอุปกรณ์จำนวนมากได้

Simplicity: มีความเรียบง่ายในการเชื่อมต่อหรือสื่อสารในโครงสร้างเครือข่ายที่เหมือนกัน

Modifiability: มีความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลงหรือความต้องการของผู้ให้บริการแม้ในขณะที่แอปพลิเคชันทำงานอยู่

Visibility: มองเห็นการสื่อสารระหว่างองค์ประกอบต่าง ๆ ภายในเครือข่ายผ่านผู้ให้บริการ

Portability: ง่ายต่อการโยกย้ายทรัพยากรต่าง ๆ ไม่ว่าจะเป็น โปรแกรม หรือข้อมูล

Reliability: มีความน่าเชื่อถือในด้านความทนทาน ความล้มเหลวหรือข้อผิดพลาดที่เกิดขึ้นในระบบ

RESTful [32] เป็นบริการที่ใช้สถาปัตยกรรมเดียวกับ REST แต่เจ้าของระบบสามารถอนุญาตให้ระบบส่งคำขอ (Request) เข้าถึงทรัพยากร (Resource) ต่าง ๆ บนเว็บได้โดยใช้ชุดคำสั่งที่ผู้ให้บริการได้กำหนดเอาไว้ล่วงหน้า การโต้ตอบของระบบที่ใช้ REST อยู่บนพื้นฐานของ HTTP ส่งคำขอไปยัง URL ที่ได้กำหนด จากนั้นผู้ให้บริการจะโต้ตอบ (Response) กลับมาเป็นกลุ่มข้อมูล (Payload) ในรูปแบบ XML, HTML, JSON หรือรูปแบบอื่น ๆ เอพีไอแบบ RESTful มีผู้ใช้บริการ (client) หรือผู้ต้องการส่งคำขอใช้ทรัพยากร และเซิร์ฟเวอร์ (Server) เป็นผู้ให้บริการทรัพยากร ชุดคำสั่งที่ใช้ในการส่งคำขอในระบบ RESTful ผ่าน HTTP มีดังนี้:

1. GET เป็นคำขอเพื่อต้องการข้อมูลจาก resource

2. POST เป็นคำขอเพื่อสร้างข้อมูลใหม่ใน resource
3. PUT เป็นคำขออัปเดตข้อมูลที่มีอยู่แล้ว หรือสร้างชุดข้อมูลใหม่ใน resource
4. DELETE เป็นคำขอลบข้อมูลที่มีอยู่แล้วใน resource

นอกจากนี้ยังมีสถานะการโต้ตอบของคำขอซึ่งจะเป็นชุดตัวเลข ได้แก่

กลุ่มของคำขอที่สำเร็จ 2xx (success code)

- 200 OK – เป็นการโต้ตอบแบบสำเร็จ (Response success) สำหรับ GET, PUT หรือ POST
- 201 Create – เป็นการโต้ตอบสำหรับข้อมูลที่ถูกสร้างขึ้นใหม่ใช้สำหรับ POST
- 204 No Content – เป็นการโต้ตอบสำหรับคำขอที่สำเร็จแต่ไม่มีการส่งข้อมูลกลับมา

กลุ่มของการเปลี่ยนเส้นทาง 3xx (redirection)

- 304 Not Modified - เป็นการบอกผู้ใช้บริการว่าได้รับการโต้ตอบแล้ว ในที่เก็บซ่อน cache และไม่จำเป็นต้องส่งข้อมูลเดิมอีกครั้ง

กลุ่มคำขอผิดพลาดของผู้ขอใช้บริการ 4xx (client error)

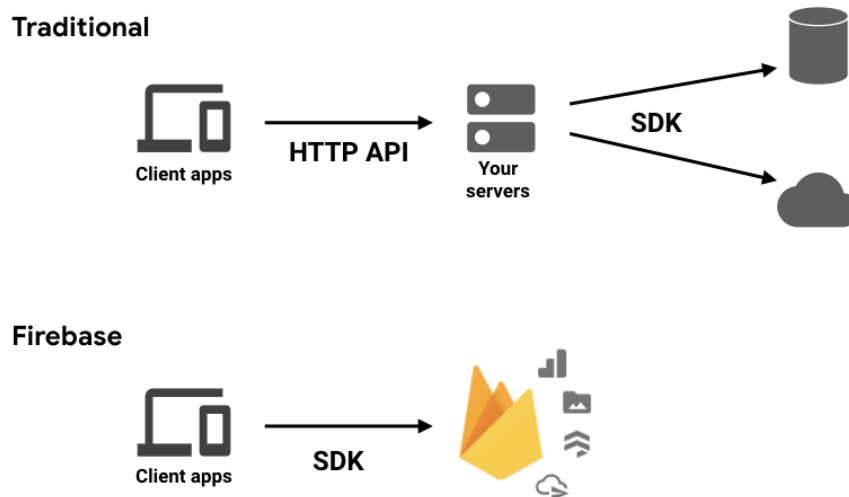
- 400 Bad Request – คำขอที่ส่งมาโดยผู้ใช้บริการนั้นไม่ได้ถูกดำเนินการ
- 401 Unauthorized – ผู้ใช้บริการไม่ได้รับอนุญาตให้เข้าถึงข้อมูลและควรต้องส่งคำขอใหม่ด้วยหนังสือรับรอง (credential)
- 403 Forbidden – บ่งบอกว่าคำขอนั้นถูกต้องและผู้ใช้บริการได้รับการอนุญาตให้เข้าถึงข้อมูล แต่มีเหตุผลบางประการที่ทำให้ผู้ใช้บริการไม่สามารถเข้าถึงข้อมูลได้
- 404 Not Found – ข้อมูลที่ได้ส่งคำขอไปนั้นไม่วางใช้งาน
- 405 Gone – ข้อมูลไม่มีอยู่หรือถูกย้ายไป

กลุ่มผู้ให้บริการเกิดข้อผิดพลาด 5xx (server error)

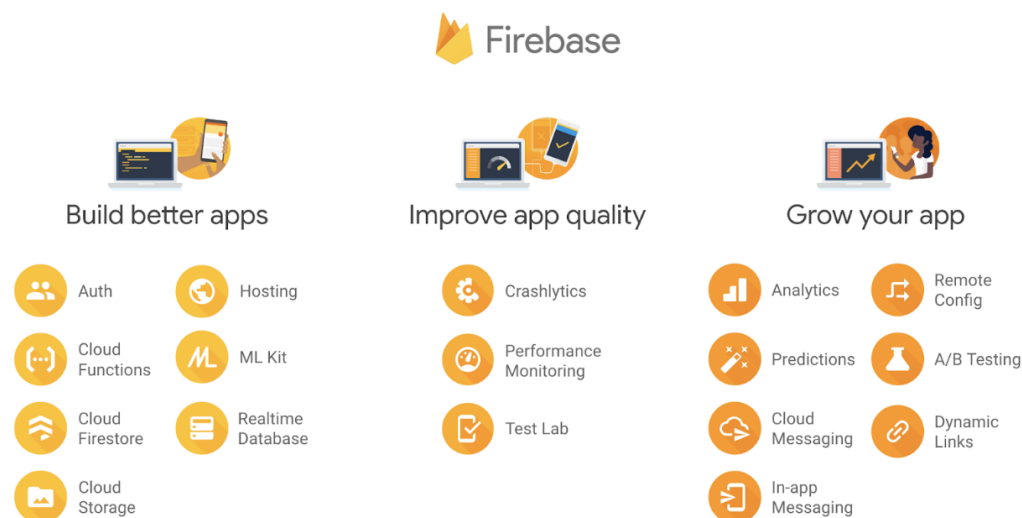
- 500 Internal Server Error – คำขอที่ส่งไปถูกต้อง แต่ผู้ให้บริการมีข้อผิดพลาดและไม่สามารถให้ข้อมูลที่ต้องการได้
- 503 Service Unavailable – ผู้ให้บริการใช้การไม่ได้ เนื่องจากอาจจะอยู่ในช่วงซ่อมบำรุงระบบ

2.1.4.5 ไฟร์เบส (Firebase)

ไฟร์เบสเป็นชุดเครื่องมือในการสร้าง ปรับปรุง และพัฒนาแอปพลิเคชัน และเป็นเครื่องมือที่จะช่วยให้ความครอบคลุมขนาดใหญ่ของการบริการที่นักพัฒนาจะต้องสร้างเอง แต่ไม่ต้องการที่จะสร้างเอง โดยได้มีการเปรียบเทียบระหว่างการสร้างระบบการจัดเก็บข้อมูลแบบเดิมกับแบบที่บริการของไฟร์เบสแสดงดังรูปที่ 25 ทำให้เห็นว่าระบบไฟร์เบสลดความซับซ้อนในการสร้างแอปพลิเคชันได้อย่างมาก ซึ่งเป็นข้อดีสำหรับนักพัฒนาที่ไม่มีความรู้ในการติดตั้งเซิร์ฟเวอร์ทำได้ง่ายขึ้น เนื่องจากต้องการที่จะมุ่งเน้นไปที่ประสบการณ์ของแอปพลิเคชันเอง รวมไปถึงการวิเคราะห์ การรับรองความถูกต้อง (authentication) การสร้างฐานข้อมูล จัดเก็บไฟล์ ส่งข้อความ เป็นต้น กล่าวคือไฟร์เบสเป็นฐานข้อมูลเรียลไทม์ซึ่งมีเอพีไอที่ช่วยให้นักพัฒนาจัดเก็บและซิงค์ข้อมูล ซึ่งบริการต่าง ๆ ที่ไฟร์เบสรองรับแสดงดังรูปที่ 26



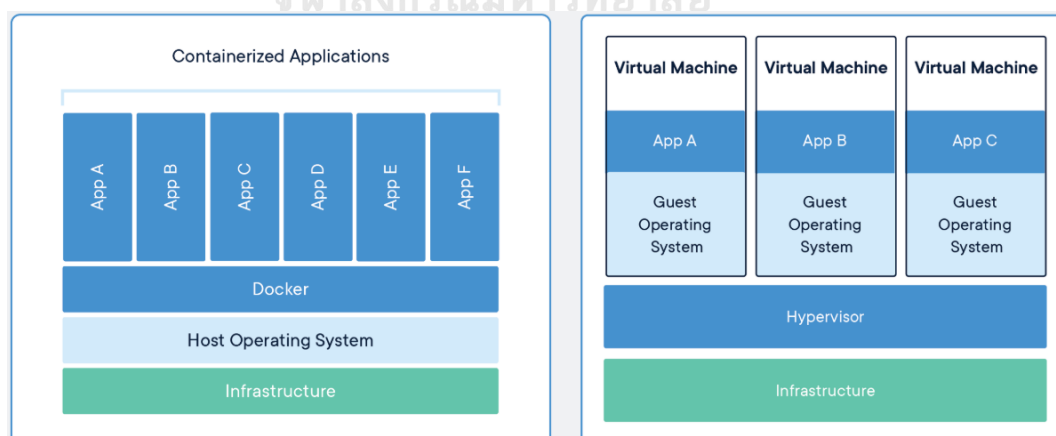
รูปที่ 25 รูปแบบการจัดเก็บข้อมูลแบบไฟร์เบส [33]



รูปที่ 26 บริการของไฟร์เบส [33]

2.1.4.6 ดอคเกอร์ Docker

Docker [34] คือเครื่องมือที่ออกแบบมาเพื่อจำลองสภาพแวดล้อมบนเครื่องเซิร์ฟเวอร์ เพื่อรันโปรแกรมที่ต้องการ มีการทำงานคล้ายกับ Virtual Machine เช่น VMWare, VirtualBox, XEN, KVM แต่ docker ไม่ได้จำลองในส่วนของ OS แต่ใช้ container ในการจำลองสภาพแวดล้อมขึ้นมาเพื่อใช้งานสำหรับ 1 บริการที่ต้องการใช้งานเท่านั้น ซึ่งสามารถใช้งานได้สะดวกและตอบสนองความต้องการของนักพัฒนาหรือผู้ดูแลระบบแสดงดังรูปที่ 27 ซึ่งเป็นการอธิบายความแตกต่างระหว่าง docker container กับ virtual machine



รูปที่ 27 การจำลองสภาพแวดล้อมด้วย docker container (ซ้าย) การจำลองสภาพแวดล้อมด้วย virtual machine (ขวา) [35]

Docker image ตัวต้นแบบของคอนเทนเนอร์ (container) ภายในประกอบด้วยแอปพลิเคชันต่าง ๆ ที่มีการติดตั้งไว้เพื่อใช้งานสำหรับการบริการนั้น ๆ รวมทั้งมีการคอนฟิก (config) ค่าต่าง ๆ ไว้ จากนั้นนำมาสร้างเป็น docker image เพื่อนำไปใช้งาน

Docker container สามารถมองได้เสมือนกล่องที่นำ docker image มาติดตั้ง เพื่อให้สามารถใช้งานบริการที่ต้องการจาก image นั้น ๆ ได้ ในคอนเทนเนอร์แต่ละตัวจะมีการใช้งาน RAM, CPU, ไฟล์คอนฟิกต่าง ๆ เป็นของแต่ละคอนเทนเนอร์เอง และสามารถสั่ง start, stop ได้ที่คอนเทนเนอร์นั้น ๆ ด้วย โดยที่ตัว Docker container นั้นจะถูกรันอยู่ในบริการของ Google Cloud Run [36]

2.1.4.7 คลาวด์รัน Cloud Run

Cloud Run [36] เป็นแพลตฟอร์มการคำนวณ (Compute Platform) ที่ถูกสร้างขึ้นโดยบริษัท Google ที่มีการจัดการทรัพยากรและปรับขนาดแอปพลิเคชันในคอนเทนเนอร์ให้มีประสิทธิภาพ ปลอดภัย และรวดเร็วต่อการใช้งาน สามารถเขียนโค้ดได้หลากหลายภาษาเช่น Go, Python, Java, Ruby, Node.js เป็นต้น มีการจัดการระบบอย่างเรียบง่ายเพื่อให้นักพัฒนาสามารถพัฒนาระบบได้อย่างมีประสิทธิภาพ สามารถใช้งานได้หลากหลาย เช่น Websites, REST APIs backend, Back-office administration เป็นต้น ซึ่งได้นำ Cloud Run ในการสร้าง REST APIs backed โดยที่ระบบนั้นได้ถูกเขียนขึ้นด้วยภาษาไพธอนซึ่งใช้ Flask [37] เป็นเว็บเฟรมเวิร์ค (web framework)

2.1.4.8 ฟลัสก์ Flask

เป็นไมโครเว็บเฟรมเวิร์คถูกเขียนจากภาษาไพธอนไม่ต้องการเครื่องมือหรือไลบรารีเฉพาะ ทำให้มีแค่เครื่องมือที่จำเป็น โครงสร้างจึงมีขนาดเล็ก โดยเขียนโค้ดไม่กี่บรรทัดก็สามารถรันได้เลย Flask ไม่มีฐานข้อมูล (Database) หรือการตรวจสอบแบบฟอร์ม (Form validation) นอกจากนี้ยังสามารถรองรับส่วนขยาย (extensions) ที่ช่วยเพิ่มความสามารถอื่นได้ ตัวอย่างเว็บไซต์ที่นำ Flask ไปใช้งานเช่น Pinterest.com, LinkedIn.com เป็นต้น ในปัจจุบันมีผู้ใช้ Flask Framework จำนวนมาก เป็นผลมาจากการใช้งานที่ง่ายและผนวกกับมีผู้ใช้งานภาษาไพธอนเพิ่มขึ้น [38] นอกจากนี้ Flask Framework ยังสามารถใช้งานร่วมกับ Hyper Text Markup Language: HTML ซึ่งเป็นภาษาที่ใช้ในการแสดงผลเอกสารบนเว็บไซต์จากพัฒนาทางด้านซอฟต์แวร์ของบริษัทไมโครซอฟท์ ทำให้ภาษา HTML เป็นภาษาหนึ่งที่ใช้เขียนโปรแกรมได้

2.1.5 การทดสอบประสิทธิภาพของแบบจำลอง

การทดสอบประสิทธิภาพของแบบจำลองโดยใช้ตารางความสำคัญในการวัดประสิทธิภาพของการเรียนรู้ของเครื่องหรือ Confusion Matrix table [38] ด้วยพารามิเตอร์ดังนี้: ประสิทธิภาพ

