

แนวทางการพัฒนาการเรียนรู้เชิงลึกโดยใช้คุณลักษณะแบบผสมเพื่อการจำแนกภาษามือ

สรวิชัย รักประเสริฐสุข

TNII

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
บัณฑิตศึกษา สาขาเทคโนโลยีวิศวกรรม
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2566

MIXED-FEATURES-BASED DEEP LEARNING APPROACH FOR SIGN LANGUAGE
CLASSIFICATION

Sorawich Rakprasertsuk

TNII

A Thesis Submitted in Partial Fulfillment of Requirements
for The Degree of Master of Engineering Program in Engineering Technology
Graduate Studies
Thai-Nichi Institute of Technology
Academic Year 2023

หัวข้อวิทยานิพนธ์

แนวทางการพัฒนาการเรียนรู้เชิงลึกโดยใช้คุณลักษณะแบบ
ผสมเพื่อการจำแนกภาษามือ

โดย

สรวิษฐ์ รักประเสริฐสุข

สาขาวิชา

เทคโนโลยีวิศวกรรม

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ดร.ชัชไชย วรรณบุญ

บัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....รองอธิการบดีฝ่ายวิชาการ

(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ดร.ณัฐกิตติ์ จิตรเอื้อตระกูล)

.....กรรมการ

(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)

.....กรรมการ

(ดร.อัฒนา เซนโตะ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร.ชัชไชย วรรณบุญ)

สรวิชัย รักประเสริฐสุข : แนวทางการพัฒนาการเรียนรู้เชิงลึกโดยใช้คุณลักษณะแบบผสม
เพื่อการจำแนกภาษามือ. อาจารย์ที่ปรึกษา : ดร.ชัชไชย วรรณบุญ, 50 หน้า.

การจำแนกภาษามือเป็นงานที่ท้าทายเนื่องจากความซับซ้อนและความหลากหลายของท่าทางและรูปร่างมือ เอกสารนี้นำเสนอวิธีการเรียนรู้เชิงลึกสำหรับการจำแนกภาษามือโดยใช้วิธีการผสมคุณลักษณะ มีข้อมูลเชิงข้อความสองชุดที่แตกต่างกันมาจากแหล่งที่มีความเป็นที่ยอมรับ คือภาพที่ถูกปรับและฮิสโตแกรมที่ถูกแปลงให้เป็นข้อมูลเข้าสู่โมเดลการเรียนรู้เชิงลึก ชุดข้อมูลภาพใช้เป็นข้อมูลเข้าสู่โมเดล convolutional neural network (CNN) เพื่อจับลักษณะของพื้นที่และคุณลักษณะ ในขณะที่ชุดข้อมูลฮิสโตแกรมถูกปรับให้เป็นข้อมูลเข้าสู่โมเดล long short-term memory (LSTM) เน้นด้านเวลาของท่าทางมือ การผสมนี้ไม่เพียงทำให้เป็นไปได้ในการวิเคราะห์ลักษณะทั้งของพื้นที่และเวลา แต่ยังช่วยให้ลดขนาดของโมเดลโดยรวม

ผลการทดลองแสดงให้เห็นการปรับปรุงที่สังเกตเห็นได้ในตัวชี้วัดความสามารถในการทำงานวิธีการที่นำเสนอสามารถทำให้เกิดการลดขนาดของโมเดล ซึ่งช่วยลดความต้องการในการประมวลผลที่ใช้ทรัพยากร นอกจากนี้ยังลดความจำเป็นในการเพิ่มข้อมูลในกระบวนการฝึกอบรมเช่นเดียวกัน วิธีการที่นำเสนอช่วยเพิ่มประสิทธิภาพโดยรวมของระบบการจำแนกภาษามือได้อย่างมีนัยสำคัญ

บัณฑิตศึกษา

สาขาวิชา เทคโนโลยีวิศวกรรม

ปีการศึกษา 2566

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

SORAWICH RAKPRASERTSUK : DEVELOPMENT OF MIXED-FEATURES-BASED DEEP LEARNING FOR SIGN LANGUAGE CLASSIFICATION. ADVISOR : DR. CHATCHAI WANNABOON, 50 PP.

Sign language classification is a challenging task due to the complexity and diversity of sign gestures, and hand shapes. This paper presents deep learning approach for sign language classification using a mixed-features method. Two distinct datasets derived from a common source, i.e., a conventional image and a transformed histogram, are used for the input of deep learning model. The image dataset serves as input for a convolutional neural network (CNN) model, capturing spatial dependencies and features, while the histogram dataset is adapted for input to a long short-term memory (LSTM) model, emphasizing temporal aspects of hand gestures. This integration not only enables a comprehensive analysis of both spatial and temporal characteristics but also facilitates a reduction in the overall model size.

The experimental results demonstrate notable improvements in efficiency metrics. The proposed method can achieve a reduction in model size, alleviating resource-intensive computational requirements. Furthermore, the need for extensive data augmentation is eliminated, streamlining the training process. Consequently, the proposed approach significantly enhances the overall efficiency of the sign language classification system.

Graduate Studies

Field of Engineering of Technology

Academic Year 2023

Student's Signature.....

Advisor's Signature.....

กิตติกรรมประกาศ

งานวิทยานิพนธ์เรื่อง “แนวทางการเรียนรู้เชิงลึกโดยใช้คุณลักษณะผสมสำหรับการจำแนกภาษามือ” ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยความกรุณาอย่างยิ่งจาก ดร.ชัชไชย วรรณบูรณ์ อาจารย์ที่ปรึกษาซึ่งได้ให้ความรู้ คำปรึกษา แนะนำแนวทาง เพื่อนำไปปรับปรุง และแก้ไขข้อบกพร่องต่างๆ เพื่อให้งานสมบูรณ์ยิ่งขึ้น พร้อมทั้งยังช่วยพัฒนาการทำงานของวิทยานิพนธ์ให้เป็นไปอย่างมีคุณภาพ และสนับสนุนเครื่องมือที่ใช้ในการทำงานวิจัยนี้ อีกทั้งยังดูแลเอาใจใส่ติดตามระยะเวลาในการทำวิทยานิพนธ์

ขอขอบพระคุณคณะกรรมการทุกท่านที่ได้ให้ความรู้ คำแนะนำ และข้อคิดที่เป็นประโยชน์ต่อการทำวิทยานิพนธ์ฉบับนี้ พร้อมทั้งให้คำแนะนำแนวทาง ปรับปรุงแก้ไข จนสามารถทำให้วิทยานิพนธ์ฉบับนี้เสร็จสิ้นสมบูรณ์ตามขอบเขตที่กำหนดไว้

นอกจากนี้ทางผู้วิจัยขอขอบคุณแรงสนับสนุนจากครอบครัว และคนรอบข้างที่คอยให้กำลังใจ ให้คำปรึกษาตลอดระยะเวลาการศึกษา จนจัดทำวิทยานิพนธ์ฉบับนี้เสร็จสิ้น

ผู้วิจัยหวังเป็นอย่างยิ่งว่า วิทยานิพนธ์ฉบับนี้จะก่อให้เกิดประโยชน์แก่ผู้ที่สนใจ และสามารถนำไปต่อยอดกับองค์ความรู้ที่เกี่ยวข้องได้ อีกทั้งสามารถก่อให้เกิดแนวคิดเพื่อใช้ในการศึกษา หรือเป็นแนวทางการทำงานวิจัยที่เกี่ยวข้องในอนาคต

สรวิชัย รักประเสริฐสุข

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญรูป.....	ฐ
 บทที่	
1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตของงานวิจัย.....	3
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1.1 ภาษามือ (American hand sign).....	4
A. องค์ประกอบของภาษามืออเมริกัน (American hand sign).....	5
2.1.2 ทฤษฎีการเรียนรู้ของเครื่องจักร (Machine Learning).....	5
A. การเรียนรู้แบบมีผู้สอน (Supervised Learning).....	6
B. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning).....	7
C. การเรียนรู้แบบการเสริม (Reinforcement Learning).....	8
D. การเรียนรู้แบบมีผู้สอน (Supervised Learning).....	8
2.1.3 Convolutional Neural Networks (CNNs).....	9
A. Convolutional Layers.....	9
B. Pooling Layers.....	9
C. Activation Functions.....	10
D. Fully Connected Layers.....	10

สารบัญ (ต่อ)

