

การเชื่อมโยงช่องว่างในการสื่อสาร: แอปพลิเคชัน
ปัญญาประดิษฐ์แบบเรียลไทม์สำหรับแปลภาษามือเป็นเสียงพูด
ภาษาไทย

BRIDGING THE COMMUNICATION GAP: A REAL-TIME AI
APPLICATION FOR SIGN LANGUAGE TO THAI SPEECH
TRANSLATION



ชนาธิป อินทร์อุดม
ธนภัทร เรืองจันทร์
นัทธพงศ์ หงษ์รัตน์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (สถิติประยุกต์)
ภาควิชาสถิติ คณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ขอสงวนสิทธิ์ในด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเพื่อการศึกษาระดับปริญญาตรีปีการศึกษา 2566

BRIDGING THE COMMUNICATION GAP: A REAL-TIME AI
APPLICATION FOR THAI SIGN LANGUAGE TO SPEECH
TRANSLATION



A SPECIAL PROBLEM SUBMITTED IN THE PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (APPLIED STATISTICS)
DEPARTMENT OF STATISTICS, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีการนำไปใช้


ACADEMIC YEAR 2023

หัวข้อปัญหาพิเศษ การเชื่อมโยงช่องว่างในการสื่อสาร: แอปพลิเคชันปัญญาประดิษฐ์แบบเรียลไทม์สำหรับแปลภาษามือเป็นเสียงพูดภาษาไทย
BRIDGING THE COMMUNICATION GAP: A REAL-TIME AI APPLICATION FOR THAI SIGN LANGUAGE TO SPEECH TRANSLATION

ชื่อนักศึกษา นายชนาธิป อินทร์อุดม รหัสนักศึกษา 63050610
นายธนภัทร เรืองจันทร์ รหัสนักศึกษา 63050628
นายนิทพงษ์ หงษ์รัตน์ รหัสนักศึกษา 63050636

ปริญญา วิทยาศาสตร์บัณฑิต (สถิติประยุกต์)
ภาควิชา สถิติ
ปีการศึกษา 2566
อาจารย์ที่ปรึกษา ผศ.ดร.พรพิมล ชัยวุฒิศักดิ์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (สถิติประยุกต์) ประจำปีการศึกษา 2566

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ดร.สุกัญญา ศรีอินมัย ประธานกรรมการ	
ดร.ธัญรดา ชัยขจรวิวัฒน์ กรรมการ	
ผศ.ดร.พรพิมล ชัยวุฒิศักดิ์ กรรมการและอาจารย์ที่ปรึกษา	<i>Pornpimol Chaiwuttisak</i>

ลิขสิทธิ์ของคณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การเชื่อมโยงช่องว่างในการสื่อสาร: แอปพลิเคชันปัญญาประดิษฐ์แบบเรียลไทม์สำหรับแปลภาษามือเป็นเสียงพูดภาษาไทย		
ชื่อนักศึกษา	นายชนาธิป อินทร์อุดม	รหัสนักศึกษา	63050610
	นายธนภัทร เรืองจันทร์	รหัสนักศึกษา	63050628
	นายนิทพงษ์ หงษ์รัตน์	รหัสนักศึกษา	63050636
ปริญญา	วิทยาศาสตร์บัณฑิต (สถิติประยุกต์)		
ภาควิชา	สถิติ		
คณะ	วิทยาศาสตร์		
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)		
ปีการศึกษา	2566		
อาจารย์ที่ปรึกษา	ผศ.ดร.พรพิมล ชัยวุฒิศักดิ์		

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาเทคโนโลยีปัญญาประดิษฐ์และการเรียนรู้เชิงลึกสำหรับออกแบบและพัฒนาแอปพลิเคชันคอมพิวเตอร์แปลภาษามือเป็นเสียงพูด โดยเป็นการประยุกต์ใช้คลังโปรแกรมมีเดียไฟฟ์เพื่อแปลภาษามือไทยโดยใช้การจับท่าทางของมือของภาษามือไทยด้วยกล้องเว็บแคมและระบุตำแหน่งสำคัญบนฝ่ามือโดยใช้เทคโนโลยีการประมวลผลโครงข่ายประสาทเทียมแบบคอนโวลูชัน 1 มิติในการจำแนกท่าทางภาษามือ เพื่อให้คนปกติสามารถสื่อสารกับผู้ที่มีความบกพร่องทางการได้ยินได้ พบว่าระบบสามารถตรวจจับตำแหน่งสำคัญและแสดงผลการแปลได้อย่างมีประสิทธิภาพ มีความแม่นยำ คิดเป็นร้อยละ 94.633

คำสำคัญ: การรู้จำภาษามือ, วิทัศน์คอมพิวเตอร์, การเรียนรู้เชิงลึก, โครงข่ายประสาทเทียมแบบคอนโวลูชันแบบ 1 มิติ (1D CNN), มีเดียไฟฟ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	BRIDGING THE COMMUNICATION GAP: A REAL-TIME AI APPLICATION FOR THAI SIGN LANGUAGE TO SPEECH TRANSLATION		
Students	Mr. Chanathip Inudom	Student ID	63050610
	Mr. Thanaphat Rueangchan	Student ID	63050628
	Mr. Nattaphong Hongrat	Student ID	63050636
Degree	Bachelor of Science (Applied Statistic)		
Department	Statistics		
School	Science		
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)		
Academic Year	2023		
Advisor	Asst.Prof.Dr. Pornpimol Chaiwuttisak		

Abstract

The research aims to investigate Artificial Intelligence and Deep Learning technologies for designing and developing a computer application that translates sign language into spoken language. It is an application of the MediaPipe library to translate Thai Sign Language by a webcam and identify key hand positions using 1-Dimensional Convolutional Neural Network (1D CNN) technology for sign language gesture recognition. The goal is to enable effective communication between hearing-impaired individuals and the general population. The system demonstrates efficient detection of key positions and accurate translation, achieving an accuracy rate of 94.633%.

Keywords: Sign Language Recognition, Computer Vision, Deep Learning, 1-Dimensional Convolutional Neural Network (1DCNN), MediaPipe

กิตติกรรมประกาศ

ปัญหาพิเศษนี้สำเร็จสมบูรณ์ด้วยดีโดยได้รับความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.พรพิมล ชัยวุฒิศักดิ์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ผู้ให้คำแนะนำ ข้อเสนอแนะ คำปรึกษา เอื้อเพื่อเอกสารต่าง ๆ ที่ใช้เป็นแนวทางในการวิเคราะห์ข้อมูลและตรวจทานแก้ไขความถูกต้องจนติดตามผลงานทุก ๆ ขั้นตอนของการดำเนินงานในการทำปัญหาพิเศษนี้จนกระทั่งเสร็จสมบูรณ์ จึงขอขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ โอกาสนี้ด้วย

ขอขอบพระคุณ ดร.สกุณา ศรีอินมัย และ ดร.ฉัตรดา ชัยขจรวัฒน์ คณะกรรมการปัญหาพิเศษที่ให้คำปรึกษา และคำแนะนำที่ทำให้ปัญหาพิเศษนี้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณคณะอาจารย์ภาควิชาสถิติทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และช่วยเหลือให้คำแนะนำในเรื่องต่าง ๆ มาโดยตลอด

สุดท้ายนี้ ขอขอบพระคุณบิดามารดา ที่คอยให้กำลังใจและสนับสนุนผู้จัดทำปัญหาพิเศษมาโดยตลอด และขอขอบคุณเพื่อน ๆ พี่น้อง ทุกคนที่ให้คำปรึกษา ช่วยเหลือการทำงานมาโดยตลอดจนปัญหาพิเศษนี้สำเร็จลุล่วงมาได้ด้วยดี

ชนาธิป อินทร์อุดม
ธนภัทร เรืองจันทร์
นัทธพงศ์ หงษ์รัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตการศึกษา.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 นิยามศัพท์เฉพาะ.....	2
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง.....	4
2.1 ภาษามือ.....	4
2.1.1 ประเภทของภาษามือ.....	4
2.1.2 ภาษามือในประเทศไทย.....	5
2.1.3 องค์ประกอบของภาษามือไทย.....	10
2.1.4 การแปลภาษามือไทย.....	10
2.2 โครงข่ายประสาทเทียม (Artificial Neural Network, ANN).....	10
2.2.1 ฟังก์ชันกระตุ้น (Activation Function).....	12
2.2.2 การฝึกฝนแบบจำลองโครงข่ายประสาทเทียม.....	12
2.2.3 ฟังก์ชันสูญเสีย (Loss Function).....	13
2.2.4 การหาค่าที่ดีที่สุด (Optimization).....	13
2.3 โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network).....	14
2.4 เครือข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network, CNN).....	15
2.4.1 ชั้นคอนโวลูชัน (Convolution Layer).....	16
2.4.2 ชั้นพูลลิง (Pooling Layer).....	18
2.4.3 ชั้นเชื่อมต่อโยงสมบูรณ์ Fully Connected Layer.....	19
2.4.4 ฟังก์ชันซอฟต์แวร์แมกซ์ (Softmax Function).....	19
2.5 โครงข่ายประสาทเทียมแบบคอนโวลูชันแบบหนึ่งมิติ (1D CNN).....	20
2.6 ทรานส์ฟอร์มเมอร์.....	21
2.7 การประมวลผลภาษาธรรมชาติ.....	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.8 ภาษาไพธอน	23
2.8.1 Open CV.....	23
2.8.2 gTts	24
2.8.3 Tensorflow	24
2.8.4 Mediapipe.....	25
2.9 Jupyter Notebook.....	26
2.10 งานวิจัยที่เกี่ยวข้อง	27
บทที่ 3 วิธีการดำเนินงานวิจัย	29
3.1 หลักการทำงาน	29
3.2 ส่วนของการสร้างระบบตรวจจับมือ (Building Phase)	32
3.2.1 ชุดข้อมูลที่นำมาฝึกฝนระบบ	33
3.2.2 คำสั่งการดึงข้อมูลค่าตำแหน่งมือและข้อมือ	35
3.2.3 คำสั่ง พิกัดตำแหน่งฝ่ามือ, ตา, จมูก, ปาก.....	38
3.3 ส่วนเรียนรู้โมเดลทำนายภาษามือ (Training Phase).....	39
3.3.1 คำสั่งการเตรียมข้อมูลตำแหน่งบนฝ่ามือและข้อมือไปใช้ในการเรียนรู้โมเดล	40
3.4 ส่วนทดสอบระบบหรือส่วนผู้ใช้งาน (Testing Phase)	45
บทที่ 4 ผลการวิจัยและการอภิปรายผล	47
4.1 การเตรียมข้อมูลเพื่อทดสอบความแม่นยำ	47
4.2 การทดสอบระบบ.....	52
4.2.1 การแสดงผลของระบบ	52
4.2.2 ผลการทดสอบประสิทธิภาพของระบบ	53
4.2.3 ปัญหาที่พบของระบบ.....	56
4.3 ขั้นตอนการใช้งานแอปพลิเคชัน	59
4.4 อภิปรายผล	61
4.4.1 เปรียบเทียบผลลัพธ์ประสิทธิภาพของความเร็วในการตรวจจับ	61
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	65
5.1 สรุปผลการวิจัย	65
5.2 ข้อจำกัดและข้อเสนอแนะ	65
5.2.1 ข้อจำกัด	65
5.2.2 ข้อเสนอแนะ.....	65
เอกสารอ้างอิง	67
ภาคผนวก.....	69
ภาคผนวก ก	70

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 ตารางผลการทดสอบประสิทธิภาพของระบบด้วย 1DCNN	53
4.2 ตารางผลการทดสอบประสิทธิภาพของระบบด้วย LSTM.....	58



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ภาษามือตามไวยากรณ์ภาษาไทย ฉัน-ไป-โรงเรียน หมายถึง “ฉันไปโรงเรียน”	5
2.2 ภาษามือตามไวยากรณ์ภาษาไทย คุณ-กิน-ข้าว-หรือ-ยัง คือ “คุณกินข้าวแล้วหรือยัง”	6
2.3 ภาษามือไทย โรงเรียน-ไป หมายถึง “ฉันไปโรงเรียน”	6
2.4 ภาษามือไทย ข้าว-กิน-แล้ว+สีหน้าด้วยเล็กคิ้ว หมายถึง “คุณกินข้าวแล้วหรือยัง”	6
2.5 การสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน	7
2.6 การเทียบเสียงพยัญชนะหรือเสียงสระในภาษาไทยที่ออกเสียงเหมือนหรือคล้ายคลึงกับเสียง พยัญชนะหรือเสียงสระในภาษาอังกฤษ	7
2.7 ตัวอย่างการเทียบเสียงพยัญชนะหรือเสียงสระในภาษาไทยที่ไม่มีในภาษาอังกฤษ	8
2.8 แบบสะกดนิ้วมือไทย	8
2.9 แบบสะกดวรรณยุกต์และสัญลักษณ์อื่น	9
2.10 การสะกดนิ้วมือไทย	9
2.11 เพอร์เซ็ปตรอน	11
2.12 Recurrent Neural Network	15
2.13 โครงสร้าง Convolution Neural Network	16
2.14 Convolution Layer	17
2.15 Pooling Operation	18
2.16 การทำฟลูลิ่ง	19
2.17 1-Dimensional Convolutional Neural Network	20
2.18 ความสัมพันธ์ไลบรารีของ OpenCV	24
2.19 Key Points ทั้ง 21 บนมือ	25
2.20 Key Points ทั้ง 33 บนร่างกาย	26
2.21 Interface การทำงานของ Jupyter Notebook	26
2.22 หน้าหลักของ Jupyter Notebook	27
3.1 Sequence Diagram แสดงการทำงานของระบบ	29
3.2 ไดอะแกรมโมเดล Transformer ร่วมกับ 1DCNN	30
3.3 Graph Diagram แสดงระบบตรวจจับภาษามือ	32
3.4 โพลีเตอร์ข้อมูลท่าทางภาษามือ	33
3.5 (ก) ความละเอียด 600 × 600 50fps, (ข) ความละเอียด 600 × 600 50fps, (ค) ความละเอียด 640 × 360 29.97fps	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 การตรวจจับท่าทางมือและใบหน้า.....	37
3.7 Block Diagram แสดงระบบการเรียนรู้โมเดลทำนายภาษามือ.....	39
3.8 Block Diagram แสดงระบบจำแนกภาษามือเป็นข้อความ และเสียง.....	45
4.1 ชุดข้อมูลวิดีโอเพื่อทดสอบความแม่นยำภาษามือไทย “หู”.....	48
4.2 ตัวอย่างโปรแกรมตรวจจับคลิปวิดีโอที่นำมาทดสอบความแม่นยำ.....	49
4.3 Console Logs ของการตรวจจับท่าทางมือ.....	51
4.4 ตัวอย่างหน้าต่างผู้ใช้งานแอปพลิเคชัน.....	52
4.5 ตัวอย่างภาษามือคำว่า “ตา”.....	54
4.6 ตัวอย่างภาษามือคำว่า “กระหายน้ำ”.....	54
4.7 ตัวอย่างภาษามือคำว่า “โปรด”.....	55
4.8 ปัญหาการตรวจจับท่าทางมือสองมือ.....	56
4.9 ปัญหาการตรวจจับท่าทางมือคำว่า “ของฉัน”.....	56
4.10 ปัญหาการตรวจจับท่าทางมือด้านมุมข้าง.....	57
4.11 ปัญหาการตรวจจับท่าทางมือด้านมุมต่ำ.....	57
4.12 ตัวไฟล์แอปพลิเคชัน.....	59
4.13 หน้าต่างแอปพลิเคชันหลังเปิดเรียกใช้งาน.....	60
4.14 หน้าต่างแอปพลิเคชันแสดงผลการแปล.....	60
4.15 ผลลัพธ์และเวลาการแปลภาษามือไทย คำว่า “โปรด” ด้วย 1DCNN และ Transformer.....	63
4.16 ผลลัพธ์และเวลาการแปลภาษามือไทย คำว่า “โปรด” ด้วย LSTM.....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การสื่อสารเป็นเรื่องที่ทุกคนในสังคมต้องการและอาจถือว่าเป็นสิ่งธรรมดาในชีวิตประจำวัน แต่สำหรับบางคน การสื่อสารกลับเป็นปัญหาที่ยากลำบาก เช่น กลุ่มผู้ที่มีความบกพร่องทางการได้ยินที่ไม่สามารถสื่อสารได้อย่างปกติ สำหรับพวกเขา การสื่อสารอาจต้องเป็นภาษามือ แต่การที่ผู้อื่นจะเข้าใจได้และตอบสนองกับภาษามืออาจไม่ง่ายเนื่องจากต้องมีความรู้ด้านภาษามือในระดับหนึ่งจึงจะสามารถสื่อสารกับผู้ที่มีความบกพร่องทางการได้ยินได้ (สมาคมคนหูหนวกแห่งประเทศไทย, 2565)

บุคคลที่มีความบกพร่องทางการได้ยิน หมายถึง ผู้ที่สูญเสียสมรรถภาพทางการได้ยินเนื่องจากอวัยวะการได้ยิน ทำให้มีผลกระทบต่อบุคคลนั้น ๆ ข้อมูลจากกระทรวงพัฒนาสังคมและความมั่นคงของมนุษย์กรมส่งเสริมและพัฒนาคุณภาพชีวิตของคนพิการในปี 2564 มีคนพิการที่ได้รับการออกบัตรประจำตัวคนพิการ จำนวน 2,138,155 คน (ร้อยละ 3.23 ของประชากรทั้งประเทศ) พบว่ามีจำนวนผู้พิการทางการได้ยินหรือสื่อความหมาย จำนวน 398,659 คน (ร้อยละ 18.65) ซึ่งเป็นอันดับที่ 2 ของจำนวนประชากรผู้พิการในประเทศไทย จากข้อมูลปี 2563 ระบุว่า ประเทศไทยมีประชากรผู้พิการทางการได้ยินกว่า 391,785 คน และจำนวนยังคงเพิ่มขึ้นต่อเนื่อง แม้จะมีข้อจำกัดด้านการได้ยิน แต่พวกเขาก็มีศักยภาพในด้านอื่นๆ และสามารถมีส่วนร่วมในสังคมได้อย่างเต็มที่ อย่างไรก็ตาม การขาดโอกาสทางการสื่อสาร อาจเป็นอุปสรรคต่อการดำรงชีวิตอย่างเต็มศักยภาพ ดังนั้น การจัดสรรทรัพยากรและการสนับสนุนอย่างเป็นรูปธรรม เช่น การส่งเสริมการเรียนรู้ภาษามือ การพัฒนาเทคโนโลยี และการสร้างความตระหนักรู้ในสังคม จึงเป็นสิ่งจำเป็นต่อการลดความเหลื่อมล้ำ และสร้างสังคมที่ทุกคนอยู่ร่วมกันอย่างเท่าเทียม (สมาคมคนหูหนวกแห่งประเทศไทย, 2565)

ในโลกที่เต็มไปด้วยเทคโนโลยีและนวัตกรรมทางดิจิทัลเรามีโอกาสในการแก้ปัญหาโดยใช้เทคโนโลยีปัญญาประดิษฐ์และการประมวลผลรูปภาพ เราสามารถที่จะสร้างแอปพลิเคชันที่ช่วยแปลงภาษามือเป็นเสียงพูด ซึ่งจะช่วยให้กลุ่มผู้พิการทางการได้ยินสามารถสื่อสารและเข้าถึงข้อมูลในโลกดิจิทัลได้อย่างสะดวกและมีความเป็นอิสระมากขึ้น ในปัจจุบันได้มีการคิดค้นเครื่องมือหรืออุปกรณ์ในการช่วยแปลความหมายของภาษามือกันมากขึ้น "ภาษามือ" คือภาษาที่ใช้ในการสื่อสารสำหรับผู้พิการทางการได้ยิน โดยใช้การผสมผสานระหว่างมือ สีหน้า และกิริยาท่าทาง เพื่อถ่ายทอดความหมายและอารมณ์ต่างๆ แทนการใช้เสียง เช่นเดียวกับภาษาพูด ภาษามือเองก็มีความหลากหลายแตกต่างกันไปในแต่ละชาติ สะท้อนถึงขนบธรรมเนียม วัฒนธรรม และลักษณะภูมิศาสตร์ที่แตกต่างกัน ตัวอย่างเช่น ภาษามือจีน ภาษามืออเมริกัน และภาษามือไทย ล้วนมีเอกลักษณ์เป็นของตัวเอง

นอกจากนี้ ภาษามือยังแบ่งออกเป็นประเภทต่างๆ ได้แก่ ภาษามือที่เกิดขึ้นเองตามธรรมชาติในกลุ่มผู้พิการทางการได้ยิน และภาษามือที่ถูกคิดค้นขึ้นเพื่อใช้ทดแทนภาษาพูดหรือภาษาเขียน ซึ่งมักจะถูกพัฒนาโดยครู ผู้ปกครอง หรือคนใกล้ชิด เพื่อให้ผู้พิการทางการได้ยินสามารถเรียนรู้และสื่อสารเรื่องราวต่างๆ ได้อย่างครอบคลุม โดยเฉพาะอย่างยิ่งเรื่องที่เป็นนามธรรม ภาษามือที่สร้างขึ้นนี้มักนำ "แบบสะกดนิ้วมือ" (Finger-Spelling) มาใช้ประกอบ ซึ่งช่วยในการสื่อสารคำศัพท์เฉพาะ เช่น ชื่อเฉพาะ ตัวอักษร วรรณยุกต์ สระ หรือแม้แต่ตัวเลข เป็นต้น (ภัทรณัฐ, 2565)

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานวิจัยนี้ได้ออกแบบสร้างโปรแกรมแปลภาษามือเป็นข้อความโดยการแปลภาษามือไทยให้เป็นข้อความและเสียงภาษาไทย โดยใช้ข้อมูลวิดีโอภาษามือไทย

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อศึกษาเทคโนโลยีปัญญาประดิษฐ์และการเรียนรู้เชิงลึก
- 1.2.2 เพื่อออกแบบและพัฒนาแอปพลิเคชันคอมพิวเตอร์แปลภาษามือเป็นเสียงพูด

1.3 ขอบเขตการศึกษา

1.3.1 ขอบเขตด้านข้อมูล

งานวิจัยนี้ใช้ชุดข้อมูลภาษามือไทย 30 คำ เพื่อใช้ในการเรียนรู้ของเครื่อง จากเว็บไซต์ th-sl.com (สมาคมคนหูหนวกแห่งประเทศไทย) โดยภาษามือที่ใช้จะเป็นคำเบื้องต้น คือ "หู" "ตา" "จมูก" "คิด" "ฉัน" "สกปรก" "หล่อ" "ผู้ชาย" "ผู้หญิง" "คาง" "สวัสดิ" "กระหายน้ำ" "ขอบคุณ" คุณ, ซื่อ, สวย, โกรธ, เท้าใหญ่, "เจ็บ" "โปรด" "ขยะ" "ของฉัน" "น้ำ" "กิน" "ดื่ม" "พ่อ" "แม่" "ใช่" "ไม่" "ช่วย"

1.3.2 ขอบเขตด้านเทคนิคที่ใช้ศึกษา

- โครงการนี้จะใช้การรับรู้ภาพและวิเคราะห์ภาพโดยใช้เทคนิค Computer Vision เพื่อแปลภาษามือให้เป็นข้อมูลที่สามารถประมวลผลได้
- ระบบการแปลงภาษามือเป็นเสียงพูดจะใช้ Deep Learning เพื่อสร้างและปรับปรุงโมเดลที่สามารถรู้จำและแปลงภาษามืออย่างแม่นยำ
- โครงการนี้จะมีการพัฒนาแอปพลิเคชันบนแพลตฟอร์มคอมพิวเตอร์เพื่อให้ผู้ใช้สามารถใช้งานระบบแปลงภาษามือได้อย่างสะดวก

1.3.3 ขอบเขตด้านเครื่องมือ

- ใช้ Jupyter Notebook ในการเขียนโปรแกรม ภาษา Python

1.4 ประโยชน์ที่คาดว่าจะได้รับ

เพื่อเสริมสร้างคุณภาพชีวิตของกลุ่มผู้พิการทางการได้ยินโดยทำให้พวกเขาสามารถสื่อสารและเข้าถึงเทคโนโลยีและข้อมูลได้อย่างมีความสะดวกและเท่าเทียมและทำให้บุคคลทั่วไปสามารถสื่อสารกับผู้พิการทางการได้ยินได้

1.5 นิยามศัพท์เฉพาะ

1. 1DCNN (1-Dimensional Convolutional Neural Networks หรือ CNN แบบ 1 มิติ) คือโมเดลของเครือข่ายประสาทเทียมที่ใช้สำหรับข้อความและสัญญาณ 1 มิติ โมเดลนี้คล้ายกับ Convolutional Neural Networks 2D โดยใช้ในการประยุกต์ใช้ต่าง ๆ เช่น การรู้จำกิจกรรมของมนุษย์ และการจำแนก ECG

2. ภาษามือไทย (Thai Sign Language หรือ ThSL) คือภาษามือทางการของประเทศไทย และใช้เป็นอย่างมากในประเทศไทย (เว็บไซต์ฐานข้อมูลมือภาษาไทย)

3. **Mediapipe** เป็นแพลตฟอร์ม (Platform) ปัญญาประดิษฐ์แบบโอเพ่นซอร์ส (Open source) ที่สามารถใช้เป็นไปป์ไลน์ (Pipeline) ตรวจสอบและรับรู้ใบหน้า มือ และท่าทางที่มีความ

ซับซ้อน โดยใช้เวลาเร่งความเร็วในการระบุและประมวลผลจึงออกมาเป็นโซลูชันที่แม่นยำและรวดเร็ว (กวิณ เต็งอำนาจ, 2563)

4. **Jupyter Notebook** เป็นเครื่องมือที่ใช้เขียนโปรแกรมสำหรับงานวิทยาศาสตร์ข้อมูล ผู้ใช้งานสามารถสร้างและแชร์เอกสารที่มีโค้ด สมการ แผนภาพ การใช้งานนั้น รวมถึงการล้างข้อมูล และการแปลงข้อมูล การจำลองเชิงตัวเลข การสร้างแบบจำลองทางสถิติ การเรียนรู้ของเครื่อง การสร้างภาพข้อมูล และอื่น ๆ (สิริรัตน์ กันทนาท, 2563)

5. **Transformer** คือโมเดลเรียนรู้เชิงลึกที่ใช้กลไก Attention เพื่อสร้างความสัมพันธ์ระหว่างคำในประโยค โดยไม่ต้องใช้ Recurrent Neural Network (RNN) หรือ Convolutional Neural Network (CNN) ที่ซับซ้อนมาก่อน Transformer ถูกนำไปใช้ในงานแปลภาษาและสร้างโมเดลภาษาขนาดใหญ่ เช่น BERT, GPT-2, T5, RoBERTa, XLM-R, mBART และ DeiT (ภควัต นาควิจิตร, 2020)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเอกสารที่เกี่ยวข้อง

สำหรับปัญหาพิเศษนี้ ผู้วิจัยมีวัตถุประสงค์เพื่อออกแบบและพัฒนาแอปพลิเคชันคอมพิวเตอร์แปลภาษามือเป็นเสียงพูด โดยใช้เทคโนโลยีวีทซ์คอมพิวเตอร์ซึ่งประกอบด้วยกล้องเว็บแคมและคลังโปรแกรมมีเดียไพพ์เพื่อตรวจจับตำแหน่งของจุดสำคัญบนฝ่ามือและใช้เครือข่ายประสาทเทียม Convolution 1 มิติ (1DCNN) เพื่อเรียนรู้ความสัมพันธ์ระหว่างลำดับของจุดสำคัญบนฝ่ามือกับท่าภาษามือซึ่งรายละเอียดและงานวิจัยที่เกี่ยวข้องดังนี้

2.1 ภาษามือ

ภาษามือเป็นอวัจนภาษาที่ใช้การเคลื่อนไหวของนิ้วมือเพื่อทำเป็นสัญลักษณ์ โดยจะใช้ร่วมกับการเคลื่อนไหวที่ของมือ แขนและร่างกาย ตลอดจนการแสดงความรู้สึกทางใบหน้าเพื่อใช้ในการสื่อสาร แทนการใช้เสียงพูด ภาษานี้ใช้ในในกลุ่มผู้พิการหรือบกพร่องทางการได้ยิน รวมถึงเพื่อน ครอบครัว และบุคคลทั่วไปซึ่งอาจไม่ได้มีความบกพร่องทางการได้ยิน

ภาษามือมักถูกเข้าใจผิดว่าเป็นภาษาสากล แต่ความจริงแล้ว ภาษามือของแต่ละชาติมีความหลากหลาย เช่นเดียวกับภาษาพูด ตัวอย่างเช่น คำว่า "พ่อ" ในภาษามือไทย ย่อมแตกต่างจากภาษามือออสเตรเลีย เช่นเดียวกับที่ภาษาพูดของทั้งสองประเทศนี้ต่างกัน โดยพื้นฐานแล้ว ภาษามือคือภาษาของคนหูหนวก ที่ใช้ท่าทาง สีหน้า และกิริยาท่าทาง ในการสื่อสารความหมาย และถ่ายทอดอารมณ์ แทนการใช้เสียง ซึ่งรูปแบบของภาษามือจะแตกต่างกันไปตามบริบททางสังคม ขนบธรรมเนียม วัฒนธรรม และลักษณะภูมิศาสตร์ เช่นเดียวกับภาษาพูด ตัวอย่างเช่น ภาษามือจีน ภาษามืออเมริกัน และภาษามือไทย ล้วนมีเอกลักษณ์เฉพาะตัว