ของการคัดแยก (Classification) ของแบบจำลอง เป็นการอธิบายเพื่อให้เข้าใจหลักการทำงาน ยกตัวอย่างหากมีรูปภาพทั้งหมด 10 รูป เป็นรูปหมา 5 รูปและรูปแมว 5 รูป ให้แบบจำลองคัดแยกว่า รูปใดเป็นหมาและรูปใดเป็นแมว จากนั้นนำผลลัพธ์ที่แบบจำลองทำนายมาใส่ตารางที่ 1 แนวตั้งคือค่าจริง (actual class) และแนวนอนคือค่าทำนาย (predicted class) เพื่อให้เข้าใจง่ายจะทำการเปลี่ยนชื่อเรียกเป็นแบบไบนารี (Binary) คือ ใช่แมวหรือไม่ใช่แมวตารางที่ 2

ตารางที่ 1 การใส่การคัดแยกลงใน Confusion Matrix

		ค่าจริง (Actual class)	
		แมว	หมา
ค่าทำนาย (Predicted class)	แมว	4	3
	หมา	1	2

ตารางที่ 2 การแปลงข้อมูลการคัดแยกเป็นแบบ Binary บน Confusion Matrix

		ค่าจริง (actual class)	
		แมว	ไม่ใช่แมว
ค่าทำนาย (Predicted class)	แมว	4 (True Positive)	3 (False Positive)
	ไม่ใช่แมว	1 (False Negative)	2 (True Negative)

เมื่อชื่อเรียกแต่ละช่องมีความหมายดังนี้

True Positive (TP): ทายว่าใช่ แล้วใช่จริง ๆ

True Negative (TN): ทายว่าไม่ใช่ แล้วไม่ใช่จริง ๆ

False Positive (FP): ทายว่าใช่ แต่จริง ๆ แล้วไม่ใช่

False Negative (FN): ทายว่าไม่ใช่ แต่จริง ๆ แล้วมันใช่

Condition Positive (P): จำนวนส่วนที่ใช่ทั้งหมดที่มีอยู่ในข้อมูล = TP+FN

Condition Negative (N): จำนวนส่วนที่ไม่ใช่ทั้งหมดที่มีอยู่ในข้อมูล = FP+TN

*หมายเหตุ True = ทายถูก, False = ทายผิด, Positive = ทายว่าใช่, Negative = ทายว่าไม่ใช่

การคำนวณหาความถูกต้อง (Accuracy) ได้จากจำนวนครั้งที่ทายถูกหารด้วยจำนวนครั้งที่ทายทั้งหมด ตามสมการที่ (6) และการคำนวณหาความแม่นยำ (Precision) ได้จากจำนวนครั้งที่ทายว่าใช่แล้วถูกต้องหารด้วยจำนวนครั้งที่ทายว่าใช่ทั้งหมดซึ่งมีชื่อเรียกอีกชื่อว่า Positive predictive value (PPV) แสดงดังสมการที่ (7) และจำนวนครั้งที่ทายว่าใช่แล้วถูกหารด้วยจำนวนที่ทายว่าใช่ทั้งหมดในข้อมูลโดยมีชื่อเรียกว่า Recall หรืออีกชื่อ True positive rate (TPR) แสดงดังสมการที่ (8) และการคำนวณที่นิยมอีกอย่างหนึ่งคือ F1-Score เป็นการหาค่าเฉลี่ยฮาร์มอนิก (Harmonic) แทนการใช้ค่าเฉลี่ยที่มีค่าที่แตกต่างกันมาก ๆ เพื่อให้ความแม่นยำมีความถูกต้องมากขึ้นสามารถคำนวณได้ดังสมการที่ (9)

$$Accuracy = \frac{TP + TN}{P + N} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1 - Score = \frac{2(precision)(recall)}{precision + recall} \quad (9)$$

2.2 งานวิจัยที่เกี่ยวข้อง

Rong Phoophuangpairaj [39] ได้ศึกษาเกี่ยวกับลักษณะของสัญญาณและพัฒนาการกระบวนการจดจำเสียงในระดับความสูงต่าง ๆ ของทุเรียนโดยปราศจากการตัดหรือทำลายผลทุเรียน กระบวนการตรวจสอบระดับความสูงใช้วิธีการเคาะที่ผลทุเรียนและบันทึกสัญญาณเสียงที่ได้จากการเคาะด้วยคอมพิวเตอร์ และการบันทึกเสียง 1 ไฟล์ประกอบด้วยผลการเคาะที่ผลทุเรียน 5 ครั้ง จากนั้นนำสัญญาณเสียงที่ได้เข้าสู่กระบวนการการเรียนรู้เชิงลึก การสกัดคุณสมบัตินเด่นของสัญญาณเสียงใช้เทคนิค Mel-Frequency Cepstral Coefficients (MFCCs) โดยแบ่งกลุ่มการเรียนรู้ออกเป็น 2 กลุ่มคือ กลุ่มทุเรียนสุก (RIPE) และกลุ่มทุเรียนที่ไม่สุก (UNRIPE) และใช้ HMM เพื่อสร้างเครื่องหมาย “UNRIPE” และ “RIPE” ขึ้นมาระหว่างการเคาะเพื่อทดสอบทุเรียน การหาความสูง-ตึบด้วยแบบจำลอง N-gram และคำนวณกลุ่มที่เป็น N-best โดยใช้ 3-best เป็นตัวบอกว่าทุเรียนสุกหรือตึบ ประสิทธิภาพความแม่นยำของแบบจำลองมีความแม่นยำมากกว่า 90% ซึ่งเป็นค่าที่มากพอสำหรับการตรวจสอบความสูง-ตึบของทุเรียน

ธีรพล พลิตวานนท์ [40] และคณะได้สร้างอุปกรณ์วัดความสุขของผลไม้แบบไม่ทำลายผล ด้วยการประมวลผลสัญญาณเสียงเคาะ โดยสัญญาณเสียงถูกบันทึกผ่านคอนเดนเซอร์และแปลงให้อยู่ในรูปของข้อมูลดิจิทัล รูปแบบการบันทึกใช้อัตราการสุ่มข้อมูลความเร็ว (Sampling rate) 44,100 Hz ที่ความละเอียด 16 บิต ข้อมูลมีความยาว 4,096 จุด นำเข้ากระบวนการแบ่งช่วงย่อยของหน้าต่าง (Windowing) ขนาด 128 จุด โดยส่วนนี้นำไปหาค่าเบี่ยงเบนมาตรฐานเพื่อหาอัตราการเปลี่ยนแปลงของข้อมูลที่เกิดขึ้นในแต่ละช่วงเวลา ในกระบวนการที่กล่าวมาข้างต้นทำให้ได้ค่าส่วนเบี่ยงเบนมาตรฐานจำนวน 32 จุด ในการเคาะหนึ่งครั้ง กระบวนการต่อไปเป็นกระบวนการตัดสินใจความสุขของผลไม้ โดยใช้ฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability Density Function: PDF) ในการทดลองกับผลไม้จำนวนสองชนิด คือ สับปะรดและทุเรียนพันธุ์หมอนทอง ผลการทดลองพบว่า อุปกรณ์สามารถคัดแยกความสุขของผลไม้ได้ความถูกต้องเฉลี่ย 89.6 เปอร์เซ็นต์ และสามารถแยกออกเป็นชนิดได้ คือ ผลสับปะรด 95.0 เปอร์เซ็นต์ ทุเรียน (น้ำหนัก 3.5-4.0กก.) 85.0 เปอร์เซ็นต์ และทุเรียน (น้ำหนัก 4.0-4.5กก.) 88.7 เปอร์เซ็นต์

Karol J. Piczak [41] ศึกษาเกี่ยวกับการจำแนกคลิปเสียงสั้นๆ ของสิ่งแวดล้อมโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network หรือ CNN) ออกแบบแบบจำลองที่ประกอบด้วย 2 ชั้น convolutional กับ max-pooling และ 2 ชั้น fully connected สำหรับการเรียนรู้ของแบบจำลอง การสกัดสัญญาณเสียงแบบ spectrograms ด้วยวิธี deltas แบบจำลองนี้ได้ทำการทดสอบกับกลุ่มข้อมูลเสียงสาธารณะจำนวน 3 กลุ่ม ที่ได้บันทึกเสียงสภาพแวดล้อมและเสียงในเมือง แบบจำลองนั้นมีประสิทธิภาพสูงกว่าการใช้วิธีสกัดสัญญาณเสียงด้วยวิธีพื้นฐาน MFCCs ซึ่งเปรียบกับการสกัดสัญญาณเสียงแบบอื่น ๆ

Fuad Noman [42] และคณะได้ศึกษาการจำแนกเสียงการเต้นของหัวใจช่วงสั้นโดยใช้เครือข่ายประสาทเทียมเชิงลึก เพื่อจำแนกรูปแบบการเต้นของหัวใจของแต่ละบุคคล แบบจำลองออกแบบเป็น 2 ส่วน คือ 1D-CNN สำหรับการเรียนรู้คุณลักษณะโดยตรงของสัญญาณเสียงของหัวใจ และแบบ 2D-CNN สำหรับรับค่า 2 มิติ time-frequency จากการสกัดคุณลักษณะแบบ MFCCs ซึ่งกลุ่มข้อมูลเสียงได้มาจาก PhysioNet CinC challenge 2016 [43] แบบจำลองที่เสนอนี้มีประสิทธิภาพสูงกว่าแบบดั้งเดิมที่ใช้ Support Vector Machine (SVM) และ Hidden Markov Model (HMM) โดยแบบจำลองที่ดีที่สุดของให้ความแม่นยำสูงสุด 91.55%

บทที่ 3

วิธีการดำเนินการวิจัย

ในบทนี้เป็นการอธิบายวิธีดำเนินงานวิจัย แบ่งออกเป็นหลักการเก็บข้อมูลเสียงเพื่อนำมาใช้ในการสร้างแบบจำลอง อัลกอริทึมที่ใช้ในการออกแบบแบบจำลองและผลการทดลองเบื้องต้น

3.1 การเตรียมชุดข้อมูลเสียงเคาะทุเรียนและกระบวนการเรียนรู้เสียงทุเรียน

การทำงานของระบบแบ่งออกเป็น 2 ส่วน ส่วนแรกคือการเตรียมชุดข้อมูลเสียงเพื่อนำมาเรียนรู้และสร้างแบบจำลอง ส่วนที่สองเป็นการนำแบบจำลองมาใช้งานบนแอปพลิเคชัน

3.1.1 การเตรียมชุดเสียงเคาะทุเรียน

ระบบการเรียนรู้ชุดข้อมูลเสียงเริ่มจากการเก็บข้อมูลเสียงเคาะทุเรียน ที่ได้รับความอนุเคราะห์ ตัวอย่างทุเรียนพันธุ์หมอนทองจากพ่อค้าคนกลางทรัพย์อนันท์ ในจังหวัดจันทบุรี ทางพ่อค้าคนกลาง ได้จัดเตรียมทุเรียนสำหรับการส่งออก ที่มีความสุกที่แตกต่างกันไปตามระยะทางการขนส่งในรูปแบบที่ 28 การจัดเตรียมสำหรับห้างสรรพสินค้าภายในประเทศ ผู้วิจัยได้สุ่มทุเรียนจำนวน 30 ลูก ประกอบด้วยทุเรียนที่คละความสุกกัน 3 ระดับ ได้แก่ ไม่สุก (ยังไม่สุกพอสำหรับการบริโภค ต้องปล่อยให้ทิ้งไว้ประมาณหนึ่งสัปดาห์ถึงจะสามารถรับประทานได้) สุกปานกลาง (ยังสุกไม่เต็มที่แต่สามารถรับประทานได้ โดยเนื้อทุเรียนข้างในจะมีความกรอบ) สุกพร้อมรับประทาน (สุกแบบพอดีผลข้างในนุ่มไม่เละ) ขณะที่เคาะเพื่อเก็บข้อมูลเสียงจะมีชาวสวนคอยระบุความสุกของทุเรียนที่กำลังเคาะอยู่

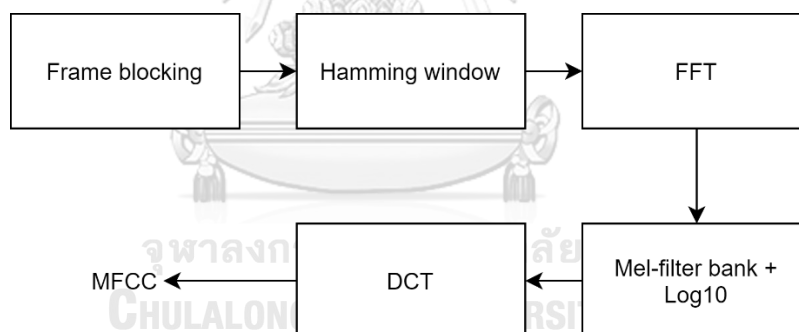


รูปที่ 28 การจัดเตรียมการขนส่งทุเรียนให้ห้างสรรพสินค้าภายในประเทศ

การเก็บข้อมูลนี้ได้ใช้สมาร์ทโฟนในการบันทึกเสียงเคาะ รูปแบบการบันทึกเสียงเป็นแบบโมโนโทน (Mono) และอัตราการสุ่มตัวอย่าง (Sampling rate) 16 กิโลเฮิร์ตซ์ (kHz) นามสกุลของไฟล์เสียงเป็นรูปแบบ Waveform Audio File Format หรือ WAV มีการสุ่มตำแหน่งในการเคาะ เพื่อให้มีข้อมูลมีความหลากหลายในการฝึกฝน การเคาะทุเรียนได้เคาะประมาณ 2-5 ครั้ง เมื่อเก็บข้อมูลเสียงเรียบร้อยขั้นตอนต่อไปคือการตัดเสียงเคาะเป็นไฟล์ที่มีเสียงเคาะ 1 ครั้งต่อ 1 ไฟล์ ด้วยโปรแกรม Audacity [44, 45] ไฟล์เสียงมีความยาว 0.3 วินาที ซึ่งครอบคลุมเสียงต่อการเคาะทุเรียนหนึ่งครั้ง เมื่อได้ไฟล์เสียงเคาะมาแล้วชุดข้อมูลสำหรับการเรียนรู้จะแบ่งเป็น 3 ชุด ตามระดับความสุกของทุเรียนที่ได้กล่าวมาข้างต้น จากนั้นเข้าสู่กระบวนการเรียนรู้ข้อมูลในขั้นตอนต่อไป

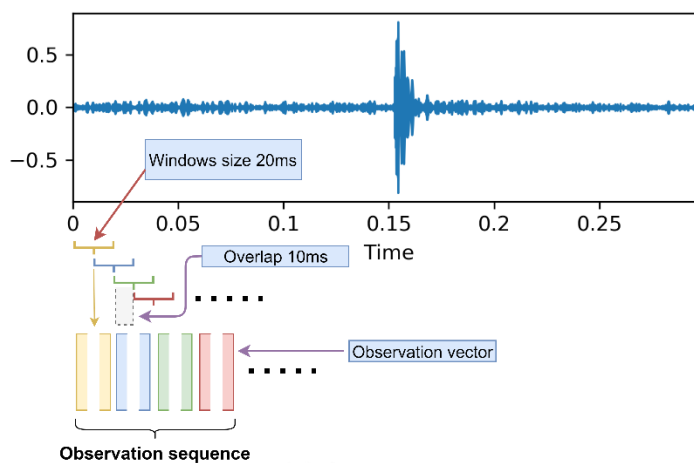
3.1.2 กระบวนการสกัดคุณลักษณะของเสียงเคาะ (Feature Extraction)

ในขั้นตอนนี้ได้กล่าวถึงการเรียนรู้เสียงเคาะทุเรียนด้วยกระบวนการเรียนรู้แบบคอนโวลูชัน ขั้นตอนแรกคือการสกัดคุณลักษณะ (Feature extraction) ของเสียงด้วยกระบวนการ MFCCs โดยได้กล่าวไว้ในหัวข้อที่ 2.1.3 การสกัดคุณลักษณะของเสียงด้วยสัมประสิทธิ์เมลฟรีควีนซีเซปสตรอล (Mel-Frequency Cepstral Coefficients (MFCCs) Feature Extraction) กระบวนการการสกัดคุณลักษณะแสดงในรูปที่ 29



รูปที่ 29 กระบวนการสกัดคุณสมบัตินเด่นของเสียงด้วยกระบวนการ MFCC

1) Frame blocking: กำหนดหน้าต่างในการเลื่อนเฟรมในขั้นตอนนี้เท่ากับ 20ms และเลื่อนไปบนสัญญาณเสียงทีละ 10ms ดังแสดงรูปที่ 30 โดยการเลื่อนแต่ละครั้งจะเก็บข้อมูลของสัญญาณที่ได้ทำการบล็อกเฟรมไปด้วย เป็นการเก็บข้อมูลในรูปแบบของเวกเตอร์เมื่อทำจนครบทั้งสัญญาณเสียงจะได้กลุ่มของข้อมูลเสียงที่จะนำไปใช้ในการบวนการขั้นตอนต่อไป