บทที่		หน้า
2	2.1.4 Long Short-Term Memory	11
	A. โครงสร้างของ LSTM	11
	2.2 การทบทวนวรรณกรรม	12
3	วิธีการดำเนินงาน	16
	3.1 แผนการดำเนินงานวิจัย	16
	3.2 ขั้นตอนการดำเนินงาน	17
	3.3 การออกแบบระบบ	18
	3.4 การทำการทดลอง	19
	3.4.1 การรวบรวมข้อมูล	19
	3.4.2 การตั้งค่าการทดลอง	19
	3.4.3 การพัฒนาแบบจำลอง	20
4	ผลการศึกษาและการอภิปรายผล	27
	4.1 ตัวชี้วัดประสิทธิภาพ	27
	4.2 การคาดเดาแบบจำลอง	31
	4.3 การวิเคราะห์เชิงคุณภาพ	31
5	สรุปผลและข้อเสนอแนะ	33
	5.1 สรุปผล	33
	5.2 ข้อเสนอแนะ	33
	บรรณานุกรม	36

สารบัญ (ต่อ)

บทที่	หน้า
ภาคผนวก	38
ภาคผนวก ก. โค้ดโปรแกรม	39
ประวัติย่อผู้วิจัย	50



สารบัญตาราง

ตาราง	หน้า
2.1 สรุปงานวิจัยที่เกี่ยวข้อง	14
3.1 แผนการดำเนินงานวิจัย	17
3.2 แสดงประสิทธิภาพของแต่ละโมเดล CNN ที่ epoch = 50	22
3.3 แสดงประสิทธิภาพของแต่ละโมเดล LSTM ที่ epoch = 50	24
4.1 สรุปตัวชี้วัดประสิทธิภาพ	30
4.2 เปรียบเทียบประสิทธิภาพของการศึกษาก่อน	32



สารบัญรูป

รูป	หน้า
2.1 แสดงรูปแบบของสัญลักษณ์มือที่แทนตัวอักษรภาษาอังกฤษทั้ง 44 ตัว	5
2.2 เครื่องมูนีวีส์โคมิเตอร์ (Mooney Viscometer).....	6
2.3 การทำงานของแกนหมุนภายในห้องบรรจุยาง	7
2.4 กราฟความหนืดมูนีที่ได้จากเครื่องมูนีวีส์โคมิเตอร์	8
2.5 ระบบสายการผลิตยางพาวด์ที่นิยมใช้ในอุตสาหกรรม.....	8
2.6 เครื่องผสมภายในระบบปิด.....	9
2.7 เครื่องผสมระบบเปิดชนิดสองลูกกลิ้ง	10
2.8 การเปรียบเทียบระหว่างตรรกศาสตร์บูลีน และตรรกศาสตร์คลุมเครือ	11
2.9 โครงสร้างของการประมวลผลแบบตรรกศาสตร์คลุมเครือ	12
3.1 รูปแบบการอนุมานแบบตรรกศาสตร์คลุมเครือ	18
3.2 วิธีการจัดองค์ประกอบแบบค่าสูงสุด-ต่ำสุด	21
3.3 วิธีการจัดองค์ประกอบแบบค่าสูงสุด-ผลคูณ.....	23
3.4 The Tsukamoto Fuzzy Model	27
4.1 การแปลงค่าความคลุมเครือเป็นค่าจริงทั่วไปด้วยวิธีถ่วงน้ำหนัก	28
4.2 ความแตกต่างของเครือข่ายประสาทในสมองมนุษย์และโครงข่ายประสาทเทียม.....	29
4.3 โครงข่ายประสาทเทียมแบบชั้นเดียว	31

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ภาษามือเป็นการสื่อสารหลักสำหรับชุมชนคนหูหนวกและมองเห็นได้ยาก ซึ่งมีการใช้ท่าทางและรูปร่างมือที่หลากหลายและมีความหมายต่างกันอย่างมาก การจำแนกสัญญาณเหล่านี้ อย่างแม่นยำและมีประสิทธิภาพมีความสำคัญอันสำคัญเนื่องจากความซับซ้อนและความหลากหลายของภาษามือตนเอง ในรอบสองสามปีที่ผ่านมา มีความก้าวหน้าที่น่าทึ่งในด้านการจำแนกภาษามือ โดยเน้นการผสมเทคโนโลยีการเรียนรู้เชิงลึก (DL) วิธีการนี้มีความแม่นยำและการส่วนทั่วไปที่สูง และสามารถรวมเข้ากับเทคโนโลยีอื่น ๆ เช่น computer vision, natural language processing, และ speech synthesis ที่นอกจากนี้ยังสามารถประยุกต์ใช้กับภาษามือต่าง ๆ เช่น ภาษามืออินเดีย, ภาษามืออเมริกัน, ภาษามือเปอร์เซีย ฯลฯ และแปลงมันเป็นภาษาพูด CNN ได้รับการนำมาใช้ในอัตราส่วนที่แตกต่างกันเนื่องจากความสามารถในการเรียนรู้ลำดับที่ซับซ้อน ซึ่งสามารถให้ความแม่นยำที่ดีกว่าในการรับรู้ท่าทางและรูปร่างมือที่หลากหลาย โมเดล deep neural network ยังมีความแข็งแกร่งต่อการเปลี่ยนแปลงในเงื่อนไขการจำลองและสามารถประมวลผลแบบ real-time ที่สำคัญสำหรับการประยุกต์ใช้ในชีวิตประจำวัน โมเดล deep neural framework สำหรับการจำแนกภาษามือต่อเนื่องได้รับการนำเสนอใน CNNs ถูกนำมาใช้งานพร้อมกับชั้นทศนิยมเวลา fusion เพื่อการตีความภาษามือต่อเนื่องผ่านวิดีโอ ซึ่งแสดงผลการปรับปรุงที่สำคัญในความแม่นยำในการจำแนกภาษามือ RNNs เป็นเครื่องมือที่มีประสิทธิภาพสูงสำหรับการจำแนกภาษามือเนื่องจากความชำนาญในการจัดการข้อมูลที่เป็นลำดับ ซึ่งเหมาะสำหรับความเคลื่อนไหวของเวลาในภาษามือ เพื่อเพิ่มความแม่นยำและเสถียรภาพในการพยากรณ์โดยรวม ensemble method ถูกใช้งานโดยการรวมกันของโมเดลที่ทำนายหลายรูปแบบ วิธีการนี้สามารถใช้ประโยชน์จากจุดแข็งของแต่ละโมเดลเพื่อสร้างการพยากรณ์สุดท้ายที่มีจะแม่นยำมากกว่าการพยากรณ์ของโมเดลเดี่ยวเพียงรายการ อย่างไรก็ตามการฝึกโมเดลหลายรูปแบบต้องใช้ทรัพยากรในการประมวลผลและเวลามากกว่าการฝึกโมเดลเดี่ยวซึ่งอาจเป็นข้อเสียสำคัญในสภาพแวดล้อมที่มีทรัพยากรจำกัด

วิธีการดั้งเดิมใช้วิธีการเดี่ยวคือ single-feature methodologies ซึ่งอาจไม่สามารถจับตัวอย่างได้ที่หลากหลายของการแสดงท่าทาง หากต้องการแก้ไขปัญหาการใช้ข้อมูลผสมในโมเดลการเรียนรู้เครื่องเสนอวิธีการที่ครอบคลุมการวิเคราะห์ข้อมูลโดยรวมโดยรวมชนิดของข้อมูลที่หลากหลาย เช่น ข้อมูลตัวเลข, ข้อมูลหมวดหมู่, ข้อมูลข้อความ, และข้อมูลภาพ ลงในโมเดลทำนายเดี่ยว

วิธีการนี้ใช้ประโยชน์จากความแข็งแกร่งและข้อมูลที่ไม่เหมือนกันที่มีอยู่ภายในรูปแบบข้อมูลที่แตกต่างกัน มีการนำเสนอกรอบการเรียนรู้ subnetworks ที่ลึกและหลากหลาย ผลการทดลองแสดงให้เห็นการปรับปรุงโดยการทำให้ subnetworks แข็งแรงและหลากหลายมากขึ้น แสดงผลการปรับปรุงในการจำแนกภาพบนชุดข้อมูล CIFAR-100 และ Tiny ImageNet