ในปัจจุบันภาษามือได้รับการยอมรับจากนักวิชาการด้านการศึกษาของคนหูหนวก ให้เป็นภาษาอย่างหนึ่ง ที่ใช้ในการติดต่อสื่อสาร ไม่ว่าจะเป็นการสื่อสารระหว่างคนหูหนวกด้วยกัน หรือระหว่างคนหูหนวกกับคนทั่วไป (จักรพันธ์, 2565)

2.1.1 ประเภทของภาษามือ

ภาษามือที่ใช้อยู่ในกลุ่มของคนหูหนวก แบ่งออกเป็น 2 ประเภท คือ

1) ภาษามือธรรมชาติ คือ ภาษาท่าทางที่คนหูหนวกเป็นผู้สร้างขึ้นและใช้ร่วมกันเพื่อสื่อความหมายให้เข้าใจในแต่ละครอบครัวหรือชุมชน เช่น พ่อแม่ พี่น้อง ญาติ เพื่อนของคนหูหนวก ซึ่งท่าทางที่คิดขึ้นมานี้มีความแตกต่างกันมากในแต่ละครอบครัวและ แต่ละท้องถิ่น ส่วนมากเป็นท่าทางเลียนแบบธรรมชาติที่จะช่วยคนหูหนวกให้มีพัฒนาการในภาษาประจำชาติเท่าเทียมกับคนปกติ

2) ภาษามือประดิษฐ์ คือ ภาษามือที่ได้จากการนำภาษามือธรรมชาติที่คนหูหนวกทำ ท่าทางแทนการพูด เรียกชื่อสิ่งของ เครื่องใช้ หรือแสดงอาการกิริยาต่าง ๆ รวมทั้งนำการสะกดนิ้วมือมาผสมด้วย บางท่าอาจได้มาจากการทำท่าเลียนแบบภาษามือของต่างประเทศ ภาษามือประดิษฐ์นั้น ได้มาจากการรวบรวมความคิดความต้องการจากคนหูหนวกในแต่ละท้องถิ่นและนักวิชาการด้านภาษา ช่วยกันคิดท่ามือสำหรับคำต่าง ๆ ขึ้น ภาษามือประดิษฐ์จึงต้องได้รับการยอมรับจากคนหูหนวก

เอกสารนี้เป็นเอกสารของงานวิจัยเพื่อการศึกษาค้นคว้า ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อน จึงจะสามารถนำไปใช้เพื่อเป็นการเรียนการสอนสำหรับเด็กหูหนวกในโรงเรียนได้ (จักรพันธ์, 2565)

2.1.2 ภาษามือในประเทศไทย

ภาษาท่าทางหรือภาษามือธรรมชาติ เป็นการสื่อสารแบบพื้นฐานที่สุด ใช้สื่อสารกันเองในครอบครัวหรือชุมชน โดยใช้การเคลื่อนไหวของร่างกายก็สามารถสื่อสารกันได้ เหมาะกับชุดคำศัพท์หรือรูปประโยคง่าย ๆ เป็นภาษาที่เข้าใจกันเองในกลุ่มของตนเองเท่านั้น

ภาษามือตามไวยากรณ์ภาษาไทย เป็นการใช้ภาษามือที่คัดลอกคำต่อคำจากประโยคภาษาไทย เพื่อสอนให้เด็กหูหนวกฝึกเขียนภาษาไทยได้ถูกต้องตามหลักไวยากรณ์ ใช้ในการสื่อสารระหว่างครูที่มีการได้ยินกับนักเรียนหูหนวก

ภาษามือไทย เป็นภาษามาตรฐานของคนหูหนวกไทย มีท่าการใช้มือที่บอกความหมายในลักษณะคำ วลี หรือประโยคที่มีการเรียงตำแหน่งของคำไม่เหมือนกันกับภาษามือตามไวยากรณ์ภาษาไทย นอกจากนี้ยังตัดบางคำที่พอจะเข้าใจกันอยู่แล้วเพื่อให้สื่อสารได้รวดเร็วขึ้นเช่น ปลา-แมวกิน หมายถึง “แมวกินปลา” โรงเรียน-ไป หมายถึง “ฉันไปโรงเรียน”

ภาษามือไทยแบบสะกดนิ้วมือ เป็นการใช้ภาษามืออีกรูปแบบหนึ่งที่ใช้กับคำที่ยังไม่ได้บัญญัติขึ้นเป็นภาษามือไทย เป็นคำนามเฉพาะ หรืออาจใช้เพื่อให้เกิดความมั่นใจว่าสื่อสารได้ถูกต้อง การสะกดนิ้วมือนี้นี้ได้ดัดแปลงมาจากสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน โดยเปรียบเทียบการออกเสียงและหลักการทางสัทศาสตร์กล่าวคือ เสียงพยัญชนะหรือเสียงสระตัวใดในภาษาไทยที่ออกเสียงเหมือนหรือคล้ายคลึงกับเสียงพยัญชนะหรือเสียงสระในภาษาอังกฤษ ก็ให้ใช้ท่าสะกดนิ้วมือเหมือนกัน เช่น [บ] ให้ใช้เหมือน b ในการสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน, [ฟ] ให้ใช้เหมือน f ในการสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน เป็นต้น ส่วนเสียงในภาษาไทยที่ไม่มีในภาษาอังกฤษ เช่น [ส] ให้ใช้เหมือน s ในการสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน, [ศ] ให้ดัดแปลงเป็น ส-1 และ [ช] ให้ ดัดแปลงเป็น ส-2 เป็นต้น (จักรพันธ์, 2565)



รูปที่ 2.1 ภาษามือตามไวยากรณ์ภาษาไทย ฉัน-ไป-โรงเรียน หมายถึง “ฉันไปโรงเรียน”
(ที่มา: จักรพันธ์, 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ภาษามือตามไวยากรณ์ภาษาไทย คุณ-กิน-ข้าว-หรือ-ยัง คือ “คุณกินข้าวแล้วหรือยัง”
(ที่มา: จักรพันธ์, 2565)

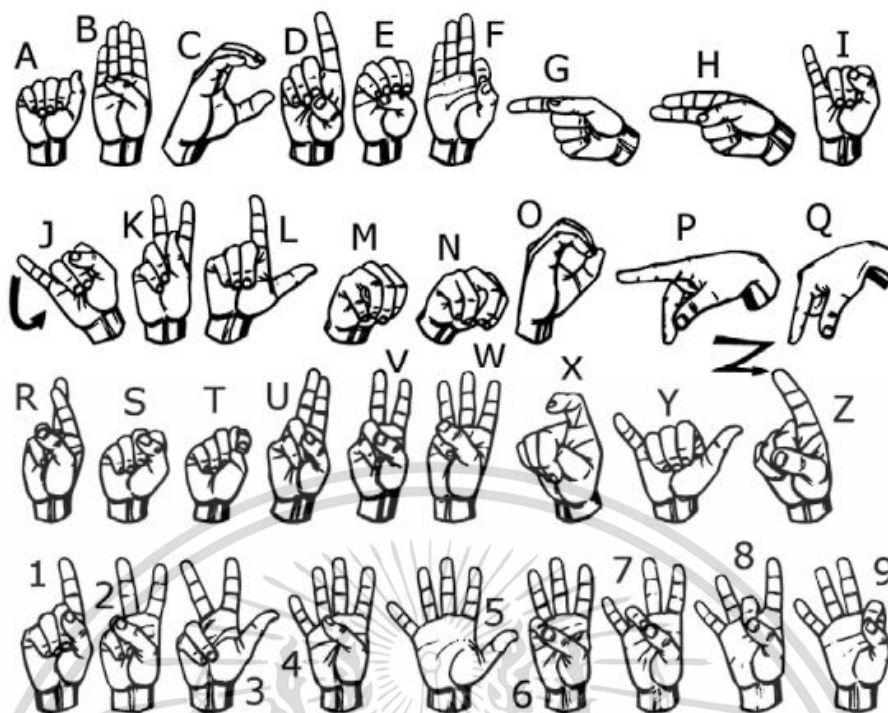


รูปที่ 2.3 ภาษามือไทย โรงเรียน-ไป หมายถึง “ฉันไปโรงเรียน”
(ที่มา: จักรพันธ์, 2565)



รูปที่ 2.4 ภาษามือไทย ข้าว-กิน-แล้ว+สีหน้าด้วยเลิกคิ้ว หมายถึง “คุณกินข้าวแล้วหรือยัง”
(ที่มา: จักรพันธ์, 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การสะกดนิ้วมือภาษาอังกฤษแบบอเมริกัน
(ที่มา: จักรพันธ์, 2565)

[b] = [บ] ทำสะกดนิ้วมือคือ

[f] = [ฟ] ทำสะกดนิ้วมือคือ

[e] = [อี] ทำสะกดนิ้วมือคือ



รูปที่ 2.6 การเทียบเสียงพยัญชนะหรือเสียงสระในภาษาไทยที่ออกเสียงเหมือนหรือคล้ายคลึงกับเสียง
พยัญชนะหรือเสียงสระในภาษาอังกฤษ
(ที่มา: จักรพันธ์, 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“ส” ใช้ทำเหมือนกับ [s]



1 ท่า

“ศ” ใช้ทำเป็น “ส - 1”



2 ท่า

“ษ” ใช้ทำเป็น “ส - 2”



2 ท่า

รูปที่ 2.7 ตัวอย่างการเทียบเสียงพยัญชนะหรือเสียงสระในภาษาไทยที่ไม่มีในภาษาอังกฤษ
(ที่มา: จักรพันธ์, 2565)

แบบสะกดนิ้วมือไทย

ประดิษฐ์โดย : คุณหญิงกมลลา ไกรฤกษ์

วาดภาพโดย นายกุลน้อย ทองน้อย

รูปที่ 2.8 แบบสะกดนิ้วมือไทย

(ที่มา: จักรพันธ์, 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แบบสะกดวรรณยุกต์ และสัญลักษณ์อื่น
(ที่มา: จักรพันธ์, 2565)



รูปที่ 2.10 การสะกดนิ้วมือไทย

(ที่มา: จักรพันธ์, 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 องค์ประกอบของภาษาไทย

ท่ามือ คือการทำมือเป็นท่าต่าง ๆ เช่น กำมือ แแบมือ กางนิ้ว จีบนิ้ว เป็นต้น ตำแหน่งของมือ คือ การวางมือหรือท่าทำมือในตำแหน่งต่าง ๆ ตำแหน่งที่ทำท่ามือ ควรจะอยู่ในรัศมีที่สามารถสังเกตได้ง่ายและชัดเจนถึงแม้จะใช้ท่ามือเดียวกันแต่คนละตำแหน่งของมือ ก็จะทำให้ความหมายที่ต่างกัน เช่น ใช้นิ้วชี้ชี้ที่หน้าอก หมายถึง “ฉัน” แต่ถ้าชี้ที่ขมับ หมายถึง “รู้” และ ถ้าชี้ออกไปที่คู่สนทนา หมายถึง “คุณ” เป็นต้น

การเคลื่อนไหวของมือ คือ การย้ายมือ เปลี่ยนมือไปในทิศทางหรือรูปแบบที่ต้องการทำมืออย่างเดียวกัน แต่เคลื่อนไหวไปในทิศทางที่ต่างกัน ความหมายก็จะแตกต่างกันด้วย เช่น มือทั้งสองตั้งขึ้น นิ้วหัวแม่มือชิดกันแล้วเลือนอกห่าง หมายถึง “เปิด” แต่ถ้าเลือนให้นิ้วหัวแม่มือชิดกัน หมายถึง “ปิด” การเคลื่อนไหวมือจะใช้มือข้างที่ถนัดเป็นมือที่เคลื่อนไหวไม่มีข้อกำหนดว่าเป็นมือซ้ายหรือมือขวาเสมอไป

การพลิกฝ่ามือ เป็นอีกส่วนที่มีผลต่อความหมาย เช่น นิ้วชิดกัน หันฝ่ามือออก ยื่นไปข้างหน้า หมายถึง “ของเขา” แต่ถ้าหันฝ่ามือเข้าหาตนเอง หมายถึง “ของฉัน”

การใช้สีหน้า เป็นการเน้นย้ำความรู้สึก ขนาด หรือปริมาณของสิ่งที่ต้องการสื่อสาร เพื่อช่วยให้เข้าใจความหมายได้ชัดเจนขึ้น (จักรพันธ์, 2565)

2.1.4 การแปลภาษาไทย

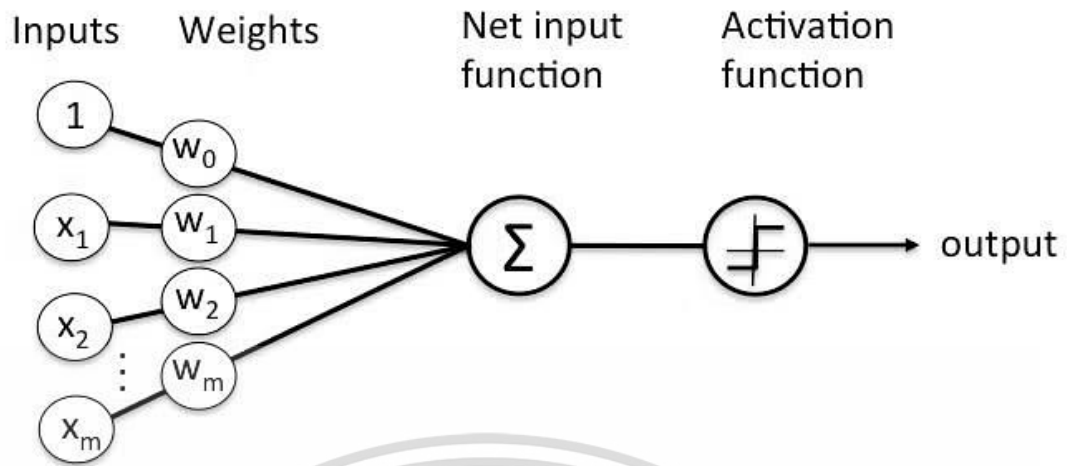
การแปล หมายถึง การถ่ายทอดความหมายและถ่ายทอดรูปแบบของการเสนอความหมายของข้อความในภาษาหนึ่งไปเป็นอีกภาษาหนึ่ง หรือการทำให้เข้าใจความหมายโดยมีวิธีการแปลที่มีหลากหลายรูปแบบ เช่น ยึดผู้เขียนเป็นหลัก ยึดสิ่งที่จะแปลเป็นหลัก หรือ ยึดผู้อ่านเป็นหลัก โดยการแปลนั้นมีเป้าหมายคือ รักษารูปแบบ รักษาความหมาย และมีกระบวนการแปลที่ครอบคลุม เนื่องจากคนพิการทางการได้ยินมีภาษาและวัฒนธรรมเป็นของตนเอง ซึ่งมีความแตกต่างจากภาษาและวัฒนธรรมของผู้ที่มีการได้ยินปกติ การแปลภาษาไทยเป็นภาษามือไทย หรือกลับกันนั้นซับซ้อนกว่าการแปลภาษาพูดทั่วไป เพราะเป็นการเชื่อมโยงวัฒนธรรมที่แตกต่าง ผู้แปลจึงต้องมีความเข้าใจอย่างลึกซึ้ง ทั้งในแง่ของภาษา บริบททางสังคม และมุมมองของทั้งสองวัฒนธรรม เพื่อให้มั่นใจว่าการแปลมีความถูกต้อง ชัดเจน เป็นธรรมชาติ และสื่อถึงความหมายที่แท้จริง

ภาษามือไทย มีโครงสร้างและไวยากรณ์เฉพาะตัว การแปลจึงไม่ใช่แค่การเปลี่ยนคำต่อคำ แต่ต้องคำนึงถึงรูปแบบภาษามือ และท่าทางประกอบที่สอดคล้องกับบริบทของภาษาไทยด้วย เทคนิคต่างๆ เช่น การเว้นจังหวะ หรือการใช้ภาษากาย ล้วนมีส่วนสำคัญในการสื่อสารให้ชัดเจน และป้องกันการตีความคลาดเคลื่อน (ศศิธร, 2560)

2.2 โครงข่ายประสาทเทียม (Artificial Neural Network, ANN)

โครงข่ายประสาทเทียมเป็นแบบจำลองที่มีการทำงานเลียนแบบการทำงานของสมองอาศัยการเรียนรู้โดยใช้หน่วยประมวลผลจำนวนมากรวมกันเป็นโครงข่ายซึ่งมีลักษณะเหมือนเซลล์สมองมนุษย์ โดยมีส่วนข้อมูลรับเข้าหรืออินพุต (Input) และส่งค่าถ่วงน้ำหนัก (Weight) ผ่านการประมวลผลแล้วจะได้ข้อมูลออกหรือเอาต์พุต (Output) (ศุภณัฐ, 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 เพอร์เซ็ปตรอน
(ที่มา: ศุภณัฐ, 2562)

โครงข่ายประสาทเทียมนั้นจะประกอบขึ้นจากเพอร์เซ็ปตรอน (Perceptron) ซึ่งเป็นหน่วยพื้นฐานที่เล็กที่สุด รูปที่ 2.11 แสดงโครงสร้างของเพอร์เซ็ปตรอน ซึ่งสามารถเขียนในรูปสมการได้ดังสมการที่ 2.1

$$y = f\left(b + \sum_{i=1}^n w_i x_i\right) \quad (2.1)$$

ข้อมูลรับเข้า (Input)

$$X = [x_1, x_2, \dots, x_n]$$

โดยที่ x_i คือ ข้อมูลรับเข้า (Input) แต่ละตัว

พารามิเตอร์ (Parameter)

$$W = [w_1, w_2, \dots, w_n], b$$

โดยที่ w_i คือ ค่าถ่วงน้ำหนัก (Weight) ดังนั้น $x_i \cdot w_i$ คือ Input ที่ถูกถ่วงน้ำหนัก
 b คือ ค่าความเอนเอียง (Bias)

เอาต์พุต (Output)

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + b$$

$$y = f(z)$$

โดยที่ z คือ ผลรวมของข้อมูลเข้าที่ถ่วงน้ำหนักแล้ว $\sum_{i=1}^n w_i x_i + b$

$f(z)$ คือ ฟังก์ชันกระตุ้น (Activation function)

y คือ ข้อมูลส่งออก (output) จากเพอร์เซ็ปตรอน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนชื่อผู้เผยแพร่โดยไม่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ฟังก์ชันกระตุ้น (Activation Function)

ฟังก์ชันกระตุ้น (Activation Function) เป็นองค์ประกอบสำคัญในโครงข่ายประสาทเทียม โดยทำหน้าที่กำหนดรูปแบบการส่งข้อมูลออกของเพอร์เซปตรอน ฟังก์ชันกระตุ้นที่นิยมใช้ในปัจจุบัน มีหลากหลายรูปแบบ อาทิเช่น ฟังก์ชันซิกมอยด์ (Sigmoid Function) ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ (Hyperbolic Tangent Function: Tanh) และฟังก์ชันเรกติไฟด์เชิงเส้น (Rectified Linear Unit Function: ReLU) (ศุภณัฐ, 2562)

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.2)$$

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \quad (2.3)$$

$$ReLU = \max(0, z) \quad (2.4)$$

โดยที่ $\sigma(z)$ คือ เอาต์พุตของฟังก์ชันซิกมอยด์
 z คือ ผลรวมเชิงเส้นของอินพุตและน้ำหนัก
 $\tanh(z)$ คือ เอาต์พุตของฟังก์ชัน Tanh
 $ReLU(z)$ คือ เอาต์พุตของฟังก์ชัน ReLU

สมการที่ 2.2 คือฟังก์ชันซิกมอยด์จะให้ผลลัพธ์อยู่ในช่วง 0 ถึง 1 สมการที่ 2.3 ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์จะให้ผลลัพธ์อยู่ในช่วง -1 ถึง 1 และ สมการที่ 2.4 ฟังก์ชันเรกติไฟด์เชิงเส้นจะให้ผลลัพธ์เป็น 0 หรือ Z

2.2.2 การฝึกฝนแบบจำลองโครงข่ายประสาทเทียม

เป้าหมายของการฝึกฝนแบบจำลองก็เพื่อให้ได้แบบจำลองที่มีความแม่นยำที่สุด ในการวัดว่าแบบจำลองสามารถทำนายผลได้ดีแค่ไหนต้องมีการนิยามฟังก์ชันต้นทุน (loss function) ขึ้นมาเพื่อจะหาความแตกต่างระหว่างค่าจริง y เปรียบเทียบกับค่าที่ทำนาย \hat{y}

ในขั้นตอนการฝึกโครงข่ายประสาทเทียม ค่าถ่วงน้ำหนัก (Weight) และค่าความเอนเอียง (Bias) จะถูกปรับเปลี่ยนเพื่อให้ได้ฟังก์ชันต้นทุนน้อยลงที่สุด ดังที่แสดงในสมการที่ 2.5

$$W^* = \arg \min_w \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, \hat{Y}^{(i)}) \quad (2.5)$$

โดยที่ y คือ ค่าจริง
 \hat{Y} คือ คีอค่าที่ทำนาย
 $L(y, \hat{Y})$ คือ ฟังก์ชันต้นทุน (Loss Function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ฟังก์ชันสูญเสีย (Loss Function)

ในการฝึกฝนแบบจำลองค่าจาก Loss Function เป็นตัวแปรที่ใช้ในการปรับค่าถ่วงน้ำหนักเพื่อหาค่าถ่วงน้ำหนักที่ดีที่สุดที่ทำให้ค่าของฟังก์ชันต้นทุนน้อยที่สุด ฟังก์ชันต้นทุนมีอยู่หลายฟังก์ชันโดยต้องเลือกใช้ให้เหมาะสมปัญหา เช่น Cross-entropy, MSE, Kullback Leibler ฯลฯ

ครอส-เอนโทรปี ลีอช (Cross-Entropy Loss) เป็นฟังก์ชันใช้บ่งชี้ประสิทธิภาพของแบบจำลองที่สามารถจำแนกได้ผิดพลาดแค่ไหน ในสมการที่ 2.6 แสดงการหา Cross-Entropy loss สำหรับการจำแนกเป็นหลายประเภท (Multi-class classification) เพื่อบอกประสิทธิภาพในการจำแนกซึ่งผลลัพธ์คือค่าความน่าจะเป็นระหว่าง 0 ถึง 1 (ศุภณัฐ, 2562)

$$L(y^{(i)}, \hat{Y}^{(i)}) = -\sum_{k=1}^K 1(y^{(i)} = k) \log(\hat{y}_k^{(i)}) \quad (2.6)$$

Mean Squared Error (MSE) เป็นฟังก์ชันที่นิยมใช้ในการวัดค่าความผิดพลาดของโมเดลสำหรับปัญหา Regression โดยเป็นการหาค่าเฉลี่ยของผลต่างกำลังสองระหว่างค่าจริง (y_i) และค่าที่โมเดลทำนาย (\hat{y}_i) ดังสมการ

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.7)$$

โดยที่ N คือจำนวนข้อมูลทั้งหมด

Kullback-Leibler Divergence (KL Divergence) เป็นฟังก์ชันที่ใช้วัดความแตกต่างระหว่างการแจกแจงความน่าจะเป็น 2 ชุด ซึ่งนิยมใช้ในปัญหา Classification โดยค่า KL Divergence จะมีค่าน้อยเมื่อการแจกแจงความน่าจะเป็นของสองชุดมีความใกล้เคียงกัน สามารถคำนวณได้จากสมการ

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \quad (2.8)$$

โดยที่ P และ Q คือการแจกแจงความน่าจะเป็นของสองชุดข้อมูล

2.2.4 การหาค่าที่ดีที่สุด (Optimization)

การหาค่าที่ดีที่สุด (Optimization) มีเป้าหมายเพื่อหาพารามิเตอร์หรือค่าถ่วงน้ำหนักที่ดีที่สุดเพื่อปรับปรุงแบบจำลองให้มีประสิทธิภาพดีขึ้น โดยวิธีการที่นิยมใช้กันก็คือการเคลื่อนลงตามความชัน (Gradient Descent Algorithm)

จากสมการที่ 2.9 เกรเดียนต์ $\nabla g(w)$ คือ เวกเตอร์ของอนุพันธ์ย่อย (Partial derivatives) ทั้งหมด ซึ่งจะบอกถึงความเปลี่ยนแปลงของ $g(w)$ เมื่อ w เปลี่ยน

$$\nabla g(w) = \left(\frac{\partial g}{\partial w_1}(w), \dots, \frac{\partial g}{\partial w_k}(w) \right) \quad (2.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $W \leftarrow W - \alpha * \frac{\partial L}{\partial W}$ เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ (2.10) ภายใต้อาณัติใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าถ่วงน้ำหนักใหม่จะหาได้จากสมการที่ 2.10

โดยที่ a คือ อัตราการเรียนรู้ (Learning Rate)

$\frac{\partial L}{\partial W}$ คือ เกรเดียนต์ของฟังก์ชันต้นทุนเทียบกับค่าถ่วงน้ำหนัก

1) Full-Batch Gradient Descent

ฟูลแบตช์เกรเดียนต์เดสเซนท์ (Full-Batch Gradient Descent) จะใช้ข้อมูลทุกจุดในการคำนวณเกรเดียนต์ ดังที่แสดงในสมการที่ 2.11 แสดงให้เห็นการคำนวณเกรเดียนต์จากข้อมูลตัวที่ 1 ถึง N

$$W \leftarrow W - \alpha * \frac{1}{N} \sum_{i=1}^N \frac{\partial L(y^{(i)}, \hat{Y}^{(i)})}{\partial W} \quad (2.11)$$

2) Stochastic Gradient Descent

สโตแคสติกเกรเดียนต์เดสเซนท์ (Stochastic Gradient Descent; SGD) จะใช้ข้อมูลเพียงจุดเดียวในการคำนวณเกรเดียนต์แทนที่ใช้ทุกจุดดังที่แสดงในสมการที่ 2.12

$$W \leftarrow W - \alpha * \frac{\partial L(y_i, \hat{Y}^{(i)})}{\partial W} \quad (2.12)$$

3) Mini-Batch Gradient Descent

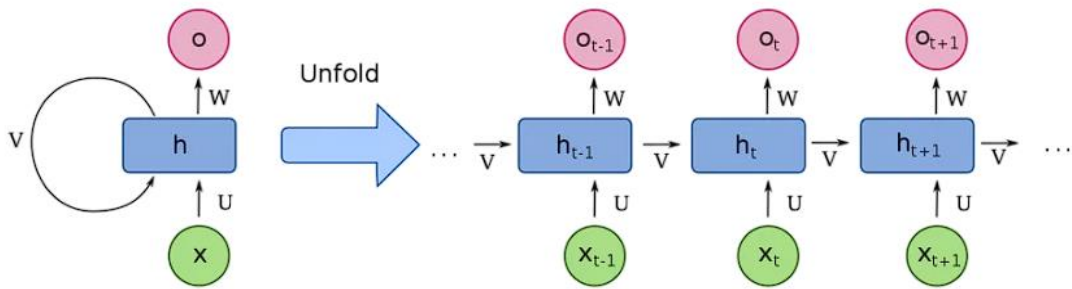
มินิแบตช์เกรเดียนต์เดสเซนท์ (Mini-Batch Gradient Descent) จะใช้ข้อมูลบางส่วนในการคำนวณเกรเดียนต์ จากสมการที่ 2.13 แสดงให้เห็นการคำนวณเกรเดียนต์จากคือข้อมูลตัวที่ k ถึง $k+m$

$$W \leftarrow W - \alpha * \sum_{i=k}^{k+m} \frac{\partial L(y_i, \hat{Y}^{(i)})}{\partial W} \quad (2.13)$$

2.3 โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network)

โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network) คือโครงข่ายประสาทเทียมแบบหนึ่ง ซึ่งจะใช้ในการประมวลผลข้อมูลที่มีลำดับ โดยใช้หลักการนำสถานะภายในของโมเดลกลับมาเป็นข้อมูลเข้าใหม่คู่กับข้อมูลเข้าแบบปกติ เรียกว่า สถานะซ่อน (Hidden State) หรือสถานะภายใน (Internal State) ช่วยให้โมเดลรู้จำรูปแบบของลำดับข้อมูลเข้า (Input Sequence) ได้ (กวินและคณะ, 2563)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 Recurrent neural network
(ที่มา: กวิน และคณะ, 2563)

จากรูปที่ 2.12 โครงข่ายประสาทเทียมแบบวนซ้ำแบบพื้นฐานในแต่ละโหนด (Node) ของโครงข่ายประสาทเทียมแบบวนซ้ำจะมีข้อมูลเข้าสองอย่าง ได้แก่ ข้อมูลเข้า ณ โหนด นั้น ๆ และผลลัพธ์ที่ได้จากการคำนวณในโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทาง คือ ผลลัพธ์ที่ออก ณ โหนดนั้น ๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าในโหนดถัดไป ข้อดีของโครงข่ายประสาทเทียมแบบวนซ้ำ คือ มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคต ซึ่งหมายถึงข้อมูลที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย โดยโครงข่ายประสาทเทียมแบบวนซ้ำมีขั้นตอนการทำงานดังที่แสดงในสมการ 2.14, 2.15

$$h_t = f(U x_t + W h_{t-1} + b) \quad (2.14)$$

$$y_t = f_y(W_y h_t + b_y) \quad (2.15)$$

โดยที่

f_h คือ Active Function ของ Hidden Layer (เช่น tanh หรือ ReLU หรือ Sigmoid function)

f_y คือ Active Function ของ Output Layer (เช่น SoftMax function)