รูปที่ 30 การทำ frame blocking ของสัญญาณเสียงเคาะทุเรียนที่มีความยาวคลิป 300ms

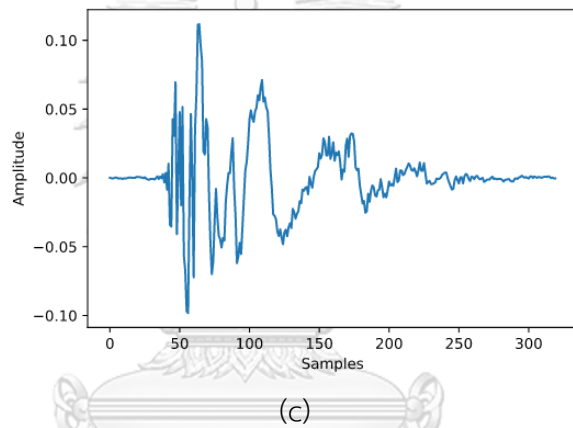
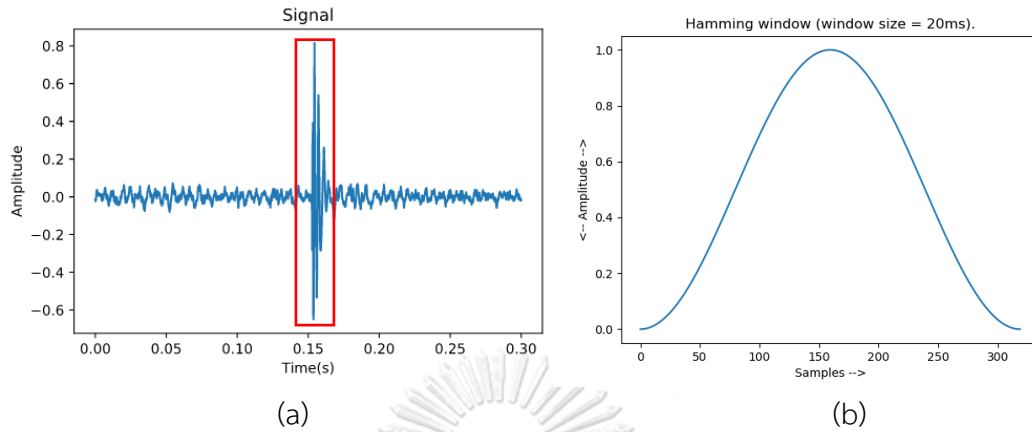
2) Hamming window เป็นส่วนที่ทำให้สัญญาณที่ต้องการมีความชัดเจนมากขึ้นและยังคงความต่อเนื่องของสัญญาณไว้ก่อนที่จะเข้ากระบวนการการแปลงฟูริเยร์แบบรวดเร็ว ในรูปที่ 31 ส่วนที่อยู่ในกรอบสีแดงคือส่วนแสดงให้เห็นว่าเมื่อสัญญาณผ่านกระบวนการแฮมมิงแล้วหน้าต่างของสัญญาณจะเป็นดังรูปที่ 31 (c) ที่ใช้อัตราการสุ่มตัวอย่าง 320 ตัวอย่างต่อหนึ่งเฟรม

3) การแปลงฟูริเยร์แบบรวดเร็วเป็นการแปลงสัญญาณเสียงในโดเมนเวลาไปเป็นโดเมนความถี่ แสดงดังรูปที่ 32 (a) เป็นการแปลงสัญญาณจากโดเมนเวลาเป็นโดเมนความถี่ (b) เป็นการยกกำลังสองของ FFT ที่เรียกว่า Power spectrum density ซึ่งเป็นการบ่งบอกถึงความหนาแน่นของความถี่นั้น ๆ ยิ่งมีค่าสูงแสดงว่าสัญญาณที่สนใจมีความถี่นั้น ๆ อยู่มากจึงให้ระบุความถี่ได้ชัดเจนขึ้น

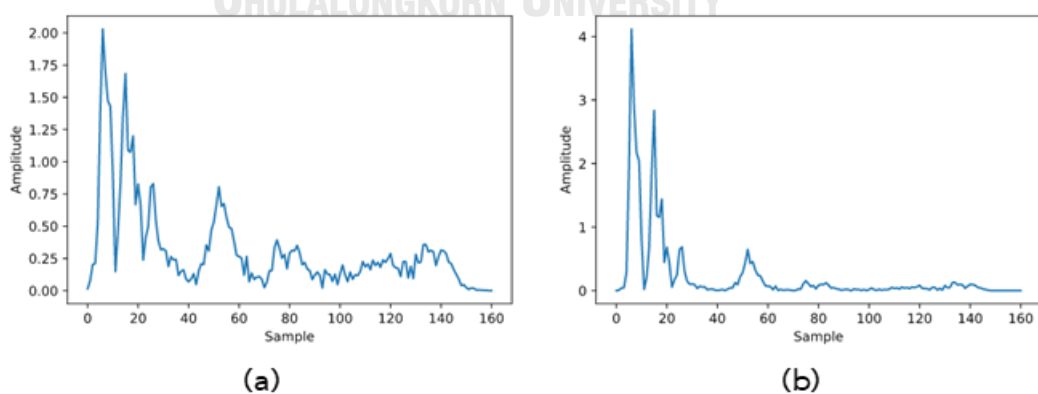
4) กระบวนการคัดกรองความถี่ด้วยเมลฟิลเตอร์แบงก์ เป็นกระบวนการกรองช่วงความถี่ที่สำคัญ โดยได้กำหนดจำนวนตัวกรองไว้ที่ 9 ตัวกรอง สัดส่วนของความถี่อ้างอิงตามสมการที่ (4) ผลลัพธ์จากการคำนวณออกมาเป็นหน้าต่างของสัญญาณตามรูปที่ 33 โดยแกนแนวตั้งจะแสดงกลุ่มของความถี่ที่สนใจที่ได้มาจากสเกลของเมล และแกนแนวนอนคือค่าของการสุ่มตัวอย่างของสัญญาณทั้งหมด

5) การแปลงโคไซน์แบบไม่ต่อเนื่องเพื่อเก็บค่าสัมประสิทธิ์ความถี่ที่สำคัญหรืออีกนัยคือเป็นกระบวนการบีบอัดข้อมูลแบบมีการสูญเสีย โดยการแปลงสัญญาณความถี่ให้เป็นองค์ประกอบเล็ก ๆ จากกระบวนการนี้จะได้ย่านชุดความถี่ที่สำคัญทั้งหมด 9 ค่า เหตุผลที่เลือกค่านี้นี้เนื่องจากจำนวนตัวแปรสำหรับการเรียนรู้ที่ไม่ใหญ่และสามารถเรียนรู้ได้อย่างรวดเร็ว นอกจากนี้ค่าสัมประสิทธิ์ 9 ค่า

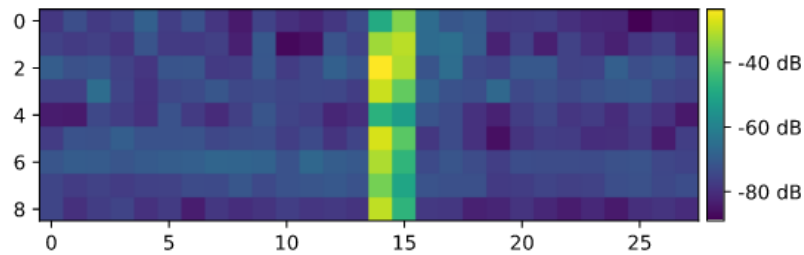
สามารถครอบคลุมความถี่ของเสียงเคาะทุเรียน สัญญาณที่ผ่านกระบวนการแปลงโคไซน์แบบไม่ต่อเนื่องจะแสดงดังรูปที่ 34



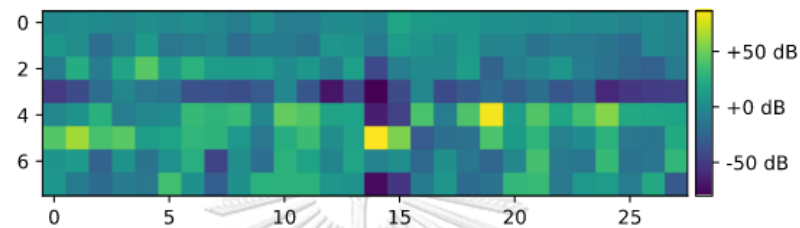
รูปที่ 31 การทำหน้าต่างแฮมมิง (a) สัญญาณเสียงเคาะต้นฉบับ (b) หน้าต่างแฮมมิงเมื่อ (c) สัญญาณเสียงหลังจากเข้ากระบวนการแฮมมิง 1 เฟรมที่มาจากกรอบสีแดงจากสัญญาณเสียงต้นฉบับ



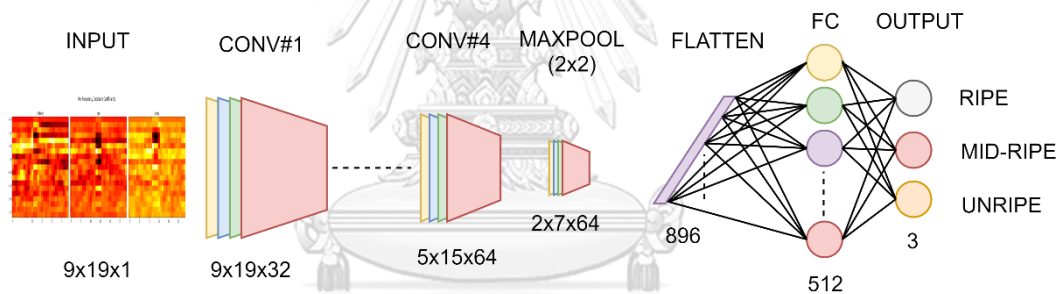
รูปที่ 32 สัญญาณที่แปลงจากโดเมนเวลาเป็นโดเมนความถี่, (a) Fast-Fourier Transform, (b) Power Spectrum Density ของกลุ่มตัวอย่างทุเรียนสุก (ripe)



รูปที่ 33 สัญญาณหลังผ่านกระบวนการเมลฟิลเตอร์แบงก์



รูปที่ 34 สัญญาณหลังจากผ่านกระบวนการแปลงโคไซน์แบบไม่ต่อเนื่อง



รูปที่ 35 โครงสร้างการเรียนรู้เสียงเคาะด้วยการเรียนรู้เชิงลึกแบบคอนโวลูชัน

3.1.3 กระบวนการเรียนรู้ของเสียงเคาะทุเรียน

การออกแบบจำลองด้วยกระบวนการเรียนรู้แบบคอนโวลูชัน เป็นกระบวนการต่อเนื่องจากหัวข้อที่ 3.1.2 กระบวนการสกัดคุณลักษณะของเสียงเคาะ (Feature Extraction) โครงสร้างการเรียนรู้แสดงในรูปที่ 35 ออกแบบชั้นการเรียนรู้เป็นคอนโวลูชัน 4 ชั้นต่อกับชั้น Fully connected และเอาต์พุตจำนวน 3 ตัวแปร ตามกลุ่มทุเรียนที่แบ่งไว้ตามตารางที่ 3 ซึ่งเป็นการอธิบายอย่างละเอียดถึงโครงสร้างภายในและจำนวนของพารามิเตอร์ในแต่ละชั้น ส่วนนี้ถูกสร้างโดย Keras Library อินพุตสำหรับกระบวนการเรียนรู้แบ่งออกเป็นกลุ่มคือ เสียงเคาะทุเรียนสุก เสียงเคาะทุเรียนสุกกลาง และเสียงเคาะทุเรียนไม่สุก และแบ่งข้อมูลสำหรับการเรียนรู้เป็น 3 ส่วนคือ ชุดข้อมูล 70 เปอร์เซ็นต์ สำหรับการเรียนรู้ (Training), ชุดข้อมูล 20 เปอร์เซ็นต์ สำหรับตรวจสอบความถูกต้อง

ของข้อมูลระหว่างเรียนรู้ข้อมูล (Validation) และ 10 เปอร์เซ็นต์ สำหรับการทดสอบ (Test) แบบจำลองที่ผ่านการเรียนรู้มาแล้วจะได้พารามิเตอร์สำหรับการเรียนรู้ทั้งหมด 525,795 พารามิเตอร์

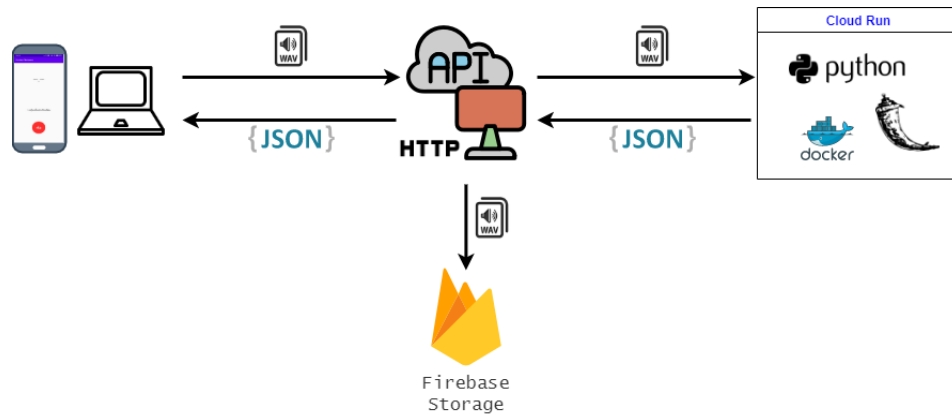
ตารางที่ 3 โครงสร้างภายในของการเรียนรู้เชิงลึกแบบคอนโวลูชัน

ชนิดของชั้น (layer type)	ขนาดของเอาต์พุต (output shape)	จำนวนพารามิเตอร์ (param)
Conv2d #1	(None, 11, 11, 32)	320
Activation #1 (ReLU)	(None, 9, 9, 32)	0
Conv2d #2	(None, 7, 7, 32)	9248
Activation #2 (ReLU)	(None, 7, 7, 32)	0
Dropout #1	(None, 7, 7, 32)	0
Conv2d #3	(None, 7, 7, 64)	18496
Activation #3 (ReLU)	(None, 7, 7, 64)	0
Conv2d #4	(None, 5, 5, 64)	36928
Activation #4 (ReLU)	(None, 5, 5, 64)	0
Max pooling2d #1	(None, 2, 2, 64)	0
Dropout #2	(None, 2, 2, 64)	0
Flatten #1	(None, 896)	0
Dense #1	(None, 512)	459264
Activation #5 (ReLU)	(None, 512)	0
Dropout #3	(None, 512)	0
Dense #2	(None, 3)	1539
Activation #6 (Softmax)	(None, 3)	0

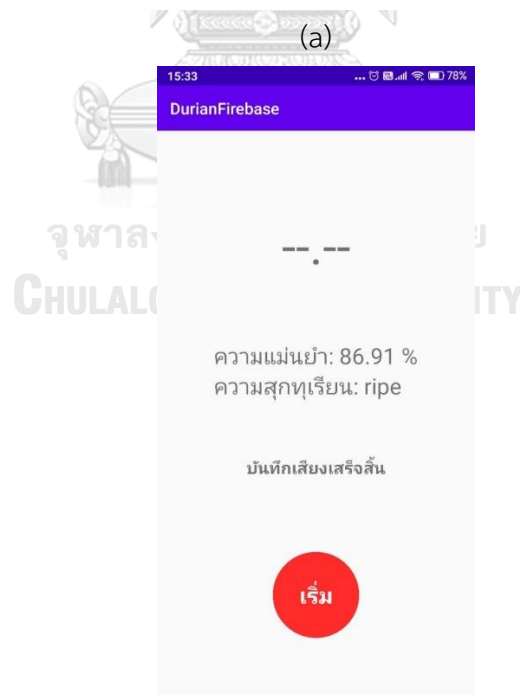
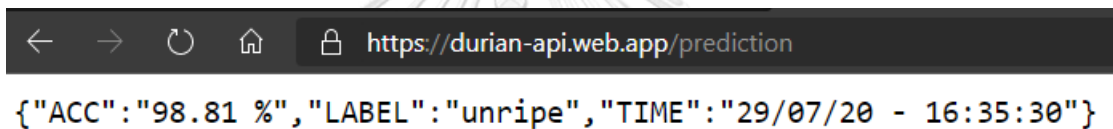
3.2 กระบวนการออกแบบแอปพลิเคชันและระบบการจำแนกเสียงทุเรียน