ดังนั้น เอกสารนี้นำเสนอโมเดลการเรียนรู้เชิงลึกที่ผสมคุณลักษณะสำหรับการจำแนกภาษามือ โมเดล CNN เหมาะสำหรับการสกัดพื้นที่และคุณลักษณะจากข้อมูลภาพ เพื่อให้ได้การวิเคราะห์ที่แข็งแกร่งของลักษณะทางกายภาพของท่าทางมือ ในเวลาเดียวกันโมเดล LSTM จะประมวลผลข้อมูลที่เกี่ยวข้องกับฮิสโตแกรมให้ได้ข้อมูลเวลาที่เปลี่ยนไปตามการสื่อสารภาษามือ วิธีการที่นำเสนอลดความจำเป็นในการเพิ่มข้อมูลอย่างมาก ทำให้กระบวนการฝึกอบรมเป็นไปอย่างราบรื่น และเสถียรภาพโมเดลโดยรวมที่แสดงให้เห็นได้ใน F1-score matrix และ confusion matrix งานนี้ยังเสนอระบบการจำแนกภาษามือที่สามารถเข้าถึงและใช้งานได้ง่ายและมีประสิทธิภาพโดยเฉพาะสำหรับการใช้งานแบบ real-time

1.2 ความสำคัญของการวิจัย

1.2.1 เพื่อพัฒนาระบบการจำแนกภาษามือที่สามารถจดจำและจำแนกท่าทางสัญลักษณ์ที่หลากหลายจากตัวอักษรภาษามืออเมริกัน (ASL) ได้อย่างแม่นยำ

1.2.2 เพื่อตรวจสอบประสิทธิภาพของเทคนิคการเรียนรู้เชิงลึก รวมถึงโครงข่ายประสาทเทียมแบบหมุนวน (CNN) เครือข่ายหน่วยความจำระยะสั้นระยะยาว (LSTM) และแบบจำลองแบบผสมในการจับภาพลักษณะเชิงพื้นที่และเชิงเวลาของท่าทางสัญลักษณ์

1.2.3 เพื่อลดขนาดของ model ที่ฝึกฝนสำหรับการนำไปใช้ในอุปกรณ์ที่มีหน่วยความจำที่จำกัด

1.3 วัตถุประสงค์ของการวิจัย

1.3.1 ปัญญาประดิษฐ์ที่ได้พัฒนาขึ้นไม่สามารถตรวจจับภาพอื่นๆ ได้นอกจากมือ

1.3.2 ปัญญาประดิษฐ์ที่ได้พัฒนาขึ้นจำแนกได้เฉพาะภาษามืออเมริกันเท่านั้น

1.3.3 ไม่สามารถตรวจภาพเคลื่อนไหวได้ ต้องเป็นรูปเท่านั้น

1.3.4 ต้องเป็นภาษา

1.3.5 ข้อมูลจำเพาะ(Google Colab)

GPU: NVIDIA Tesla T4

CUDA Cores: 2560

GPU Memory: 16 GB GDDR6

Memory Bandwidth: 320 GB/s

Compute Capability: 7.5

CPU: Intel Xeon Processor

RAM: Typically ranges from 12.7 GB

Storage: 78.2 GB

1.4 ขอบเขตของการวิจัย

1.4.1 การพัฒนาระบบการจำแนกภาษามือที่แม่นยำและมีประสิทธิภาพสามารถปรับปรุงการเข้าถึงสำหรับชุมชนคนหูหนวก

1.4.2 ลดค่าใช้จ่ายจากอุปกรณ์ที่ทำนายตัวอักษรที่มีราคาสูง





บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะนำเสนอทฤษฎีที่เกี่ยวข้องกับแนวทางการพัฒนาการเรียนรู้เชิงลึกโดยใช้คุณลักษณะแบบผสมเพื่อการจำแนกภาษามือ และการทบทวนวรรณกรรม ผู้วิจัยได้ทำการศึกษาเอกสาร ทฤษฎี และงานวิจัยต่างๆ ที่เกี่ยวข้อง โดยทฤษฎีที่เกี่ยวข้องมีดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ภาษามือ (American hand sign) [1]

ภาษามือ, โดยเฉพาะภาษามืออเมริกัน, เป็นวิธีการสื่อสารที่สำคัญสำหรับชุมชนคนหูหนวกและคนหูตึง มันประกอบด้วยการใช้หลายลักษณะของมือ เคลื่อนไหว และการแสดงอารมณ์ใบหน้าเพื่อสื่อความหมาย การเข้าใจความซับซ้อนของภาษามืออเมริกันเป็นสิ่งสำคัญสำหรับการพัฒนาระบบจำแนกภาษามือที่แม่นยำและมีประสิทธิภาพ ในปีหลังนี้ การผสมเทคโนโลยีการเรียนรู้ของเครื่องจักร โดยเฉพาะการใช้ระบบการเรียนรู้เชิงลึกได้เปลี่ยนแปลงวิธีการจำแนกภาษามือโดยมอบความแม่นยำและประสิทธิภาพสูงในการระบุและแปลความหมายของเครื่องหมายมือ วิทยานิพนธ์นี้มีวัตถุประสงค์ที่จะสำรวจหลักการทฤษฎีและการประยุกต์ใช้ของการเรียนรู้ของเครื่องจักรโดยเฉพาะการใช้โครงสร้างของระบบการเรียนรู้เชิงลึกเช่น Convolutional Neural Networks (CNNs) และ Long Short-Term Memory (LSTM) networks รวมทั้งวิธีการผสมโมเดลที่นวัตกรรมขึ้น เพื่อที่จะนำมาใช้ในการจำแนกภาษามือ นอกจากนี้ วิทยานิพนธ์นี้ยังสำรวจงานวิจัยที่เกี่ยวข้องในสาขาวิชาภาษามือ โดยเน้นไปที่ความก้าวหน้า ความท้าทาย และทิศทางในอนาคต

A. องค์ประกอบของภาษามืออเมริกัน (American hand sign) [2]

ภาษามือแบบอเมริกัน (ASL) เป็นภาษาที่มีชีวิตชีวาและสมบูรณ์ซึ่งอาศัยการผสมผสานระหว่างสัญญาณที่แสดงออก การแสดงออกทางสีหน้า และการเคลื่อนไหวร่างกายในการสื่อความหมาย ป้ายด้วยมือซึ่งประกอบขึ้นจากรูปทรงมือ การเคลื่อนไหว และตำแหน่ง แสดงถึงคำวลี และแนวคิดภายในตำแหน่งเฉพาะด้านหน้าร่างกายหรือภายในพื้นที่ลงนามของผู้ลงนาม สิ่งสำคัญที่สุดคือการแสดงออกทางสีหน้ามีบทบาทสำคัญใน ASL โดยถ่ายทอดข้อมูลทางไวยากรณ์ ความแตกต่างเล็กน้อย และอารมณ์ สำนวนเหล่านี้เกี่ยวข้องกับการเคลื่อนไหวของคิ้ว ตา แก้ม และปาก ซึ่งสามารถเปลี่ยนแปลงความหมายของสัญลักษณ์หรือประโยคได้ นอกจากนี้ ASL ยังรวมสัญญาณที่ไม่ใช่แบบแมนนวล เช่น การเคลื่อนไหวของศีรษะ ท่าทางของร่างกาย และการขยับไหล่ เพื่อปรับปรุงไวยากรณ์ ความชัดเจน และการเน้นย้ำ การอ้างอิงเชิงพื้นที่เป็นพื้นฐานใน ASL โดยที่มือที่โดดเด่นของผู้ลงนามมักจะแสดงถึงหัวข้อหรือจุดเน้นหลักของ