W_h คือ Weight Matrix ของ Hidden Layer

U_h คือ Hidden-State-to-Hidden-State Matrix

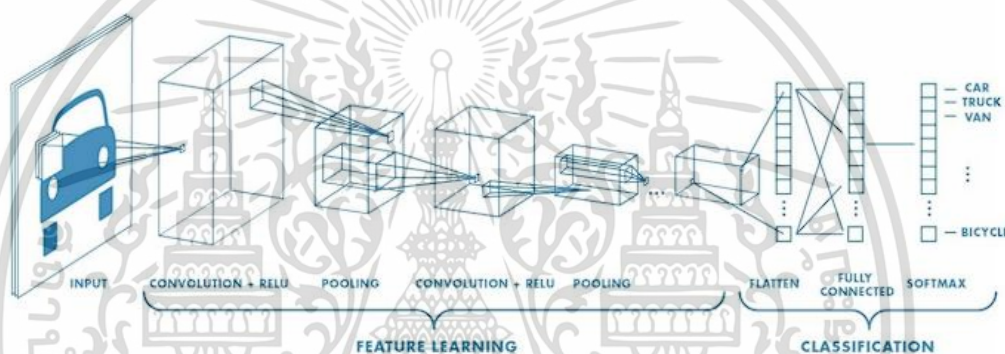
จากสมการจะเห็นว่า การคำนวณสถานะซ่อนที่เวลา $t(h_t)$ นั้นจะต้องใช้ 2 ตัวแปรที่สำคัญ คือสถานะซ่อนก่อนหน้า (h_{t-1}) และข้อมูลเข้า ณ ตอนนั้น (x_t)

2.4 เครือข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network, CNN)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network; CNN) เป็นโครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network) ที่ประกอบด้วยเซตของฟิลเตอร์ (Filter) ซึ่งฟิลเตอร์เหล่านี้จะใช้เพื่อทำคอนโวลูชันกับอินพุต และในระหว่างขั้นตอนการฝึกฝน (Train Neural Network) เป็นอีกส่วนหนึ่งที่สำคัญในการเลือกฟิลเตอร์ที่เหมาะสม เมื่ออยู่ในขั้นตอนการฝึกฝนแล้ว ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Network) พิลเตอร์เหล่านี้ก็จะถูกปรับค่าเพื่อให้ได้ฟิลเตอร์ที่เหมาะสมที่สุด โดยมีจุดประสงค์คือการสกัดฟีเจอร์จากภาพ (Feature Extraction)

โครงข่ายประสาทเทียมแบบคอนโวลูชันได้รับความนิยมอย่างมากหลังจาก Krizhevsky ได้นำเสนอ AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) ซึ่งสามารถเอาชนะในการแข่งขัน ImageNet Large Scale Visual Recognition Challenge (ILSVRC) โดยสามารถทำคะแนนได้สูงกว่าเทคนิคแบบดั้งเดิม และ CNN Based Architecture ยังถูกพัฒนาอย่างต่อเนื่อง เห็นได้จาก Top 5 Error Rate ของชุดข้อมูล ImageNet ลดจาก ~25% จนเหลือ ~2.25% ภายใน 5 ปีหลังจาก AlexNet ถูกนำเสนอ ซึ่ง error rate ที่ ~2.5 % นั้นถือว่าน้อยกว่า human error อีกด้วย ปัจจุบัน CNN ถูกประยุกต์ไปใช้ในคอมพิวเตอร์วิทัศน์ (Computer vision) ไม่ว่าจะเป็นการทำ Image Classification, Object Detection, Image and Video Recognition ฯลฯ ปัจจุบัน CNN ถูกประยุกต์ไปใช้ในคอมพิวเตอร์ วิทัศน์ (Computer Vision) (ศุภณัฐ, 2562)



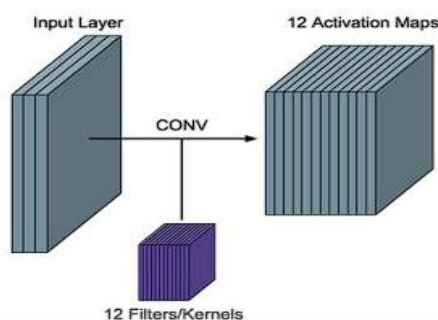
รูปที่ 2.13 โครงสร้าง Convolution Neural Network
(ที่มา: ศุภณัฐ, 2562)

จากรูปที่ 2.13 ได้แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยเริ่มต้นที่ชั้นข้อมูลเข้า (Input Layer) จะรับข้อมูลเข้าเป็นรูปภาพเพื่อเข้าสู่การดำเนินการที่ชั้นต่าง ๆ โดยสามารถจัดประเภทของชั้นต่าง ๆ ได้เป็นชั้นคอนโวลูชัน (Convolution Layer), ชั้นพูลลิ่ง (Pooling Layer), ชั้นเชื่อมโยงโดยสมบูรณ์ (Fully Connected Layer)

2.4.1 ชั้นคอนโวลูชัน (Convolution Layer)

ชั้นคอนโวลูชัน (Convolution Layer) มีไว้เพื่อทำการคอนโวลูชันระหว่างอินพุต (Input) กับฟิลเตอร์/เคอร์เนล (Filter/Kernel) ผลลัพธ์ที่ได้คือฟีเจอร์แมป/แอกทิเวชันแมป (Feature Map/Activation Map) ดังที่แสดงในรูปที่ 2.14 โดยที่ฟิลเตอร์จะมีการสุ่มและปรับเปลี่ยนในขั้นตอนการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 Convolution Layer

(ที่มา: ศุภณัฐ, 2562)

ขนาดอินพุต ($W_{input} \times H_{input} \times D_{input}$) จะถูกกำหนดโดยความกว้าง, ความสูงและความลึก โดยปกติแล้วขนาดความกว้างและความสูงจะมีขนาดเท่ากันที่ชั้นอินพุต (Input Layer) ความลึกของอินพุตจะมาจากจำนวน Color Depth เช่น หากเป็นภาพสี RGB ความลึก (D_{input}) จะเท่ากับ 3 และในชั้นที่ลึกขึ้น (Hidden Layer) ความลึก (D_{input}) จะเท่ากับจำนวนฟิลเตอร์ที่ใช้ในชั้นก่อนหน้าที่ชั้นคอนโวลูชัน (Convolution Layer) จะมีพารามิเตอร์สำหรับฟิลเตอร์ดังต่อไปนี้

จำนวนฟิลเตอร์ (K) จำนวนฟิลเตอร์ที่ใช้ จะส่งผลต่อความลึกของเอาต์พุต (Feature Map/Activation Map) ดังที่แสดงในรูปที่ 2.14 ความลึกของเอาต์พุต (12 Activation map) จะเท่ากับจำนวนฟิลเตอร์ที่ใช้ (12 Filter)

ขนาดฟิลเตอร์ ($F \times F$) เป็นการกำหนดขนาดความกว้างและความสูงของฟิลเตอร์ เช่น ฟิลเตอร์ขนาด 3×3 , 5×5 , 7×7 , 11×11 เป็นต้น

Stride (S) ฟิลเตอร์จะมีการขยับเพื่อดำเนินการคอนโวลูชันกับอินพุต ดังนั้นจะต้องมีการกำหนดขนาดของ Stride เพื่อกำหนดการขยับของฟิลเตอร์

Zero-Padding (P) คือการกำหนดว่าให้มีการเติม 0 เข้าไปที่บริเวณขอบของอินพุตหรือไม่ ปกติแล้วถ้าไม่มีการทำ Padding ผลที่ได้จากการคอนโวลูชันจะทำให้ได้ฟีเจอร์แมพที่มีขนาดเล็กกว่าขนาดอินพุต จึงมีทางเลือกในการทำ Padding ในกรณีที่ต้องการรักษาขนาดของฟีเจอร์แมพไม่ให้ลดลงและจำนวน Padding จะหาได้จากสมการที่ 2.16 เมื่อต้องการรักษาขนาดของเอาต์พุตให้เท่ากับขนาดของอินพุต

$$p = \frac{f-1}{2} \quad (2.16)$$

โดยที่ p คือ ขนาดของ Padding p

f คือ ขนาดของเคอร์เนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุต (Output) ที่ได้จากการทำคอนโวลูชันจะเรียกว่า ฟีเจอร์แมพหรือแอคทิเวชันแมพ (Feature Map; Activation Map) ซึ่งจะมีขนาดเป็น $(W_{output} \times H_{output} \times D_{output})$ โดยที่

$$W_{output} = ((W_{input} - F + 2P) / S) + 1 \quad (2.17)$$

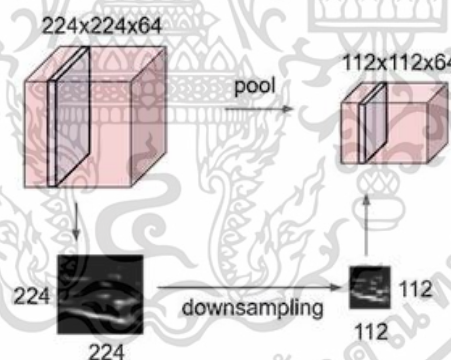
$$H_{output} = ((H_{input} - F + 2P) / S) + 1$$

$$D_{output} = K$$

ผลที่ได้จากการทำคอนโวลูชันจะต้องผ่านฟังก์ชันกระตุ้น (Activation Function) เพื่อทำการปรับค่าที่ได้จากการทำคอนโวลูชันและส่งต่อไปเป็นอินพุตในชั้นถัดไป โดยฟังก์ชันกระตุ้นที่นิยมใช้ใน CNN มีหลายฟังก์ชัน เช่น ฟังก์ชันนเรกติไฟด์เชิงเส้น (ReLU) เป็นต้น

2.4.2 ชั้นพูลลิ่ง (Pooling Layer)

ชั้นพูลลิ่ง (Pooling Layer) มักวางต่อจากชั้นคอนโวลูชัน จะถูกใช้เพื่อลดขนาดของฟีเจอร์แมพ (Feature Map, Activation Map) กล่าวคือฟีเจอร์แมพที่ได้จากชั้นคอนโวลูชัน จะถูกส่งไปยังชั้นพูลลิ่งเพื่อทำการลดขนาดความกว้างและความสูง ทำให้ข้อมูลที่ส่งไปยังชั้นถัดไปมีขนาดเล็กลง ซึ่งจะเป็นการลดพารามิเตอร์ไปด้วย



รูปที่ 2.15 Pooling Operation

(ที่มา: ศุภณัฐ, 2562)

ในชั้นพูลลิ่งจะรับข้อมูลเข้าซึ่งจะมีขนาดเป็น $(W_{input} \times H_{input} \times D_{input})$ และจะมีพารามิเตอร์ที่ต้องกำหนดดังต่อไปนี้

Pool size ขนาดของพูล $F \times F$

Stride (S) เพื่อกำหนดการขยับของ Pool

เอาต์พุตจากชั้นพูลลิ่งจะมีขนาดเป็น $(W_{output} \times H_{output} \times D_{output})$ โดยที่

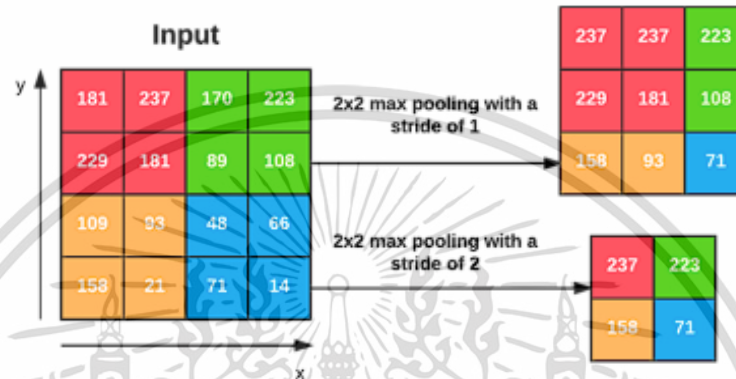
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W_{output} = ((W_{input} - F) / S) + 1 \quad (2.18)$$

$$H_{output} = ((H_{input} - F / S) + 1)$$

$$D_{output} = D_{input}$$

การดำเนินการบนชั้นพูลลิ่งสามารถทำได้ 2 วิธีคือพูลค่ามากที่สุด(Max Pooling) และพูลค่าเฉลี่ย (Average Pooling)



รูปที่ 2.16 การทำพูลลิ่ง
(ที่มา: ศุภณัฐ, 2562)

จุดประสงค์ของชั้นคอนโวลูชันและชั้นพูลลิ่งนั้นก็เพื่อทำการสกัดพีเจอร์จากภาพ โดยในชั้นก่อนแรก ๆ จะได้พีเจอร์ระดับล่าง เช่น เส้นในชั้นที่ลึกขึ้นลงไปก็จะเป็นการสกัดพีเจอร์ที่ซับซ้อนขึ้น ซึ่งจำนวนชั้นคอนโวลูชันและชั้นพูลลิ่งที่เพิ่มขึ้นก็จะส่งผลให้ความลึกของชั้นโครงข่ายเพิ่มขึ้น

2.4.3 ชั้นเชื่อมโยงสมบูรณ์ Fully Connected Layer

ชั้นเชื่อมโยงสมบูรณ์ (Fully Connected Layer) จะอยู่ในส่วนท้ายของเครือข่ายซึ่งอาจจะมี 1-2 ชั้นก่อนส่งข้อมูลเข้า Softmax Function โดยให้ทุก Neuron ในชั้นนี้จะต่อกับพีเจอร์แมพก่อนหน้าแบบ Fully-Connected โดยจะต้องทำการ (Flatten) ให้เป็นเวกเตอร์หนึ่งมิติก่อน

2.4.4 ฟังก์ชันซอฟต์แวร์แม็กซ์ (Softmax Function)

ฟังก์ชันซอฟต์แวร์แม็กซ์ (Softmax Function) จะถูกวางไว้ที่ชั้นสุดท้ายเพื่อหาความน่าจะเป็นของแต่ละคลาส คลาสที่มีความน่าจะเป็นสูงสุดคือผลการทำนาย จากสมการที่ 2.19 คะแนนความน่าจะเป็นของแต่ละคลาส โดย z_i คือ Logit Score ในเวกเตอร์ z ความน่าจะเป็นของแต่ละคลาสหาได้จากเอกซ์โพเนนเชียลของ Logitหารด้วยผลรวมของเอกซ์โพเนนเชียลของ Logit ทุกตัว ผลลัพธ์ที่ได้คือคะแนนความน่าจะเป็นของแต่ละคลาส ซึ่งผลรวมของความน่าจะเป็นของแต่ละคลาสจะมีค่าเท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^K \exp(z_k)} \quad (2.19)$$

โดยที่ z_i คือ ค่าอินพุตตัวที่ i

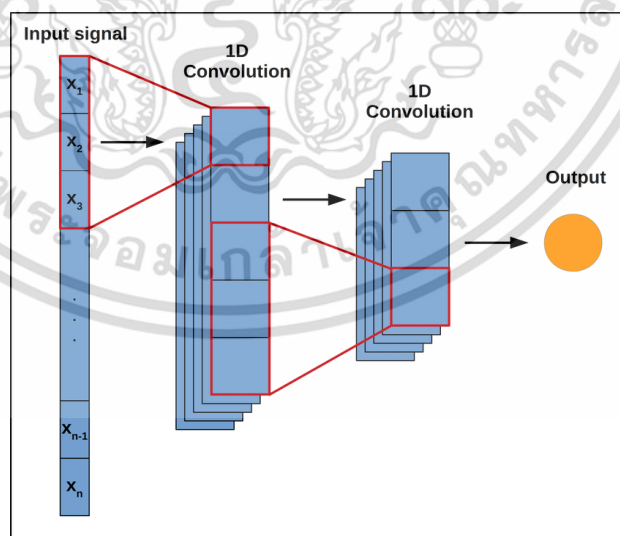
k คือ จำนวนคลาส

$\text{softmax}(z_i)$ คือ ความน่าจะเป็นที่อินพุตจะเป็นคลาสที่ i

2.5 โครงข่ายประสาทเทียมแบบคอนโวลูชันแบบหนึ่งมิติ (1D CNN)

1D CNN เป็นโมเดลในการเรียนรู้ของเครื่องที่ใช้สำหรับข้อมูลเชิงลำดับ ซึ่งมีลักษณะเป็นแถวเดียว เช่น ข้อมูลเสียง, ข้อมูลเวลา, หรือข้อมูลที่เรียงลำดับอื่น ๆ 1D CNN ทำการคำนวณผ่านชั้น Convolutional Layer ที่จะสแกนผ่านข้อมูลเชิงลำดับโดยใช้ฟิลเตอร์หรือเคอร์เนล เพื่อเรียนรู้ลักษณะเด่นของข้อมูลในรูปแบบต่าง ๆ จากนั้นจะนำผลลัพธ์ที่ได้ไปใช้ในงานต่าง ๆ เช่น การจำแนกประเภท, การตรวจจับลักษณะ, หรือการทำนายเหตุการณ์ต่าง ๆ ในข้อมูลเชิงลำดับ 1D CNN มีประโยชน์อย่างมากในการประมวลผลสัญญาณและการวิเคราะห์ข้อมูลที่มีโครงสร้างเชิงลำดับ เนื่องจากสามารถจับคู่ลักษณะเฉพาะที่ซ่อนอยู่ในข้อมูลได้อย่างมีประสิทธิภาพ และยังช่วยลดความซับซ้อนของโมเดลและเพิ่มความสามารถในการทำนายผลลัพธ์ได้อย่างแม่นยำมากขึ้น

1D CNN (1-Dimensional Convolutional Neural Network) เป็นโมเดลเชิงลึกที่ใช้ในการประมวลผลข้อมูลที่มีลำดับหรือชุดข้อมูลต่อเนื่อง เช่น ชุดข้อมูลเวลา (Time Series Data) หรือสัญญาณสังเคราะห์ (Synthetic Signals) เป็นต้น โดยเป็นการประยุกต์ใช้หลักการของ CNN ใน 2 มิติ มาสู่ข้อมูลที่มีลำดับเพื่อนำมาใช้ในงานต่าง ๆ เช่น การจำแนกประเภท (Classification) หรือการทำนาย (Prediction) ที่มีลำดับเป็นต้น



รูปที่ 2.17 1-Dimensional Convolutional Neural Network

(ที่มา: Mohammad, R.K., 2023)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1D CNN มีโครงสร้างพื้นฐานที่ประกอบด้วยการใช้งานต่างๆ ดังนี้

- **Input Layer (ชั้นข้อมูลเข้า):** รับข้อมูลเข้ามาในรูปแบบของชุดข้อมูลที่มีลำดับหรือลำดับของค่าต่อเนื่อง

- **Convolutional Layer (ชั้นคอนโวลูชัน):** ชั้นนี้ประมวลผลข้อมูลโดยใช้การคอนโวลูชันกับ filter (หรือ kernel) ที่มีขนาดเล็กและเคลื่อนที่ตามลำดับของข้อมูล เพื่อสร้าง feature maps ที่สกัดคุณลักษณะที่สำคัญออกมาจากข้อมูล

- **Activation Function (ฟังก์ชันการกระตุ้น):** หลังจากนั้นข้อมูลผ่านการคอนโวลูชัน เราจะใช้ฟังก์ชันการกระตุ้น เช่น ReLU, sigmoid, tanh เป็นต้น เพื่อเพิ่มความซับซ้อนและความสัมพันธ์ที่ไม่เชิงเส้นให้กับข้อมูล

- **Pooling Layer (ชั้นการสรุป):** ชั้นนี้มีประโยชน์ในการลดขนาดของ feature maps ที่สร้างขึ้น และสรุปคุณลักษณะที่สำคัญ โดยการใช้งาน pooling operations เช่น max pooling หรือ average pooling เป็นต้น

- **Fully Connected Layer (ชั้นเชื่อมต่อ):** หลังจากการสรุปข้อมูลและคุณลักษณะจากชั้นก่อนหน้า ชั้นเชื่อมต่อจะใช้ข้อมูลเหล่านี้ในการทำนายผลลัพธ์ ซึ่งมักจะประกอบด้วยเน็ตเวิร์กของโหนด (neurons) ที่เชื่อมต่อกันเพื่อสร้างโมเดลที่สามารถทำนายผลลัพธ์ต่าง ๆ ได้

สมการที่ 2.20 ที่ใช้ในการคำนวณใน 1D CNN นั้นมีลักษณะคล้ายกับสมการใน 2D CNN โดยแต่ละชั้นจะมีการทำงานต่างกันตามลำดับของการประมวลผล สำหรับ Convolutional Layer สมการสำคัญคือ

$$Z_i = f \left(\sum_{j=0}^{K-1} W_j \cdot X_{i+j} + b \right) \quad (2.20)$$

โดยที่ Z_i คือ ค่าใน Output Sequence ที่ตำแหน่ง i

f คือ ฟังก์ชันการกระตุ้น เช่น Relu, Sigmoid, Tanh เป็นต้น

W_j คือ ค่า Weight ใน Filter ที่ตำแหน่ง j

X_{i+j} คือ ค่าใน Input Sequence ที่ตำแหน่ง $i+j$

b คือ ค่า Bias

2.6 ทรานส์ฟอร์มเมอร์ (Transformer)

ทรานส์ฟอร์มเมอร์เป็นสถาปัตยกรรมการเรียนรู้เชิงลึกที่ถูกนำเสนอในงานวิจัย "Attention is All You Need" โดย Vaswani และคณะ ในปี 2017 โดยมีความสำคัญและได้รับความนิยมอย่างมากในด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing, NLP) และงานที่เกี่ยวข้องกับข้อมูลลำดับอื่นๆ เนื่องจากมีความมีประสิทธิภาพและประสิทธิภาพ

ทรานส์ฟอร์มเมอร์มีโครงสร้างหลักที่สำคัญดังนี้:

1. กลไกการให้ความสนใจ (Attention Mechanism): เป็นหัวใจหลักของทรานส์ฟอร์มเมอร์ ซึ่งช่วยให้โมเดลสามารถเน้นที่ส่วนต่างๆ ของอินพุตลำดับที่มีความสำคัญในการคาดเดาค่าของเอาต์พุต โดยไม่ต้องใช้การคำนวณที่ซับซ้อนเหมือนใน RNNs หรือ LSTMs

2. การเข้ารหัสเชิงตำแหน่ง (Positional Encoding): เนื่องจากทรานส์ฟอร์มเมอร์ไม่มีลำดับเอกสารนี้ภายในตัวเอง จึงต้องเพิ่มการเข้ารหัสเชิงตำแหน่งเข้าไปในอินพุตลำดับ เพื่อให้โมเดลสามารถทราบตำแหน่งสัมพัทธ์ของแต่ละส่วนของข้อมูลในลำดับได้

3. ตัวเข้ารหัส (Encoder) และตัวถอดรหัส (Decoder): ทรานส์ฟอร์มเมอร์ประกอบด้วยส่วนตัวเข้ารหัสและตัวถอดรหัสที่ซ้อนกันหลายชั้น ตัวเข้ารหัสจะเปลี่ยนอินพุตลำดับไปเป็นการแทนค่าที่มีบริบท (contextual representation) ในขณะที่ตัวถอดรหัสจะทำการแทนค่านี้ออกมาเพื่อสร้างเอาต์พุตลำดับ

4. การประมวลผลขนาน (Parallel Processing): ทรานส์ฟอร์มเมอร์สามารถประมวลผลข้อมูลได้อย่างขนาน ทำให้สามารถฝึกฝนได้เร็วกว่าโมเดลที่ใช้ลำดับเช่น RNNs และ LSTMs

ด้วยคุณสมบัติเหล่านี้ ทรานส์ฟอร์มเมอร์ได้กลายเป็นพื้นฐานสำหรับโมเดลการประมวลผลภาษาธรรมชาติที่ทันสมัยหลายรุ่น เช่น BERT, GPT-3, และ T5

2.7 การประมวลผลภาษาธรรมชาติ

การประมวลผลภาษาธรรมชาติหรือ Natural Language Processing (NLP) เป็นสาขาหนึ่งของ Artificial Intelligence (AI) หรือปัญญาประดิษฐ์ที่ทำให้คอมพิวเตอร์เข้าใจภาษามนุษย์ เพื่อวัตถุประสงค์ด้านการสื่อสารและวิเคราะห์ข้อมูลที่เป็นภาษา สาเหตุที่ต้องพัฒนา NLP เนื่องจากคอมพิวเตอร์ถูกออกแบบมาให้เหมาะสมกับการเข้าใจข้อมูลตัวเลขหรือรหัสที่มีความหมายน้อยหนึ่งชัดเจน ซึ่งไม่ตรงกับวิธีการสื่อสารของมนุษย์ที่อาศัยภาษาเป็นหลัก และภาษามีความซับซ้อนกว่ารหัสที่ใช้กับคอมพิวเตอร์อย่างมาก NLP จึงเกิดขึ้นเพื่อลดช่องว่างในการสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์ NLP รองรับทั้งการอ่านและการฟังโดยอาศัยเทคโนโลยีอื่นเข้าช่วย เช่น Visual Recognition สำหรับอ่านข้อความและใช้ Voice Recognition สำหรับฟังเสียง รวมถึง เทคโนโลยีอื่น ๆ เพื่อแสดงผลให้มนุษย์ได้รับทราบ รวมถึงสามารถถ่ายทอดข้อมูลกลับเพื่อสื่อสารกับมนุษย์ทั้งในรูปแบบของข้อความและเสียงซึ่งกระบวนการเรียนรู้ภาษาของ NLP มีพื้นฐานมาจากการทำงานของ AI แต่ต่างกันตรงที่นำมาใช้กับภาษา ซึ่งมีความซับซ้อนและเป็นนามธรรม ปัจจุบัน NLP มีกระบวนการเรียนรู้ภาษาทั้งหมด 6 ขั้นตอน ดังนี้

1) **Morphological Level:** การเข้าใจตัวอักษร NLP จะถอดคำออกเป็นตัวอักษร หาพยัญชนะ สระ ตัวสะกด เพื่อให้เกิดความแม่นยำในขั้นตอนต่อไป

2) **Lexical Level:** การเข้าใจคำ หลังจากผสมตัวอักษรแล้ว จากนั้นเริ่มหาความหมายของคำนั้น ๆ เพื่อเตรียมสำหรับการทำความเข้าใจทั้งประโยค

3) **Syntactic Level:** การเข้าใจประโยค อ้างอิงจากการเข้าใจคำและลำดับโครงสร้างตามมาตรฐานที่ระบุโดยผู้เชี่ยวชาญหรือแบบแผนที่ได้เรียนมา

4) **Semantic Level:** การเข้าใจบริบทของคำในประโยค เข้าใจถึงความหมายของคำที่ใช้ในประโยคซึ่งอยู่นอกเหนือโครงสร้างตามมาตรฐานภาษา

5) **Discourse Level:** การเข้าใจความเชื่อมโยงของประโยค เข้าใจผลกระทบของประโยคก่อนหน้าต่อความหมายของประโยคที่อ่านอยู่รวมถึงเข้าใจลำดับการใช้คำในประโยคซึ่งให้ความหมายที่แตกต่างกันด้วย

6) **Pragmatic Level:** การเข้าใจความหมายของคำและประโยคอ้างอิงจากสถานการณ์หรือฐานความรู้เดิม ซึ่งอาจไม่ได้ระบุอยู่ในเนื้อหา นั้น ๆ เพื่อให้สามารถตีความได้ใกล้เคียงกับมนุษย์ที่สามารถเชื่อมโยงข้อมูลใหม่เข้ากับความรู้เดิมได้ตลอดเวลา นอกจากการทำความเข้าใจในแต่ละจุด

แล้ว NLP ยังมีช่องทางการเรียนภาษาได้อีก 3 รูปแบบ ซึ่งจำลองมาจากการเรียนภาษาของมนุษย์ ไม่ว่าจะเป็นได้แก่ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Symbolic เป็นพื้นฐานของการเข้าใจภาษาของมนุษย์ โดย AI ต้องทำความเข้าใจ คำศัพท์ จนถึงโครงสร้างของภาษานั้น ๆ ซึ่งนักพัฒนาสามารถนำความรู้จากผู้เชี่ยวชาญใส่ลงไป ใน AI ได้โดยตรง

- Statistical หลังจากเรียนรู้พื้นฐานของภาษาแล้ว จากนั้น AI จะเก็บข้อมูลการใช้ภาษา และ นำมาวิเคราะห์รูปแบบด้วยวิธีการทางสถิติเช่น การดูความถี่ของคำที่ใช้ วิธีเรียงลำดับประโยคที่ พบบ่อย แล้วนำมาสังเคราะห์เป็นความรู้ใหม่ วิธีนี้จะช่วยให้ AI ปรับปรุงภาษาได้ตามความนิยมใน ปัจจุบัน และเข้าใจการใช้ภาษาในสาขาเฉพาะทางด้านวิทยาศาสตร์ การเงิน หรือเอกสารวิชาการต่าง ๆ เป็นต้น 11 - Connectionist AI การนำกระบวนการเรียนรู้ภาษาในรูปแบบ Statistical มา เชื่อมโยงกับ รูปแบบ Symbolic เพื่อให้การสื่อสารมีความเข้าใจที่สมบูรณ์มากขึ้น โดยเริ่มจากความรู้ เดิมที่เก็บได้ ในรูปแบบ Symbolic และดัดแปลงด้วยข้อมูลใหม่ที่ได้รับจากรูปแบบ Statistical (อนุเทพ และคณะ, 2565)