แอปพลิเคชันในการพัฒนาได้ออกแบบการส่งข้อมูลผ่านระบบ HTTP Protocol อยู่ใน รูปแบบเอพีไอ ผู้ใช้บริการเพียงอัดเสียงเคาะทุเรียนด้วยสมาร์โฟน จากนั้นตัวแอปพลิเคชันจะ ทำหน้าที่ส่งเสียงที่เป็นไฟล์นามสกุล .wav ส่งคำขอไปยัง RESTful APIs จากนั้นตัวเอพีไอทำหน้าที่ส่ง ไฟล์เสียงไปยังตัวบริการ Cloud Run ที่สร้างไว้ทำการประมวลผลและให้ผลลัพธ์ในรูปแบบของ JSON ข้อมูลภายในประกอบด้วยเปอร์เซ็นต์ความถูกต้อง (Accuracy: ACC) กลุ่มความสุกทุเรียน (Label) เวลาและวันที่ (Time) ที่ทำการจำแนก และส่งข้อมูลผ่าน APIs เพื่อส่งต่อข้อมูลไปยัง

ผู้ให้บริการ ระบบสามารถใช้ได้ทั้งบนแอปพลิเคชันมือถือในระบบปฏิบัติการ Android ภาพรวมการทำงานแสดงดังรูปที่ 36 และเว็บไซต์ <https://durian-api.web.app/> ที่แสดงในรูปที่ 37



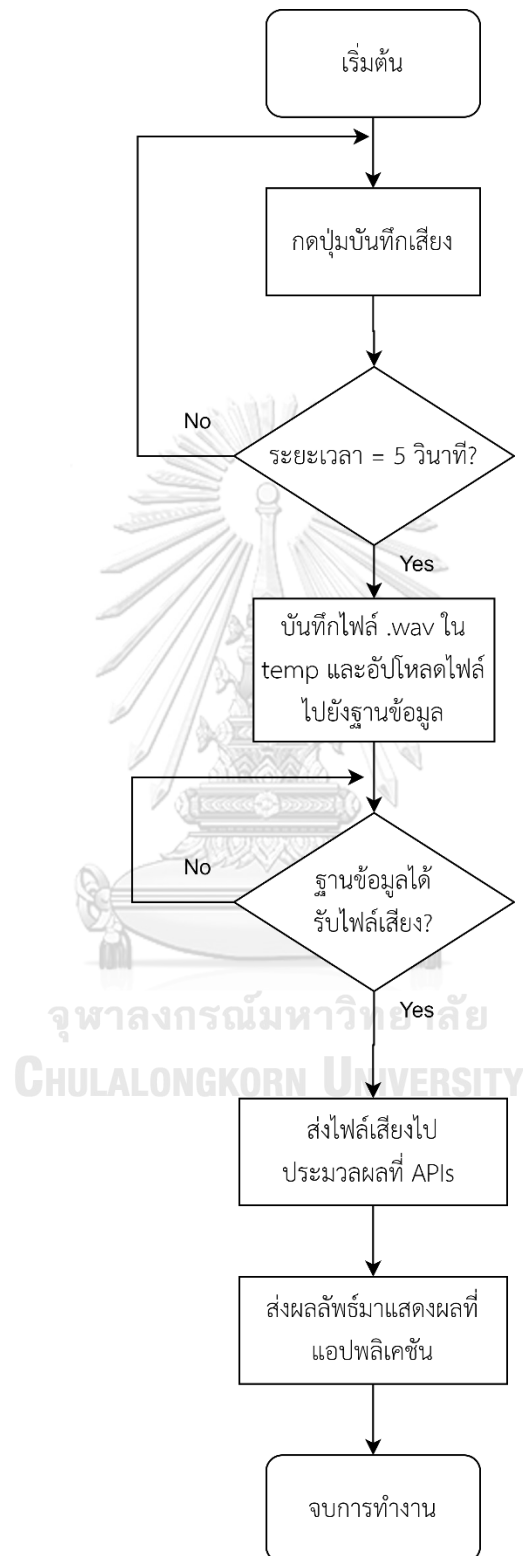
รูปที่ 36 ภาพรวมแสดงการทำงานของระบบการจำแนกเสียงทุเรียน



(b)

รูปที่ 37 รูปแบบการจำแนกความสุกทุเรียน (a) ผ่านเว็บ APIs และ (b) ผ่านแอปพลิเคชันมือถือ

3.2.1 การออกแบบการทำงานแอปพลิเคชัน



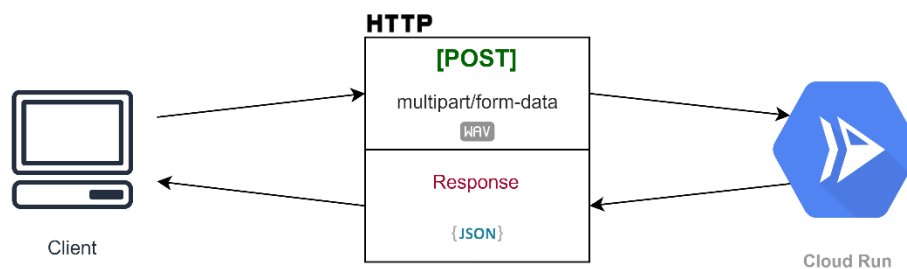
รูปที่ 38 Flow Chart แสดงการทำงานของแอปพลิเคชัน

การทำงานของแอปพลิเคชัน ขั้นตอนแรกผู้ใช้กดปุ่มบันทึกเสียงและเคาะทุเรียน ระยะเวลาในการบันทึก 5 วินาที การเคาะทุเรียนควรเคาะอย่างน้อย 2 ครั้งต่อหนึ่งไฟล์ขึ้นไป และแอปพลิเคชันรอรับการบันทึกเสียงจากผู้ใช้งานจนกว่าจะครบ 5 วินาที จากนั้นทำการบันทึกเสียงที่ได้ให้อยู่ในรูปแบบ waveform ที่มีนามสกุลเป็น .wav โดยที่ไฟล์นี้ถูกบันทึกอยู่ในโฟลเดอร์ชั่วคราว Temporary Folder เมื่อบันทึกเสร็จไฟล์เสียงจะถูกส่งไปจัดเก็บในฐานข้อมูลของบริการ Google Firebase Storage ในการจัดเก็บ เมื่อบันทึกในฐานข้อมูลไฟล์เสียงจะถูกส่งไปยัง APIs สำหรับการประมวลผลสำหรับการจำแนกกลุ่มความสุกทุเรียน ที่แบ่งเป็น 3 กลุ่ม ได้แก่ ripe: ทุเรียน (เนื้อทุเรียนมีความนุ่มไปจนถึงนุ่มมาก), mid-ripe: ทุเรียนสุกปานกลาง (เนื้อทุเรียนมีความกรอบ) และ unripe: ทุเรียนที่ยังไม่สุก (ยังไม่สามารถรับประทานได้ต้องรอประมาณ 2 – 5 วัน ถึงจะสามารถนำมารับประทานได้) เมื่อระบบทำการจำแนกเสร็จเรียบร้อยแล้วจะทำการส่งผลการจำแนกมาอยู่ในรูปแบบ JSON เพื่อให้สามารถแสดงผลผ่านตัวแอปพลิเคชันและเว็บไซต์ได้ เป็นอันจบการทำงานใน 1 ครั้ง Flow Chart การทำงานแสดงดังรูปที่ 38

3.2.2 การออกแบบระบบ RESTful APIs

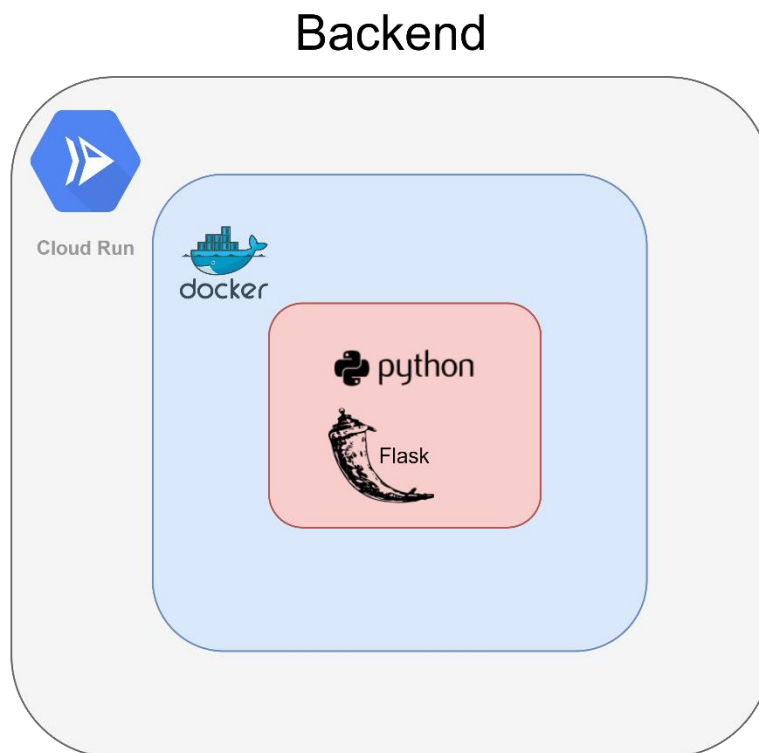
การออกแบบ RESTful APIs แสดงดังรูปที่ 39 ได้ออกแบบระบบให้สามารถใช้ได้แค่ 1 ชุดคำสั่งเท่านั้น นั่นคือ POST ซึ่งได้อธิบายไว้ในหัวข้อที่ 2.1.4.4 REST และ RESTful API เป็นการส่งไฟล์เสียงผ่านโปรโตคอล HTTP ตาม URL ต่อไปนี้ <https://durian-api.web.app/> โดยจะเข้าถึงระบบเบื้องหลัง (Backend) ซึ่งได้ใช้บริการของ Google Cloud Run แสดงดังรูปที่ 40 ภายใน Cloud Run มีการสร้าง Docker container เพื่อสร้างสภาพแวดล้อม (Environment) สำหรับการรันชุดคำสั่งภาษา ไพธอน โดยมี Flask web framework สำหรับติดต่อสื่อสารกับ RESTful APIs เมื่อ Cloud Run ประมวลผลเสร็จเรียบร้อยแล้วจะโต้ตอบ (Response) กลับมาผ่าน URL <https://durian-api.web.app/prediction> ในรูปแบบของ JSON ดังแสดงในรูปที่ 37 (a)

RESTful APIs



รูปที่ 39 การสื่อสารผ่านระบบ RESTful APIs

3.2.3 โครงสร้างการทำงานของระบบเบื้องหลัง Backend



รูปที่ 40 โครงสร้างระบบเบื้องหลัง Backend

ระบบเบื้องหลังออกแบบสำหรับการรันไพธอนไว้บน Cloud Run ภายใต้ Cloud Run จะทำการรัน Docker container เอาไว้ จากนั้นเขียนสคริปต์ด้วยภาษาไพธอนเพื่อเรียกใช้งานไลบรารี Flask web framework เพื่อรอคำขอจากผู้ใช้งานจากฝั่ง RESTful APIs ภาพรวมของโครงสร้าง Backend แสดงดังรูปที่ 40 ขั้นตอนการสร้างโครงสร้าง Backend เป็นไปตามขั้นตอนดังนี้:

ขั้นตอนแรกสร้างโฟลเดอร์สำหรับใช้ในเซิร์ฟเวอร์ที่กำหนดชื่อเป็น src ภายในโฟลเดอร์ประกอบด้วยโฟลเดอร์ย่อยตามรูปที่ 41 จากนั้นสร้างไฟล์ index.html สำหรับแสดงบนหน้าเว็บในโฟลเดอร์ root/src/templates/index.html เมื่อทำการสร้างไฟล์ index.html เรียบร้อย จากนั้นสร้างไฟล์ที่มีชื่อว่า app.py ให้อยู่ในโฟลเดอร์ root/src/app.py ซึ่งเป็นการสร้างตัวไฟล์สคริปต์เพื่อเรียกใช้ไฟล์ pred.py โดยที่ไฟล์จะอยู่ภายใต้ root/src/backend/ ซึ่งเป็นสคริปต์ไฟล์สำหรับจำแนกความสูงของทุเรียน จากนั้นสร้าง Dockerfile ภายใต้โฟลเดอร์ root/Dockerfile ซึ่งเป็นไฟล์ชุดคำสั่งสำหรับการสร้างสภาพแวดล้อม (Environment) เพื่อให้ Python script สามารถรันได้ภายในไฟล์มีโครงสร้างดังรูปที่ 42