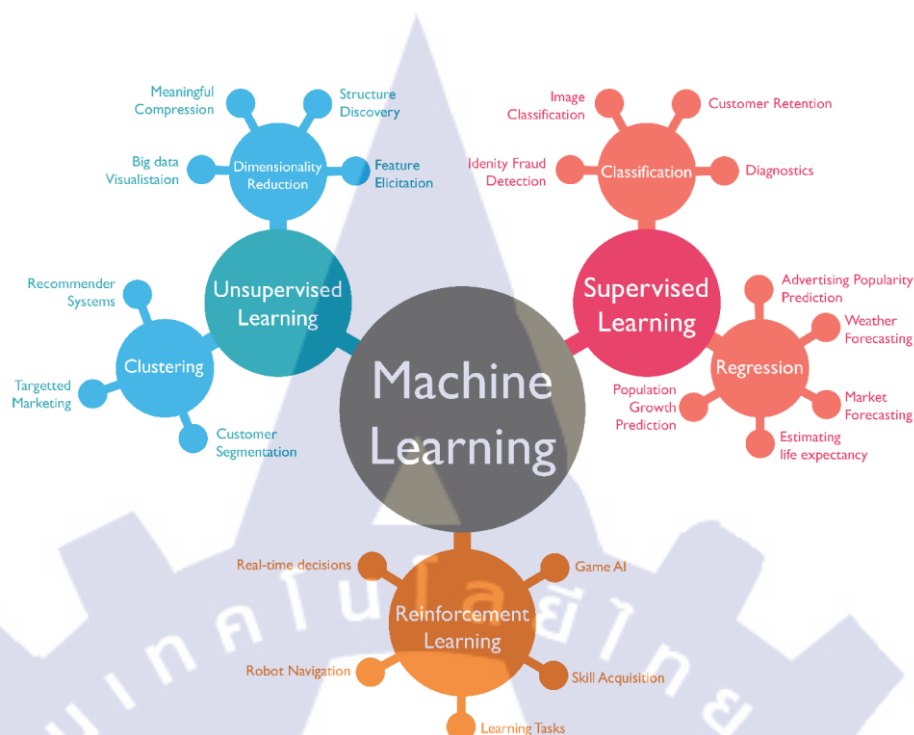
ประโยค และทิศทางภายในพื้นที่ลงนามบ่งบอกถึงความสัมพันธ์ระหว่างเอนทิตีและการกระทำ ASL ยังรวมถึงท่าทางและสัญลักษณ์ที่เป็นสัญลักษณ์ด้วย โดยที่สัญญาณบางอย่างแสดงถึงแนวคิดที่สื่อออกมาเป็นภาพ ซึ่งช่วยให้เกิดความเข้าใจ ด้วยรูปแบบไวยากรณ์ ไวยากรณ์ และวัฒนธรรมที่แตกต่างกัน ASL ทำหน้าที่เป็นสื่อกลางในการสื่อสารและอัตลักษณ์ที่มีชีวิตชีวาภายในชุมชนคนหูหนวก ครอบคลุมรูปแบบวรรณกรรมและการแสดงออกทางศิลปะที่หลากหลาย



รูปที่ 2.1 แสดงรูปแบบของสัญลักษณ์มือที่แทนตัวอักษรภาษาอังกฤษทั้ง 44 ตัว [3]

2.1.2 ทฤษฎีการเรียนรู้ของเครื่องจักร (Machine Learning) [4]

การเรียนรู้ของเครื่องเป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่เน้นการพัฒนาอัลกอริทึมและโมเดลที่สามารถเรียนรู้จากข้อมูลและทำการพยากรณ์หรือตัดสินใจโดยไม่ต้องโปรแกรมอย่างชัดเจน มันรวมถึงเทคนิคและวิธีการต่างๆ ที่เหมาะสมกับประเภทของข้อมูลและงานที่แตกต่างกัน ประเภทหลักๆ ของการเรียนรู้ของเครื่องมี การเรียนรู้แบบมีผู้สอน (Supervised Learning), การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) และ การเรียนรู้แบบการเสริม (Reinforcement Learning)

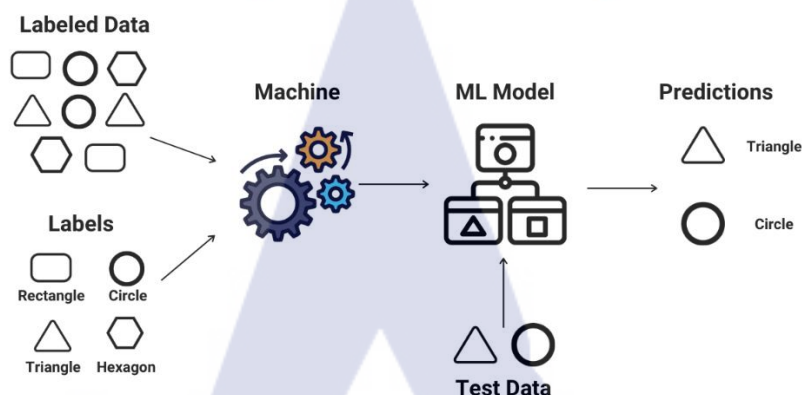


รูปที่ 2.2 แสดงประเภทของ Machine learning [5]

A. การเรียนรู้แบบมีผู้สอน (Supervised Learning)

การเรียนรู้แบบมีผู้สอนเป็นการฝึกโมเดลบนชุดข้อมูลที่มีป้ายชื่อ (labeled dataset) โดยที่แต่ละข้อมูลนั้นมีการแบ่งกลุ่มหรือการติดป้ายไว้ล่วงหน้า จุดมุ่งหมายคือให้โมเดลเรียนรู้การแมพ (mapping) ระหว่างข้อมูลนำเข้ากับผลลัพธ์หรือเป้าหมาย เพื่อให้โมเดลจะสามารถทำการทำนายได้ถูกต้องบนข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน งานที่พบบ่อยในการเรียนรู้แบบมีผู้สอนรวมถึงการจำแนกประเภทและการสร้างรูปภาพ เมื่อพูดถึงการจำแนกประเภทในบริบทของการจำแนกภาษามือ อัลกอริทึมการเรียนรู้แบบมีผู้สอนสามารถฝึกโมเดลบนชุดข้อมูลที่มีป้ายชื่อของภาษามือเพื่อระบุและจำแนกสัญญาณมือต่างๆ

Supervised Learning

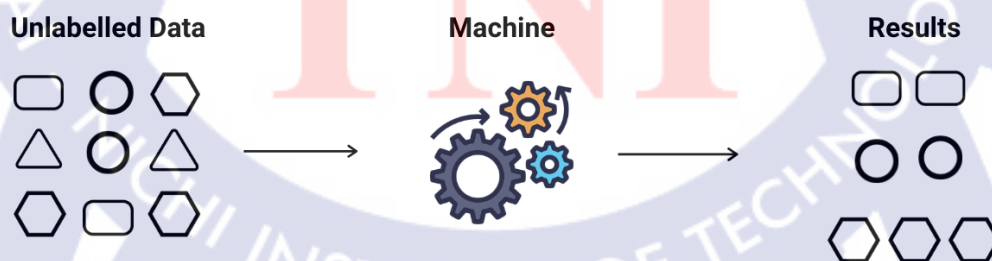


รูปที่ 2.3 แสดงกระบวนการเรียนรู้ของ Supervised learning [6]

B. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

การเรียนรู้แบบไม่มีผู้สอนเกี่ยวกับการฝึกโมเดลบนชุดข้อมูลที่ไม่มีป้ายชื่อ โดยที่เป้าหมายคือการค้นหารูปแบบ โครงสร้าง หรือความสัมพันธ์ในข้อมูลโดยไม่มีการสอนแบบชัดเจน การแบ่งกลุ่มและการลดขนาดของมิติ (dimensionality reduction) เป็นงานที่พบในการเรียนรู้แบบไม่มีผู้สอน ในบริบทของการจำแนกภาษามือ เทคนิคการเรียนรู้แบบไม่มีผู้สอนอาจถูกใช้เพื่อสำรวจโครงสร้างที่มีอยู่ในภาษามือและหาความคล้ายคลึงหรือการจัดกลุ่มของสัญญาณมือต่างๆ

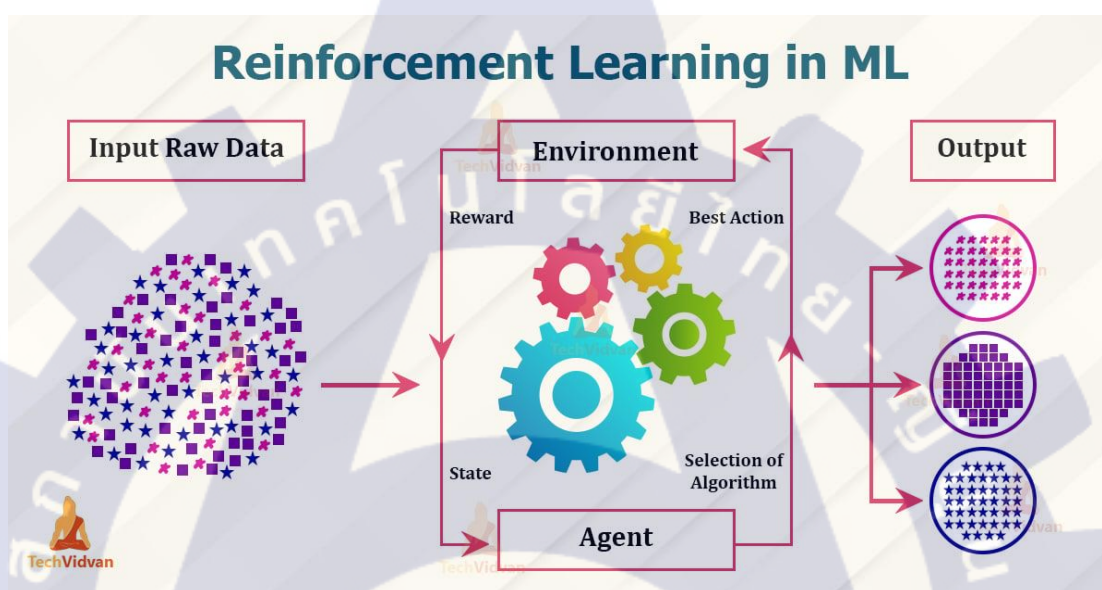
Unsupervised Learning



รูปที่ 2.4 แสดงกระบวนการเรียนรู้ของ Unsupervised learning [6]