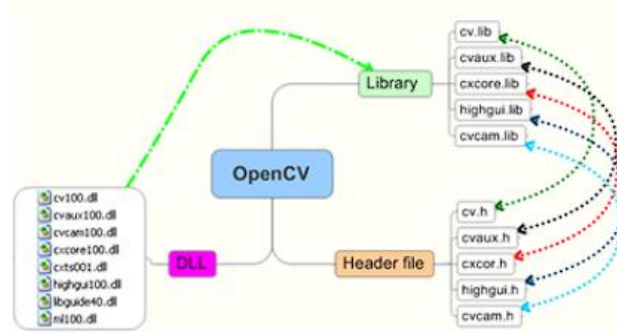
2.8 ภาษาไพธอน (Python)

ภาษาไพธอน (Python) คือ ภาษาโปรแกรมคอมพิวเตอร์ระดับสูง ซึ่งถูกออกแบบมาให้ เป็น ภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ใน ส่วน ของการแปลงชุดคำสั่งที่เขียนให้เป็นภาษาเครื่องไพธอน (Python) มีการทำงานแบบ Interpreter คือ เป็นการแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ ทำงานตาม ที่ ต้องการ นอกจากนี้ภาษาไพธอน (Python) ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้ หลากหลาย ประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มี การนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม คือ สามารถรันภาษา ไพธอน ได้ทั้งบนระบบ Unix, Linux, WindowsNT, Windows 2000, Windows XP หรือแม้แต่ ระบบ FreeBSD โดยภาษาไพธอน (Python) นี้เป็น Open Source โค้ด ของไพธอน ถูกสร้างขึ้นมา จากภาษาซี ทำให้สามารถที่จะนำ Python มาพัฒนาโปรแกรมได้โดยไม่ต้องเสียค่าใช้จ่าย

2.8.1 OpenCV

OpenCV ย่อมาจาก Open source Computer Vision เป็นไลบรารีสำหรับใช้ในการ ประมวลผลภาพ (Image Processing) ซึ่งเป็นไลบรารี โอเพนซอร์ส (Open Source) สามารถดาวน์โหลดใช้งานได้ฟรี ไลบรารีต่าง ๆ ของ OpenCV ได้พัฒนาขึ้นโดย บริษัทอินเทล (Intel) จุดเด่นใน ด้านความสามารถของไลบรารี OpenCV คือความสามารถในการประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหว เช่น ภาพจากกล้องวิดีโอหรือไฟล์วิดีโอ โดยไม่จำกัดที่ฮาร์ดแวร์เฉพาะ ทำให้ OpenCV สามารถนำไปพัฒนาโปรแกรมร่วมกับภาษาโปรแกรมอื่น ๆ ได้ นอกจากนี้ OpenCV ยังมี ฟังก์ชันที่ใช้สำหรับจัดการข้อมูลภาพและการประมวลผลภาพพื้นฐาน โดยการเรียกใช้ฟังก์ชันต่าง ๆ ของ OpenCV จะต้องผ่านไฟล์ส่วนหัว (Header file) และลิงก์กับไลบรารีต่าง ๆ รวมถึง DLL (Dynamic Link Library) โดยมีความสัมพันธ์ดังภาพที่ 2.18 ส่วนใหญ่จะถูกนำไปใช้พัฒนาการ แสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) อีกทั้งยังสนับสนุนเฟรม เวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/Py Torch และ Caffe (ณัญญา, 2564)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงความสัมพันธ์ไลบรารีของ OpenCV
(ที่มา: ัญญา, 2564)

OpenCV ประกอบด้วยไลบรารีหลัก 4 ส่วน ดังนี้:

1. **core:** โมดูลพื้นฐาน ประกอบด้วยฟังก์ชันการประมวลผลภาพขั้นพื้นฐาน โครงสร้างข้อมูล อัลกอริธึมพื้นฐาน และฟังก์ชันเกี่ยวกับเมทริกซ์
2. **imgproc:** โมดูลการประมวลผลภาพ ประกอบด้วยอัลกอริธึมการประมวลผลภาพ เช่น การกรอง การแปลงทางเรขาคณิต การตรวจจับขอบ การแบ่งส่วนภาพ และอื่นๆ
3. **highgui:** โมดูลส่วนติดต่อผู้ใช้ ประกอบด้วยฟังก์ชันสำหรับการแสดงภาพ การอ่านและเขียนภาพจากไฟล์ การจัดการหน้าต่าง และการโต้ตอบกับผู้ใช้
4. **imgcodecs:** โมดูลตัวแปลงสัญญาณภาพ ประกอบด้วยฟังก์ชันสำหรับการอ่านและเขียนภาพในรูปแบบไฟล์ต่างๆ เช่น JPEG, PNG, TIFF และอื่นๆ

2.8.2 gTTS

gTTS เป็นโมดูลที่ดึง Google TTS (Text-to-Speech) API เข้ามาใช้งาน โดยจะสร้างไฟล์เสียงสังเคราะห์ขึ้นมาในรูปแบบ mp3 ไม่จำกัดความยาวรองรับภาษาต่าง ๆ รวมถึงภาษาไทย ด้วย เวลาใช้งานต้องเรียกใช้อินเทอร์เน็ต (กวิณ และคณะ, 2563)

2.8.3 Tensorflow

สถาปัตยกรรม Tensorflow สามารถแบ่งเป็น 3 ส่วน ได้แก่ การเตรียมประมวลผลข้อมูล การสร้างแบบจำลอง ฟังก์ชันและประเมินแบบจำลอง ส่วนประกอบของ Tensorflow

1) Tensor การคำนวณทั้งหมดจึงเกี่ยวข้องกับเวกเตอร์ และ เมทริกซ์หลายมิติ ที่มีข้อมูลอยู่หลายหลากชนิด ค่าทั้งหมดในหนึ่ง Tensor จะมีขนาดของข้อมูลแตกต่างกันไปที่เรียกว่า shape

2) Graph จะเป็นตัวรวบรวมและอธิบายชุดการคำนวณทั้งหมดในระหว่างการฝึกสามารถทำงานผ่าน CPUs และ GPUs ได้หลายตัว ทั้งยังทำงานผ่านมือถือได้ ความสามารถในการพหุ ทำให้สามารถหีบใช้งานได้อย่างทันที และสามารถบันทึก graph เพื่อดำเนินการต่อในอนาคต การคำนวณทั้งหมดใน Graph เกิดจาก Tensor ที่เชื่อมไว้ด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.4 Mediapipe

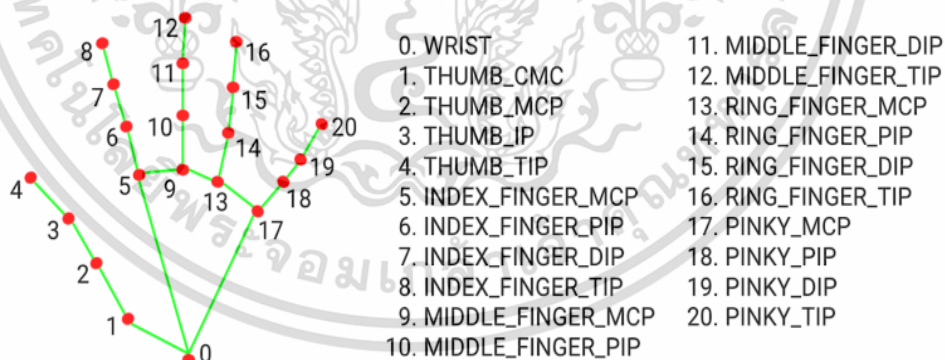
Mediapipe เป็นแพลตฟอร์ม (Platform) ปัญญาประดิษฐ์แบบโอเพนซอร์ส (Open Source) ที่สามารถใช้เป็นไปป์ไลน์ (Pipeline) ตรวจสอบและรับรู้ใบหน้า มือ และท่าทางที่มีความซับซ้อน โดยใช้การเร่งความเร็วในการระบุและประมวลผล จึงออกมาเป็นโซลูชันที่แม่นยำและรวดเร็ว

Mediapipe Hands เป็นการเรียนรู้ของเครื่อง (Machine Learning) สำหรับการตรวจจับฝ่ามือและนิ้วมือได้อย่างแม่นยำ ซึ่งจะทำให้การระบุตำแหน่งข้อต่อต่าง ๆ บริเวณมือทั้งหมด 21 จุด ในรูปแบบสามมิติ โดยมีโมเดล อยู่ 2 ส่วน คือ

1) The Palm Detector Model ทำหน้าที่ในการตรวจจับตำแหน่งฝ่ามือ มีการใช้ Single Shot Multibox Detector ในการตรวจจับฝ่ามือ ซึ่งสามารถตรวจจับฝ่ามือที่มีหลากหลายขนาดเมื่อเทียบกับขนาดของภาพและยังสามารถตรวจจับฝ่ามือในขณะที่ถูกบดบังได้อีกด้วย มีความแม่นยำเฉลี่ย 95.7% ในการตรวจจับฝ่ามือ

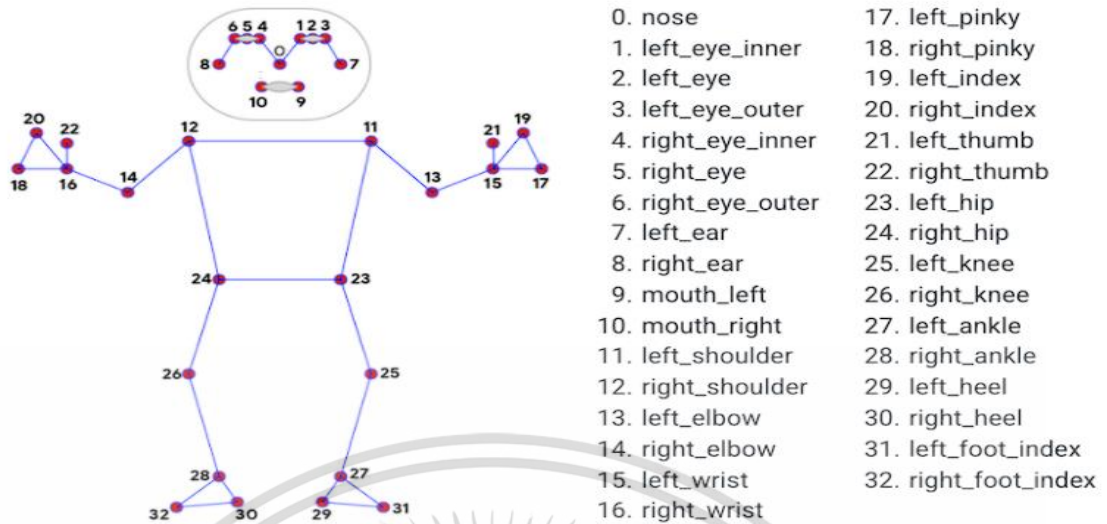
2) Hand Landmark Detector Model หลังจากใช้ The Palm Detector ในการตรวจจับฝ่ามือ จะนำภาพฝ่ามือนั้นมาระบุตำแหน่งบนฝ่ามือและข้อนิ้วมือ 21 จุด และยังทำการลากเส้นเชื่อมแต่ละจุดให้กลายเป็นโครงมือโดยมีการใช้รูปภาพท่าทางของประมาณ 30000 ภาพมาใช้ในการเรียนรู้ เพื่อสร้างโมเดลให้ เกิดการทำนายอย่างแม่นยำ (กวิน และคณะ, 2563)

Mediapipe ทำงานแลกเปลี่ยนกันระหว่างการตรวจจับจุดทั้งสามจุด โดยประสิทธิภาพของการทำงานจะขึ้นอยู่กับความเร็ว และคุณภาพของการแลกเปลี่ยนข้อมูล เมื่อรวมการตรวจจับทั้งสามเข้าด้วยกัน จะได้เป็นเทคโนโลยีที่ทำงานร่วมกันเป็นหนึ่งเดียว โดยสามารถจับ Key Points ของภาพเคลื่อนไหวได้ถึง 540+ จุด (ส่วนของท่าทาง 33 จุด มือข้างละ 21 จุด และส่วนใบหน้า 468 จุดซึ่งเป็นระดับที่ไม่เคยทำได้มาก่อน และสามารถประมวลผลได้เกือบจะเรียลไทม์ในการแสดงผลทางโทรศัพท์มือถือ (ณัฐยา, 2564)



รูปที่ 2.19 Key Points ทั้ง 21 บนมือ
(ที่มา: ณัฐยา, 2564)

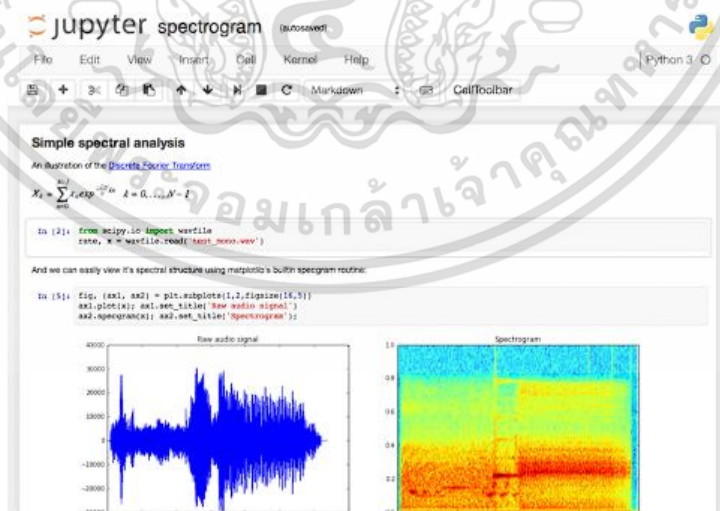
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 Key Points ทั้ง 33 บนร่างกาย (ที่มา: ัญญา, 2564)

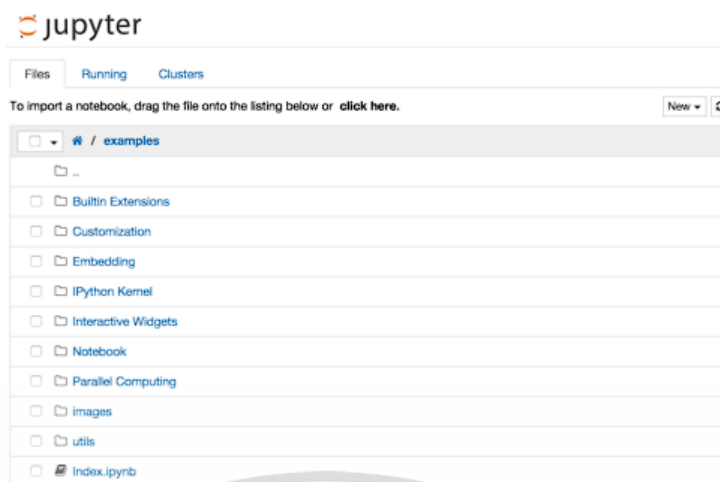
2.9 Jupyter Notebook

Jupyter Notebook เป็น Application รูปแบบ Server-Client ที่ยอมรับการสร้าง แก้ไข และรัน Notebook Document บน Web Browser โดย Application รองรับการรันบนคอมพิวเตอร์โดยไม่ต้องใช้อินเทอร์เน็ต สำหรับการใช้งาน Jupyter Notebook นั้น ตัวโปรแกรมจะมี Dashboard ที่ใช้ควบคุมไฟล์เช่น เปลี่ยนชื่อไฟล์ คัดลอกไฟล์ ยอมรับการเปิด และปิด Notebook Document



รูปที่ 2.21 Interface การทำงานของ Jupyter Notebook

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 หน้าหลักของ Jupyter Notebook
(ที่มา: ปวีวรรณ, 2561)

2.10 งานวิจัยที่เกี่ยวข้อง

จักรพันธ์ (2565) ประยุกต์ใช้คลังโปรแกรมมีเดียไฟฟ์เพื่อแปลภาษามือไทยโดยใช้การจับภาพท่ามือของภาษามือไทยด้วยกล้องเว็บแคมและระบุตำแหน่งสำคัญบนฝ่ามือด้วยคลังโปรแกรมมีเดียไฟฟ์ เปรียบเทียบผลลัพธ์ที่ได้กับแฟ้มข้อความในรูปแบบตารางที่ได้บันทึกตำแหน่งสำคัญและคำศัพท์ไว้จำนวน 35 คำ ใช้ขั้นตอนวิธีการค้นหาเพื่อนบ้านที่ใกล้ที่สุด K ตัว เพื่อเลือกคำศัพท์ที่ใกล้เคียงกับท่ามือมากที่สุด ประมวลผลด้วยชุดคำสั่งภาษาไพทอนและแสดงผลการแปลบนจอภาพ ในทันทีการประมวลผลใช้เพียงหน่วยประมวลผลกลางของคอมพิวเตอร์ทั่วไปจึงไม่จำเป็นต้องใช้ระบบคอมพิวเตอร์ที่มีประสิทธิภาพสูง พบว่าระบบสามารถตรวจจับตำแหน่งสำคัญและแสดงผลการแปลได้อย่างมีประสิทธิภาพ มีความแม่นยำร้อยละ 85.71

ภักธนัฐ (2565) ศึกษาประสิทธิภาพของการรู้จำภาษามือไทยด้วย Long Short-Term Memory (LSTM) อัลกอริทึมบนเทคนิคโครงกระดูกของ MediaPipe ในการทดลองสามารถแบ่งออกได้ 3 หัวข้อใหญ่: (1) การสะกดด้วยนิ้วสำหรับตัวเลข (1-9), (2) ภาษามือธรรมชาติ จำนวน 20 คำ และ (3) โปรแกรมแปลภาษามือด้วยซอฟต์แวร์ โดยผลการทดลองแสดงความแม่นยำของการสะกดด้วยนิ้วและแบบภาษามือธรรมชาติด้วยโปรแกรมแปลภาษามืออยู่ระหว่าง 94% - 99% และ 80.5% ตามลำดับ

ในช่วงสิบปีที่ผ่านมา, Convolutional Neural Networks (CNNs) ได้กลายเป็นมาตรฐานสำหรับการดำเนินการทาง Computer Vision และ Machine Learning ต่าง ๆ ไปแล้ว เป็น Feedforward Artificial Neural Networks (ANNs) ที่มีชั้นโครงข่ายสลับระหว่างชั้น Convolutional และ Subsampling โดย CNNs มี Deep 2D ซึ่งประกอบด้วยชั้นซ่อนที่หลายชั้น และพารามิเตอร์จำนวนล้านตัวที่สามารถเรียนรู้วัตถุและรูปแบบที่ซับซ้อนได้ หากมีการเทรนด้วยข้อมูลปริมาณใหญ่พอและถูกต้อง CNNs จะเป็นเครื่องมือหลักสำหรับการประยุกต์ใช้ทางวิศวกรรมต่าง ๆ สำหรับสัญญาณแบบ 2 มิติ เช่น ภาพและเฟรมวิดีโอ แต่อาจไม่เป็นทางเลือกที่เหมาะสมในการประยุกต์ใช้สำหรับสัญญาณแบบ 1 มิติ โดยเฉพาะเมื่อข้อมูลที่มีการเทรนมีน้อยหรือเฉพาะในการใช้งาน ในการแก้ไขปัญหา 1D CNNs ได้รับการเสนอเมื่อเร็ว ๆ นี้และได้รับการยอมรับในระดับการ

ทำงานที่ดีที่สุด ในหลาย ๆ ประเภทของการประยุกต์ใช้ เช่น การจำแนกประเภทข้อมูลชีวภาพที่ได้รับ การปรับแต่งและการวินิจฉัยโรคในขั้นต้น การตรวจสอบสุขภาพโครงสร้าง การตรวจจับและระบุความผิดปกติในอิเล็กทรอนิกส์และการตรวจจับข้อผิดพลาดในเครื่องยนต์ไฟฟ้า ข้อดีหลักอีกอย่างคือสามารถนำมาใช้ในการดำเนินงานฮาร์ดแวร์แบบ Real-Time และราคาประหยัดได้เนื่องจากโครงสร้างขนาดเล็กและเรียบง่ายของ 1D CNNs ที่ดำเนินการ Convolution เพียงแค่ 1 มิติเท่านั้น

Kiranyaz et al. (2021) นำเสนอภาพรวมที่ครอบคลุมเกี่ยวกับโครงสร้างทั่วไปและหลักการของ 1D CNNs พร้อมกับการประยุกต์ใช้หลักๆ ในการประยุกต์ใช้ทางวิศวกรรม การแสดงประสิทธิภาพระดับที่ดีที่สุดถูกสรุปด้วยคุณสมบัติที่เป็นเอกลักษณ์ของมัน ชุดข้อมูลทดสอบและซอฟต์แวร์หลักของ 1D CNNs ที่ใช้ในการประยุกต์ใช้เหล่านั้นถูกแบ่งปันในเว็บไซต์ที่ได้รับการกำหนดไว้ แม้ว่าจะยังไม่มีความเกี่ยวข้องกับการทบทวน 1D CNN และการนำไปใช้ในงานวิจัย แต่บทความนี้ก็ช่วยเติมเต็มช่องว่างนี้ได้

กวิณ และคณะ (2563) ได้ศึกษาและวิเคราะห์จำแนกท่าทางภาษามือโดยใช้วิธีทางเทคโนโลยีการประมวลผลโครงข่ายประสาทเทียมแบบวนซ้ำการวิเคราะห์การจำแนกภาษามือนั้นมีความสำคัญเนื่องจากภาษามือเป็นส่วนหนึ่งของการสื่อสารและใช้ในการถ่ายทอดความหมายเมื่อเทียบกับการพูดซึ่งจะทำให้คนปกติสามารถสื่อสารกับผู้ที่มีความผิดปกติทางการได้ยินโดยการทำงานแบ่งเป็น 3 ส่วนคือการเก็บวิดีโอและแปลงเป็นข้อมูล, การเทรนโมเดล, การทำนายซึ่งจากวิดีโอเป็นข้อมูลที่มีความต่อเนื่องเป็นลำดับการเทรนโมเดลจึงต้องใช้โมเดลที่เป็นโครงข่ายประสาทเทียมแบบวนซ้ำและการทำนายจะกำหนดเงื่อนไขเพื่อให้ได้ผลลัพธ์ตามความหมายที่ถูกต้องรวมถึงกำหนดลักษณะของผลลัพธ์ที่ต้องการทั้งข้อความเสียงและภาพแอนิเมชัน

ปวิวรรต (2561) ได้ใช้เทคนิคการเรียนรู้ของเครื่องในการทำนายประสิทธิภาพของ Jupyter Notebook ที่ทำงานอยู่บน JupyterHub โดยใช้ข้อมูลจาก Notebook's CPU Profile, Notebook's Memory Profile, จำนวนผู้ใช้งาน และค่าเฉลี่ยการหน่วงเวลาระหว่าง Cell โดยใช้เกณฑ์ชี้วัดเป็น Response Time สำหรับการวัดความแม่นยำที่ได้จากการทำนายของเทคนิคการเรียนรู้ของเครื่อง ผู้วิจัยจะใช้ Mean Absolute Error (MAE) และ Mean Absolute Percentage Error (MAPE) ในการวัดความแม่นยำ ผลลัพธ์ที่ได้แสดงให้เห็นว่า Random Forest Regression Model มีความสามารถในการทำนายประสิทธิภาพของ Jupyter Notebook ที่ทำงานอยู่บน JupyterHub ได้ดีโดยค่า MAE ได้ 13.679 วินาทีและ MAPE ได้ 9.732% และ KNN Regression Model ไม่สามารถทำนายภาพประสิทธิภาพของ Jupyter Notebook ที่ทำงานอยู่บน JupyterHub ได้โดยค่า MAE ได้ 128.907 วินาทีและ MAPE ได้ 95.057%

Sanullah et al. (2021) บทความนี้แนะนำเสนอโครงสร้างของแอปพลิเคชันชื่อ Sign4PSL ซึ่งแปลงประโยคเป็นภาษามือของปากีสถาน (Pakistan Sign Language - PSL) สำหรับคนหูหนวกด้วยการแสดงผลด้วยตัวละครภาพเสมือน งานวิจัยนี้มีจุดมุ่งหมายในการพัฒนาแอปพลิเคชันที่เป็นส่วนตัวและสามารถใช้ซ้ำได้บนแพลตฟอร์มใด ๆ รวมถึงเว็บและมือถือ โดยมีความสามารถในการแปลงข้อความออฟไลน์ได้ โปรแกรม Sign4PSL ฝังพอใจในฐานความรู้ที่เก็บรักษาคำศัพท์ PSL และรหัสของพวกเขาในระบบบรรทัด โปรแกรม Sign4PSL รับข้อความภาษาอังกฤษเป็นข้อมูลนำเข้า ดำเนินการแปลงเป็น PSL ผ่านการรหัสภาษามือและแสดงท่ามือให้ผู้ใช้ด้วยตัวละครเสมือน ระบบถูกทดสอบกับนักเรียนหูหนวกที่โรงเรียนพิเศษ ผลลัพธ์แสดงให้เห็นว่านักเรียนสามารถเข้าใจเรื่องราวที่

เอกสารนี้เป็นเอกสารต้นฉบับที่ส่งมาเพื่อพิจารณาตีพิมพ์ในวารสารวิชาการของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและขอสงวนสิทธิ์ในเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

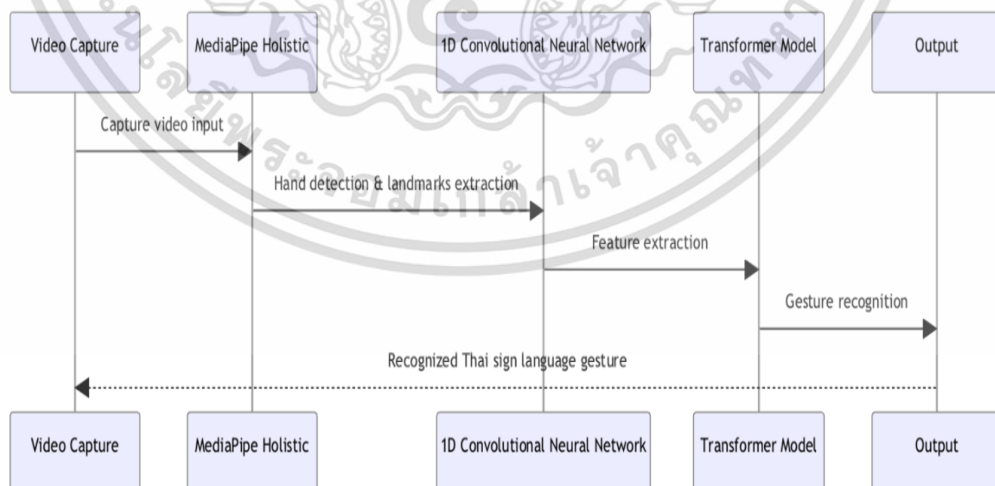
วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงวิธีดำเนินการวิจัยของการสร้างและหลักการทำงานของตัวโมเดล โดยใช้เทคโนโลยีวีทัศน์คอมพิวเตอร์ซึ่งประกอบด้วยกล้องเว็บแคมและคลังโปรแกรมมีเดียเพื่อตรวจจับตำแหน่งของจุดสำคัญบนฝ่ามือและใช้เครือข่ายประสาท Convolutional 1 มิติ (1DCNN) เพื่อเรียนรู้ความสัมพันธ์ระหว่างลำดับของจุดสำคัญบนฝ่ามือกับท่าภาษามือ 1DCNN จะวิเคราะห์ลำดับของพิกัด x,y ของจุดสำคัญบนฝ่ามือเพื่อเปรียบเทียบกับจุดสำคัญที่ตรวจจับได้จากฐานข้อมูล ถ้าภาษามือไทยที่ใช้ “ท่ามือ” เป็นตัวบ่งชี้หลังจากนั้นได้ประมวลผลลัพธ์โดยใช้ชุดคำสั่งภาษาไพทอนเพื่อแสดงผลของคำหรือตัวสะกดที่แปลได้ซึ่งขั้นตอนการดำเนินการวิจัยเป็นออกเป็น 4 ส่วน ดังนี้