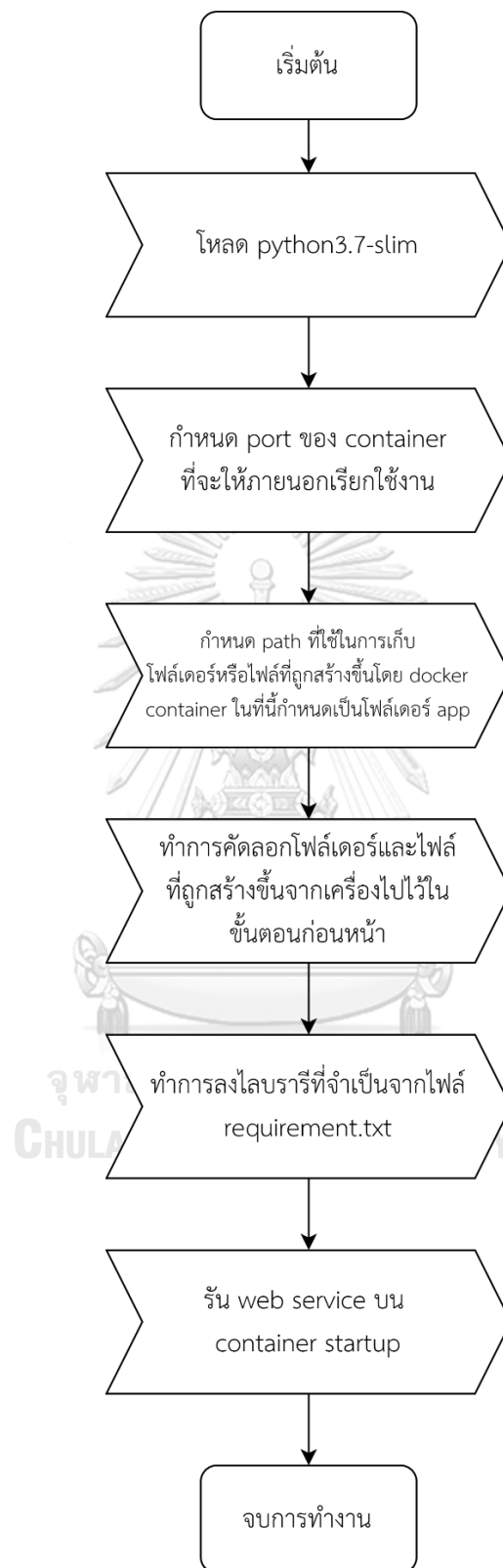
ต่อมาอธิบายการทำงานของไฟล์ `pred.py` ซึ่งเป็นส่วนของ Flask web framework ที่คอยรับและส่งข้อมูลผ่าน RESTful APIs ซึ่งเป็นส่วนทำงานอยู่เบื้องหลังที่คอยทำนายเสียงเคาะทุเรียนที่รับมาแล้วก็ส่งผลลัพธ์ออกไปยังผู้ใช้บริการโดยได้แสดงการทำงานดังรูปที่ 43

```

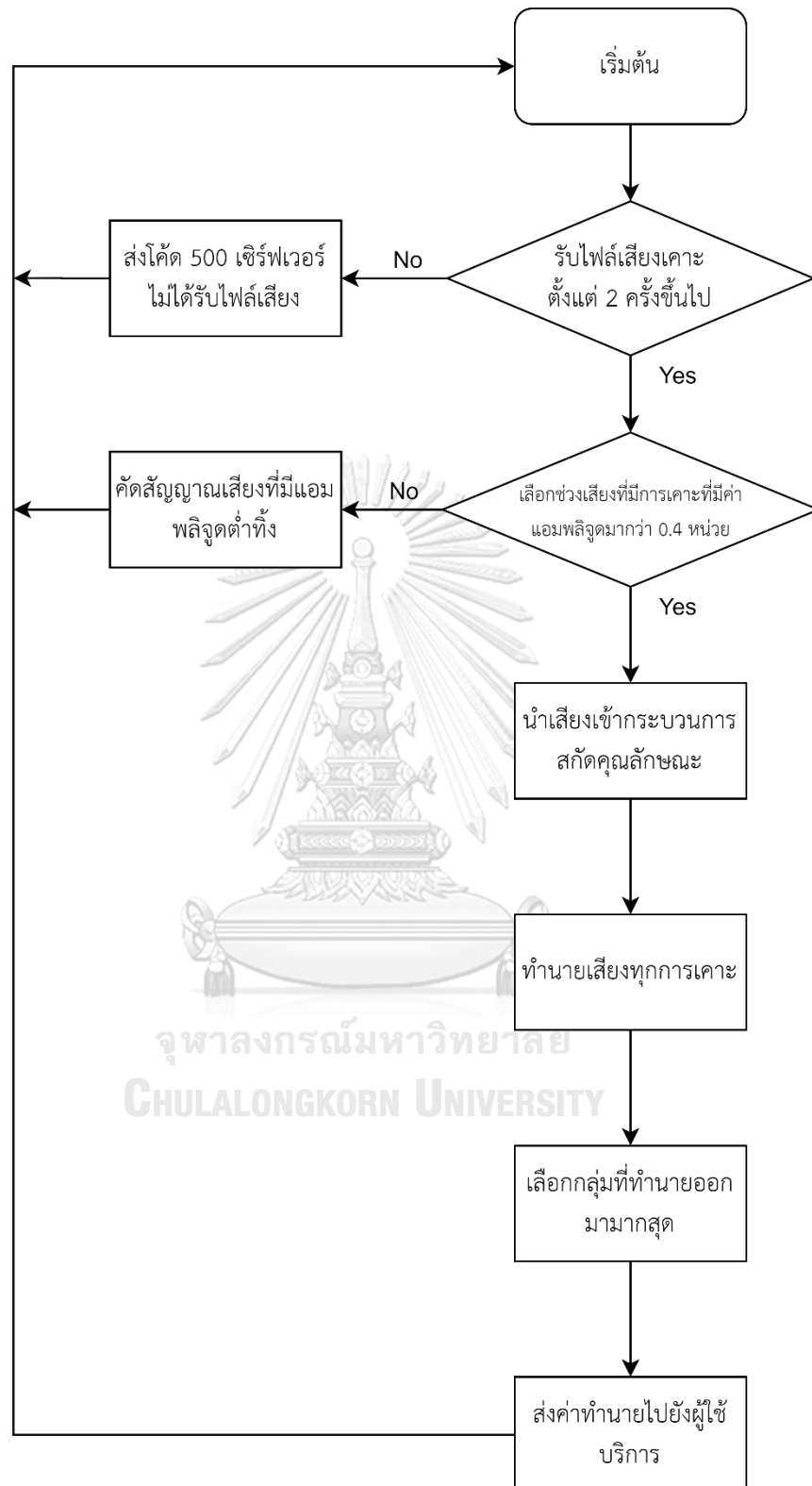
+---Build Server (root dir)
|   Dockerfile
|   requirements.txt
|   firebase.json
|   .firebaserc
|
+---src
|   app.py
|
+---backend
|   |   pred.py
|
+---db
|   |   c_firebase.py
|   |   key.json
|
+---model
|   mfcc.tflite
|
+---static
|   +---css
|   |   durian.jpg
|   |   style.css
|   +---img
|   |   background.jpg
|   |   favicon.svg
|   \---js
|   |   script.js
|
\---templates
    index.html

```

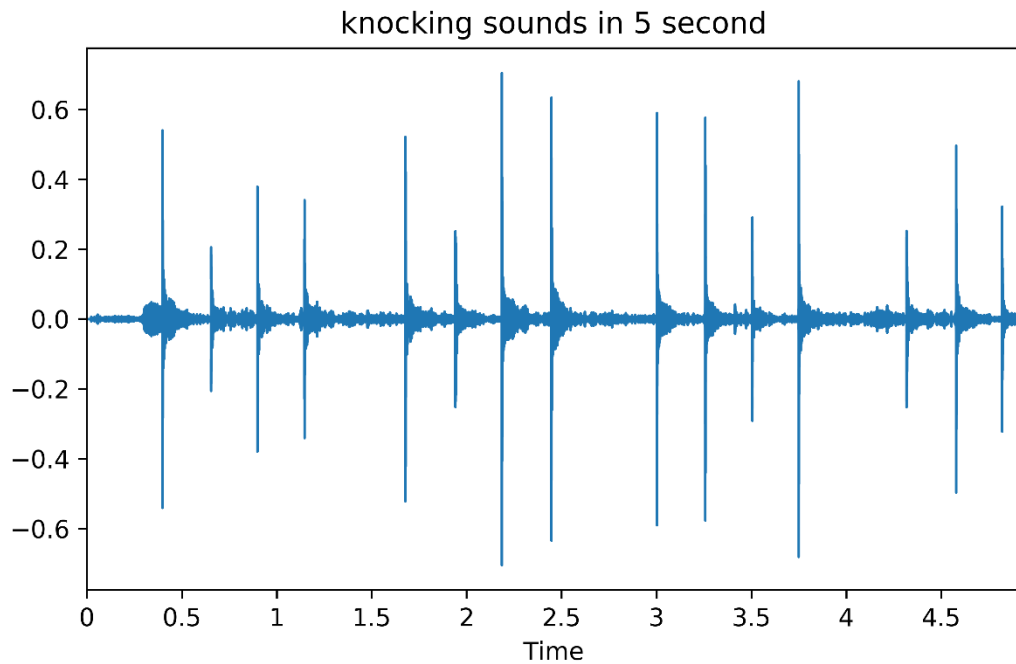
รูปที่ 41 โครงสร้างไฟล์เดอร์สำหรับการสร้างระบบเบื้องหลัง



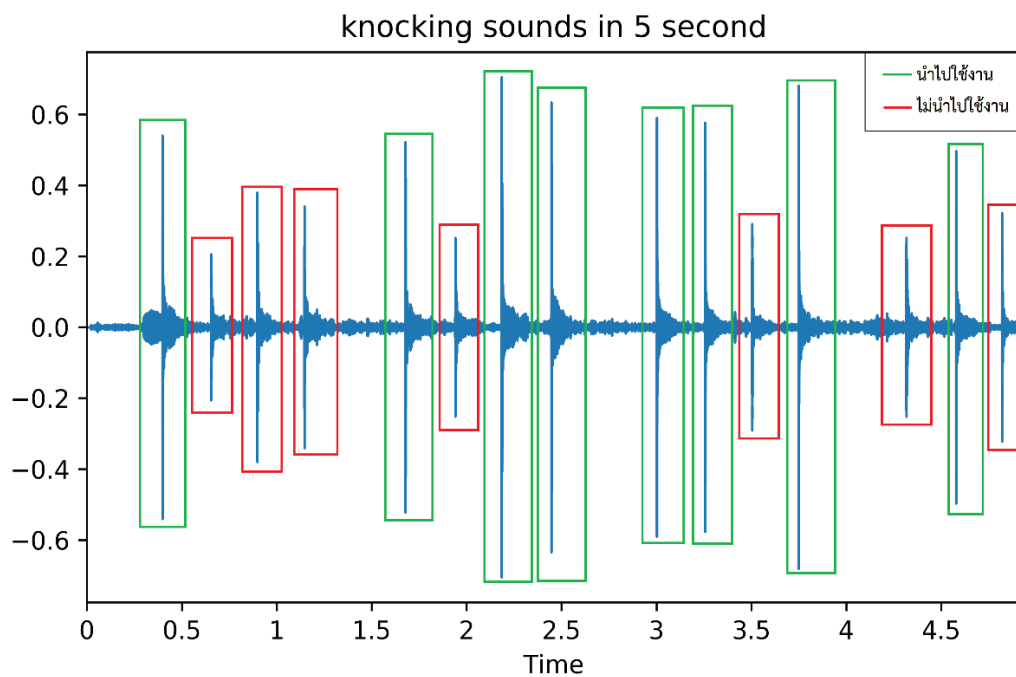
รูปที่ 42 Flow chart สำหรับการสร้าง Docker container



รูปที่ 43 การทำงานของไฟล์ pred.py ที่อยู่ในส่วนเบื้องหลัง



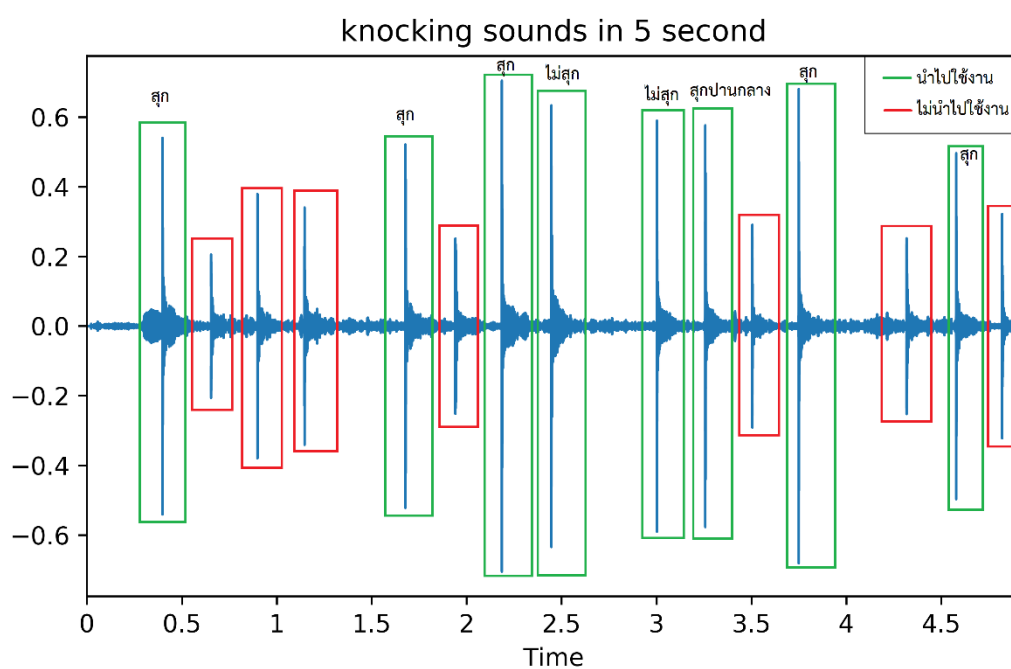
รูปที่ 44 สัญญาณเสียงเคาะที่ได้รับจากผู้ให้บริการระยะเวลา 5 วินาที



รูปที่ 45 การเลือกเสียงเคาะสำหรับไปทำนาย

การเลือกสัญญาณเสียงที่ได้รับจากผู้ให้บริการจะแสดงดังรูปที่ 44 จากนั้นได้เลือกเสียงเคาะที่มีค่าแอมพลิจูดมากกว่า 0.4 หน่วย แสดงในกรอบสีเขียวและเสียงเคาะที่ค่าแอมพลิจูดไม่ถึง 0.4 หน่วย

แสดงในกรอบสีแดงซึ่งจะไม่เอาไปเข้ากระบวนการการทำนาย แสดงดังรูปที่ 45 จากนั้นจะเป็นการแยกกลุ่มในแต่ละเสียงเคาะหากกลุ่มไหนมีการทำนายได้มากที่สุด ก็จะเลือกกลุ่มนั้นส่งไปให้ผู้ให้บริการ แสดงตัวอย่างดังรูปที่ 46 การทำนายที่ออกมาจะเป็นกลุ่มทุเรียนสุก ซึ่งจากรูปจะเห็นว่ากลุ่มทุเรียนสุกมีการทำนายออกมา 5 ครั้ง กลุ่มทุเรียนไม่สุกทำนายออกมา 2 ครั้ง และกลุ่มทุเรียนสุกปานกลางทำนายออกมา 1 ครั้ง ซึ่งกลุ่มทุเรียนสุกทำนายออกมาได้มากกว่าจึงเลือกกลุ่มทุเรียนสุก



รูปที่ 46 การระบุกลุ่มความสุขหลังจากเลือกเสียงเคาะ

ขั้นตอนการคัดแยกกลุ่มทุเรียนในแต่ละเสียงเคาะ

1: นับจำนวนเสียงเคาะ:

เก็บค่าจำนวนเสียงเคาะไว้

2: ทำนายเสียงเคาะ:

จำนวนกลุ่มทุเรียนสุก และค่าเปอร์เซ็นต์ที่ความถูกต้องได้จากการทำนาย

จำนวนกลุ่มทุเรียนสุกปานกลาง และค่าเปอร์เซ็นต์ที่ความถูกต้องได้จากการทำนาย

จำนวนกลุ่มทุเรียนไม่สุก และค่าเปอร์เซ็นต์ที่ความถูกต้องได้จากการทำนาย

3: ส่งค่ากลุ่มทำนาย:

กรณีที่ค่าทำนายแยกได้ชัดเจน:

เปรียบเทียบจำนวนการทำนายแล้วส่งค่าไปยังผู้ให้บริการ

กรณีที่ค่าจำนวนการทำนายบางกลุ่มเท่ากัน:

เปรียบเทียบค่าเปอร์เซ็นต์ความถูกต้อง:

กลุ่มไหนมากที่สุดให้ส่งค่ากลุ่มนั้นไปยังผู้ใช้บริการ

4: รอรับไฟล์เสียงใหม่จากผู้ใช้บริการ



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 4

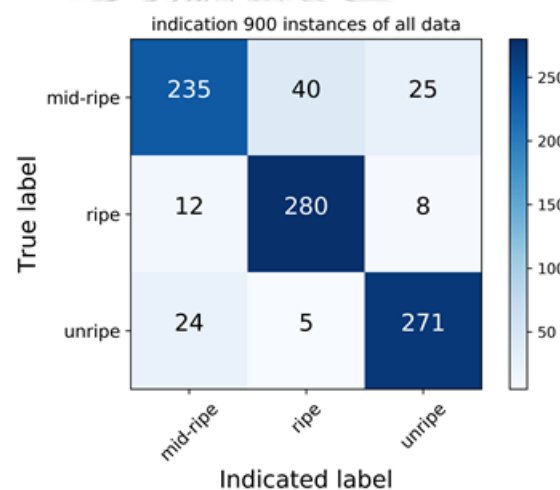
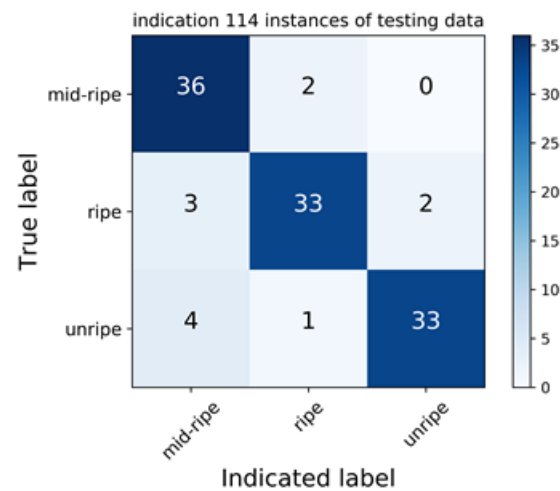
การทดลองและประเมินผล

4.1 การทดลองหาแบบจำลองที่ดีที่สุด

การพัฒนาและทดสอบแบบจำลองการเรียนรู้เสียงเคาะทุเรียนพัฒนาบนภาษา Python 3.7 และคอมพิวเตอร์ระบบปฏิบัติการ Windows 10 Pro ความเร็วหน่วยประมวลผลกลาง CPU 2.5 GHz Intel® Core™ i7-6500U หน่วยความจำ (RAM) 8 GB หน่วยประมวลผลกราฟฟิก NVIDIA GTX 950M มีหน่วยประมวลผล 640 CUDA หน่วยความจำ (GRAM) 4 GB ซึ่งได้ประสิทธิภาพการทำนาย 90.78 เปอร์เซ็นต์ ที่ทดสอบกับข้อมูล Validation และ 89.47 เปอร์เซ็นต์กับข้อมูล Test การทดสอบได้แบ่งรอบการเรียนรู้ (epoch) ทั้งหมด 6 ครั้ง ครั้งแรกกำหนดรอบการเรียนรู้เท่ากับ 50 epoch จากนั้นเพิ่มจำนวนรอบการเรียนรู้ครั้งละ 50 epoch จนถึงครั้งที่ 6 ได้จำนวนรอบการเรียนรู้เท่ากับ 300 epoch และผลการทดลองแสดงในตารางที่ 4 จะเห็นครั้งที่ 3 ได้ความแม่นยำสูงสุดจึงเลือกมาใช้ทดสอบประสิทธิภาพในการทำนาย ประสิทธิภาพของแบบจำลองแสดงในรูปที่ 4.7 ที่ใช้ตาราง Confusion Matrix วัดประสิทธิภาพ ภาพ (a) เป็นการทดสอบด้วยชุดข้อมูลทดสอบที่มีทั้งหมด 114 ตัวอย่างเสียง ภาพ (b) เป็นการทดสอบประสิทธิภาพด้วยตัวอย่างทั้งหมด 900 ตัวอย่างเสียง โดยตัวเลขในเส้นทแยงมุมจากซ้ายไปขวาคือจำนวนตัวอย่างที่ทำนายถูกต้อง