C. การเรียนรู้แบบการเสริม (Reinforcement Learning)

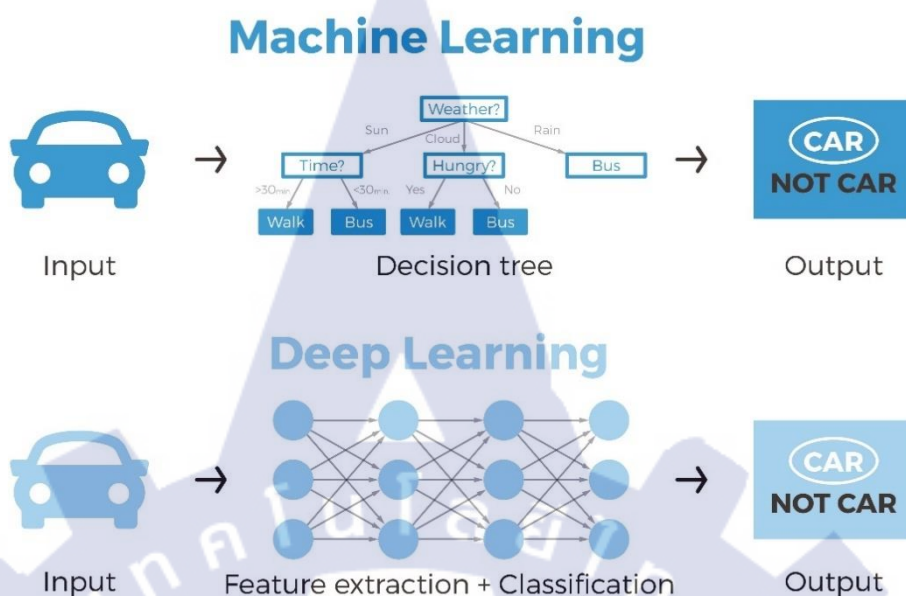
การเรียนรู้แบบการเสริมเป็นการฝึกตัวแทนให้ปฏิสัมพันธ์กับสิ่งแวดล้อมเพื่อสร้างการกระทำที่สามารถสร้างรางวัลรวมเป็นสูงสุด ตัวแทนจะเรียนรู้ผ่านการลองผิดลองถูกโดยรับข้อมูลตอบกลับจากสิ่งแวดล้อมในรูปแบบของรางวัลหรือโทษ การเรียนรู้แบบการเสริมมักถูกใช้ในสภาพแวดล้อมที่เป็นไปได้และมีการปฏิสัมพันธ์ เช่น การแปลงภาษามือหรือแอปพลิเคชันการสื่อสารที่เกี่ยวข้องกับภาษามือ



รูปที่ 2.5 แสดงกระบวนการเรียนรู้ของ Reinforcement Learning[7]

D. การเรียนรู้แบบเชิงลึก (Deep Learning) [4]

การเรียนรู้ลึกเป็นส่วนหนึ่งของการเรียนรู้ของเครื่องที่เน้นการพัฒนาโมเดลเชิงลึกที่มีชั้นของความเสมอภายใน โมเดลการเรียนรู้ลึก เช่น โครงข่ายประสาทเทียมคอนโวลูชัน (Convolutional Neural Networks: CNNs) และ โครงข่ายประสาทเทียมแบบรีเคอร์เรนท์ (Recurrent Neural Networks: RNNs) ได้รับความสำเร็จที่น่าทึ่งในหลายด้าน เช่น การมองเห็นด้านคอมพิวเตอร์ การประมวลผลภาษาธรรมชาติและการรับรู้เสียง ในบริบทของการจำแนกภาษามือ โมเดลการเรียนรู้ลึกเหมาะสมอย่างยิ่งสำหรับการจับตามรูปแบบของสัญญาณมือที่ซับซ้อน ทำให้สามารถเรียนรู้และจำแนกสัญญาณมือต่างๆ ได้อย่างมีประสิทธิภาพและแข็งแกร่ง



รูปที่ 2.6 แสดงการเปรียบเทียบระหว่าง Machine Learning และ Deep Learning [8]

2.1.3 Convolutional Neural Networks (CNNs)[9]

เป็นกลุ่มของโครงข่ายประสาทเทียมลึกที่ออกแบบมาเฉพาะสำหรับการประมวลผลข้อมูลแบบตารางที่มีโครงสร้างเช่นภาพ พวกเขาได้เปลี่ยนแปลงงานต่าง ๆ ในวิทัศน์คอมพิวเตอร์เช่นการจำแนกภาพ การตรวจจับวัตถุ และการเส็กเมนเทชัน CNNs ได้รับแรงบันดาลใจมาจากสมองทางสายตาของมนุษย์และประกอบด้วยชั้นหลาย ๆ ชั้นที่ประมวลผลข้อมูลนำเข้าต่าง ๆ ชนิดต่าง ๆ

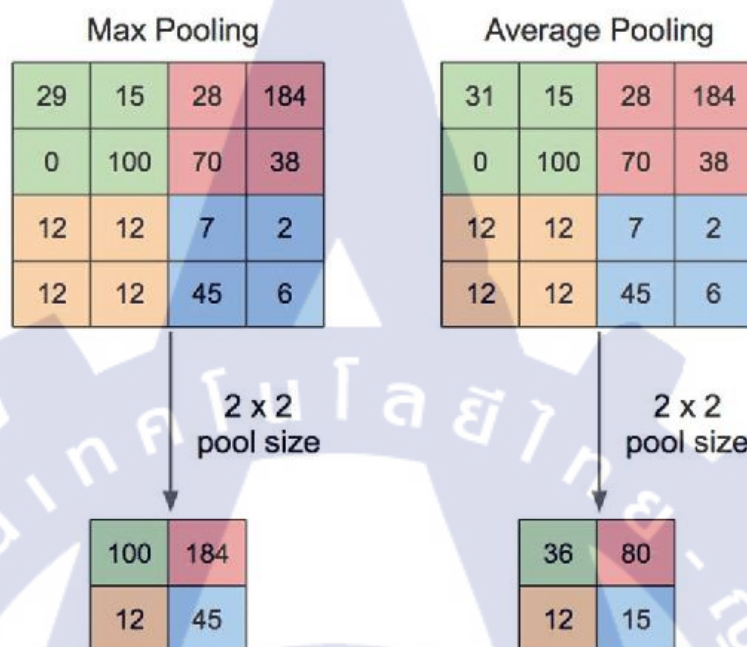
A. Convolutional Layers

โครงสร้างหลักของ CNN คือชั้นแบบหมุนวน(convolutional layers) ซึ่งใช้ชุดตัวกรองที่เรียนรู้ได้ (หรือที่เรียกว่าเคอร์เนล) กับรูปภาพอินพุต ฟิลเตอร์เหล่านี้จะวนเวียนอยู่กับภาพที่นำเข้า โดยแยกคุณสมบัติต่างๆ เช่น ขอบ พื้นผิว และลวดลาย เลเยอร์แบบหมุนวนมีลักษณะเฉพาะด้วยพารามิเตอร์ เช่น ขนาดตัวกรอง การก้าวย่าง และช่องว่างภายใน ซึ่งกำหนดขนาดเชิงพื้นที่ของแผนผังคุณลักษณะเอาต์พุต

B. Pooling Layers

เลเยอร์พูลลิ่ง(Pooling Layers)ถูกใช้เพื่อลดขนาดเชิงพื้นที่ของแผนผังคุณลักษณะ(features map)ในขณะที่ยังคงรักษาข้อมูลที่สำคัญที่สุดไว้แบ่งออกเป็น 2 ประเภทได้แก่
1.การพูลลิ่งสูงสุด(Max Pooling)เลือกค่าสูงสุดจากทุก ๆ พื้นที่ 2.การพูลลิ่งเฉลี่ย(Average Pooling)

คำนวณค่าเฉลี่ยจากทุก ๆ พื้นที่ การพูลลิ่งช่วยให้การแสดงผลที่เรียนรู้โดยเครื่องมากขึ้นและมั่นใจมากขึ้นต่อการเปลี่ยนแปลงขนาดเล็ก ๆ และการบิดเบือนในภาพนำเข้า