3.1 หลักการทำงาน

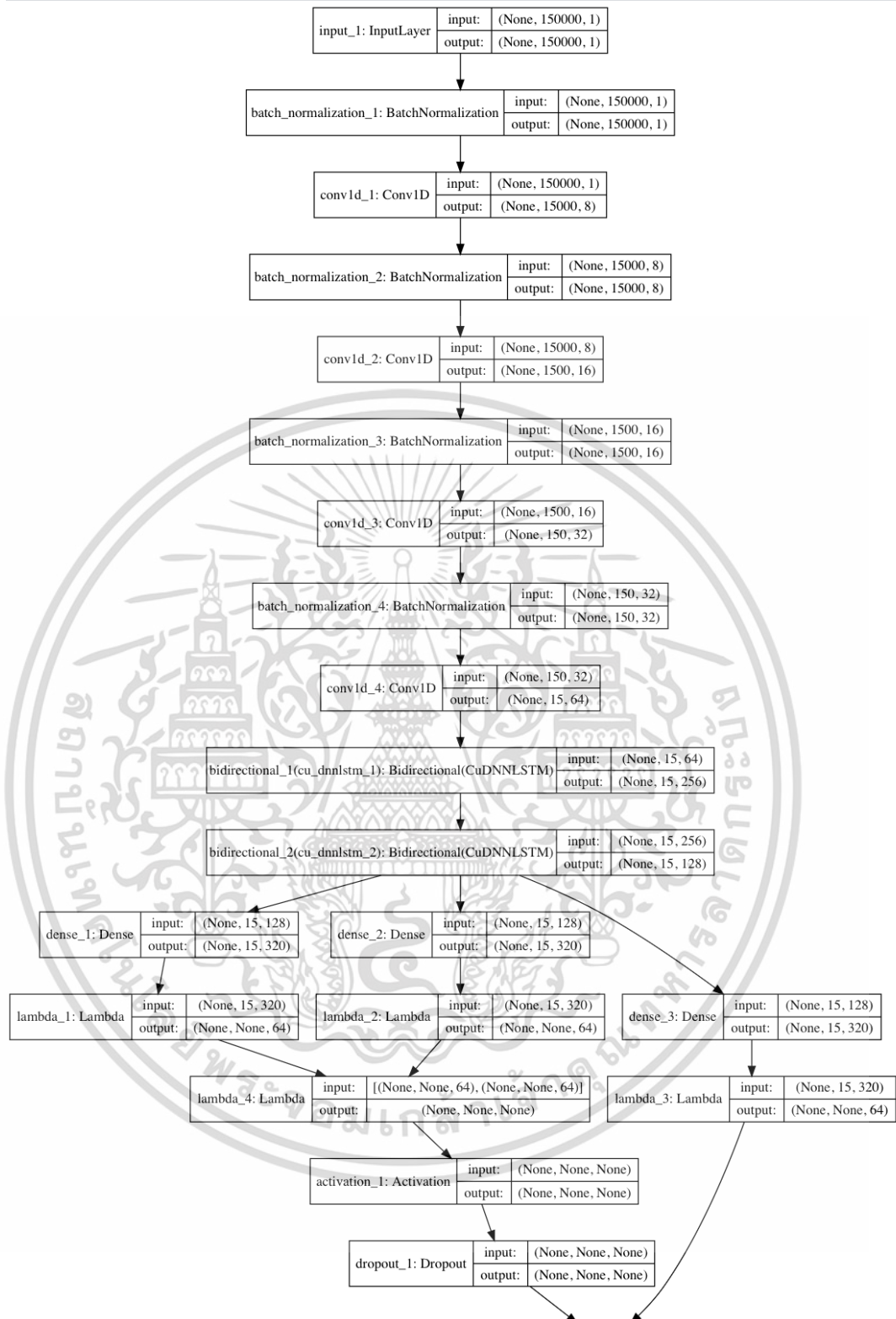
จากรูปที่ 3.1 Sequence Diagram แสดงการทำงานของระบบแปลภาษามือเป็นเสียงพูด โดยจะประกอบด้วย 3 ส่วนหลักคือ ส่วนสร้างระบบจับภาพมือ (Building Phase), ส่วนเรียนรู้โมเดลทำนายภาษามือ (Training Phase) และส่วนทดสอบระบบหรือส่วนผู้ใช้งาน (Testing Phase)

เริ่มต้นด้วยการรับภาพหรือวิดีโอจากกล้องผ่านคำสั่ง OpenCV จากนั้นระบบจะตรวจจับมือโดยใช้ MediaPipe Holistic ซึ่งรวมเอาอัลกอริทึม Mediapipe Detection และ `mp_holistic.Holistic` เข้าด้วยกัน สามารถตรวจจับร่างกายและสร้างเส้นประสาทกับจุดสำคัญบนมือแบบเรียลไทม์ได้ เมื่อได้ตำแหน่งนิ้วมือ 21 จุดแล้ว ข้อมูลเหล่านี้จะถูกนำไปเรียนรู้กับโมเดล Tensorflow ซึ่งจะเรียนรู้ลักษณะเฉพาะของท่าทางภาษามือ เพื่อทำนายว่าผู้ใช้กำลังสื่อคำอะไร สุดท้ายจะแปลงข้อความเป็นเสียงพูดให้ผู้ใช้ได้ยิน



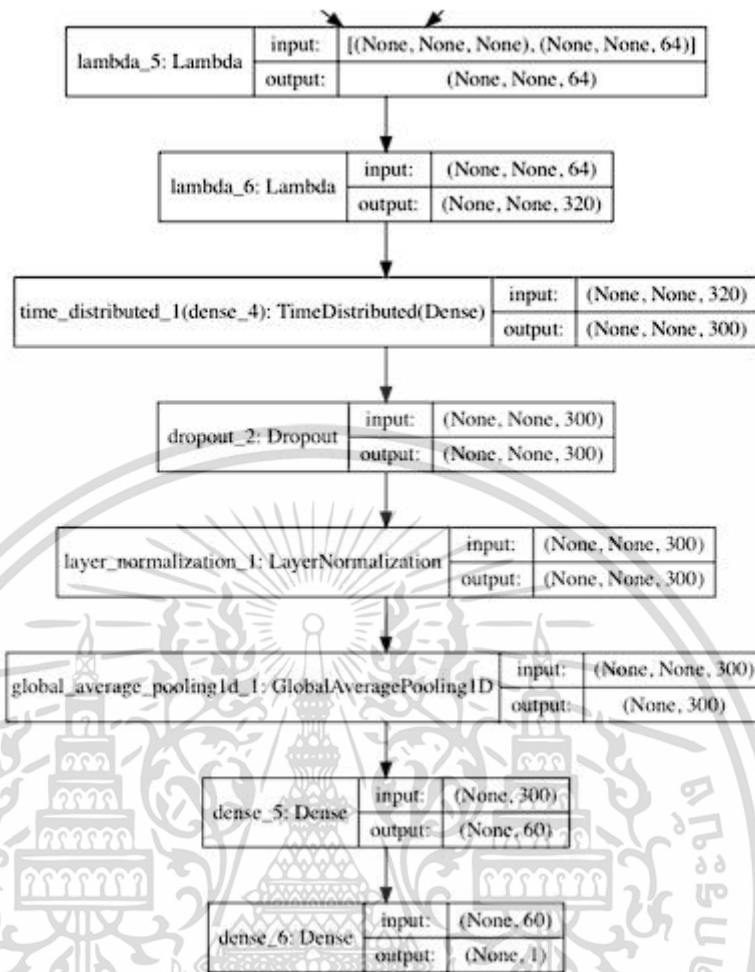
รูปที่ 3.1 Sequence Diagram แสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ไดอะแกรมโมเดล Transformer ร่วมกับ 1D CNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

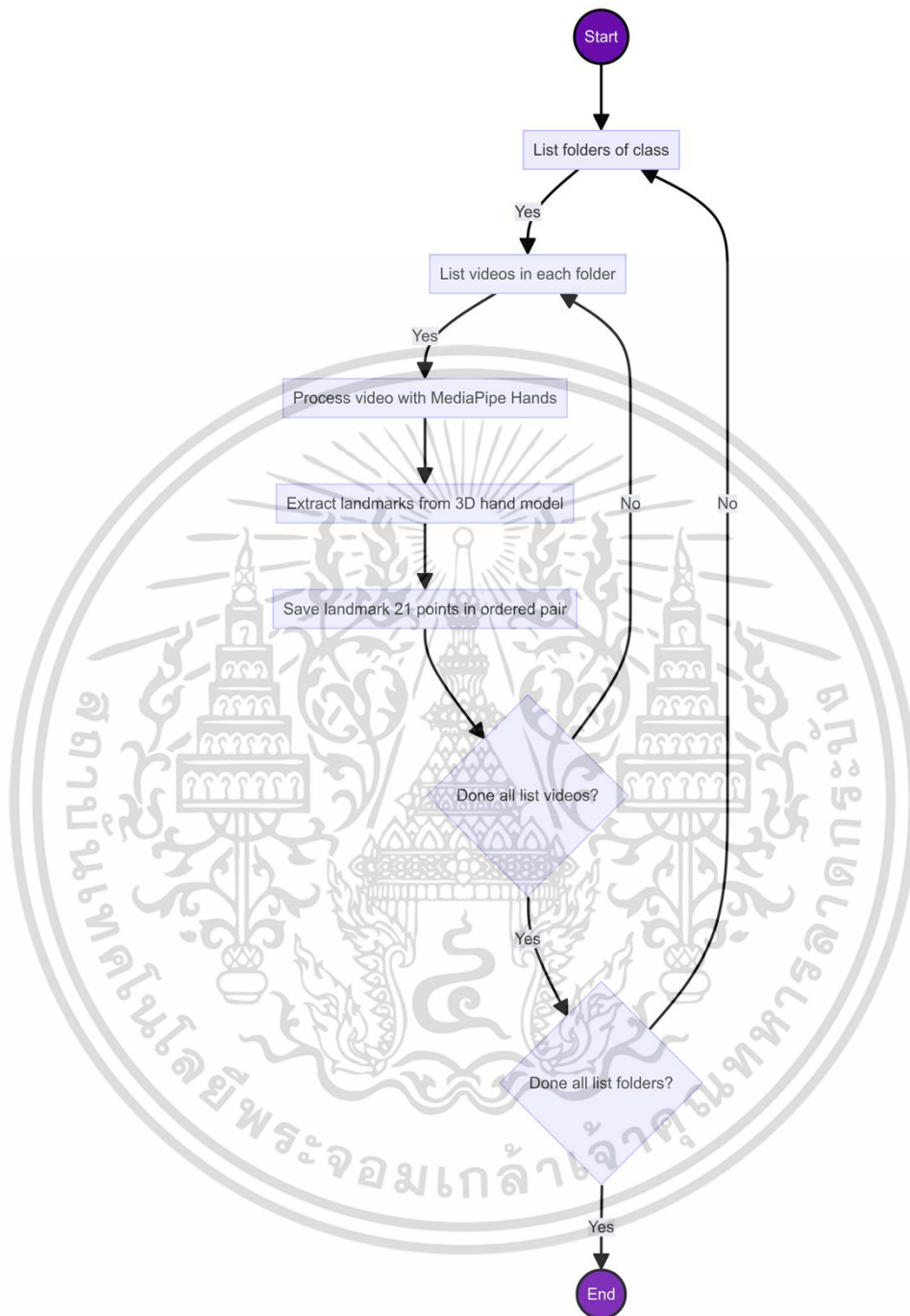


รูปที่ 3.2 ไดอะแกรมโมเดล Transformer ร่วมกับ 1D CNN (ต่อ)

จากรูปที่ 3.2 ไดอะแกรมโมเดลนี้แสดงโครงสร้างโมเดลการเรียนรู้เชิงลึกที่รวมเลเยอร์คอนโวลูชัน 1 มิติหลายชั้น, เลเยอร์ LSTM สองทิศทาง, และเลเยอร์หนาแน่น เลเยอร์คอนโวลูชันสกัดคุณลักษณะเฉพาะจากลำดับอินพุต, เลเยอร์ LSTM จับการพึ่งพาตามเวลา, และเลเยอร์หนาแน่นทำนายผลลัพธ์สุดท้าย เลเยอร์การปรับมาตรฐานและดรอปเอาต์ช่วยให้กระบวนการฝึกฝนมีเสถียรภาพและเป็นปกติมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนของการสร้างระบบตรวจจับมือ (Building Phase)



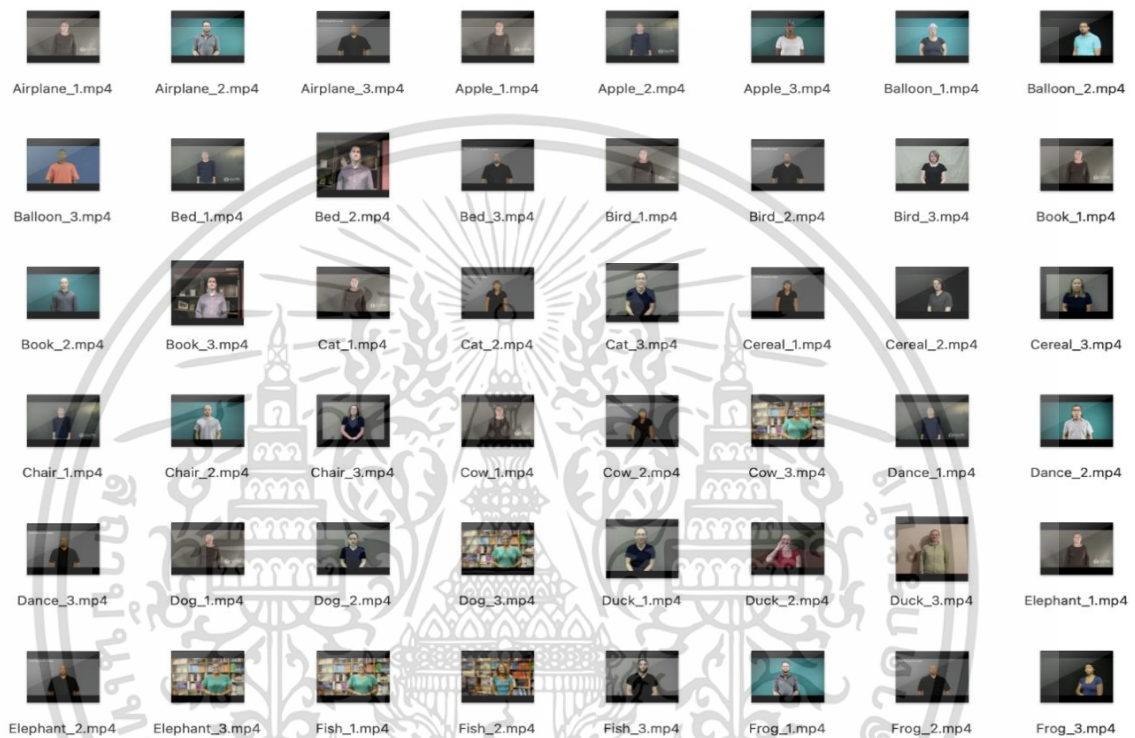
รูปที่ 3.3 Graph Diagram แสดงระบบตรวจจับภาษามือ

จากรูปที่ 3.3 กระบวนการตรวจจับจุดสำคัญบนมือเริ่มต้นด้วยการรวบรวมวิดีโอตัวอย่างท่าทางภาษามือจากโพลเดอร์ประเภทต่างๆ ระบบจะนำวิดีโอเหล่านี้มาประมวลผลทีละรายการ โดยใช้เฟรมเวิร์ก MediaPipe Holistic ซึ่งเป็นเครื่องมือที่มีประสิทธิภาพในการตรวจจับและติดตามตำแหน่งของมือในวิดีโอ

ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่ ไม่มีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตรวจจับมือได้แล้ว ระบบจะสกัดจุดสำคัญ 21 จุด จากโมเดล 3 มิติของมือที่ได้จากการประมวลผล จุดสำคัญเหล่านี้เป็นตำแหน่งบนมือที่สำคัญ เช่น ปลายนิ้ว ข้อนิ้ว และข้อมือ ตำแหน่งของจุดสำคัญจะถูกบันทึกในรูปแบบพิกัดคู่อันดับ (x,y)

เมื่อประมวลผลครบทุกวิดีโอในโฟลเดอร์ปัจจุบัน ระบบจะเปลี่ยนไปประมวลผลโฟลเดอร์ทำทางถัดไป จนกระทั่งครบทุกโฟลเดอร์ ข้อมูลพิกัดจุดสำคัญที่ได้จากทุกวิดีโอจะถูกจัดเก็บเพื่อนำไปใช้ในขั้นตอนการเรียนรู้โมเดลจำแนกทำทางภาษามือต่อไป



รูปที่ 3.4 โฟลเดอร์ข้อมูลทำทางภาษามือ

3.2.1 ชุดข้อมูลที่นำมาฝึกฝนระบบ

วิดีโอที่นำมาใช้นี้มาจากสมาคมคนหูหนวกแห่งประเทศไทย และ Signing Savvy โดยมีความละเอียด, อัตราเฟรม และความยาวที่หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค)

รูปที่ 3.5 (ก) ความละเอียด 600×600 50fps, (ข) ความละเอียด 600×600 50fps, (ค) ความละเอียด 640×360 29.97fps

3.2.2 คำสั่งการดึงข้อมูลค่าตำแหน่งมือและข้อมือ

```
from .config import MAX_LEN, POINT_LANDMARKS
import tensorflow as tf

def tf_nan_mean(x, axis=0, keepdims=False):
    return tf.reduce_sum(tf.where(tf.math.is_nan(x), tf.zeros_like(x), x), axis=axis,
    keepdims=keepdims) / tf.reduce_sum(tf.where(tf.math.is_nan(x), tf.zeros_like(x),
    tf.ones_like(x)), axis=axis, keepdims=keepdims)

def tf_nan_std(x, center=None, axis=0, keepdims=False):

    if center is None:
        center = tf_nan_mean(x, axis=axis, keepdims=True)
    d = x - center
    return tf.math.sqrt(tf_nan_mean(d * d, axis=axis, keepdims=keepdims))

class Preprocess(tf.keras.layers.Layer):
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def __init__(self, max_len=MAX_LEN, point_landmarks=POINT_LANDMARKS,
**kwargs):
    super().__init__(**kwargs)
    self.max_len = max_len
    self.point_landmarks = point_landmarks

def call(self, inputs):

    if tf.rank(inputs) == 3:
        x = inputs[None,...]
    else:
        x = inputs

    mean = tf_nan_mean(tf.gather(x, [17], axis=2), axis=[1,2], keepdims=True)
    mean = tf.where(tf.math.is_nan(mean), tf.constant(0.5,x.dtype), mean)
    x = tf.gather(x, self.point_landmarks, axis=2) #N,T,P,C
    std = tf_nan_std(x, center=mean, axis=[1,2], keepdims=True)
    x = (x - mean)/std

    if self.max_len is not None:
        x = x[:, :self.max_len]
    length = tf.shape(x)[1]
    x = x[...,:2]

    dx = tf.cond(tf.shape(x)[1]>1, lambda:tf.pad(x[:,1:] - x[:, :-1],
[[0,0],[0,1],[0,0],[0,0]]), lambda:tf.zeros_like(x))

    dx2 = tf.cond(tf.shape(x)[1]>2, lambda:tf.pad(x[:,2:] - x[:, :-2],
[[0,0],[0,2],[0,0],[0,0]]), lambda:tf.zeros_like(x))

    x = tf.concat([
        tf.reshape(x, (-1,length,2*len(self.point_landmarks))),
        tf.reshape(dx, (-1,length,2*len(self.point_landmarks))),
        tf.reshape(dx2, (-1,length,2*len(self.point_landmarks))),
    ], axis = -1)

    x = tf.where(tf.math.is_nan(x),tf.constant(0.,x.dtype),x)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 return x
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากชุดคำสั่งที่ 1 ในการประมวลผลข้อมูลสำหรับรู้จำท่าทางภาษามือด้วย TensorFlow นั้น จำเป็นต้องจัดการกับข้อมูลที่อาจมีค่าขาดหายไปหรือไม่สมบูรณ์ ซึ่งได้ใช้ฟังก์ชัน `tf_nan_mean` และ `tf_nan_std` เพื่อคำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของเทนเซอร์โดยอัตโนมัติละเว้นค่า NaN ที่อาจปรากฏในข้อมูล ส่วนขั้น `Preprocess` ซึ่งเป็นขั้นที่สร้างขึ้นเองใน TensorFlow Keras ทำหน้าที่ประมวลผลข้อมูล Input ล่วงหน้าโดยการปรับค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน เพื่อให้ได้ข้อมูลที่มีคุณภาพดีขึ้นสำหรับการฝึกอบรมและทำนายผลของโมเดล โดยเฉพาะสำหรับงานที่ใช้ลำดับจุดสัญญาณ (Point Landmark Sequences) แบบภาษามือไทย



รูปที่ 3.6 การตรวจจับท่าทางมือ และใบหน้า

จากรูปที่ 3.6 นำเสนอผลลัพธ์จากการใช้ MediaPipe ในการตรวจจับและทำเครื่องหมายตำแหน่งสำคัญบนใบหน้าและมือของผู้ใช้งาน ซึ่งเป็นพื้นฐานสำคัญของกระบวนการรู้จำภาษามือ ระบบจะวิเคราะห์ภาพจากกล้องแบบเรียลไทม์ และใช้ MediaPipe ซึ่งเป็นไลบรารี Machine learning สำหรับการประมวลผลภาพและวิดีโอ ในการระบุตำแหน่งสำคัญบนใบหน้า เช่น มุมตา ปลายจมูก มุมปาก และขอบใบหน้า รวมถึงตำแหน่งสำคัญบนมือ เช่น ปลายนิ้ว ข้อนิ้ว รอยพับ และรูปทรงของฝ่ามือตำแหน่งสำคัญเหล่านี้ จะถูกทำเครื่องหมายบนภาพด้วยจุดสีฟ้า และเชื่อมต่อกันเป็นโครงร่าง เพื่อแสดงให้เห็นถึงโครงสร้างของใบหน้าและมืออย่างชัดเจนข้อมูลตำแหน่งสำคัญที่ได้ ซึ่งแทนด้วยค่าพิกัด (x, y) ของแต่ละจุดจะถูกป้อนเข้าสู่โมเดลการรู้จำภาษามือ เพื่อนำไปใช้ในการแปลความหมายของท่าทางมือต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 คำสั่งพิกัดตำแหน่งฝ่ามือ, ตา, จมูก, ปาก

```

import numpy as np

THRESH_HOLD = 0.5

SEQ_LEN = 30
ROWS_PER_FRAME = 543
MAX_LEN = 384
CROP_LEN = MAX_LEN
NUM_CLASSES = 250
PAD = -100.

NOSE=[
    1,2,98,327
]
LNOSE = [98]
RNOSE = [327]
LIP = [ 0,
    61, 185, 40, 39, 37, 267, 269, 270, 409,
    291, 146, 91, 181, 84, 17, 314, 405, 321, 375,
    78, 191, 80, 81, 82, 13, 312, 311, 310, 415,
    95, 88, 178, 87, 14, 317, 402, 318, 324, 308,
]
LLIP = [84,181,91,146,61,185,40,39,37,87,178,88,95,78,191,80,81,82]
RLIP = [314,405,321,375,291,409,270,269,267,317,402,318,324,308,415,310,311,312]

POSE = [500, 502, 504, 501, 503, 505, 512, 513]
LPOSE = [513,505,503,501]
RPOSE = [512,504,502,500]

REYE = [
    33, 7, 163, 144, 145, 153, 154, 155, 133,
    246, 161, 160, 159, 158, 157, 173,
]
LEYE = [
    263, 249, 390, 373, 374, 380, 381, 382, 362,
    466, 388, 387, 386, 385, 384, 398,
]

LHAND = np.arange(468, 489).tolist()
RHAND = np.arange(522, 543).tolist()

POINT_LANDMARKS = LIP + LHAND + RHAND + NOSE + REYE + LEYE #+POSE

NUM_NODES = len(POINT_LANDMARKS)
CHANNELS = 6*NUM_NODES
if __name__ == "__main__":
    #ปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

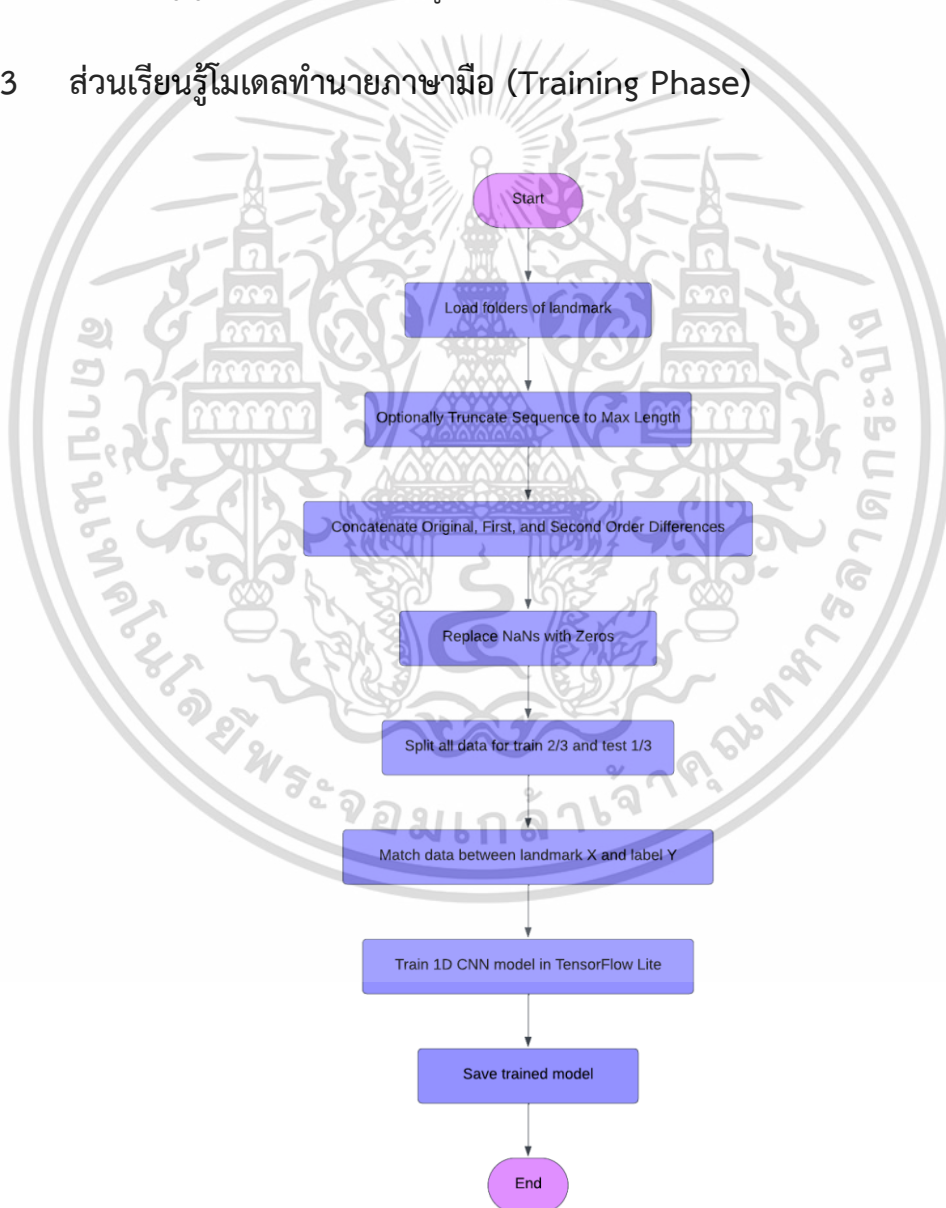
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งชื่อผู้จัดทำเอกสารนี้ไว้ที่หน้าแรกของเอกสารทุกครั้งที่มีการนำไปใช้

```
print(NUM_NODES)
print(CHANNELS)
```

ชุดคำสั่งที่ 3.2.3 นี้เป็นการเตรียมข้อมูลสำหรับการประมวลผลภาพใบหน้าโดยใช้จุดสังเกต (Landmark) บนใบหน้า โดยมีการกำหนดค่าต่าง ๆ เช่น ค่า Threshold สำหรับการตัดสินใจ, จำนวนเฟรมในลำดับ, จำนวนจุด Landmark ในแต่ละเฟรม, ความยาวสูงสุดของข้อมูลที่จะใช้, จำนวนคลาสที่จะจำแนก, และค่าสำหรับการเติมข้อมูลให้เต็ม นอกจากนี้ยังมีการกำหนดดัชนีของจุด Landmark ต่าง ๆ บนใบหน้า เช่น จมูก, ริมฝีปาก, ตา, และมือ โดยแยกตามตำแหน่งซ้าย-ขวา จากนั้นจึงรวมดัชนีของจุด Landmark ที่สนใจทั้งหมด และคำนวณจำนวนจุด Landmark และจำนวนช่องสัญญาณของข้อมูล สุดท้ายโค้ดจะพิมพ์ค่าจำนวนจุด Landmark และจำนวนช่องสัญญาณออกมาเมื่อไฟล์นี้ถูกรันโดยตรง

3.3 ส่วนเรียนรู้โมเดลทำนายภาษามือ (Training Phase)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น **รูปที่ 3.7 Block Diagram แสดงระบบการเรียนรู้โมเดลทำนายภาษามือ**

จากรูปที่ 3.7 ผังงานนี้แสดงขั้นตอนการประมวลผลข้อมูล landmark โดยเริ่มจากการโหลดข้อมูล จำนวนค่าความแตกต่างอันดับที่ 1 และ 2 รวมข้อมูลทั้งหมดเข้าด้วยกัน จัดการค่า NaN (Not a Number) โดยแทนที่ด้วยเลขศูนย์ แบ่งข้อมูลออกเป็นชุดสำหรับการเทรนและการทดสอบ จับคู่ข้อมูล landmark (X) กับ label (Y) ที่เกี่ยวข้อง ทำการเทรนโมเดล 1D Convolutional Neural Network (CNN) โดยใช้ TensorFlow Lite และสุดท้ายบันทึกโมเดลที่เทรนเสร็จแล้ว