ตารางที่ 4 การทดสอบเพิ่มรอบการเรียนรู้ของแบบจำลอง

No.	Epoch	Acc-val (%)	Acc-test (%)
1	50	88.03	82.50
2	100	89.94	85.60
3	150	90.78	89.47
4	200	90.47	83.34
5	250	90.78	88.60
6	300	89.10	85.97



CHULALONGKORN UNIVERSITY (b)

รูปที่ 47 ประสิทธิภาพในการทำนายของแบบจอง (a) ทดสอบกับชุดข้อมูลทดสอบ (Test) ทั้งหมด 114 ตัวอย่างเสียง (b) ทดสอบกับชุดข้อมูลทั้งหมด 900 ตัวอย่างเสียง

4.2 การทดสอบประสิทธิภาพของแอปพลิเคชันจำแนกความสุกทุเรียน

ในการทดลองและทดสอบประสิทธิภาพการคัดแยกความสุกของแอปพลิเคชันเราได้นำทุเรียนกลุ่มละ 1 ลูกมาสำหรับการทดสอบ ในการทดสอบเราได้สุ่มตำแหน่งการเคาะทุเรียนจำนวน 50 ครั้ง ในแต่ละครั้งจะเคาะตั้งแต่ 2 ครั้งขึ้นไปจนครบ 5 วินาที เมื่อครบ 5 เราจะได้กลุ่มทุเรียนได้ทำการคัดแยกมา จากนั้นจะทำการบันทึกว่ากลุ่มทุเรียนที่แอปพลิเคชันได้ทำนายออกมาเป็นกลุ่มอะไร ทำไปจนครบ 50 ครั้ง จากนั้นทำการเปลี่ยนทุเรียนลูกใหม่จนครบทุเรียน 3 ลูก การทดลองโดยการสุ่มตำแหน่งของการเคาะ จากนั้นจะทำการปลอกทุเรียนเพื่อให้แน่ใจว่าทุเรียนที่นำมาตรงกลุ่มความ

สูงที่ต้องการ โดยที่การทดลองทั้งหมดจะแสดงดังตารางที่ 5 นอกจากนี้เรายังได้ทำการทดสอบตัวแอปพลิเคชันในสถานการณ์จริงโดยไปที่หน้าร้านขายทุเรียนที่อยู่ตามข้างถนน ในการทดสอบนั้นจะมีสัญญาณเสียงรบกวนจากท้องถนน เช่น เสียงรถที่กำลังวิ่ง เสียงของคนที่อยู่ในบริเวณนั้น เป็นต้น การทดสอบครั้งนี้ได้รับความอนุเคราะห์จากคนขายทุเรียนเพื่อทดสอบการเคาะทุเรียนและระบุกลุ่มทุเรียนที่กำลังทดสอบว่ามีความสุขอยู่ในกลุ่มไหน ซึ่งเป็นพันธุ์หมอนทองทั้งหมด ทุเรียนที่ได้ทำการทดสอบนั้นจะสุ่มจากทางร้านที่วางขายอยู่ซึ่งมีครบทุกกลุ่มความสุขที่ต้องการ จากนั้นได้ทำการทดลองกลุ่มละ 10 ครั้ง จำนวนทุเรียนทั้งหมด 6 ลูก โดยการทดลองจะแสดงดัง

ตารางที่ 7 การคำนวณประสิทธิภาพในการคัดแยกนั้นจะใช้ค่า Precision, Recall และ F1-score ที่ได้กล่าวไว้ในหัวข้อ 2.1.5 การทดสอบประสิทธิภาพของแบบจำลอง โดยได้แสดงไว้ในตารางที่ 6 และตารางที่ 8 และแสดงประสิทธิภาพในการคัดแยกกลุ่มความสุขทุเรียนในสภาพแวดล้อมที่มีเสียงรบกวนที่แตกต่างกันดังรูปที่ 48 และรูปที่ 49

ตารางที่ 5 การทดลองการแยกกลุ่มทุเรียนในแต่ละกลุ่มละ 1 ลูก จำนวน 50 ครั้ง

		ค่าทำนาย (Prediction)		
		ไม่สุข	สุขปานกลาง	สุข
ค่าจริง (Actual)	ไม่สุข	48	0	2
	สุขปานกลาง	7	43	0
	สุข	3	2	45

ตารางที่ 6 การวัดผลการคัดแยกความสุขของแอปพลิเคชันผลลัพธ์จากตารางที่ 5

กลุ่มทุเรียน	Precision	Recall	F1-score	จำนวนการทดลอง (ครั้ง)
สุข	0.83	0.96	0.89	50
สุขปานกลาง	0.96	0.86	0.91	50
ไม่สุข	0.96	0.90	0.93	50

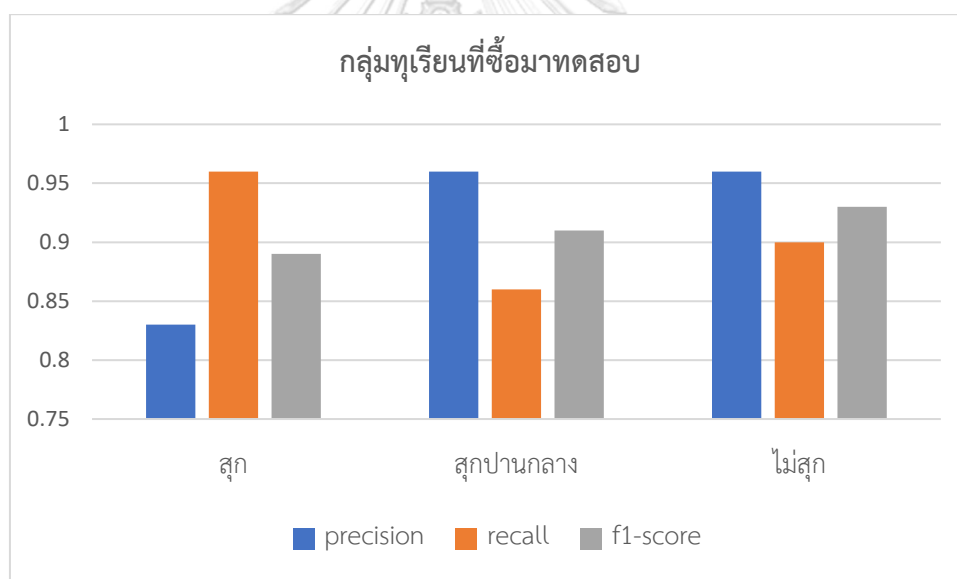
ตารางที่ 7 การทดลองการแยกกลุ่มทุเรียนในร้านค้าจำนวน 6 ลูก จำนวนลูกละ 10 ครั้ง

ทุเรียนลูกที่	กลุ่มทุเรียน	สุข	สุขปานกลาง	ไม่สุข
1	สุข	8	0	2
2	สุข	8	1	1

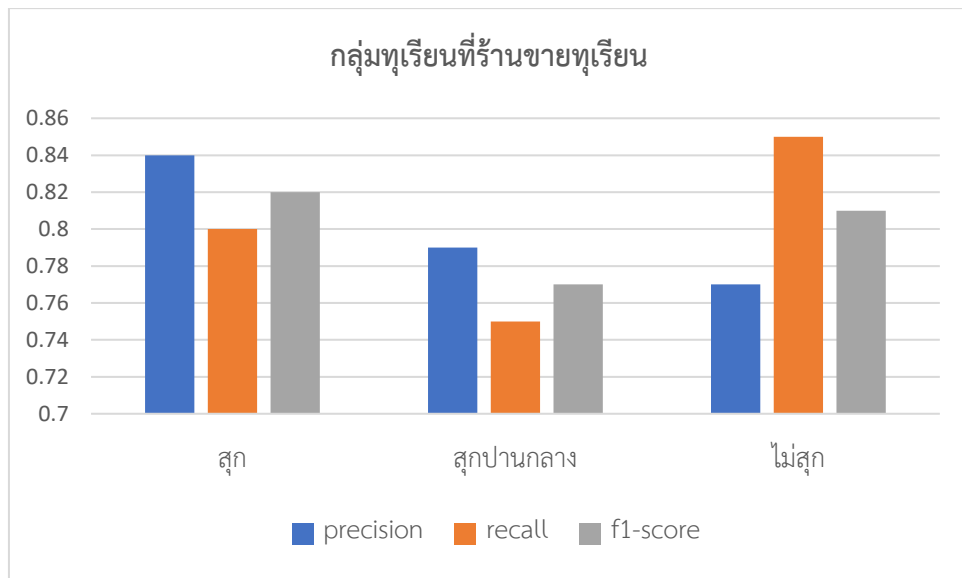
3	สุกปานกลาง	1	8	1
4	สุกปานกลาง	1	7	2
5	ไม่สุก	0	1	9
6	ไม่สุก	0	2	8

ตารางที่ 8 การวัดผลการตัดแยกความสุขของแอปพลิเคชันผลลัพธ์จากตารางที่ 7

กลุ่มทุเรียน	Precision	Recall	F1-score	จำนวนการทดลอง (ครั้ง)
สุก	0.84	0.8	0.82	20
สุกปานกลาง	0.79	0.75	0.77	20
ไม่สุก	0.77	0.85	0.81	20



รูปที่ 48 ประสิทธิภาพในการตัดแยกความสุขของทุเรียนในแต่ละกลุ่มในกรณีที่มีเสียงรบกวนน้อย



รูปที่ 49 ประสิทธิภาพในการคัดแยกความสุกของทุเรียนในแต่ละกลุ่มในกรณีที่มีเสียงรบกวนมาก



บทที่ 5

บทสรุปและอภิปราย

เนื้อหาในส่วนนี้เป็นการสรุปรายละเอียดทั้งหมดของวิทยานิพนธ์ ซึ่งประกอบด้วยการสร้างแบบจำลองและการทำงานของแอปพลิเคชันคัดแยกความสุขของทุเรียน รวมทั้งการอภิปรายปัญหาในการใช้งานแอปพลิเคชันที่ได้ทำการทดลองในบทที่ 4 การทดลองและประเมินผล

5.1 สรุปผลการสร้างแบบจำลอง

ในการสร้างแบบจำลองเราได้ทดลองหาค่าพารามิเตอร์ที่เหมาะสมกับชุดข้อมูล ซึ่งในการฝึกแบบจำลองแต่ละครั้งเราได้ทำการปรับพารามิเตอร์ต่าง ๆ ซึ่งได้จำนวนชั้นคอนโวลูชันจำนวน 4 ชั้น ที่มีขนาดชุดข้อมูลขาเข้าเป็น 9×19 ค่าพารามิเตอร์ต่าง ๆ ดังตารางที่ 3 จากนั้นก็นำแบบจำลองมาใช้ในการสร้างแอปพลิเคชัน

ปัญหาที่เกิดขึ้นระหว่างการฝึกแบบจำลองเนื่องจากจำนวนของชุดข้อมูลของเรายังมีน้อย เพื่อให้แบบจำลองไม่เกิดการ Overfit ของชุดข้อมูล จึงจำเป็นต้องตัดข้อมูลบางส่วนที่ไม่จำเป็นออกโดยการใช้ฟังก์ชัน Dropout ที่ติดมากับไลบรารี Keras แต่ถ้าหาตัดข้อมูลออกมากเกินไปจะทำให้ตัวแบบจำลองนั้นมีความถูกต้องในการคัดแยกความสุขทุเรียนที่ลดลง จากรูปที่ 49 แสดงให้เห็นว่าเมื่อนำตัวแบบจำลองไปใช้ในการจำแนกจริง ๆ จะเห็นว่าการจำแนกกลุ่มทุเรียนสุกกับไม่สุกนั้นมีประสิทธิภาพในการจำแนกมากกว่า 80 เปอร์เซ็นต์ ส่วนการจำแนกกลุ่มทุเรียนสุกปานกลางนั้นมีประสิทธิภาพยังด้อยกว่าทั้งสองกลุ่ม ทั้งนี้อาจจะเป็นเพราะว่าชุดของข้อมูลในการเรียนรู้ยังมีน้อยไปหรือการปรับพารามิเตอร์ในการเรียนรู้ยังไม่เหมาะสมกับชุดของข้อมูลที่มี

5.2 สรุปผลการใช้งานและปัญหาของแอปพลิเคชัน

การใช้งานแอปพลิเคชันในการคัดแยกความสุขของทุเรียนนั้นสามารถทำงานได้ทั้งสถานที่ที่มีเสียงรบกวนหรือไม่มีเสียงรบกวน ประกอบด้วยมีการใช้งานที่ง่ายเพียงกดปุ่มบันทึกเสียงจากนั้นใช้ไมค์เคาะที่ผลทุเรียนตั้งแต่ 2 ครั้งขึ้นไปในระยะเวลา 5 วินาที จากนั้นรอประมาณ 1-5 วินาที ส่วนนี้ขึ้นอยู่กับความเร็วอินเทอร์เน็ตที่ใช้บริการอยู่ ตัวแอปพลิเคชันก็จะแสดงผลการคัดแยกออกผ่านตัวแอปพลิเคชัน หากผู้ใช้บริการต้องการจะทำการคัดแยกอีกก็สามารถทำซ้ำได้

ปัญหาที่เกิดจากการทดลองใช้งานแอปพลิเคชันคือตัวแอปพลิเคชันไม่สามารถบอกได้ว่าอันไหนคือเสียงที่เกิดจากสภาพแวดล้อม เช่น เสียงที่เกิดจากมนุษย์ เสียงที่เกิดจากรถยนต์ หรือเสียงรบกวนอื่น ๆ ซึ่งถ้าหากเราทำการบันทึกทั้ง ๆ ที่ยังไม่ได้เคาะทุเรียนตัวแอปพลิเคชันก็ยังคงคัดแยกความสุขทุเรียนเหมือนเดิม ซึ่งประสิทธิภาพในการคัดแยกความสุขในกรณีที่มีเสียงรบกวนน้อยสามารถคัดแยกความสุขในแต่ละกลุ่มได้ถูกต้องเกิน 80 เปอร์เซ็นต์ แต่กรณีที่มีเสียงรบกวน

ค่อนข้างมากความถูกต้องในการคัดแยกจะลดลง 1-3 เปอร์เซ็นต์ และอีกปัญหาที่พบคือเนื่องจากระบบการคัดแยกความสุขทุเรียนนั้นได้ใช้ระบบการสื่อสารผ่านเอพีไอทำให้ในการคัดแยกแต่ละครั้งผู้ให้บริการจำเป็นต้องมีการเชื่อมต่อกับอินเทอร์เน็ต (Internet) ตลอดการใช้งาน ถ้าหากขณะตอนกำลังจะส่งไฟล์เสียงไปยังเซิร์ฟเวอร์แล้วระบบอินเทอร์เน็ตมีปัญหา จะให้การคัดแยกความสุขทุเรียนไม่สามารถคัดแยกได้จำเป็นต้องทำการบันทึกเสียงแล้วส่งไปยังเซิร์ฟเวอร์ใหม่อีกครั้ง นอกจากนี้ความเร็วในการรับ - ส่งข้อมูลผ่านอินเทอร์เน็ตยังมีผลต่อระยะเวลาในการคัดแยกความสุขเช่นกัน อาจจะทำให้การคัดแยกช้าประมาณ 2 - 5 วินาที หลังจากการบันทึกเสียงเสร็จ