รูปที่ 2.7 แสดงการเปรียบเทียบระหว่าง Max Pooling และ Average Pooling [10]

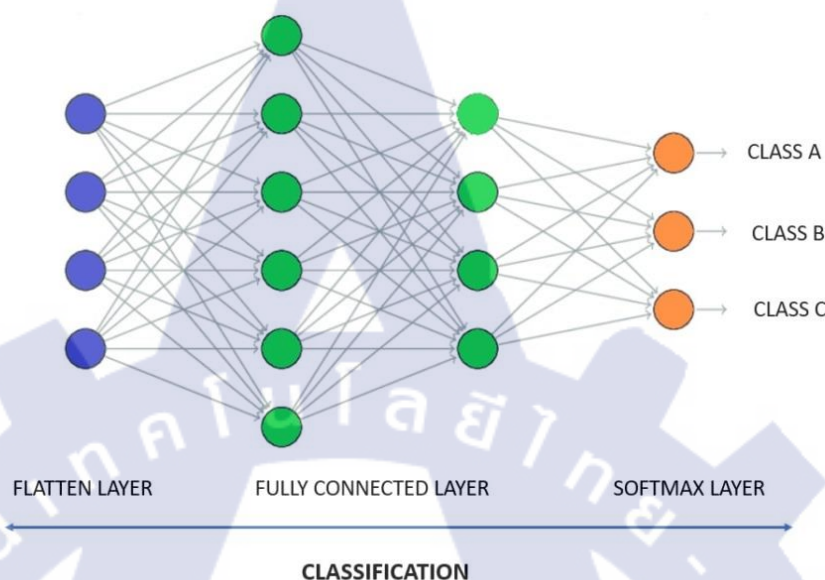
C. Activation Functions:

ฟังก์ชันการกระตุ้น(Activation Functions)ทำให้เกิดความไม่เป็นเชิงเส้นใน CNN ซึ่งทำให้เครื่องเรียนรู้เกิดความสัมพันธ์ที่ซับซ้อนและทำการแปลงที่ไม่เป็นเชิงเส้น ฟังก์ชันการกระตุ้นที่ได้รับความนิยมรวมถึง ReLU (Rectified Linear Unit), sigmoid, และ tanh ReLU ได้รับการใช้งานอย่างกว้างขวางใน CNNs เนื่องจากความง่ายและประสิทธิภาพในการลดปัญหาการไล่ระดับสีที่หายไป

D. Fully Connected Layers [11]

ในตอนท้ายของสถาปัตยกรรม CNN โดยทั่วไปจะใช้เลเยอร์ที่เชื่อมต่อกันอย่างสมบูรณ์ตั้งแต่หนึ่งเลเยอร์ขึ้นไปเพื่อดำเนินการให้เหตุผลระดับสูง(high-level reasoning) และการตัดสินใจ(decision-making)ตามคุณสมบัติที่แยกออกมา เลเยอร์เหล่านี้เชื่อมต่อเซลล์ประสาท(neuron) ทุกตัวในชั้นหนึ่งกับทุกเซลล์ประสาทในชั้นถัดไป ช่วยให้เครือข่ายสามารถเรียนรู้การแสดงผลข้อมูลอินพุตแบบลำดับชั้นที่ซับซ้อนได้ โดยทั่วไปแล้วเลเยอร์ที่เชื่อมต่อกันอย่างสมบูรณ์จะตามด้วย

ฟังก์ชันการเปิดใช้งาน softmax ในงานการจำแนกประเภทเพื่อสร้างการแจกแจงความน่าจะเป็นเหนือคลาสเอาต์พุต



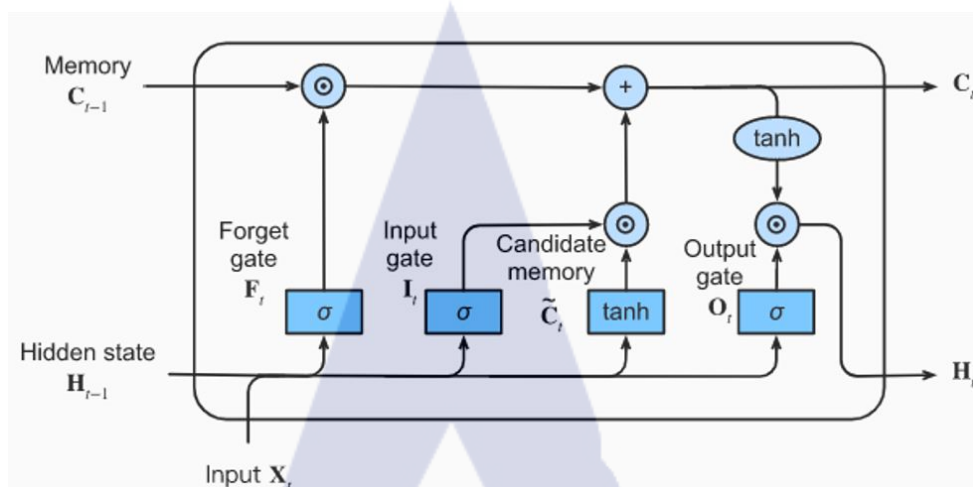
รูปที่ 2.8 แสดงการทำงานของ Softmax layer สำหรับการทำนาย [12]

2.1.4 Long Short-Term Memory [13]

Long Short-Term Memory (LSTM) เป็นสถาปัตยกรรมโครงข่ายประสาทเทียม (RNN) ชนิดหนึ่งที่ถูกออกแบบมาเพื่อเอาชนะข้อจำกัดของ RNN แบบดั้งเดิมในการจับภาพการพึ่งพาระยะยาวและการจัดการการไล่ระดับสีที่หายไป LSTM มีประสิทธิภาพเป็นพิเศษในการประมวลผลและการเรียนรู้จากข้อมูลตามลำดับ ทำให้เหมาะสำหรับงานต่างๆ เช่น การรู้จำคำพูด การประมวลผลภาษาธรรมชาติ และในบริบทของการจำแนกภาษามือ การจับการเปลี่ยนแปลงชั่วคราวของท่าทางสัญญาณ

A. โครงสร้างของ LSTM

นิวตกรรมที่สำคัญของเครือข่าย LSTM อยู่ที่โครงสร้างเซลล์แบบมีประตู ซึ่งช่วยให้เครือข่ายสามารถเลือกเก็บหรือลืมข้อมูลเมื่อเวลาผ่านไปได้ เซลล์ LSTM ประกอบด้วยองค์ประกอบหลักสามส่วน ได้แก่ ประตูอินพุต(input gate) ประตูลืม(forget gate) และประตูผลลัพธ์(output gate) ซึ่งแต่ละส่วนควบคุมโดยฟังก์ชันการเปิดใช้งานซิกมอยด์(sigmoid activation functions) และมาพร้อมกับฟังก์ชันการเปิดใช้งาน Tanh เพื่อควบคุมการไหลของข้อมูล



รูปที่ 2.9 แสดงโครงสร้างภายในของ Long Short-Term Memory [14]

ประตูนำเข้า (Input Gate): ประตูนี้จะกำหนดจำนวนข้อมูลใหม่ที่จะอนุญาตให้เข้าสู่สถานะเซลล์ โดยจะรวมข้อมูลจากอินพุตปัจจุบันและสถานะที่ซ่อนอยู่ก่อนหน้านี้ โดยส่งข้อมูลที่เกี่ยวข้องไปยังสถานะเซลล์เพื่อจัดเก็บ

ประตูลืม (Forget Gate): ประตูนี้จะตัดสินใจว่าข้อมูลใดจากสถานะของเซลล์ก่อนหน้านี้ควรถูกลบทิ้งหรือลืม โดยจะพิจารณาทั้งอินพุตปัจจุบันและสถานะที่ซ่อนอยู่ก่อนหน้านี้ ทำให้ LSTM สามารถเลือกเก็บข้อมูลที่สำคัญในขณะที่ละทิ้งข้อมูลที่ไม่เกี่ยวข้องหรือล้าสมัยไป

สถานะเซลล์ (Cell State): สถานะของเซลล์ทำหน้าที่เป็นสายพานลำเลียงที่นำข้อมูลข้ามขั้นตอนของเวลา ได้รับการแก้ไขโดยประตูนำเข้าและประตูลืม เพื่อให้มั่นใจว่าข้อมูลที่เกี่ยวข้องได้รับการเก็บรักษาและเผยแพร่ไปตามกาลเวลา