3.3.1 คำสั่งการเตรียมข้อมูลตำแหน่งบนฝ่ามือและข้อนิ้วมือไปใช้ในการเรียนรู้โมเดล

```
from .utils import Preprocess
from .config import MAX_LEN, CHANNELS, NUM_CLASSES
import tensorflow as tf
```

```
class ECA(tf.keras.layers.Layer):
    def __init__(self, kernel_size=5, **kwargs):
        super().__init__(**kwargs)
        self.supports_masking = True
        self.kernel_size = kernel_size
        self.conv = tf.keras.layers.Conv1D(1, kernel_size=kernel_size, strides=1,
padding="same", use_bias=False)
```

```
    def call(self, inputs, mask=None):
        nn = tf.keras.layers.GlobalAveragePooling1D()(inputs, mask=mask)
        nn = tf.expand_dims(nn, -1)
        nn = self.conv(nn)
        nn = tf.squeeze(nn, -1)
        nn = tf.nn.sigmoid(nn)
        nn = nn[:,None,:]
        return inputs * nn
```

```
class LateDropout(tf.keras.layers.Layer):
    def __init__(self, rate, noise_shape=None, start_step=0, **kwargs):
        super().__init__(**kwargs)
        self.supports_masking = True
        self.rate = rate
        self.start_step = start_step
        self.dropout = tf.keras.layers.Dropout(rate, noise_shape=noise_shape)
```

```
    def build(self, input_shape):
        super().build(input_shape)
        agg = tf.VariableAggregation.ONLY_FIRST_REPLICA
        self._train_counter = tf.Variable(0, dtype="int64", aggregation=agg,
trainable=False)
```

```
    def call(self, inputs, training=False):
        x = tf.cond(self._train_counter < self.start_step, lambda:inputs,
lambda:self.dropout(inputs, training=training))
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return x

class CausalDWConv1D(tf.keras.layers.Layer):
    def __init__(self,
                 kernel_size=17,
                 dilation_rate=1,
                 use_bias=False,
                 depthwise_initializer='glorot_uniform',
                 name="", **kwargs):
        super().__init__(name=name, **kwargs)
        self.causal_pad = tf.keras.layers.ZeroPadding1D((dilation_rate*(kernel_size-
1),0),name=name + '_pad')
        self.dw_conv = tf.keras.layers.DepthwiseConv1D(
            kernel_size,
            strides=1,
            dilation_rate=dilation_rate,
            padding='valid',
            use_bias=use_bias,
            depthwise_initializer=depthwise_initializer,
            name=name + '_dwconv')
        self.supports_masking = True

    def call(self, inputs):
        x = self.causal_pad(inputs)
        x = self.dw_conv(x)
        return x

def Conv1DBlock(channel_size,
               kernel_size,
               dilation_rate=1,
               drop_rate=0.0,
               expand_ratio=2,
               se_ratio=0.25,
               activation='swish',
               name=None):
    if name is None:
        name = str(tf.keras.backend.get_uid("mbblock"))
    def apply(inputs):
        channels_in = tf.keras.backend.int_shape(inputs)[-1]
        channels_expand = channels_in * expand_ratio
        skip = inputs
        x = tf.keras.layers.Dense(
            channels_expand,
            use_bias=True,
            activation=activation,
            name=name + '_expand_conv')(inputs)

```

เอกสารนี้เป็นเอกสารของ CausalDWConv1D(kernel_size, dilation_rate=dilation_rate, ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นแต่กรณีเห็นชอบและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        use_bias=False,
        name=name + '_dwconv')(x)

    x = tf.keras.layers.BatchNormalization(momentum=0.95, name=name + '_bn')(x)

    x = ECA()(x)

    x = tf.keras.layers.Dense(
        channel_size,
        use_bias=True,
        name=name + '_project_conv')(x)

    if drop_rate > 0:
        x = tf.keras.layers.Dropout(drop_rate, noise_shape=(None,1,1), name=name
+ '_drop')(x)

    if (channels_in == channel_size):
        x = tf.keras.layers.add([x, skip], name=name + '_add')
    return x

return apply

class MultiHeadSelfAttention(tf.keras.layers.Layer):
    def __init__(self, dim=256, num_heads=4, dropout=0, **kwargs):
        super().__init__(**kwargs)
        self.dim = dim
        self.scale = self.dim ** -0.5
        self.num_heads = num_heads
        self.qkv = tf.keras.layers.Dense(3 * dim, use_bias=False)
        self.drop1 = tf.keras.layers.Dropout(dropout)
        self.proj = tf.keras.layers.Dense(dim, use_bias=False)
        self.supports_masking = True

    def call(self, inputs, mask=None):
        qkv = self.qkv(inputs)
        qkv = tf.keras.layers.Permute((2, 1, 3))(tf.keras.layers.Reshape((-1,
self.num_heads, self.dim * 3 // self.num_heads))(qkv))
        q, k, v = tf.split(qkv, [self.dim // self.num_heads] * 3, axis=-1)
        attn = tf.matmul(q, k, transpose_b=True) * self.scale

        if mask is not None:
            mask = mask[:, None, None, :]

        attn = tf.keras.layers.Softmax(axis=-1)(attn, mask=mask)
        attn = self.drop1(attn)

        x = attn @ v

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาและการวิจัยเท่านั้น
 ไม่ควรนำเอกสารนี้ไปเผยแพร่หรือใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร

```

return x

def TransformerBlock(dim=256, num_heads=4, expand=4, attn_dropout=0.2,
drop_rate=0.2, activation='swish'):
    def apply(inputs):
        x = inputs
        x = tf.keras.layers.BatchNormalization(momentum=0.95)(x)
        x =
MultiHeadSelfAttention(dim=dim,num_heads=num_heads,dropout=attn_dropout)(x)
        x = tf.keras.layers.Dropout(drop_rate, noise_shape=(None,1,1))(x)
        x = tf.keras.layers.Add()([inputs, x])
        attn_out = x

        x = tf.keras.layers.BatchNormalization(momentum=0.95)(x)
        x = tf.keras.layers.Dense(dim*expand, use_bias=False, activation=activation)(x)
        x = tf.keras.layers.Dense(dim, use_bias=False)(x)
        x = tf.keras.layers.Dropout(drop_rate, noise_shape=(None,1,1))(x)
        x = tf.keras.layers.Add()([attn_out, x])
        return x
    return apply

class TFLiteModel(tf.Module):
    def __init__(self, islr_models):
        super(TFLiteModel, self).__init__()
        self.prep_inputs = Preprocess()
        self.islr_models = islr_models

    @tf.function(input_signature=[tf.TensorSpec(shape=[None, 543, 3],
dtype=tf.float32, name='inputs')])
    def __call__(self, inputs):
        x = self.prep_inputs(tf.cast(inputs, dtype=tf.float32))
        outputs = [model(x) for model in self.islr_models]
        outputs = tf.keras.layers.Average()(outputs)[0]
        return {'outputs': outputs}

def get_model(max_len=MAX_LEN, dropout_step=0, dim=192):
    inp = tf.keras.Input((max_len,CHANNELS))
    x = inp
    ksize = 17

    # Stem layers
    x = tf.keras.layers.Dense(dim, use_bias=False,name='stem_conv')(x)
    x = tf.keras.layers.BatchNormalization(momentum=0.95,name='stem_bn')(x)

    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = TransformerBlock(dim,expand=2)(x)

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นผู้ที่มีเหตุอันสมควรและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
x = TransformerBlock(dim,expand=2)(x)

if dim == 384: #for the 4x sized model
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = TransformerBlock(dim,expand=2)(x)

    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = Conv1DBlock(dim,ksize,drop_rate=0.2)(x)
    x = TransformerBlock(dim,expand=2)(x)

x = tf.keras.layers.Dense(dim*2,activation=None,name='top_conv')(x)
x = tf.keras.layers.GlobalAveragePooling1D()(x)
x = LateDropout(0.8, start_step=dropout_step)(x)
x = tf.keras.layers.Dense(NUM_CLASSES,name='classifier', activation = 'softmax')(x)
return tf.keras.Model(inp, x)

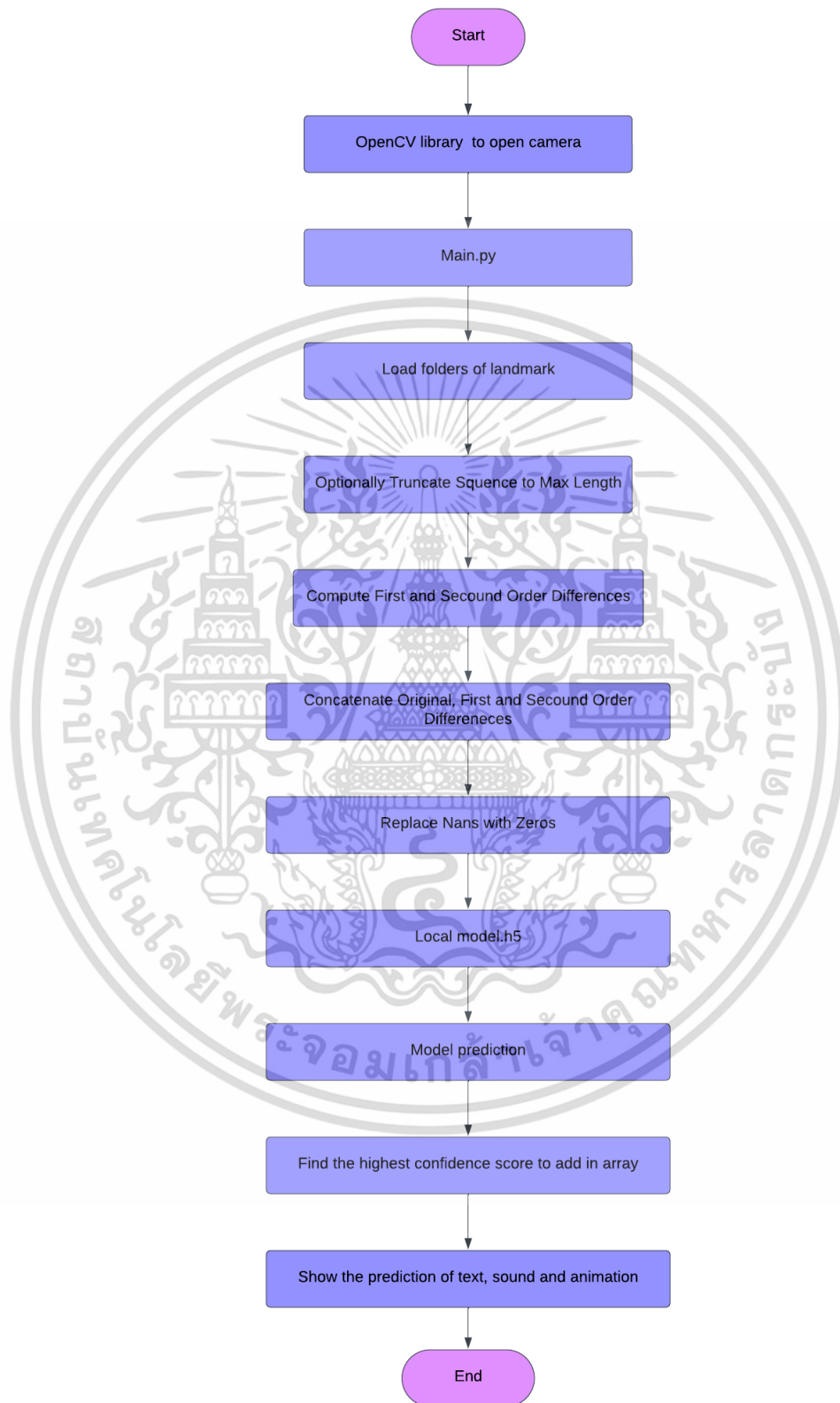
```

จากชุดคำสั่งที่ 3.3.1 เป็นการใช้โครงสร้างของโมเดล Transformer เพื่อการประมวลผลลำดับข้อมูลที่มีคุณลักษณะเฉพาะจากภาษามือ โมเดลดังกล่าวถูกออกแบบมาเพื่อรับมือกับข้อมูลที่มีค่าว่างหรือ NaN ที่อาจปรากฏในชุดข้อมูล แทนที่จะปรับขนาดเฟรมวิดีโอให้เท่ากัน โมเดลจะคำนวณค่าเฉลี่ยและ ส่วนเบี่ยงเบนมาตรฐานโดยละเว้นค่า NaN ทำให้โมเดลสามารถจัดการกับชุดข้อมูลที่มีความยาวแตกต่างกันได้โดยไม่สูญเสียข้อมูลที่สำคัญ ข้อมูลถูกนำมาฝึกสอน (Train) และทดสอบ (Test) โดยใช้วิธีการแบ่งอัตราส่วน 2 ต่อ 3 สำหรับการฝึกสอน และ 1 ต่อ 3 สำหรับการทดสอบ เพื่อประเมินประสิทธิภาพของโมเดลอย่างเหมาะสม

โครงข่ายคอนโวลูชันแบบหนึ่งมิติ (1D Convolutional Neural Network) หรือ 1DCNN ถูกประยุกต์ใช้สร้างโมเดลในการจำแนกภาษามือที่มีลำดับการเคลื่อนไหวของมือในท่าทางต่างๆ โมเดลที่สร้างขึ้นนี้ประกอบด้วยชั้นต่าง ๆ มากมายที่ทำงานร่วมกัน เช่น ชั้นเชื่อมต่อเต็มรูปแบบ (Dense Layer) ที่มีฟังก์ชันกระตุ้นเป็นซอฟต์แวร์ (Softmax) สำหรับการจำแนกหมวดหมู่ของข้อมูล โดยใช้ categorical_crossentropy เป็นฟังก์ชันการสูญเสียในการปรับปรุงน้ำหนักของโครงข่ายให้สามารถจำแนกข้อมูลได้อย่างแม่นยำ ประกอบด้วยชั้นของการคำนวณความสนใจแบบเอนกประสงค์ (Multi-Head Self-Attention) และ Transformer Block ที่ช่วยให้โมเดลสามารถจับคู่ลักษณะเฉพาะของข้อมูลที่มีลำดับและเรียนรู้ความสัมพันธ์ระยะไกลภายในข้อมูลได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนทดสอบระบบหรือส่วนผู้ใช้งาน (Testing Phase)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 3.8 Block Diagram แสดงระบบจำแนกภาษามือเป็นข้อความ และเสียง นำไปใช้

จากรูปที่ 3.8 แผนภาพนี้แสดงถึงขั้นตอนการทำงานของระบบรู้จำภาษามือ เริ่มต้นด้วยการเรียกใช้ไลบรารี OpenCV เพื่อเปิดใช้งานกล้อง จากนั้น ระบบจะทำการโหลดไฟล์ main.py ซึ่งเป็นโปรแกรมหลัก และทำการโหลดไฟล์เตอร์ ที่เก็บข้อมูลตำแหน่งสำคัญบนมือ (Landmark)

ข้อมูลตำแหน่งเหล่านี้ อาจถูกตัดทอน (Truncate) ให้มีความยาวไม่เกินค่าสูงสุดที่กำหนด ก่อนที่ระบบจะคำนวณค่าความแตกต่าง (Difference) อันดับที่หนึ่ง และอันดับที่สอง ของข้อมูล จากนั้น นำข้อมูลทั้งสามส่วน (ข้อมูลดิบ, ค่าความแตกต่างอันดับที่ 1 และค่าความแตกต่างอันดับที่ 2) มารวมกัน หากพบค่า NaN (Not a Number) ในข้อมูล ระบบจะแทนที่ด้วยค่าศูนย์ ก่อนที่โมเดล (Model.h5) จะถูกโหลดขึ้นมาเพื่อทำนายผล หลังจากโมเดลทำนายผลลัพธ์แล้ว ระบบจะค้นหาค่าความเชื่อมั่น (Confidence Score) ที่สูงที่สุด และเพิ่มเข้าไปในอาร์เรย์ สุดท้าย ระบบจะแสดงผลลัพธ์ของการทำนาย ในรูปแบบของข้อความ เสียง และแอนิเมชัน ก่อนจะสิ้นสุดการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

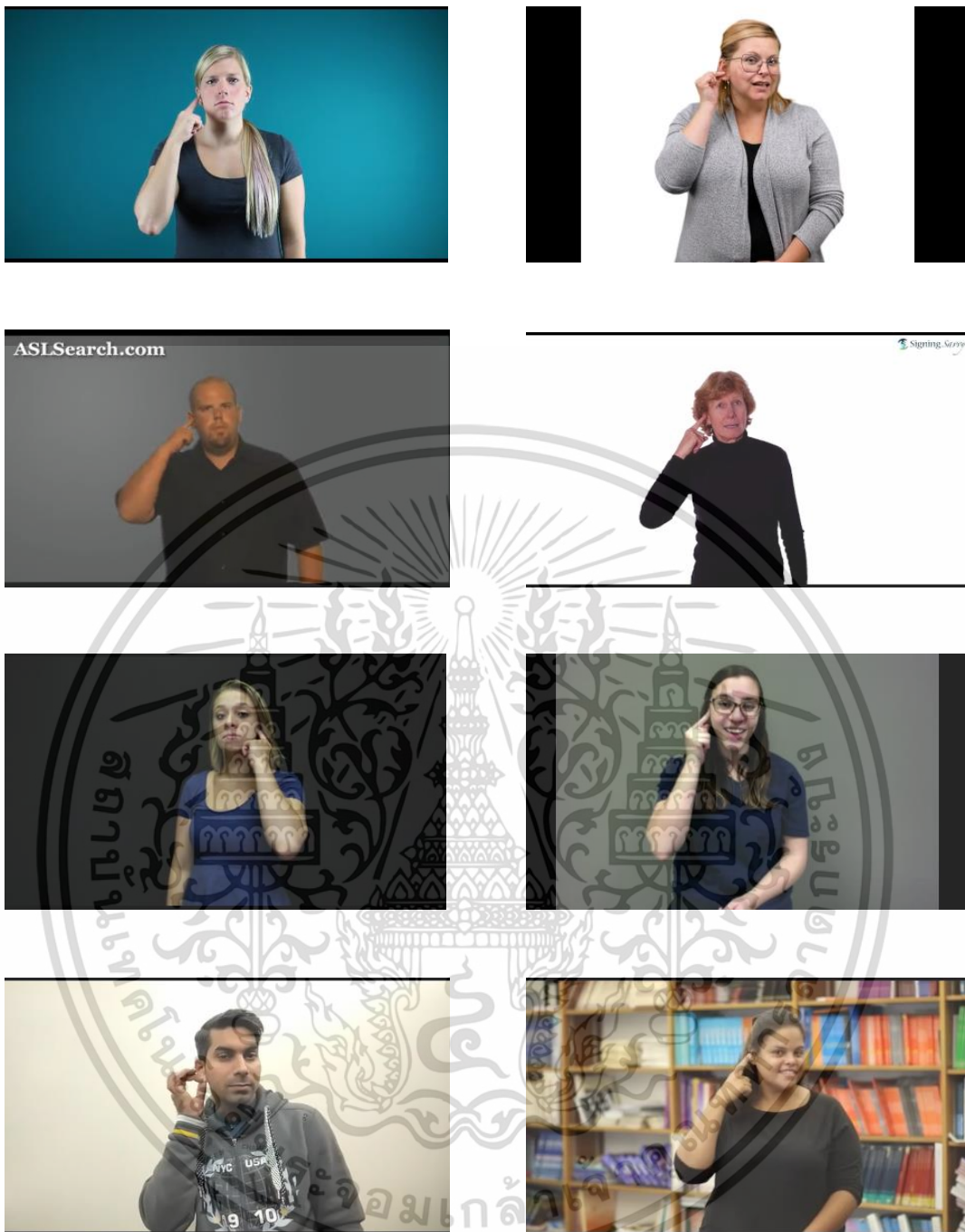
ผลการวิจัยและการอภิปรายผล

ในการวิจัยครั้งนี้ ผู้ศึกษาได้ประยุกต์ใช้เทคโนโลยีวีทัศน์คอมพิวเตอร์ โดยใช้กล้องเว็บแคม ร่วมกับคลังโปรแกรมมีเดียไฟฟ์ เพื่อตรวจจับตำแหน่งของจุดสำคัญบนฝ่ามือ และนำข้อมูลที่ได้ไปประมวลผลด้วยโครงข่ายประสาทเทียม Convolutional 1 มิติ (1D CNN) ซึ่งทำหน้าที่เรียนรู้ความสัมพันธ์ระหว่างลำดับของจุดสำคัญบนฝ่ามือกับท่าภาษามือ โดย 1D CNN จะวิเคราะห์ลำดับของพิกัด x, y ของจุดสำคัญ เพื่อเปรียบเทียบกับฐานข้อมูลท่าภาษามือไทย โดยใช้ "ท่ามือ" เป็นตัวบ่งชี้ จากนั้นจึงประมวลผลผลลัพธ์อีกครั้งโดยใช้ชุดคำสั่งภาษาไพทอน เพื่อแสดงผลลัพธ์เป็นคำศัพท์หรือตัวสะกดที่แปลได้

จากการทดสอบพบว่าระบบสามารถแปลคำศัพท์และตัวสะกดได้อย่างรวดเร็ว สามารถตรวจจับการเคลื่อนไหวของท่ามือได้ทันที และมีความแม่นยำในการแปลอยู่ที่ร้อยละ 94.663

4.1 การเตรียมข้อมูลเพื่อทดสอบความแม่นยำ

ผู้วิจัยได้เลือกใช้ชุดข้อมูลภาษามือไทยที่จะมาทำการทดสอบความแม่นยำ ประกอบไปด้วยคำศัพท์ทั้งหมด 19 คำ จาก 30 คำ ของชุดข้อมูลที่นำมาทดลอง เนื่องจาก 19 คำนี้ เป็นคำที่กักทายเป็นเบื้องต้นและใช้ในชีวิตประจำวัน และมีท่าที่ซับซ้อนและไว ประกอบด้วยคำดังต่อไปนี้ "สวัสดี" "กระหายน้ำ" "ขอบคุณ" "หุ" "ตา" "คาง" "โปรด" "คิด" "จมูก" "ผู้ชาย" "ผู้หญิง" "เจ็บ" "เท่าไร" "คุณ" "ฉันทัน" "ชื่อ" "สกปรก" "สวย" และ "หล่อ" โดยได้บันทึกคลิปวิดีโอท่าทางภาษามือจำนวน 30 คลิปต่อคำศัพท์แต่ละคำ และนำเข้าสู่ระบบ โดยระบบจะทำการจับท่าทางจากวิดีโอ และนำไปประมวลผลเพื่อเปรียบเทียบกับฐานข้อมูล เพื่อทดสอบความแม่นยำในการแปลความหมายของท่าทางภาษามือ



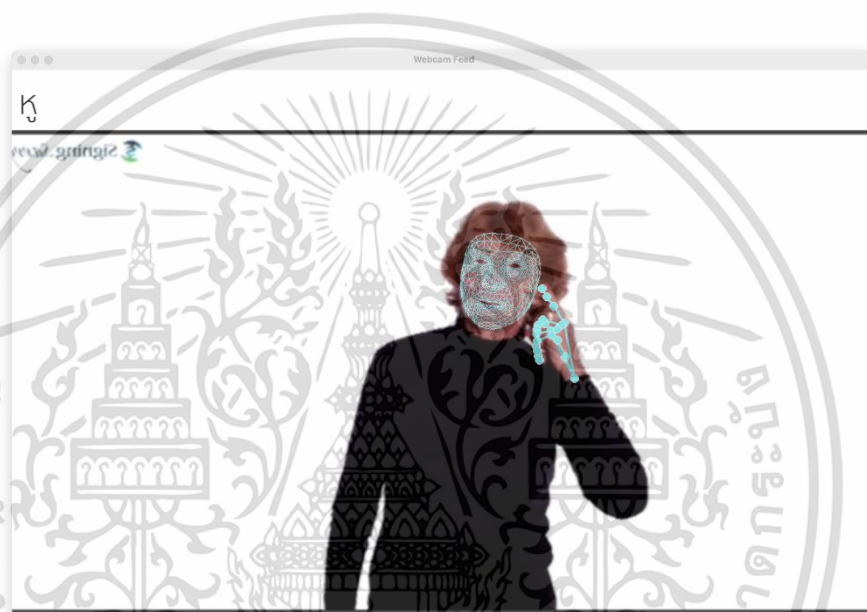
รูปที่ 4.1 ชุดข้อมูลวิดีโอเพื่อทดสอบความแม่นยำภาษามือไทย “หู”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 แสดงตัวอย่างชุดวิดีโอเพื่อทดสอบความแม่นยำภาษามือสำหรับคำว่า "หู" ซึ่งเป็นหนึ่งในคำศัพท์ที่ใช้ในการทดสอบระบบรู้จำภาษามือ โดยรูปภาพแสดงให้เห็นถึงความหลากหลายของการแสดงท่าทาง "หู" ในภาษามือ ซึ่งอาจแตกต่างกันในด้านของมุมกล้อง, บุคคล, เพศ, และลักษณะท่าทางประกอบ เช่น การแสดงออกทางสีหน้า

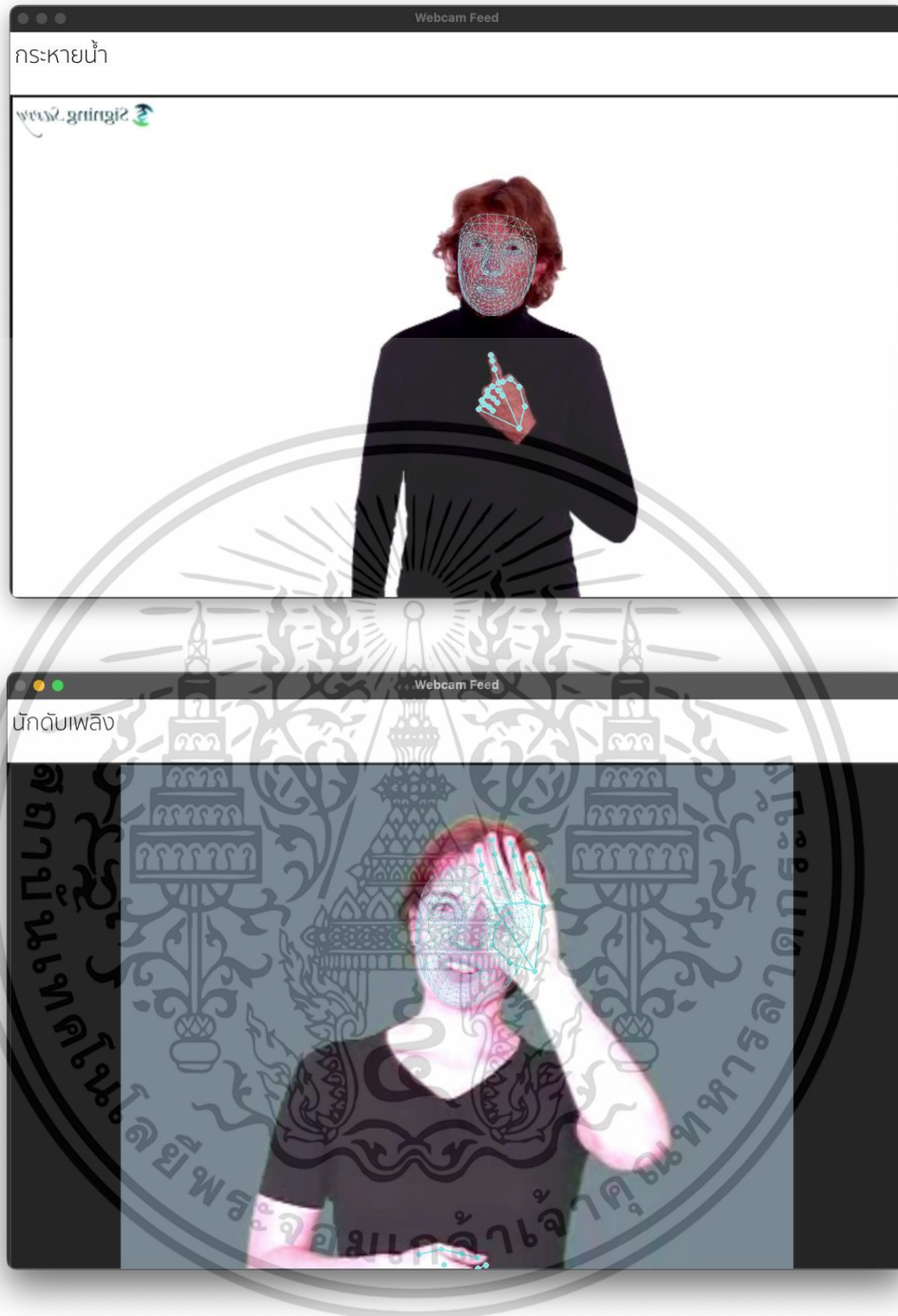
ตัวอย่างเช่น บางภาพแสดงให้เห็นถึงการใช้มือเดียวสัมผัสที่หู ในขณะที่บางภาพใช้สองมือสัมผัสที่หูทั้งสองข้าง นอกจากนี้ ตำแหน่งของมือและนิ้ว ก็อาจแตกต่างกันไปในแต่ละบุคคล

ความหลากหลายของท่าทางภาษามือ ดังที่แสดงในรูปที่ 4.1 เป็นความท้าทายอย่างหนึ่งในการพัฒนาระบบรู้จำภาษามือ เพราะระบบต้องสามารถแยกแยะและทำความเข้าใจท่าทางที่หลากหลายเหล่านี้ได้อย่างถูกต้อง