5.3 แนวทางวิจัยในขั้นถัดไป

หลังจากแอปพลิเคชันสามารถจำแนกความสุขได้ทั้งหมด 3 กลุ่มแล้ว แนวทางวิจัยในขั้นถัดไปคือการเพิ่มจำนวนกลุ่มของเสียงที่เป็นกลุ่มทุเรียนพันธุ์อื่น ๆ และมีกลุ่มของเสียงสภาพแวดล้อมทั่วไปที่สามารถรวบรวมการจำแนกของทุเรียนได้ เช่น เสียงพูด เสียงรถยนต์ หรือเสียงอื่น ๆ เพื่อให้ตัวแอปพลิเคชันสามารถจำแนกได้แม่นยำมากขึ้น หรือไม่ก็เป็นส่วนของการปรับเปลี่ยนของแบบจำลองการเรียนรู้เพื่อให้เหมาะสมกับชุดข้อมูล นอกจากนี้ยังพัฒนาในส่วนของการบอกระดับความสุขทุเรียนเป็นระดับความนุ่มของเนื้อทุเรียนได้เช่นกัน

บรรณานุกรม

- [1] สำนักวิจัยเศรษฐกิจการเกษตร. "ส่องเส้นทางตลาดทุเรียนในภาคตะวันออกจากเกษตรกรสู่ลิ่งและสหกรณ์เพื่อกระจายผลผลิต." <https://www.moac.go.th/news-preview-411191791075> (accessed 2019).
- [2] สำนักงานมาตรฐานสินค้าเกษตรและอาหารแห่งชาติกระทรวงเกษตรและสหกรณ์, "มาตรฐานสินค้าเกษตร (ทุเรียน)," (in ไทย), vol. เล่ม 131 ตอนพิเศษ 31 ง, 13 กุมภาพันธ์ 2014. [Online]. Available: https://www.acfs.go.th/standard/download/DURIAN_new.pdf.
- [3] K. Patel. "MNIST Handwritten Digits Classification using a Convolutional Neural Network (CNN)." towards data science. <https://towardsdatascience.com/mnist-handwritten-digits-classification-using-a-convolutional-neural-network-cnn-af5fafbc35e9> (accessed 23 May, 2020, 2020).
- [4] A. Deshpande. "A Beginner's Guide To Understanding Convolutional Neural Networks." Engineering at Forward | UCLA CS '19. <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/> (accessed Jan 16, 2020).
- [5] N. Phongchit. "Convolutional Neural Network (CNN) คืออะไร." Data Wow. <https://bit.ly/37Ogv9c> (accessed Jan 16, 2020).
- [6] O. Dürr, B. Sick, and E. Murina, *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. MEAP began June 2019, 2019, p. 252.
- [7] A. Dertat. "Applied Deep Learning - Part 4: Convolutional Neural Networks." towards data science. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#8efa> (accessed Dec 19, 2019).
- [8] T. KIDS. "Fully-Connected Layer with dynamic input shape." medium. <https://medium.com/@tecokids.monastir/fully-connected-layer-with-dynamic-input-shape-70c869ae71af> (accessed 24 Jan, 2020).
- [9] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

- [10] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 1997*, vol. 181: Citeseer, p. 185.
- [11] P. S. Foundation. "Python 3.8.1 documentation." <https://docs.python.org/3/> (accessed 2 Feb, 2020).
- [12] ค. ณัฐวัตร, คู่มือเขียนโปรแกรมด้วยภาษาไพธอน *Python*, พิมพ์ครั้งที่ 1. ed. โปรววิชั่น, 2018.
- [13] P. Tongpradit. "อะไรคือ Tensorflow." สมาคมโปรแกรมเมอร์ไทย Thai Programmer Association. <https://www.thaiprogrammer.org/2018/12/มาทำความรู้จัก-tensorflow/> (accessed 2 Feb, 2020).
- [14] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [15] Chollet and Fran. "Keras: The Python Deep Learning library." GitHub. <https://keras.io/> (accessed 2 Feb, 2020).
- [16] T. T. D. Team *et al.*, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.
- [17] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing Mel-frequency cepstral coefficients on the power spectrum," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 7-11 May 2001 2001, vol. 1, pp. 73-76 vol.1, doi: 10.1109/ICASSP.2001.940770.
- [18] R. Jang. "MFCC." <http://mirllab.org/jang/books/audioSignalProcessing/speechFeatureMfcc.asp?title=12-2%20MFCC> (accessed 20 Jan, 2020).
- [19] J. O and S. III. "Hamming Window." in *Spectral Audio Signal Processing*. https://ccrma.stanford.edu/~jos/sasp/Hamming_Window.html (accessed 24 Jan, 2020).
- [20] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," *IEEE Spectrum*, vol. 4, no. 12, pp. 63-70, 1967, doi: 10.1109/MSPEC.1967.5217220.
- [21] S. Saksamudre and R. Deshmukh, "Comparative Study of Isolated Word Recognition System for Hindi Language," *International Journal of Engineering Research & Technology*, vol. 4, pp. 536-540, 07/08 2015, doi:

10.17577/IJERTV4IS070443.

- [22] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90-93, 1974.
- [23] R. Vergin, D. O'Shaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Transactions on speech and audio processing*, vol. 7, no. 5, pp. 525-532, 1999.
- [24] Google. "What is Android." <https://www.android.com/what-is-android/> (accessed 30 Jan, 2020).
- [25] G. Developers. "Android Studio." <https://developer.android.com/studio/> (accessed 30 Jan, 2020).
- [26] T. L. Foundation@. "What Is Linux?" <https://www.linux.com/what-is-linux/> (accessed 30 Jan, 2020).
- [27] D. Jemerov and S. Isakova, *Kotlin in action*, Shelter Island, NY: Manning Publications, 2017. [Online]. Available: <http://proquest.safaribooksonline.com/9781617293290>.
- [28] S. Samuel and S. Bocutiu, *Programming Kotlin*. Birmingham, UK: Packt Publishing (in English), 2017.
- [29] S. Annopornchai. "API คืออะไร." <https://saixii.com/what-is-api/> (accessed 1 Feb, 2020).
- [30] jirawoot. "REST API คืออะไร." <https://www.mindphp.com/forums/viewtopic.php?f=29&t=58244> (accessed 2020).
- [31] R. T. Fielding. "Architectural Styles and the Design of Network-based Software Architectures." <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (accessed 2020).
- [32] NICH. "RESTful API คืออะไร." http://www.4x-treme.com/restful-api-คืออะไร/?lang=th&cli_action=1594825734.488 (accessed 2020).
- [33] D. Stevenson. "What is Firebase? The complete story, abridged." <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (accessed 2020).

- [34] A. Kajornwongwattan. "มาใช้งาน Docker กันเถอะ." <https://developers.ascendcorp.com/docker-starter-342b61a1c915> (accessed 2020).
- [35] D. Inc. "What is a Container?" © 2020 Docker Inc. All rights reserved. <https://www.docker.com/resources/what-container> (accessed 2020).
- [36] G. Inc. "Cloud Run." <https://cloud.google.com/run> (accessed 2020).
- [37] A. Ronacher. "Flask web development, one drop at a time." <https://palletsprojects.com/p/flask/> (accessed 2020).
- [38] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.
- [39] R. Phoophuangpairaj, "Durian Ripeness Striking Sound Recognition Using N-gram Models with N-best Lists and Majority Voting," Cham, 2014: Springer International Publishing, in *Recent Advances in Information and Communication Technology*, pp. 167-176.
- [40] T. Palitwanon, P. Somboonyod, and W. Bundit, "Nondestructive maturity detector of fruits by using knock-sound processing," Bachelor, Electrical Engineering, Burapha University, 2008.
- [41] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 17-20 Sept. 2015 2015, pp. 1-6, doi: 10.1109/MLSP.2015.7324337.
- [42] F. Noman, C. Ting, S. Salleh, and H. Ombao, "Short-segment Heart Sound Classification Using an Ensemble of Deep Convolutional Neural Networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12-17 May 2019 2019, pp. 1318-1322, doi: 10.1109/ICASSP.2019.8682668.
- [43] C. Liu *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological Measurement*, vol. 37, pp. 2181-2213, 11/21 2016, doi: 10.1088/0967-3334/37/12/2181.
- [44] *Audacity(R): Free Audio Editor and Recorder [Computer application]*. (2019). Accessed: 27 Jan. [Online]. Available: <https://audacityteam.org/>

- [45] D. Mazzoni and R. Dannenberg, "Audacity [software]," *The Audacity Team, Pittsburg, PA, USA, 2000.*



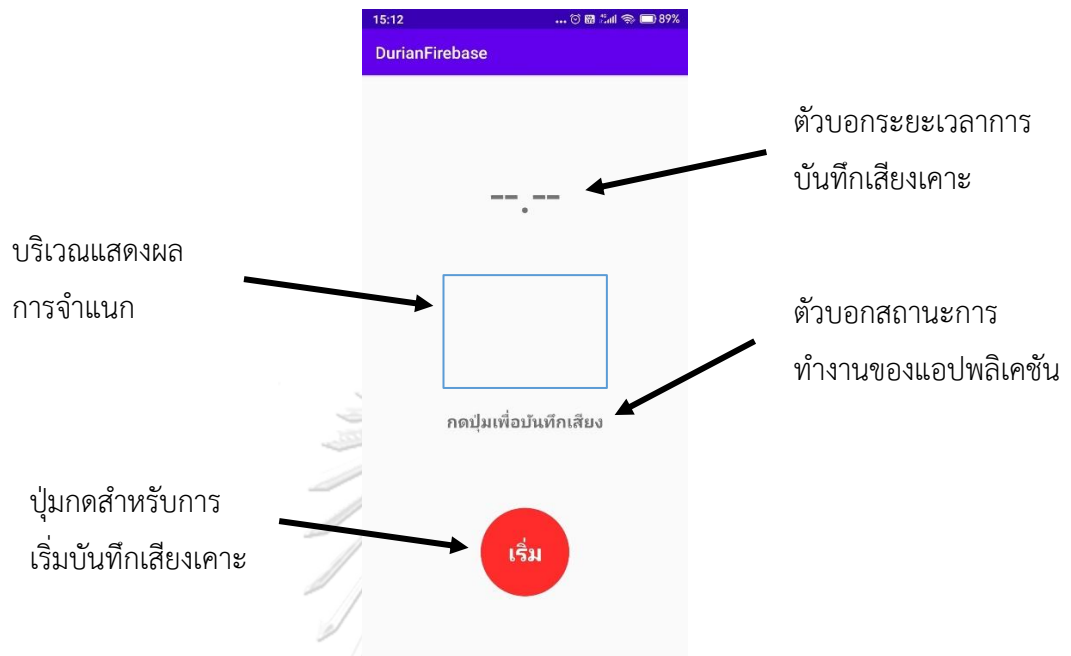


ภาคผนวก

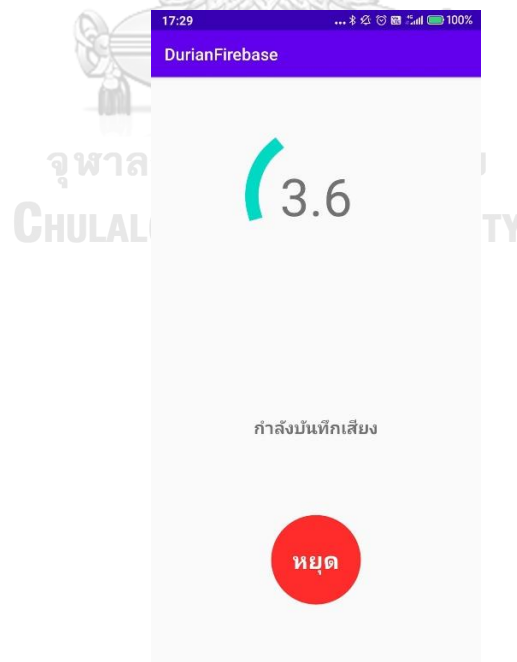
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาคผนวก ก

หน้าต่างของแอปพลิเคชัน และการลองใช้งานแอปพลิเคชันในพื้นที่จริง



รูปที่ ก. 1 หน้าต่างเริ่มต้นของแอปพลิเคชัน



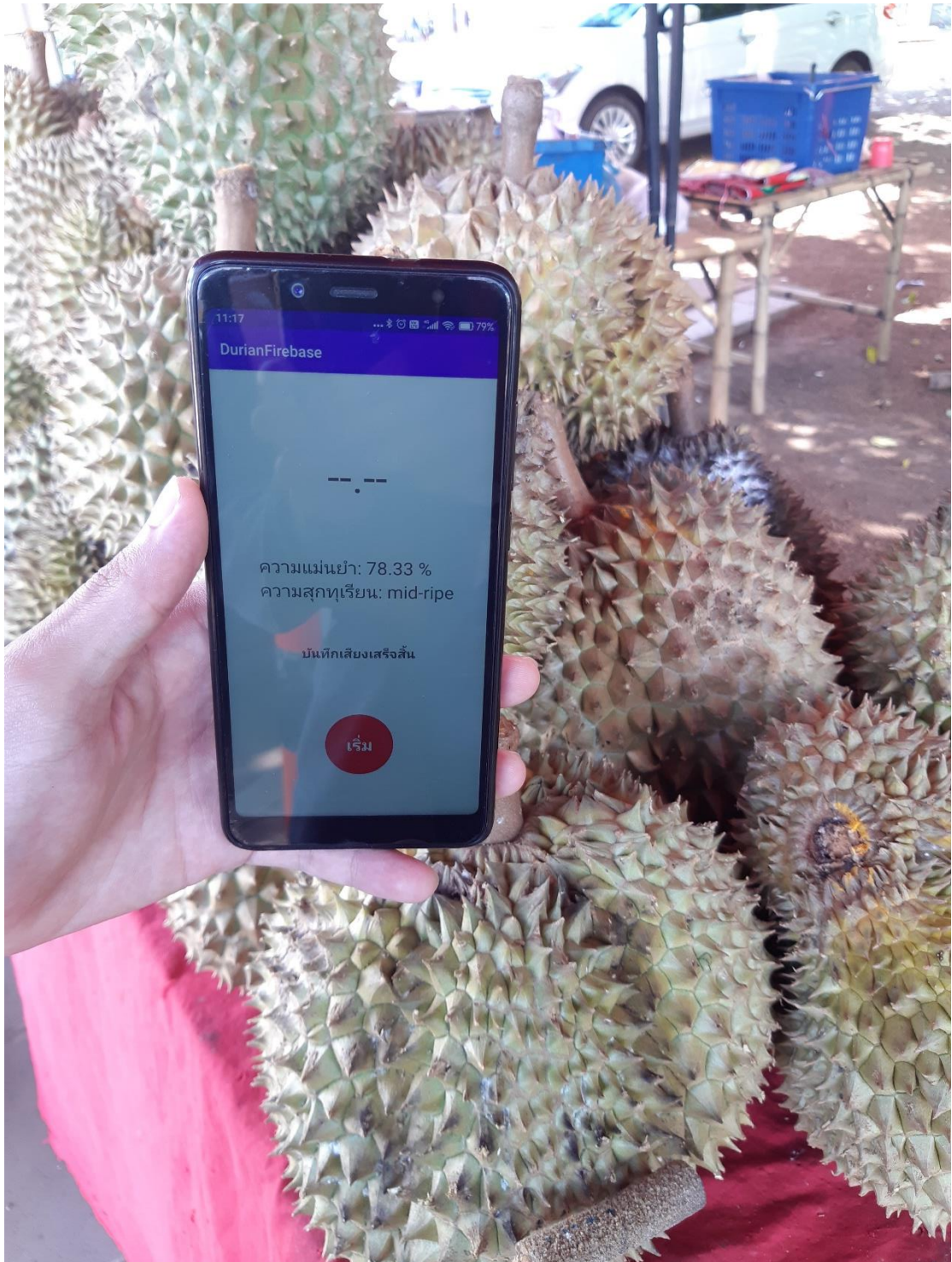
รูปที่ ก. 2 หน้าต่างการบันทึกเสียงของแอปพลิเคชัน