ประตูผลลัพธ์ (Output Gate): ประตูผลลัพธ์ควบคุมการไหลของข้อมูลจากสถานะเซลล์ไปยังสถานะที่ซ่อนอยู่ในปัจจุบัน โดยจะกำหนดว่าส่วนของสถานะเซลล์ที่ควรเปิดเผยต่อเอาต์พุตโดยพิจารณาจากอินพุตปัจจุบันและสถานะที่ซ่อนอยู่ก่อนหน้านี้

2.2 การทบทวนวรรณกรรม

ผู้วิจัยได้ทำการทบทวนงานวิจัยที่เกี่ยวข้องโดยการหาข้อมูลจากฐานข้อมูลมาตรฐานต่างๆ และทำการหาบทความต่างๆ ที่เพิ่มเติมที่เกี่ยวข้องกับหัวข้องานวิจัย ซึ่งได้พบงานวิจัยที่เกี่ยวข้อง สรุปได้ตามตารางที่ 2.1 ดังต่อไปนี้

ตารางที่ 2.2 สรุปงานวิจัยที่เกี่ยวข้อง

ลำดับ	ผู้แต่ง	ปีที่พิมพ์	ชื่อเรื่อง
1	M. M. Hasan et al. [15]	2020	Classification of Sign Language Characters by Applying a Deep Convolutional Neural Network
2	B. P. P. Kumar and M. B. Manjunatha [16]	2017	A Hybrid Gesture Recognition Method for American Sign Language
3	A. Mannan et al. [17]	2023	Hypertuned Deep Convolutional Neural Network for Sign Language Recognition
4	S. RAVI et al [18]	2018	Sign language recognition with multi feature fusion and ANN classifier

M. M. Hasan และคณะ[15] ได้ทำการแนะนำโมเดลโครงข่ายประสาทเทียมเชิงลึก (CNN) สำหรับการรู้จำภาษามือ โดยเน้นไปที่ภาษามือแบบอเมริกัน (ASL) โดยเฉพาะ ด้วยชุดข้อมูลของท่าทางตัวอักษร ASL โมเดลที่นำเสนอได้รับความแม่นยำโดยรวมที่น่าประทับใจถึง 97.62% ซึ่งมีประสิทธิภาพเหนือกว่าวิธีการก่อนหน้านี้ เช่น CNN, Support Vector Machines, Random Forest และ Capsule Networks สถาปัตยกรรมแบบจำลองประกอบด้วยชั้น Convolutional หกชั้น ตามด้วยชั้นรวมสามชั้น โดยมีการใช้ฟังก์ชันการเปิดใช้งาน ReLU เครือข่ายได้รับการฝึกอบรมเป็นเวลา 50 ครั้ง โดยใช้เครื่องมือเพิ่มประสิทธิภาพ Adam ด้วยอัตราการเรียนรู้ 0.0001 และการสูญเสียเอนโทรปีข้ามหมวดหมู่ การประมวลผลล่วงหน้าเกี่ยวข้องกับการแปลงภาพ RGB เป็นระดับสีเทาและปรับขนาดเป็นขนาดคงที่ 28x28 พิกเซล มีการใช้การออกกลางคันเพื่อป้องกันไม่ให้เกิดการติดตึงมากเกินไป ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของแบบจำลอง CNN ที่เสนอในการจดจำท่าทางตัวอักษร ASL อย่างถูกต้อง ซึ่งมีส่วนทำให้เกิดความก้าวหน้าในเทคโนโลยีการรู้จำภาษามือ

B. P. P. Kumar และคณะ[16] ได้ศึกษาเกี่ยวกับระบบการรู้จำท่าทางมือแบบใหม่สำหรับภาษามือแบบอเมริกัน (ASL) โดยใช้วิธีไฮบริดที่รวมวิธี Local Binary Pattern (LBP) super pixels (SP) และวิธี Speeded Up Robust Features (SURF) ระบบนี้มีจุดมุ่งหมายเพื่อปรับปรุงการสื่อสารระหว่างคนหูหนวกและผู้ได้ยินผ่านสัญญาณท่าทาง ใช้ตัวแยกประเภท K-Nearest Neighbor (KNN) และ Support Vector Machine (SVM) สำหรับการจดจำท่าทางด้วยตัวอักษรเดียว วิธีการนี้เกี่ยวข้องกับการประมวลผลล่วงหน้าและการแบ่งส่วน การตรวจจับและติดตามมือ การจดจำท่าทาง

มือ และการจำแนกท่าทาง ความท้าทายในการจดจำท่าทางรวมถึงการเปลี่ยนแปลงของขนาด ตำแหน่ง การวางแนว การส่องสว่าง และพื้นหลัง ระบบที่นำเสนอจัดการกับความท้าทายเหล่านี้โดยการแยกคุณลักษณะต่างๆ และใช้เทคนิคการจำแนกประเภทที่มีประสิทธิภาพ ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของวิธีการดังกล่าว โดยมีความแม่นยำสูงในการจดจำท่าทาง ASL สถาปัตยกรรมของระบบรวมการรับภาพแบบเรียลไทม์ การประมวลผลล่วงหน้า การแยกคุณสมบัติ การจัดหมวดหมู่ และการแสดงท่าทางที่จดจำได้ พารามิเตอร์ประสิทธิภาพ เช่น ความแม่นยำ ความไว ความแม่นยำ อัตราผลลบลง (FNR) และอัตราการค้นพบลง (FDR) ได้รับการประเมินเพื่อประเมินประสิทธิภาพของระบบ

A. Mannan และคณะ[17] ได้ทำการศึกษาแนวทางการเรียนรู้เชิงลึกโดยใช้โครงข่ายประสาทเทียมเชิงลึก (DeepCNN) เพื่อจดจำท่าทางตัวอักษรภาษามืออเมริกัน (ASL) การรับรู้ ASL เผชิญกับความท้าทายเนื่องจากความคล้ายคลึงและความซับซ้อนระหว่างคลาส โมเดล DeepCNN ที่นำเสนอได้รับการปรับปรุงด้วยการเพิ่มข้อมูล โดยขยายชุดข้อมูลการฝึกอบรมแบบเทียม ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพที่เพิ่มขึ้นอย่างต่อเนื่อง โดยมีการปรับปรุงความแม่นยำสูงสุดถึง 19.84% เมื่อเทียบกับแนวทางที่มีอยู่ ชุดข้อมูลที่ใช้เป็นชุดข้อมูล Modified National Institute of Standards and Technology (MNIST) ซึ่งมีตัวอักษร ASL สถาปัตยกรรม DeepCNN ประกอบด้วย multiple convolutional layers, max-pooling, dropout, และ dense layers การฝึกอบรมและการตรวจสอบแสดงให้เห็นถึงการปรับปรุงอย่างต่อเนื่อง โดยมี validation accuracy ในการตรวจสอบสูงสุดถึง 99.96% การทดสอบข้อมูลที่มองไม่เห็นมีความแม่นยำ 99.67% การวิเคราะห์เปรียบเทียบแสดงให้เห็นถึงประสิทธิภาพที่เหนือกว่าเมื่อเปรียบเทียบกับวิธีการพื้นฐานต่างๆ โดยมีความแม่นยำเพิ่มขึ้นตั้งแต่ 3.26% ถึง 19.84% แนวทางที่นำเสนอแสดงให้เห็นถึงคำแนะนำสำหรับการรับรู้ ASL ที่แข็งแกร่ง โดยจัดการกับความท้าทายในการใช้งานในชีวิตจริง

S. RAVI และคณะ[18] ได้ทำการศึกษาการจดจำการเคลื่อนไหวที่ซับซ้อนของมนุษย์ โดยเฉพาะอย่างยิ่งท่าทาง Indian sign language (ISL) จากลำดับวิดีโอ เสนออัลกอริทึมการแบ่งส่วนที่รวมคุณสมบัติจาก discrete wavelet transform (DWT) และ local binary pattern (LBP) เพื่อแยกกรูปร่างมือจากเฟรมวิดีโอที่ต่อเนื่องกัน คุณสมบัติที่แยกออกมาได้รับการตรวจสอบโดยใช้ artificial neural network (ANN) ซึ่งมีประสิทธิภาพเหนือกว่าคุณสมบัติล้ำสมัยอื่นๆ เช่น HOG, SIFT และ SURF วิธีการที่นำเสนอนี้มีอัตราการรับรู้ถึง 92.79% ในชุดข้อมูล ISL ซึ่งเหนือกว่างานที่มีอยู่ วิธีการของบทความนี้เกี่ยวข้องกับการระบุผู้ลงนาม การแยกคุณลักษณะโดยใช้ DWT และ LBP การผสมผสานคุณลักษณะของ Haar และ LBP และการจำแนกประเภทโดยใช้ ANN ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของแนวทางที่นำเสนอ โดยพีเจอร์ DLBP มีอัตราการรับรู้สูงสุดที่ 95.49%