รูปที่ 4.2 ตัวอย่างโปรแกรมตรวจจับคลิปวิดีโอที่นำมาทดสอบความแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ตัวอย่างโปรแกรมตรวจจับคลิปวิดีโอที่นำมาทดสอบความแม่นยำ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sign-language - Python main3 copyyy thissssss2.py - 80x24
Detected word: หู
Start detecting word
Detected word: หู
Start detecting word
Detected word: หู
Start detecting word
Detected word: หู
Start detecting word
No word detected
Sentence reset
Start detecting word
No word detected
Start detecting word
No word detected
Start detecting word
No word detected
Start detecting word
Detected word: หู
Start detecting word
No word detected
Sentence reset
Start detecting word
Detected word: หู

```

รูปที่ 4.3 Console Logs ของการตรวจจับท่าทางมือ

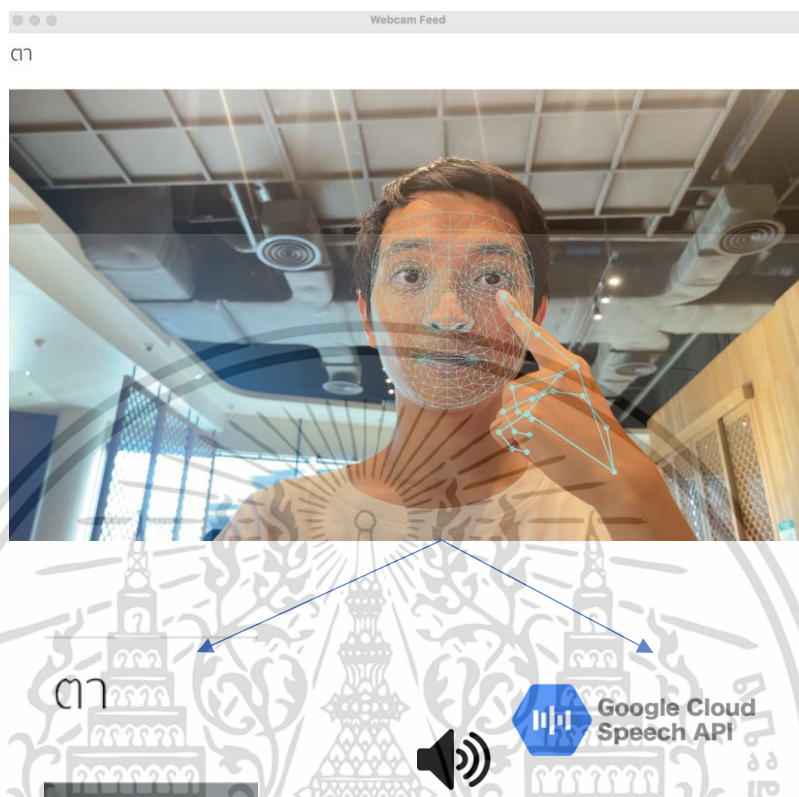
รูปที่ 4.2 แสดงให้เห็นถึงอินเทอร์เฟซของระบบรู้จำภาษามือ โดยในภาพจะเห็นได้ว่าระบบกำลังประมวลผลท่าทางมือของผู้ใช้งาน ซึ่งแสดงผลเป็นโครงร่างจุดสีฟ้า ซ้อนทับบนภาพวิดีโอ จุดสีฟ้าเหล่านี้ แสดงถึงตำแหน่งสำคัญของมือ เช่น ปลายนิ้ว ข้อนิ้ว และฝ่ามือ ซึ่งระบบจะนำไปใช้เป็นข้อมูล ในการวิเคราะห์และแปลความหมายของภาษามือ

รูปที่ 4.3 แสดงผลลัพธ์การประมวลผลของระบบ ในรูปแบบของข้อความ (Console Logs) ซึ่งแสดงให้เห็นถึงสถานะการทำงานของระบบ เช่น "Start detecting word" (เริ่มต้นการตรวจจับคำ) ซึ่งหมายความว่า ระบบกำลังเริ่มต้นกระบวนการ ในการวิเคราะห์ลำดับท่าทางมือ เพื่อระบุคำศัพท์ และ "Detected word: '...'" (ตรวจพบคำว่า '...') ซึ่งหมายความว่า ระบบได้ทำการวิเคราะห์ท่าทางมือ และประมวลผลออกมาเป็นคำศัพท์เรียบร้อยแล้ว ภาพทั้งสองนี้ แสดงให้เห็นถึงการทำงานของระบบ ตั้งแต่การรับข้อมูลภาพ การประมวลผลท่าทาง ไปจนถึงการแสดงผลในรูปแบบที่เข้าใจง่าย ซึ่งสอดคล้องกับวัตถุประสงค์ของระบบที่มุ่งอำนวยความสะดวกในการสื่อสารระหว่างผู้บกพร่องทางการได้ยิน กับบุคคลทั่วไป โดยระบบสามารถตรวจจับและแปลความหมายของภาษามือ ได้อย่างถูกต้องและมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบระบบ

4.2.1 การแสดงผลของระบบ



รูปที่ 4.4 ตัวอย่างหน้าต่างผู้ใช้งานแอปพลิเคชัน

จากรูปที่ 4.4 แสดงตัวอย่างการใช้งานของระบบจำแนกภาษามือ ซึ่งได้รับการออกแบบให้เรียบง่าย และใช้งานง่าย เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้อย่างสะดวก ดังรูปคือคำว่า "ตา" ระบบจะแสดงผลเป็น 2 ส่วน คือ ส่วนที่แสดงคำแปลของภาษามือเป็นแถบข้อความด้านซ้าย จากทางด้านบน และ ส่วนที่เป็นเสียงพูดออกมาจากคำแปลที่ได้ ส่วนหน้าจอผู้ใช้งานนี้ จะแสดงผลแบบเรียลไทม์ โดยแสดงภาพจากกล้อง พร้อมกับผลลัพธ์การประมวลผลของระบบ ซึ่งแสดงผลเป็นโครงร่างจุดสีฟ้า ซ้อนทับบนภาพวิดีโอ จุดสีฟ้าเหล่านี้ แสดงถึงตำแหน่งสำคัญบนใบหน้า เช่น ดวงตา จมูก และริมฝีปาก และแสดงถึงตำแหน่งสำคัญบนมือ เช่น ปลายนิ้ว ข้อนิ้ว และฝ่ามือ การแสดงผลตำแหน่งสำคัญบนใบหน้าและมือ ช่วยให้ผู้ใช้งาน สามารถตรวจสอบความถูกต้อง ของการตรวจจับของระบบได้ทันที ซึ่งเป็นประโยชน์ สำหรับการปรับตำแหน่งมือและใบหน้า ให้เหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

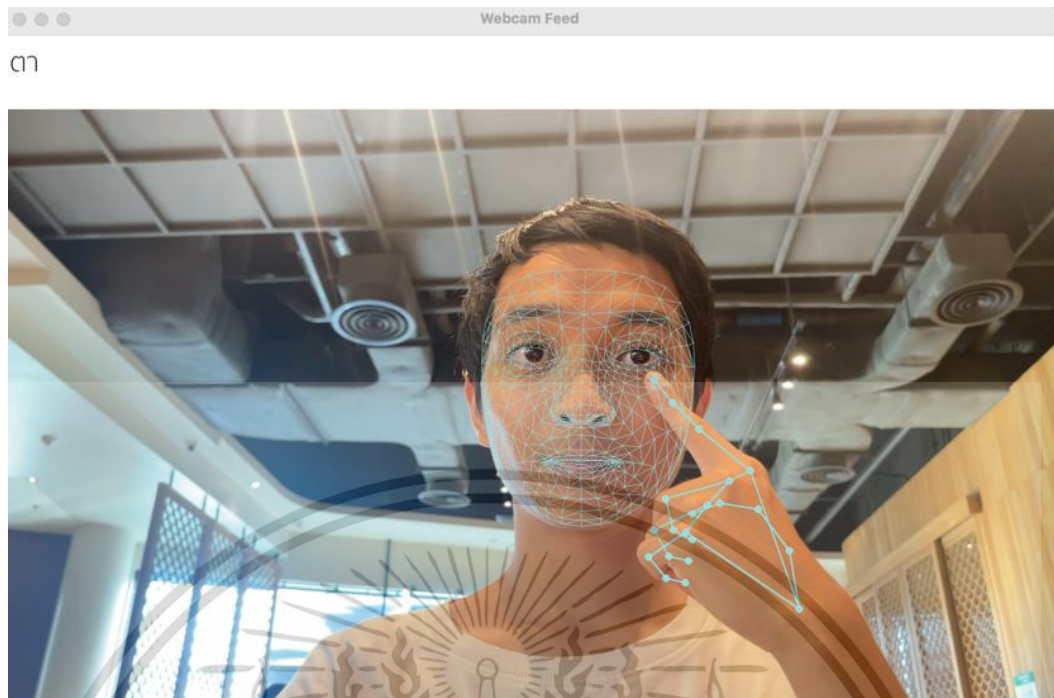
4.2.2 ผลการทดสอบประสิทธิภาพของระบบ

ตารางที่ 4.1 ตารางผลการทดสอบประสิทธิภาพของระบบ

คำแปลภาษามือ	วิดีโอทั้งหมด	การทำนายถูกต้อง	ร้อยละ (%)
สวัสดี	30	29	96.66
กระหายน้ำ	30	28	93.33
ขอบคุณ	30	28	93.33
หู	30	30	100
ตา	30	30	100
คาง	30	29	96.66
โปรด	30	26	86.66
คิด	30	30	100
จุมก	30	30	100
ผู้ชาย	30	29	96.66
ผู้หญิง	30	29	96.66
เจ็บ	30	25	83.33
เท่าไร	30	26	86.66
คุณ	30	28	93.33
ฉัน	30	30	100
ชื่อ	30	28	93.33
สกปรก	30	29	96.66
สวย	30	28	93.33
หล่อ	30	29	96.66

ตารางที่ 4.1 แสดงผลการทดสอบประสิทธิภาพของระบบจำแนกภาษามือ พบว่าระบบที่พัฒนาขึ้นนี้ มีความสามารถในการแปลภาษามือไทยได้อย่างถูกต้องในระดับที่น่าพอใจ โดยมีความแม่นยำอยู่ที่ 94.663% โดยคำที่ระบบสามารถแปลทำทางออกมาได้อย่างถูกต้อง 100% มี 5 คำ คือ "หู" "ตา" "จุมก" "คิด" "ฉัน" เนื่องจากเป็นท่าทางที่ง่ายและมีความชัดเจน เช่นคำว่า "หู" "ตา" "จุมก" การแสดงท่าทางคือการชี้ไปที่จุดนั้นจึงทำให้ระบบสามารถแปลผลออกมาได้ถูกต้องทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ตัวอย่างภาษามือคำว่า “ตา”

ต่อมาคือคำที่ระบบสามารถแปลผลได้ถูกต้องระดับปานกลางที่ 96.66% และ 93.33% โดยมีทั้งหมด 11 คำ คือ "สกปรก" "หล่อ" "ผู้ชาย" "ผู้หญิง" "คาง" "สวีสวี" "กระหายน้ำ" "ขอบคุณ" "คุณ" "ชื่อ" "สวย" ซึ่งคำเหล่านี้ระบบก็สามารถแปลผลออกมาได้อย่างถูกต้องเช่นกันแต่ก็ไม่ถูกทั้งหมดเนื่องจากคำเหล่านี้มีท่าทางที่แตกต่างจาก 5 คำแรกที่มีการแสดงท่าทางเป็นการชี้ไปที่จุดนั้น แต่ 11 คำนี้จะเริ่มใช้ท่าทางที่มีการขยับมือในการทำ ยกตัวอย่างเช่นคำว่า "กระหายน้ำ" เป็นท่าที่นำนิ้วชี้ไปที่คอและลากนิ้วลงมาที่อก เนื่องจากมีท่าทางที่ต้องขยับมือในการทำ จึงจำเป็นต้องทำท่าทางให้ถูกต้องระบบจะสามารถแปลผลออกมาได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีรูปที่ 4.6 ตัวอย่างภาษามือคำว่า “กระหายน้ำ” สารทุกครั้งที่มีการนำไปใช้

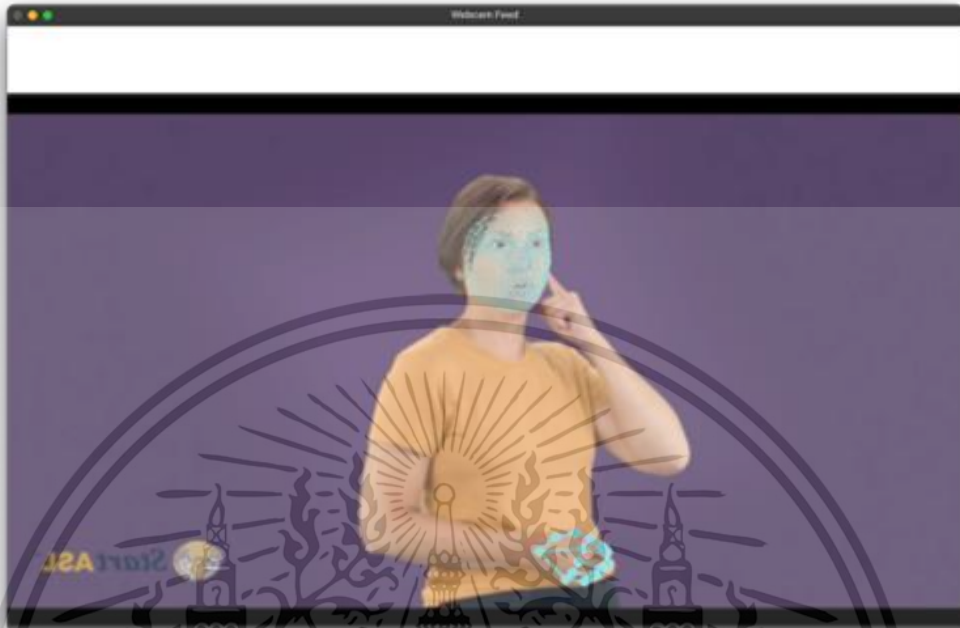
และสุดท้ายคือคำที่ระบบแปลผลได้ถูกต้องระดับน้อยที่ 86.66% และ 83.33% โดยมีทั้งหมด 3 คำ คือ "โปรด" "เท่าไร" "เจ็บ" คำเหล่านี้จะเหมือนกับ 11 คำ ที่มีการใช้ท่าทางขยับมือในการทำ แต่ก็ยังมีหลายปัจจัยที่ทำให้ 3 คำนี้แปลผลออกมาได้น้อยกว่าคำอื่น เช่น คำว่า "โปรด" จะมีท่าทางที่คล้ายกับคำว่าของฉัน คือ การที่นำมือไปแนบที่หน้าอก ซึ่งคำว่าโปรดจะเป็นการที่นำมือแนบไปที่อก และลูบวนให้เป็นวงกลมทำให้ระบบอาจเกิดความสับสนในการแปลได้ จากที่กล่าวมาข้างต้น ปัจจัยที่ทำให้ระบบไม่สามารถแปลผลออกมาได้อาจเป็นเพราะการแสดงท่าทางที่ไม่ถูกต้อง หรือ ท่าทางไม่เหมือนกับคลิปวิดีโอที่นำมาเทรนในระบบ แต่ก็ยังมีอีกหลายปัจจัยที่ทำให้ระบบไม่สามารถแปลผลออกมาได้ เช่น การเอียงของมุมกล้อง ตำแหน่งการยืนในการแสดงท่าทาง หรือการหันฝ่ามือไม่ตรงกับกล้อง ซึ่งอาจทำให้ระบบเกิดความสับสนในการจำแนกได้เช่นกัน



รูปที่ 4.7 ตัวอย่างภาษามือคำว่า “โปรด”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอียงมือ การเอียงของมุมกล้องที่ต่างจากข้อมูลที่ฝึกฝนระบบ (รูปที่ 4.10, 4.11) หรือการหันฝ่ามือ ซึ่งอาจทำให้ระบบเกิดความสับสนในการจำแนกได้เช่นกัน



รูปที่ 4.10 ปัญหาการตรวจจับท่าทางมือด้านมุมข้าง



รูปที่ 4.11 ปัญหาการตรวจจับท่าทางมือด้านมุมต่ำ

เพื่อเพิ่มประสิทธิภาพของระบบจำแนกภาษามือ ให้สามารถแปลความหมายได้อย่างถูกต้อง และแม่นยำมากยิ่งขึ้น ควรพิจารณาแนวทางในการปรับปรุงหลายด้าน เช่น การเพิ่มปริมาณข้อมูล เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อประโยชน์ทางการศึกษาเท่านั้น เมื่อผู้เขียนได้ศึกษาเอกสารนี้แล้วพบว่ากรณีศึกษาที่น่าสนใจอีกทั้งยังมีประเด็นที่ควรศึกษาและต้องคำนึงถึงใจของเอกสารที่ควรนำมาใช้

สำหรับการฝึกสอนโมเดล ยังมีข้อมูลสำหรับการฝึกสอนมากเท่าไร โมเดลก็จะยังสามารถเรียนรู้และแยกแยะท่าทางภาษามือได้อย่างแม่นยำมากขึ้นเท่านั้น โดยเฉพาะอย่างยิ่ง ควรเพิ่มข้อมูลที่ครอบคลุมความหลากหลายในการแสดงท่าทางของคำแปลแต่ละคำ อีกทั้งการปรับปรุงเทคนิคการตรวจจับตำแหน่งสำคัญบนมือ ก็เป็นสิ่งสำคัญ โดยอาจพิจารณาใช้เทคนิคหรือโมเดลที่สามารถตรวจจับตำแหน่งสำคัญบนมือได้อย่างละเอียดและแม่นยำมากยิ่งขึ้น เช่น การใช้ 3D Hand Pose Estimation นอกจากนี้ การพิจารณาข้อมูลบริบทในการแปลความหมาย ก็เป็นสิ่งที่ควรพิจารณา โดยระบบอาจใช้ข้อมูลบริบท เช่น ท่าทางใบหน้า หรือการเคลื่อนไหวร่างกาย ประกอบการวิเคราะห์ เพื่อช่วยในการแปลความหมายของท่าทางมือ ได้อย่างแม่นยำยิ่งขึ้น อย่างไรก็ตามระบบที่พัฒนาขึ้นนี้ยังคงมีข้อจำกัดในเรื่องของความสามารถในการรู้จำคำศัพท์ภาษามือที่อยู่นอกเหนือจากชุดข้อมูลที่ใช้ในการฝึกสอน และประสิทธิภาพของระบบยังคงขึ้นอยู่กับปัจจัยภายนอก เช่น สภาพแสง และอัตราเฟรมของกล้อง ถึงแม้ว่าระบบจำแนกภาษามือที่พัฒนาขึ้นจะยังคงมีข้อจำกัด แต่ผลการทดสอบแสดงให้เห็นถึงศักยภาพในการนำไปประยุกต์ใช้เพื่อช่วยเหลือผู้บกพร่องทางการได้ยินในการสื่อสารกับบุคคลทั่วไป ซึ่งการปรับปรุงและพัฒนาตามแนวทางที่กล่าวมา จะช่วยให้ระบบมีความแม่นยำและมีประสิทธิภาพมากยิ่งขึ้นในอนาคต

ตารางที่ 4.2 ตารางผลการทดสอบประสิทธิภาพของระบบ LSTM

คำแปลภาษามือ	วิธีโอทั้งหมด	การทำนายถูกต้อง	ร้อยละ (%)
สวัสดี	30	27	90.00
กระหายน้ำ	30	26	86.66
ขอบคุณ	30	27	90.00
หุ	30	28	93.33
ตา	30	30	100.00
คาง	30	27	90.00
โปรด	30	24	80.00
คิด	30	28	93.33
จมูก	30	30	100.00
ผู้ชาย	30	27	90.00
ผู้หญิง	30	27	90.00
เจ็บ	30	23	76.66
เท่าไร	30	25	83.33
คุณ	30	26	86.66
ฉัน	30	28	93.33

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ตามการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ตารางผลการทดสอบประสิทธิภาพของระบบ LSTM (ต่อ)

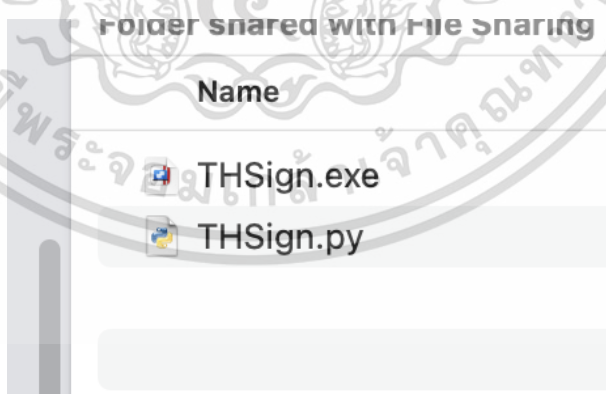
คำแปลภาษา มือ	วิดีโอทั้งหมด	การทำงาน ถูกต้อง	ร้อยละ (%)
ชื่อ	30	26	86.66
สกปรก	30	27	90.00
สวย	30	26	86.66
หล่อ	30	27	90.00

จากตารางที่ 4.2 ซึ่งแสดงผลการทดสอบประสิทธิภาพของระบบจำแนกภาษามือ พบว่าระบบที่ใช้ LSTM นี้ มีความสามารถในการแปลภาษามือไทยได้อย่างถูกต้องในระดับที่น่าพอใจ โดยมีความแม่นยำอยู่ที่ 89.82% ซึ่งเมื่อเทียบกับระบบ 1D CNN + Transformer ที่มีความแม่นยำอยู่ที่ 94.66% แสดงให้เห็นว่าระบบ LSTM ยังมีประสิทธิภาพในการแปลภาษามือในระดับที่ดี แต่ยังคงมีความแม่นยำน้อยกว่า 1D CNN + Transformer

โดยคำที่ระบบ LSTM สามารถแปลท่าทางออกมาได้อย่างถูกต้อง 100% มี 2 คำ คือ "ตา" และ "จมูก" เนื่องจากเป็นท่าทางที่ง่ายและมีความชัดเจน เช่นคำว่า "ตา" และ "จมูก" การแสดงท่าทางคือการชี้ไปที่จุดนั้นจึงทำให้ระบบสามารถแปลผลออกมาได้ถูกต้องทั้งหมด สำหรับคำว่า "หู" "คิด" และ "ฉัน" ระบบสามารถแปลได้ถูกต้อง 28 ครั้งจากทั้งหมด 30 ครั้ง

4.3 ขั้นตอนการใช้งานแอปพลิเคชัน

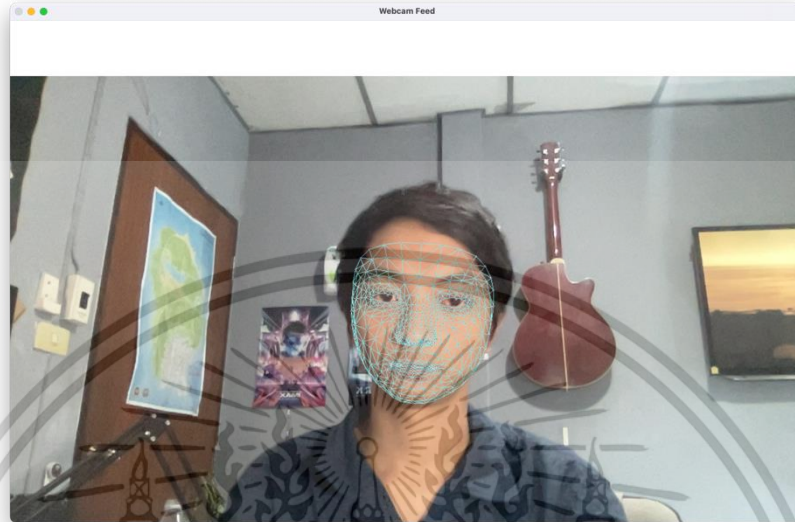
1. ตัวไฟล์แอปพลิเคชัน สามารถเรียกใช้ได้ด้วยไฟล์หลักสองไฟล์นี้ (รูปที่ 4.12): THSign.exe: ไฟล์สำหรับเรียกใช้งานแอปพลิเคชัน บนระบบปฏิบัติการ Windows และ THSign.py: ไฟล์สำหรับเรียกใช้งานแอปพลิเคชัน บนทุกระบบปฏิบัติการที่ทำการติดตั้ง Python ไว้



รูปที่ 4.12 ตัวไฟล์แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อเริ่มต้นใช้งานแอปพลิเคชัน (รูปที่ 4.13) ระบบจะเปิดใช้งานกล้อง และแสดงภาพแบบเรียลไทม์ ในขั้นตอนนี้ระบบจะทำการตรวจจับใบหน้าของผู้ใช้งาน และแสดงผลเป็นโครงร่างจุดสำคัญ (Landmark) บนใบหน้า เช่น มุมตา ปลายจมูก และ มุมปาก



รูปที่ 4.13 หน้าต่างแอปพลิเคชันหลังเปิดเรียกใช้งาน

3. เมื่อผู้ใช้แสดงท่าทางภาษามือ ระบบจะทำการตรวจจับมือจากกล้อง และแสดงผลเป็นโครงร่างจุดสำคัญบนมือ เช่น ปลายนิ้ว ข้อนิ้ว และ ฝ่ามือ พร้อมกันนี้ ระบบจะทำการแปลความหมายของท่าทางภาษามือ และแสดงผลลัพธ์เป็นข้อความและทำการเล่นเสียงพูดภาษาไทย ดังในรูปที่ 4.14 แสดงคำว่า "สวัสดี"



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามรูปที่ 4.14 หน้าต่างแอปพลิเคชันแสดงผลลัพธ์การแปลทุกครั้งที่มีการนำไปใช้

4.4 อภิปรายผล

ในการจำแนกภาษามือ โมเดล 1D CNN ที่เสริมด้วยสถาปัตยกรรม Transformer แสดงให้เห็นถึงประสิทธิภาพที่โดดเด่นในด้านความเร็วในการประมวลผล โดยเฉพาะอย่างยิ่งเมื่อต้องจัดการกับลำดับข้อมูลที่ยาวขึ้น ความสามารถในการประมวลผลแบบขนานของทั้ง 1D CNN และ Transformers ช่วยให้สามารถวิเคราะห์ส่วนต่าง ๆ ของลำดับข้อมูลนำเข้าได้พร้อมกัน ซึ่งส่งผลให้การประมวลผลรวดเร็วและมีประสิทธิภาพ นอกจากนี้ กลไก Attention ที่ Transformer ใช้ก็ยังช่วยเพิ่มประสิทธิภาพในการคำนวณ โดยการโฟกัสไปที่ส่วนสำคัญของข้อมูลนำเข้า ยิ่งไปกว่านั้น การมีอยู่ของการปรับแต่งสำหรับ 1DCNN และ Transformers บนฮาร์ดแวร์ GPU ยิ่งเสริมความได้เปรียบในด้านความเร็ว แม้ว่าโมเดล LSTM จะมีประสิทธิภาพในงานประมวลผลแบบลำดับที่ไม่ยาวมาก แต่ในกรณีของการแปลภาษามือแบบเรียลไทม์ ที่ต้องการความเร็วในการตอบสนอง 1DCNN ที่เสริมด้วย Transformer จึงเป็นทางเลือกที่เหมาะสมกว่า อย่างไรก็ตาม การเปรียบเทียบประสิทธิภาพของโมเดล ควรพิจารณาถึงบริบทเฉพาะ เช่น ความยาวของลำดับข้อมูล และรายละเอียดของการใช้งาน เช่น ฟังก์ชันการกระตุ้น และจำนวนชั้น

4.4.1 การเปรียบเทียบผลลัพธ์ประสิทธิภาพในการตรวจจับ

จากการทดสอบ 5 ครั้งกับระบบที่ใช้ 1DCNN และ Transformer ด้วยการทำท่าทางคำว่า “โปรด” ซึ่งเป็นคำที่มีท่าทางเป็นลำดับ โดยหมุนมือเบาที่หน้าอกแล้ววน หลังจากระบบจับท่าทางมือได้ ระบบจะทำการบันทึกรูปภาพพร้อมเวลา ซึ่งมีผลลัพธ์เฉลี่ยอยู่ที่ 1.47 วินาที ดังรูปที่ 4.14



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

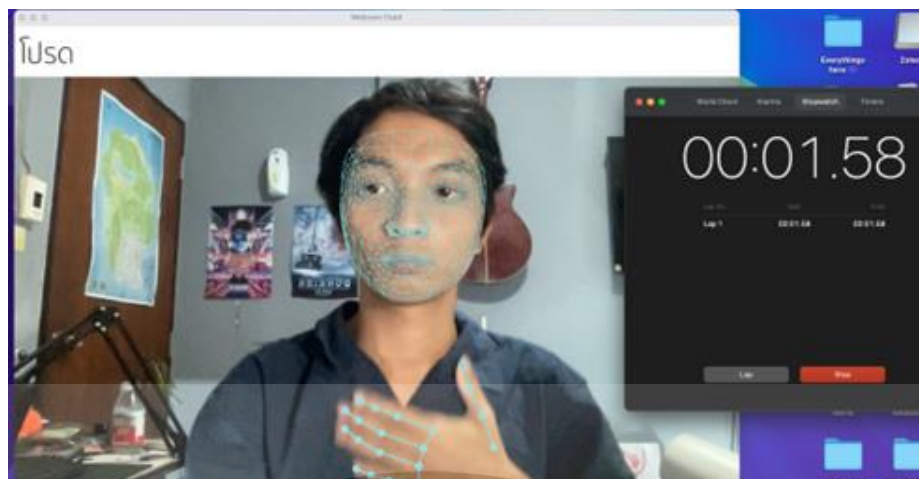


(ค)



(ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(จ)

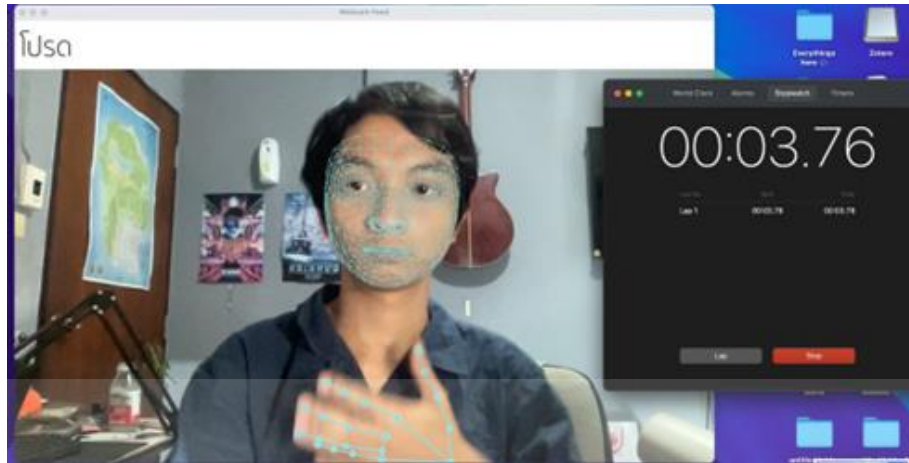
รูปที่ 4.15 ผลลัพธ์และเวลาการแปลภาษามือไทยคำว่า “โปรด” ด้วย 1DCNN และ Transformer (ก) ได้ผลลัพธ์ที่ 1.71 วินาที, (ข) ได้ผลลัพธ์ที่ 1.29 วินาที, (ค) ได้ผลลัพธ์ที่ 1.60 วินาที, (ง) ได้ผลลัพธ์ที่ 1.21 วินาที และ (จ) ได้ผลลัพธ์ที่ 1.58 วินาที

จากการทดสอบ 5 ครั้งกับระบบที่ใช้ LSTM ด้วยการทำท่าทางคำว่า “โปรด” ซึ่งเป็นคำที่มีท่าทางเป็นลำดับ โดยหมุนมือเบาที่หน้าอกแล้ววน หลังจากระบบจับท่าทางมือได้ ระบบจะทำการบันทึกรูปภาพพร้อมเวลา ซึ่งมีผลลัพธ์เฉลี่ยอยู่ที่ 3.35 วินาที ดังรูปที่ 4.15

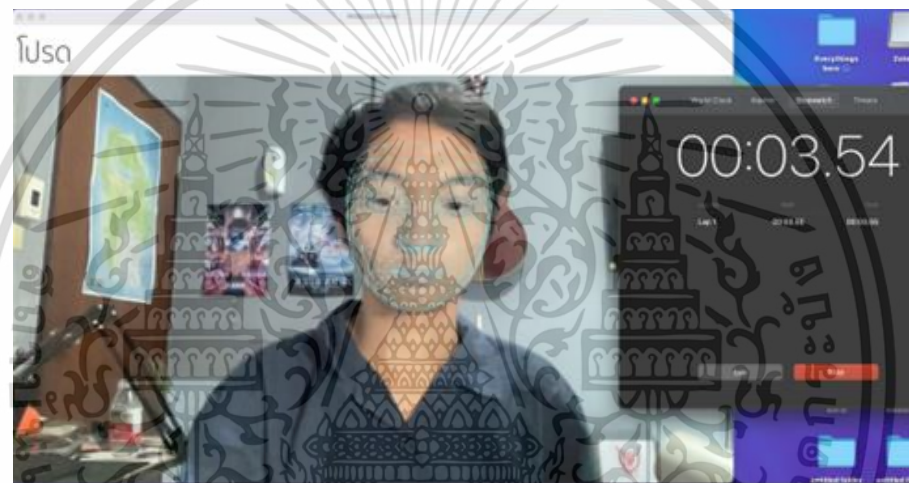


(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)



(ค)



(ง)

รูปที่ 4.16 ผลลัพธ์และเวลาการแปลภาษามือไทยคำว่า “โปรด” ด้วย LSTM
 เอกสารนี้เรื่อง... (ก) ได้ผลลัพธ์ที่ 3. วินาที, (ข) ได้ผลลัพธ์ที่ 1.29 วินาที, (ค) ได้ผลลัพธ์ที่ 1.60 วินาที, (ง) ได้ผลลัพธ์ที่ 1.21 วินาที และ (จ) ได้ผลลัพธ์ที่ 1.58 วินาที

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาแอปพลิเคชันคอมพิวเตอร์แปลภาษามือเป็นเสียงพูด เพื่อช่วยลดช่องว่างทางการสื่อสารระหว่างผู้ที่มีบกพร่องทางการได้ยินกับคนปกติ โดยที่ผู้วิจัยได้ใช้เทคโนโลยีวิทัศน์คอมพิวเตอร์ซึ่งประกอบด้วยกล้องเว็บแคมและคลังโปรแกรมมีเดียไพพ์เพื่อตรวจจับตำแหน่งของจุดสำคัญบนฝ่ามือ และใช้เครือข่ายประสาทคอนโวลูชันแบบ 1 มิติ (1DCNN) เพื่อเรียนรู้ความสัมพันธ์ระหว่างลำดับของจุดสำคัญบนฝ่ามือกับท่าภาษามือ โดยที่ 1DCNN จะวิเคราะห์ลำดับของพิกัด x, y ของจุดสำคัญบนฝ่ามือเพื่อเปรียบเทียบจุดสำคัญที่ตรวจจับได้กับฐานข้อมูลท่าภาษามือไทยที่ใช้ "ท่ามือ" เป็นตัวบ่งชี้ หลังจากนั้นจะประมวลผลผลลัพธ์โดยใช้ชุดคำสั่งภาษาไพธอนเพื่อแสดงผลของคำหรือตัวสะกดที่แปลได้ พบว่าสามารถแปลคำศัพท์และตัวสะกดได้อย่างรวดเร็ว ตรวจจับการเคลื่อนไหวของท่ามือได้ทันที และมีความแม่นยำในการแปลอยู่ที่ร้อยละ 94.663

5.2 ข้อจำกัดและข้อเสนอแนะ

5.2.1 ข้อจำกัด

1) ในขั้นต้น ผู้วิจัยได้พัฒนาต้นแบบแอปพลิเคชันโดยใช้ชุดข้อมูลภาษามือไทย 30 คำ ซึ่งเป็นคำที่ง่ายเบื้องต้น เช่น สวัสดี ขอบคุณ และ ระบายน้ำ การมุ่งเน้นไปที่คำศัพท์พื้นฐานเหล่านี้ ช่วยให้ระบบสามารถแปลคำศัพท์ที่ใช้งานบ่อยได้อย่างแม่นยำ ในอนาคต ผู้วิจัยมีแผนที่จะขยายชุดข้อมูลให้ครอบคลุมคำศัพท์และประโยคที่หลากหลายมากขึ้น เพื่อเพิ่มความสามารถในการแปลภาษา และรองรับการใช้งานในสถานการณ์ที่ซับซ้อนยิ่งขึ้น

2) เพื่อให้ระบบแปลภาษามือได้อย่างมีประสิทธิภาพสูงสุด ผู้ใช้งานควรแสดงท่าทางให้ชัดเจน โดยพยายามรักษามุมมือและตำแหน่งให้ตรงกับกล้อง และอยู่ในสภาพแสงที่เหมาะสม ผู้วิจัยกำลังพัฒนาเทคนิคการประมวลผลภาพ เพื่อลดผลกระทบจากความแปรปรวนของมุมกล้องและสภาพแสง รวมถึงเพิ่มความสามารถในการตรวจจับท่าทางที่ซับซ้อน เช่น การบิดมือ เพื่อรองรับการใช้งานที่หลากหลายมากขึ้น

5.2.2 ข้อเสนอแนะ

คอมพิวเตอร์วิทัศน์ที่พัฒนาขึ้นสำหรับการศึกษานี้สามารถรู้จำภาษามือไทยจากชุดคำศัพท์และชุดตัวสะกดนิ้วมือไทยได้ในระดับที่ค่อนข้างดี อย่างไรก็ตาม ชุดคำศัพท์และชุดตัวสะกดที่ใช้ศึกษายังมีลักษณะเป็นเพียงภาพนิ่งของนิ้วมือที่ยังไม่ได้เคลื่อนไหวอย่างต่อเนื่อง ดังนั้นการรู้จำจึงยังจำกัดอยู่เพียงที่ "ท่ามือ" เท่านั้น ในการพัฒนาต่อไปจะเพิ่มความสามารถในการรู้จำภาษามือไทยโดยคำนึงถึงองค์ประกอบอื่นๆ ของภาษามือ อาทิ ตำแหน่งของมือ การเคลื่อนไหวของมือ การพลิกฝ่ามือ และการแสดงออกทางสีหน้า

นอกจากนี้สำหรับการวิจัยในอนาคตจะปรับปรุงการแสดงผลลัพธ์ในรูปแบบเสียงให้เร็วและสัมพันธ์กับการแปล เพื่อให้สามารถแปลได้อย่างต่อเนื่อง ลดปัญหาที่เกิดจากการหยุดการทำงานของไพธอนเนื่องจากการเรียกใช้ชุดคำสั่งแปลงข้อความเป็นเสียง และจะแสดงผลลัพธ์ของคำที่

แปลแล้วเป็นแถบเลื่อนที่ด้านล่างเพื่อความสะดวกของผู้ใช้ที่ต้องการย้อนกลับไปดูพร้อมกับพัฒนา
คอมพิวเตอร์วิทัศน์บนแพลตฟอร์มอื่นๆ เช่น เว็บแอปพลิเคชัน (Web Application), แอนดรอยด์
(Android) และ ไอโอเอส (iOS) เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- กวิณ เต็งอำนวยการ, ธัญพิสิษฐ์ พรหมมี, และ ชิดารัตน์ โชคชวลิต (2563). การจำแนกภาษามือด้วย
โครงข่ายแบบวนซ้ำ. ภาควิชาวิศวกรรมอิเล็กทรอนิกส์คณะวิศวกรรมศาสตร์. กรุงเทพฯ:
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- จักรพันธ์ สุทธิแพทย์ (2565). การพัฒนาการแปลภาษามือไทยด้วยคลังโปรแกรมมีเดียไพพ์. คณะ
วิศวกรรมศาสตร์. กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย.
- จิรายุ พัทธกันต์สันกุล, ดิษย์ฉัตร เปียลาวัฒน์, และ ปีย์วรานูช เบญญาบุณยาพนธ์. (2565).
การเปรียบเทียบประสิทธิภาพวิธีการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกใน
การตรวจจับจดหมายอิเล็กทรอนิกส์ไม่พึงประสงค์. ภาควิชาสถิติ คณะวิทยาศาสตร์.
กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- ณัฏยา เปลี่ยนวงษ์ (2564). ระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้เชิงลึกบนข้อมูลบางส่วน.
สารนิพนธ์ (วิศวกรรมข้อมูลขนาดใหญ่). กรุงเทพฯ: มหาวิทยาลัยธุรกิจบัณฑิต.
- ชนกฤต เอี่ยมเต็ง และ ธนากร สุวรรณกรกิจ (2563). การตรวจจับการยื่นมือจับสิ่งของโดยใช้กล้อง.
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์. กรุงเทพฯ: สถาบันเทคโนโลยี
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- ปวีรบรรต ปธานราชภูรี (2561). การทำนายประสิทธิภาพการทำงานของ JUPYTER NOTEBOOK
บน JUPYTERHUB โดยใช้เทคนิคการเรียนรู้ของเครื่อง. กรุงเทพฯ: มหาวิทยาลัยธุรกิจ
บัณฑิต.
- ปานรวี ชุ่มนิม และ ปราณี มณีรัตน์ (2563). การพัฒนาแอปพลิเคชันระบบการแปลภาษามือให้กับ
ผู้พิการทางการได้ยิน. สาขาวิชาเทคโนโลยีสารสนเทศคณะเทคโนโลยีสารสนเทศ.
มหาวิทยาลัยศรีปทุม.
- ภัทรณัฐ ศรีบุญเรือง, พันธกรณ์ ทะนันชัย, ศุภนทร์เทพ แวกระโทก, วุฒิชัย วิจิตรกุลสวัสดิ์, และ
ปราโมทย์ อนันต์วรพงษ์ (2565). โปรแกรมแปลภาษามือเป็นข้อความและเสียงพูดโดย
การใช้วิธีการระบุพิกัดตำแหน่งด้วยมีเดียไพพ์.
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะวิศวกรรมศาสตร์. กรุงเทพฯ:
มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ.
- ศศิธร ทรัพย์วัฒนไพศาล, ส.ท., กานต์ อรรถยุกติ, ราชภูรี บุญญา, พงษ์ส ศุภจรรยา และ พร พรรณ
สมบูรณ์ (2560). เทคนิคที่ใช้ในการแปลคำอธิบายศัพท์: กรณีคำศัพท์ด้านเทคโนโลยีสิ่ง
อำนวยความสะดวก. in การสัมมนาวิชาการระดับชาติด้านคนพิการ ครั้งที่ 9 ปี 2560.
- ศุภณัฐ จิตวัฒน์สกุล. (2562). การจำแนกภาพขวดแบบเซตเปิดด้วยโครงข่ายประสาทเทียมแบบ
คอนโวลูชัน. กรุงเทพฯ: มหาวิทยาลัยศรีนครินทรวิโรฒ.
- อนุเทพ ชมพูธวัช, สิริกัญญา ชมพูธวัช (2565). วิจัยพัฒนาระบบสนทนาอัตโนมัติและแพลตฟอร์ม
ระบบผู้ช่วยส่วนตัวอัจฉริยะ. กรุงเทพฯ: กรมสอบสวนคดีพิเศษ กระทรวงยุติธรรม.
- อมร พงษ์สว่าง, สิทธิพร สุนทร (2564). ปัจจัยที่ส่งผลต่อคุณภาพชีวิตของคนพิการได้ยินหรือสื่อ
ความหมายในจังหวัดสระบุรี. มหาสารคาม: มหาวิทยาลัยราชภัฏมหาสารคาม.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเข้าถึงที่ผิดกฎหมายหากมีการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- Kiranyaz, S., Ince, T., & Gabbouj, M. (2015). **Real-time patient-specific ECG classification by 1-D convolutional neural networks**. *IEEE Transactions on Biomedical Engineering*, 63(3), 664-675. doi:10.1109/TBME.2015.2468589.
- Mohammad, R.K., (2023). **Wind field forecasting using a novel method based on convolutional neural networks and bidirectional LSTM**. Faculty of Naval Aviation. Malek Ashtar University of Technology. Iran.
- Muhammad, S., Babar, A., Muhammad, K., Tauqeer, S., Mehdi, H., Mohd Hilmi, H., Norshakirah, A., (2021). **A Real-Time Automatic Translation of Text to Sign Language**. Multan. Bahauddin Zakariya University.
- Ratchakrit, A., Stephen, M., Scott, P., (2022). **Survey on Thai NLP Language Resources and Tools**. School of Computing and Communications. Lancaster University Lancaster LA1 4WA.
- Ratchakrit, A., Stephen, M., Scott, P., (2022). **Survey on Thai NLP Language Resources and Tools**. School of Computing and Communications. Lancaster University Lancaster LA1 4WA.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is All You Need**. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6000-6010). doi:10.48550/arXiv.1706.03762.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

.....
 ชุดคำสั่งภาษา Python ที่ใช้ในการเขียนโปรแกรม
 แอปพลิเคชันแปลภาษามือเป็นเสียงพูดด้วยเทคโนโลยีปัญญาประดิษฐ์