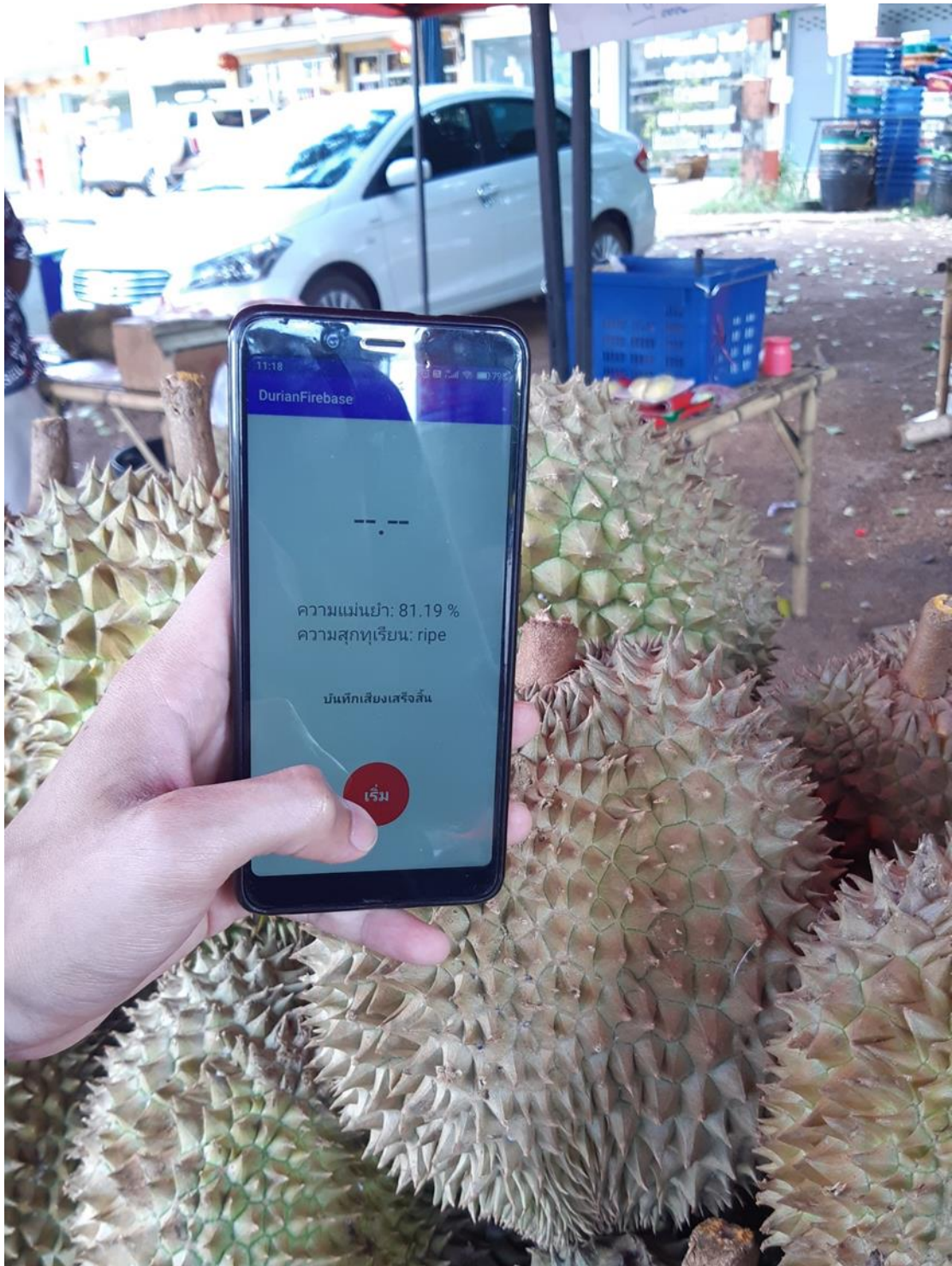
รูปที่ ก. 3 หน้าต่างเมื่อการจำแนกความสุกสำเร็จของแอปพลิเคชัน



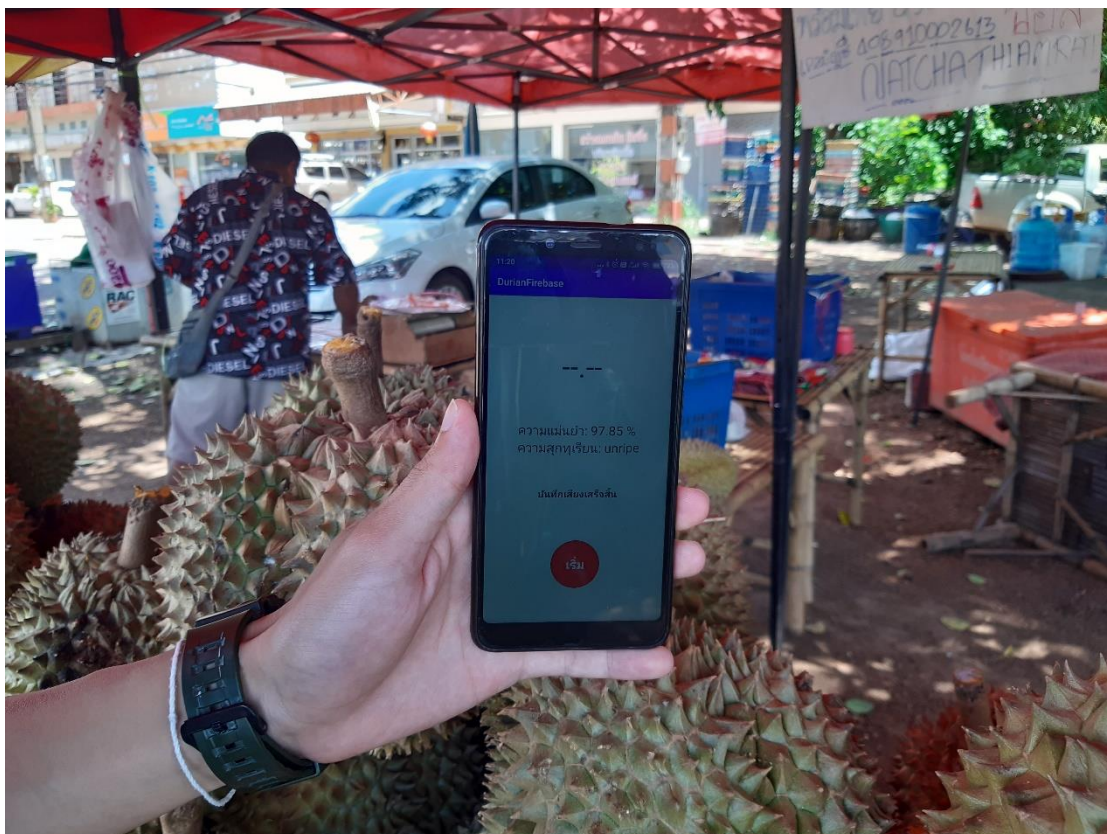
รูปที่ ก. 4 ลงพื้นที่เพื่อทดสอบประสิทธิภาพการทำงานของแอปพลิเคชันตามร้านขายทุเรียน



รูปที่ ก. 5 ผลทดสอบความสุกของกลุ่มทุเรียนสุกปานกลาง



รูปที่ ก. 6 ผลทดสอบความสุกของกลุ่มทุเรียนสุก



รูปที่ ก. 7 ผลทดสอบความสุกของกลุ่มทุเรียนไม่สุก

ภาคผนวก ข

โค้ดสำหรับการสร้างและทดสอบแบบจำลองการเรียนรู้เสียงเคาะทุเรียน

1. โค้ดสำหรับกำหนดค่าตัวแปรในการใช้สร้างแบบจำลอง ชื่อไฟล์ cfg.py

```

import os
class Config:
    def __init__(self, mode='conv', nfilt=26, nfeat=19, nfft=512,
rate=16000):
        self.mode = mode
        self.nfilt = nfilt
        self.nfeat = nfeat
        self.nfft = nfft
        self.rate = rate
        self.step = int(rate / 10)
        self.model_path = os.path.join('models', mode + '.model')
        self.val_path = os.path.join('models', mode + '.model')
        self.p_path = os.path.join('pickles', mode + '.p')

```

2. โค้ดสำหรับสร้างแบบจำลองด้วยกระบวนการการเรียนรู้คอนโวลูชัน ชื่อไฟล์ model.py

```

import os
from scipy.io import wavfile
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import keras
from keras.layers import Conv2D, MaxPooling2D, Flatten, LSTM, Activation
from keras.layers import Dropout, Dense, TimeDistributed
from keras.models import Sequential
from keras.utils import to_categorical, plot_model
from sklearn.utils.class_weight import compute_class_weight
from tqdm import tqdm
from python_speech_features import mfcc
import pickle
from keras.callbacks import ModelCheckpoint, TensorBoard, EarlyStopping
from cfg import Config
import time
config = Config(mode = 'conv')
def check_data():
    if os.path.isfile(config.p_path):

```

```

        print('Loading existing data for {} model'.format(config.mode))
        with open(config.p_path, 'rb') as handle:
            tmp = pickle.load(handle)
            return tmp
    else:
        return None

def build_rand_feat():
    tmp = check_data()
    if tmp:
        return tmp.data[0], tmp.data[1]

x = []
y = []

_min, _max = float('inf'), -float('inf')
for _ in tqdm(range(n_samples)):
    rand_class = np.random.choice(class_dist.index, p=probab_dist)
    file = np.random.choice(df[df.label == rand_class].index)
    rate, wav = wavfile.read('clean_data/' + file)
    label = df.at[file, 'label']
    rand_index = np.random.randint(0, wav.shape[0] - config.step)
    sample = wav[rand_index: rand_index + config.step]
    x_sample = mfcc(sample, rate, numcep=config.nfeat,
                    nfilt=config.nfilt, nfft=config.nfft)

    _min = min(np.amin(x_sample), _min)
    _max = max(np.amax(x_sample), _max)
    x.append(x_sample)
    y.append(classes.index(label))
#    print(classes.index(label))
config.min = _min
config.max = _max
x, y = np.array(x), np.array(y)
# y = np.array(y)
x = (x - _min) / (_max - _min)
if config.mode == 'conv':
    x = x.reshape(x.shape[0], x.shape[1], x.shape[2], 1)
elif config.mode == 'time':
    x = x.reshape(x.shape[0], x.shape[1], x.shape[2])
y = to_categorical(y, num_classes=3)
# print(y)
config.data = (x, y)

```



```

with open(config.p_path, 'wb') as handle:
    pickle.dump(config, handle, protocol=2)

return x, y

def get_conv_model():
    model = Sequential()
    # CONV2D 1
    model.add(Conv2D(32, (3, 3), padding='same',
                    input_shape=input_shape))
    model.add(Activation('relu'))

    # CONV2D 2
    model.add(Conv2D(32, (3, 3)))
    model.add(Activation('relu'))
    model.add(Dropout(0.25))

    # CONV2D 3
    model.add(Conv2D(64, (3, 3), padding='same'))
    model.add(Activation('relu'))

    # CONV2D 4
    model.add(Conv2D(64, (3, 3)))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))

    # Flatten layer
    model.add(Flatten())
    model.add(Dense(512))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    model.add(Dense(3))
    model.add(Activation('softmax'))

    model.summary()
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['acc'])

    return model

df = pd.read_csv('csv_file/df_train.csv')
df.set_index('fname', inplace=True)

```

```

for f in df.index:
    rate, signal = wavfile.read('clean_data/' + f)
    df.at[f, 'length'] = signal.shape[0] / rate

classes = list(np.unique(df.label))
class_dist = df.groupby(['label'])['length'].mean()

n_samples = 2 * int(df['length'].sum() / 0.1)
prob_dist = class_dist / class_dist.sum()
choices = np.random.choice(class_dist.index, p=prob_dist)

if config.mode == 'conv':
    x, y = build_rand_feat()
    y_flat = np.argmax(y, axis=1)
    input_shape = (x.shape[1], x.shape[2], 1)
    model = get_conv_model()

class_weight = compute_class_weight('balanced', np.unique(y_flat), y_flat)

checkpoint = ModelCheckpoint(config.model_path, monitor='val_acc',
                             verbose=1, mode='max',
                             save_best_only=True, save_weights_only=False,
                             period=1)

es = EarlyStopping(monitor='val_loss', mode='min', verbose=0, patience=5)

history = model.fit(x, y, epochs=300,
                   batch_size=32,
                   shuffle=True,
                   validation_split=0.2,
                   callbacks=[checkpoint],
                   class_weight=class_weight
                   )

model.save(config.model_path)
# Plot training & validation accuracy values
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='lower right')
plt.tight_layout()

```



```

# plt.savefig('fig/acc14.png', dpi=600)
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='lower right')
plt.tight_layout()
# plt.savefig('fig/loss14.png', dpi=600)
plt.show()

```

3. โค้ดสำหรับประเมินผลประสิทธิภาพของแบบจำลองการเรียนรู้ ชื่อไฟล์ prediction.py

```

import pickle
import os
import numpy as np
from tqdm import tqdm
from scipy.io import wavfile
from python_speech_features import mfcc
from keras.models import load_model
import pandas as pd
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import itertools

def build_predictions(audio_dir):
    y_true = []
    y_pred = []
    fn_prob = {}

    print('Extracting feature from audio')
    for fn in tqdm(os.listdir(audio_dir)):
        rate, wav = wavfile.read(os.path.join(audio_dir, fn))
        label = fn2class[fn]
        c = classes.index(label)
        y_prob = []

        for i in range(0, wav.shape[0]-config.step, config.step):
            sample = wav[i:i+config.step]
            x = mfcc(sample, rate, numcep=config.nfeat,
                    nfilt=config.nfilt, nfft=config.nfft)

```

```

x = (x - config.min) / (config.max - config.min)
if config.mode == 'conv':
    x = x.reshape(1, x.shape[0], x.shape[1], 1)

y_hat = model.predict(x, batch_size=10, verbose=0)

y_prob.append(y_hat)
y_pred.append(np.argmax(y_hat))
y_true.append(c)
fn_prob[fn] = np.mean(y_prob, axis=0).flatten()
return y_true, y_pred, fn_prob

df = pd.read_csv('csv_file/df_test.csv')
classes = list(np.unique(df.label))
fn2class = dict(zip(df.fname, df.label))
p_path = os.path.join('pickles', 'conv.p')

with open(p_path, 'rb') as handle:
    config = pickle.load(handle)

model = load_model(config.model_path)

y_true, y_pred, fn_prob = build_predictions('data_test')
acc_score = accuracy_score(y_true=y_true, y_pred=y_pred)
print('predict score:', acc_score)

# y_probs = []
# for i, row in df.iterrows():
#     y_prob = fn_prob[row.fname]
#     y_probs.append(y_prob)
#     for c, p in zip(classes, y_prob):
#         df.at[i, c] = p

# y_pred = [classes[np.argmax(y)] for y in y_probs]
# df['y_pred'] = y_pred

# df.to_csv('test_pre.csv', index=False)

cm = confusion_matrix(y_true, y_pred)

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion Matrix',
                          cmap=plt.cm.Blues):

```

```

"""
This function prints and plots the confusion matrix.
Normalization can be applied by setting `normalize=True`.
"""
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classesd))
plt.xticks(tick_marks, classesd, rotation=45)
plt.yticks(tick_marks, classesd)

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.ylabel('True label')
plt.xlabel('Predicted label')

plot_confusion_matrix(cm, classes, title='Prediction 144 instances of
testing data')
plt.tight_layout()
plt.savefig('pred/Confusion_matrix17.png', dpi=600)
plt.show()

```

ประวัติผู้เขียน

ชื่อ-สกุล	นายเวียงชัย คาระมาตย์
วัน เดือน ปี เกิด	11 เมษายน 2538
สถานที่เกิด	จังหวัดร้อยเอ็ด
วุฒิการศึกษา	สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพาในปีการศึกษา 2560 ในสังกัดห้องปฏิบัติการไฟฟ้าและการสื่อสารเพื่อการประดิษฐ์ (ECIL) ด้วยโครงการแบบจำลองระบบจัดเก็บและเรียกคืนสินค้าอัตโนมัติ ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาโทบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ผลงานตีพิมพ์	W. Kharamat, M. Wongsaisuwan and N. Wattanamongkhon, "Durian Ripeness Classification from the Knocking Sounds Using Convolutional Neural Network," 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 2020, pp. 1-4, doi: 10.1109/iEECON48109.2020.229571.
รางวัลที่ได้รับ	ผลงานเข้าร่วมประกวด "ชุดคอมพิวเตอร์อัจฉริยะแบบพกพา สำหรับแก้สถานะตัวเหลืองในทารกแรกคลอด" (ได้รับรางวัลรองชนะเลิศ) ในวันนักประดิษฐ์แห่งชาติ 2-6 กุมภาพันธ์ 2559 ผลงานเข้าร่วมประกวด "ชุดชาร์จเจอร์เพื่อลดการเสียชีวิตจาก Carbon-monoxide ในรถยนต์" ในงาน Thailand Research Expo 18-21 สิงหาคม 2559