```

from .utils import Preprocess
from .config import MAX_LEN, CHANNELS, NUM_CLASSES
import tensorflow as tf

class ECA(tf.keras.layers.Layer):
    def __init__(self, kernel_size=5, **kwargs):
        super().__init__(**kwargs)
        self.supports_masking = True
        self.kernel_size = kernel_size
        self.conv = tf.keras.layers.Conv1D(1, kernel_size=kernel_size, strides=1,
padding="same", use_bias=False)

    def call(self, inputs, mask=None):
        nn = tf.keras.layers.GlobalAveragePooling1D()(inputs, mask=mask)
        nn = tf.expand_dims(nn, -1)
        nn = self.conv(nn)
        nn = tf.squeeze(nn, -1)
        nn = tf.nn.sigmoid(nn)
        nn = nn[:,None,:]
        return inputs * nn

class LateDropout(tf.keras.layers.Layer):
    def __init__(self, rate, noise_shape=None, start_step=0, **kwargs):
        super().__init__(**kwargs)
        self.supports_masking = True
        self.rate = rate
        self.start_step = start_step
        self.dropout = tf.keras.layers.Dropout(rate, noise_shape=noise_shape)

    def build(self, input_shape):
        super().build(input_shape)
        self.agg = tf.VariableAggregation(ONLY_FIRST_REPLICA)

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ของเอกสารทุกครั้งที่มีการนำไปใช้

```
self._train_counter = tf.Variable(0, dtype="int64", aggregation=agg,
trainable=False)
```

```
def call(self, inputs, training=False):
    x = tf.cond(self._train_counter < self.start_step, lambda:inputs,
lambda:self.dropout(inputs, training=training))
    if training:
        self._train_counter.assign_add(1)
    return x
```

```
class CausalDWConv1D(tf.keras.layers.Layer):
    def __init__(self,
kernel_size=17,
dilation_rate=1,
use_bias=False,
depthwise_initializer='glorot_uniform',
name="", **kwargs):
    super().__init__(name=name, **kwargs)
    self.causal_pad = tf.keras.layers.ZeroPadding1D((dilation_rate*(kernel_size-
1),0),name=name + '_pad')
    self.dw_conv = tf.keras.layers.DepthwiseConv1D(
kernel_size,
strides=1,
dilation_rate=dilation_rate,
padding='valid',
use_bias=use_bias,
depthwise_initializer=depthwise_initializer,
name=name + '_dwconv')
    self.supports_masking = True
```

```
def call(self, inputs):
    x = self.causal_pad(inputs)
    x = self.dw_conv(x)
    return x
```

```
def Conv1DBlock(channel_size,
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นเห็นสมควรให้เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        drop_rate=0.0,
        expand_ratio=2,
        se_ratio=0.25,
        activation='swish',
name=None):
    if name is None:
        name = str(tf.keras.backend.get_uid("mbblock"))
    def apply(inputs):
        channels_in = tf.keras.backend.int_shape(inputs)[-1]
        channels_expand = channels_in * expand_ratio

        skip = inputs

        x = tf.keras.layers.Dense(
            channels_expand,
            use_bias=True,
            activation=activation,
            name=name + '_expand_conv')(inputs)

        x = CausalDWConv1D(kernel_size,
            dilation_rate=dilation_rate,
            use_bias=False,
            name=name + '_dwconv')(x)

        x = tf.keras.layers.BatchNormalization(momentum=0.95, name=name + '_bn')(x)

        x = ECA()(x)

        x = tf.keras.layers.Dense(
            channel_size,
            use_bias=True,
            name=name + '_project_conv')(x)
import numpy as np

THRESH_HOLD = 0.5

```

เอกสารนี้เป็นลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ROWS_PER_FRAME = 543
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAX_LEN = 384
CROP_LEN = MAX_LEN
NUM_CLASSES = 250
PAD = -100.

NOSE=[
    1,2,98,327
]
LNOSE = [98]
RNOSE = [327]
LIP = [ 0,
    61, 185, 40, 39, 37, 267, 269, 270, 409,
    291, 146, 91, 181, 84, 17, 314, 405, 321, 375,
    78, 191, 80, 81, 82, 13, 312, 311, 310, 415,
    95, 88, 178, 87, 14, 317, 402, 318, 324, 308,
]
LLIP = [84,181,91,146,61,185,40,39,37,87,178,88,95,78,191,80,81,82]
RLIP = [314,405,321,375,291,409,270,269,267,317,402,318,324,308,415,310,311,312]

POSE = [500, 502, 504, 501, 503, 505, 512, 513]
LPOSE = [513,505,503,501]
RPOSE = [512,504,502,500]

REYE = [
    33, 7, 163, 144, 145, 153, 154, 155, 133,
    246, 161, 160, 159, 158, 157, 173,
]
LEYE = [
    263, 249, 390, 373, 374, 380, 381, 382, 362,
    466, 388, 387, 386, 385, 384, 398,
]

LHAND = np.arange(468, 489).tolist()
RHAND = np.arange(522, 543).tolist()

POINT_LANDMARKS = LIP + LHAND + RHAND + NOSE + REYE + LEYE #+POSE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใด ทั้งสิ้น อีกทั้งหากมีข้อสงสัยหรือต้องการอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHANNELS = 6*NUM_NODES

if __name__ == "__main__":
    print(NUM_NODES)
    print(CHANNELS)
from .config import ROWS_PER_FRAME, SEQ_LEN
import json
import cv2
import numpy as np
import mediapipe as mp

SEQUENCE_LENGTH = SEQ_LEN
ROWS_PER_FRAME = ROWS_PER_FRAME

mp_holistic = mp.solutions.holistic
mp_drawing = mp.solutions.drawing_utils

def mediapipe_detection(image, model):
    converted_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    converted_image.flags.writeable = False
    results = model.process(converted_image)
    converted_image.flags.writeable = True
    return cv2.cvtColor(converted_image, cv2.COLOR_RGB2BGR), results

def draw_landmarks(image, results):
    if results.face_landmarks:
        mp_drawing.draw_landmarks(image, results.face_landmarks,
            mp_holistic.FACEMESH_TESSELATION,
            mp_drawing.DrawingSpec(color=(0, 0, 0), thickness=1,
            circle_radius=0),
            mp_drawing.DrawingSpec(color=(227, 224, 113), thickness=1,
            circle_radius=0))
    if results.left_hand_landmarks:
        mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
            mp_holistic.HAND_CONNECTIONS,
            mp_drawing.DrawingSpec(color=(227, 224, 113), thickness=3,
            circle_radius=3),

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mp_drawing.DrawingSpec(color=(227, 224, 113), thickness=2,
circle_radius=2))
    if results.right_hand_landmarks:
        mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
        mp_drawing.DrawingSpec(color=(227, 224, 113), thickness=3,
circle_radius=3),
        mp_drawing.DrawingSpec(color=(227, 224, 113), thickness=2,
circle_radius=2))

```

```

def extract_coordinates(results):
    def get_landmark_array(landmarks, size):
        return np.array([[lm.x, lm.y, lm.z] for lm in landmarks.landmark] if
landmarks else np.zeros(size * 3))

```

```

    face = get_landmark_array(results.face_landmarks, 468)
    lh = get_landmark_array(results.left_hand_landmarks, 21)
    pose = get_landmark_array(results.pose_landmarks, 33)
    rh = get_landmark_array(results.right_hand_landmarks, 21)

    return np.concatenate([face, lh, pose, rh])

```

```

def load_json_file(json_path):
    try:
        with open(json_path, 'r') as f:
            return json.load(f)
    except FileNotFoundError:
        print(f"File not found: {json_path}")
    except json.JSONDecodeError:
        print(f"Error decoding JSON from file: {json_path}")
    return {}

```

```

from .config import MAX_LEN, POINT_LANDMARKS
import tensorflow as tf

```

```

def tf_nan_mean(x, axis=0, keepdims=False):
    return tf.reduce_sum(tf.where(tf.math.is_nan(x), tf.zeros_like(x), x), axis=axis,
keepdims=keepdims) / tf.reduce_sum(tf.where(tf.math.is_nan(x), tf.zeros_like(x),
tf.ones_like(x)), axis=axis, keepdims=keepdims))

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะในรูปแบบใดก็ตาม หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และต้องอย่างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
def tf_nan_std(x, center=None, axis=0, keepdims=False):
```

```
    if center is None:
```

```
        center = tf_nan_mean(x, axis=axis, keepdims=True)
```

```
    d = x - center
```

```
    return tf.math.sqrt(tf_nan_mean(d * d, axis=axis, keepdims=keepdims))
```

```
class Preprocess(tf.keras.layers.Layer):
```

```
    def __init__(self, max_len=MAX_LEN, point_landmarks=POINT_LANDMARKS,
**kwargs):
```

```
        super().__init__(**kwargs)
```

```
        self.max_len = max_len
```

```
        self.point_landmarks = point_landmarks
```

```
    def call(self, inputs):
```

```
        if tf.rank(inputs) == 3:
```

```
            x = inputs[None,...]
```

```
        else:
```

```
            x = inputs
```

```
        mean = tf_nan_mean(tf.gather(x, [17], axis=2), axis=[1,2], keepdims=True)
```

```
        mean = tf.where(tf.math.is_nan(mean), tf.constant(0.5,x.dtype), mean)
```

```
        x = tf.gather(x, self.point_landmarks, axis=2) #N,T,P,C
```

```
        std = tf_nan_std(x, center=mean, axis=[1,2], keepdims=True)
```

```
        x = (x - mean)/std
```

```
        if self.max_len is not None:
```

```
            x = x[:, :self.max_len]
```

```
        length = tf.shape(x)[1]
```

```
        x = x[...,:2]
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโครงการวิจัยเพื่อการพัฒนาเมืองนวัตกรรมดิจิทัล
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นผู้ที่ไม่เห็นด้วยกับเงื่อนไขและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dx2 = tf.cond(tf.shape(x)[1]>2,lambda:tf.pad(x[:,2:] - x[:,:-2],
[[0,0],[0,2],[0,0],[0,0]]),lambda:tf.zeros_like(x))
```

```
x = tf.concat([
    tf.reshape(x, (-1,length,2*len(self.point_landmarks))),
    tf.reshape(dx, (-1,length,2*len(self.point_landmarks))),
    tf.reshape(dx2, (-1,length,2*len(self.point_landmarks))),
], axis = -1)
x = tf.where(tf.math.is_nan(x),tf.constant(0.,x.dtype),x)
return x
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้วิจัย



ชนาธิป อินทร์อุดม (มาร์ค)

ที่อยู่: 5/11 หมู่ 3 ต.บางเสาธง อ.บางเสาธง จ.สมุทรปราการ 10570

เบอร์โทรศัพท์: 095-293-8807

Email: mark19082544@gmail.com

ประวัติการศึกษา

- ระดับมัธยมศึกษา โรงเรียนนวมินทราชินูทิศ เตรียมอุดมศึกษาพัฒนาการ
- ระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาสถิติประยุกต์ ภาควิชาสถิติ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



ธนภัทร เรืองจันทร์ (เจ)

ที่อยู่: 11/41 หมู่ 7 ต.บางเมือง อ.เมือง จ.สมุทรปราการ 10270

เบอร์โทรศัพท์: 095-614-5755

Email: jicupjoke0@hotmail.com

ประวัติการศึกษา

- ระดับมัธยมศึกษา โรงเรียนสมุทรปราการ
- ระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาสถิติประยุกต์ ภาควิชาสถิติ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



นัทพงษ์ หงษ์รัตน์ (ไอซ์)

ที่อยู่: 139/110 คู้บอน 27 แยก 25 เขตบางเขน แขวงท่าแร้ง กรุงเทพมหานคร 10220

Email: nattaponghongrat@gmail.com

ประวัติการศึกษา

- ระดับมัธยมศึกษา โรงเรียนนวมินทราชูทิศ กรุงเทพมหานคร
- ระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาสถิติประยุกต์ ภาควิชาสถิติ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



งานทะเบียนคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
คำรับรองเล่มปัญหาพิเศษ

วันที่ 26 เดือน มิถุนายน พ.ศ. 2567

ข้าพเจ้า นายชนาธิป อินทร์อุดม รหัสนักศึกษา 63050610
นายธนภัทร เรืองจันทร์ รหัสนักศึกษา 63050628
นายณัฏพงษ์ หงษ์รัตน์ รหัสนักศึกษา 63050636

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา สถิติประยุกต์ ภาควิชา สถิติ ขอรับรองว่าปัญหา
พิเศษ เรื่อง

ชื่อภาษาไทย การเชื่อมโยงช่องว่างในการสื่อสาร: แอปพลิเคชันปัญญาประดิษฐ์แบบ
เรียลไทม์สำหรับแปลภาษามือเป็นเสียงพูดภาษาไทย

ชื่อภาษาอังกฤษ BRIDGING THE COMMUNICATION GAP: A REAL-TIME AI
APPLICATION FOR SIGN LANGUAGE TO THAI SPEECH
TRANSLATION

ปีการศึกษา 2566

เป็นผลงานวิจัยที่มีได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อน
เรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่ม
ปัญหาพิเศษฉบับสมบูรณ์แล้ว
โปรแกรมอักขราวิสุทธิ์ 0.56 %

ลงชื่อ.....*ชนาธิป อินทร์อุดม*..... ลงชื่อ.....*ธนภัทร*..... ลงชื่อ.....*ณัฏพงษ์*.....
(นายชนาธิป อินทร์อุดม) (นายธนภัทร เรืองจันทร์) (นายณัฏพงษ์ หงษ์รัตน์)
นักศึกษา นักศึกษา นักศึกษา

ข้าพเจ้า ผศ.ดร.พรพิมล ชัยวุฒิศักดิ์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ได้ตรวจสอบปัญหาพิเศษของ
นักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์
จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ *Pornpimol Chaiwuttisak*
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
อาจารย์ที่ปรึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Plagiarism Checking Report

Created on 2024-06-25 02:55:03 at 02:55 AM

Submission Information

ID	SUBMISSION DATE	SUBMITTED BY	ORGANIZATION	FILENAME	STATUS	SIMILARITY INDEX
3799729	Jun 25, 2024 at 02:51 AM	63050636@kmitl.ac.th	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	newstedited_Special_Mol(Comment 18 June 2024).docx	Completed	0.56 %

Match Overview

NO.	TITLE	AUTHOR(S)	SOURCE	SIMILARITY INDEX
1	https://www.rs.mahidol.ac.th/rs-journal/vol.1/v.1-1-006.pdf	rs.mahidol.ac.th	rs.mahidol.ac.th_nutch	0.30 %
2	การนับหอกระตาศด้วยกล้องไอพี กรณีศึกษาระบบงานขนส่งของ SCGL สาขาบ้านโป่ง	อรุณพล พลานนท์	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	0.26 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้