นอกจากนี้ การเปรียบเทียบกับโมเดลคุณลักษณะและตัวแยกประเภทอื่น ๆ จะตรวจสอบความเหนือกว่าของวิธีการที่เสนอ



วิธีการดำเนินงาน

ในบทนี้ผู้วิจัยจะอธิบายถึงขั้นตอนการดำเนินการวิจัยในขั้นตอนและแนวทางการดำเนินการเพื่อให้บรรลุวัตถุประสงค์ของการศึกษา ส่วนนี้ให้คำอธิบายโดยละเอียดเกี่ยวกับการรวบรวมข้อมูลการประมวลผลล่วงหน้า การพัฒนาแบบจำลอง การฝึกอบรม การประเมิน และวิธีการวิเคราะห์ที่ใช้ในการวิจัย

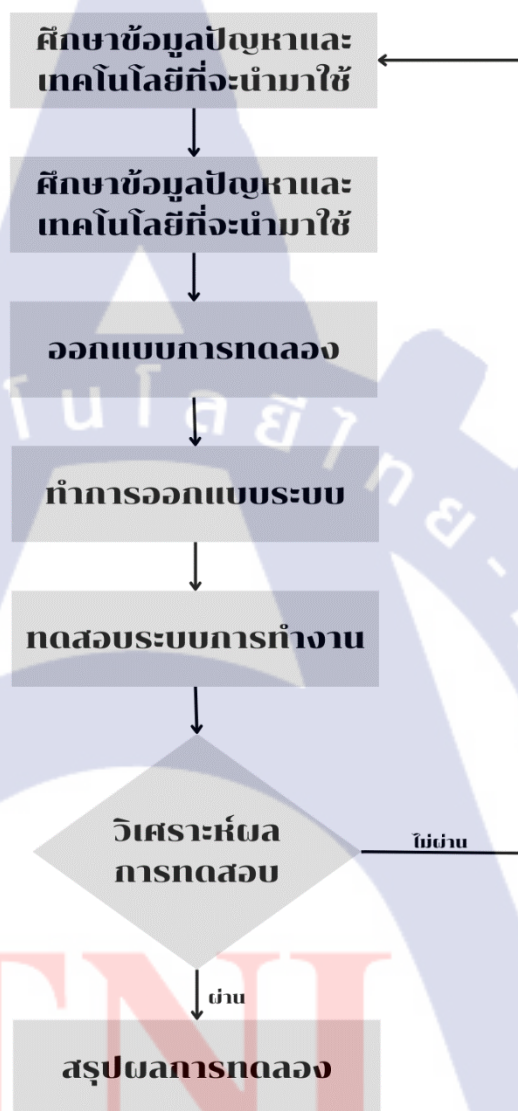
3.1 แผนการดำเนินงานวิจัย

ตารางที่ 3.1 แผนงานและระยะเวลาในการดำเนินงาน

หัวข้องาน	2566												2567					
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6
1.ศึกษาปัญหา และเทคโนโลยี																		
2.ศึกษาทฤษฎีและงานวิจัย																		
3.ออกแบบระบบการทำงาน																		
4.สออบโครงร่างวิทยานิพนธ์																		
5.พัฒนาระบบการทำงาน																		
6.ทดสอบระบบการทำงาน																		
7.วิเคราะห์ผลที่ได้																		
8.สรุปผลและทำเล่มวิทยานิพนธ์																		
9.สอบป้องกันวิทยานิพนธ์																		
10.นำเสนอผลงานทางวิชาการ																		

3.2 ขั้นตอนการดำเนินงาน

ผู้วิจัยได้ทำการศึกษา และหาข้อมูลของงานวิจัย โดยมีขั้นตอนการดำเนินงานดังนี้



รูปที่ 3.1 แผนผังขั้นตอนการดำเนินงาน

3.3 การออกแบบระบบ

ในการดำเนินงานวิจัยนี้ทางผู้จัดทำได้ทำการเตรียมชุดข้อมูลครอบคลุมท่าทางตัวอักษรที่หลากหลายในภาษามืออเมริกัน (ASL) ยกเว้นตัวอักษร 'J' และ 'Z' จำนวน 27,455 รูป จะแบ่งออกเป็นชุดการฝึกอบรม(training)และชุดตรวจสอบ(Validate) ชุดฝึกประกอบด้วย 80% ของตัวอย่างทั้งหมด ในขณะที่ชุดตรวจสอบประกอบด้วย 20% ซึ่งข้อมูลที่ใช้ทดสอบ(Test) นั้นจะแยกออกมาจำนวน 7,172 รูป ซึ่งการทำสอบนี้จะทำการวัดค่า Accuracy, precision, recall, และ F1-score.

3.3.1 Accuracy

เป็นค่าความแม่นยำจากการทดสอบซึ่งเป็นที่นิยมใช้กันมากที่สุดในปัจจุบันในการทดสอบ Model ของ Machine Learning ต่างๆ บนโลกนี้ ต่างก็จะนิยมใช้ค่านี้เป็นหลักเพื่อบ่งชี้ถึงประสิทธิภาพของ Model ที่ได้พัฒนาขึ้น

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

โดยที่ค่า True Positive คือ จำนวนข้อมูลที่ถูกจำแนกได้อย่างถูกต้องในกลุ่มของประเภทที่พวกเราต้องการ และค่า True Negative คือ จำนวนข้อมูลที่ถูกจำแนกถูกต้องในกลุ่มของประเภทที่ไม่ใช่เป้าหมาย

3.3.2 Precision

เป็นค่าความแม่นยำที่ใช้ในกรณีที่เกิดเหตุการณ์ False Positive มีปริมาณมาก โดยเฉพาะกับเหตุการณ์ที่ traffic บน server สูง แต่เป็น traffic แบบปกติธรรมดาทั่วไปไม่ใช่การกระทำ ตัว Precision ก็จะสามารถนำมาเป็นตัวชี้วัดว่า model นี้มีความสามารถในการตรวจจับค่า False Positive มากแค่ไหน

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

โดยที่ค่า True Positive คือ จำนวนข้อมูลที่ถูกจำแนกได้อย่างถูกต้องในกลุ่มของประเภทที่พวกเราต้องการ, ค่า True Negative คือ จำนวนข้อมูลที่ถูกจำแนกถูกต้องในกลุ่มของประเภทที่ไม่ใช่เป้าหมาย และค่า False Positives คือจำนวนข้อมูลที่ถูกทำนายผิดในกลุ่มของประเภทที่เราสนใจ

3.3.3 Recall

Recall เป็นค่าความแม่นยำที่ใช้ในกรณีที่เกิด False Negative มีปริมาณจำนวน
มากๆ

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

โดยที่ค่า True Positive คือ จำนวนข้อมูลที่ถูกจำแนกได้อย่างถูกต้องในกลุ่มของ
ประเภทที่พวกเราต้องการ, ค่า True Negative คือ จำนวนข้อมูลที่ถูกจำแนกถูกต้องในกลุ่มของ
ประเภทที่ไม่ใช่เป้าหมาย และค่า False Positives คือจำนวนข้อมูลที่ถูกทำนายผิดในกลุ่มของ
ประเภทที่เราสนใจ

3.3.4 F1-score

F1-score เป็นค่าที่นำ Recall และ Precision มาทำงานร่วมกันโดยที่จะเน้นการวัด
ค่า ความแม่นยำที่เป็น True-Negative เป็นหลัก

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

โดยที่ Precision เป็นค่าความแม่นยำที่ใช้ในกรณีที่เกิดเหตุการณ์ False Positive
และค่า Recall เป็นค่าความแม่นยำที่ใช้ในกรณีที่เกิด False Negative มีปริมาณจำนวนมากๆ

3.4 การทำการทดลอง

ภายในระบบการติดตามนี้ จะแบ่งการทำงานออกเป็น 3 ส่วน คือ การรวบรวมข้อมูล การ
ตั้งค่าการทดลอง และการพัฒนาแบบจำลอง โดยในการเขียนคำสั่งบน Software as a
Service(SaaS) ของ google colab ซึ่งจะอธิบายรายละเอียดดังนี้

3.4.1 การรวบรวมข้อมูล

บทความนี้ได้ทำการรวบรวมชุดข้อมูล MNIST ภาษามือ รวมถึงแหล่งข้อมูลจากเว็บไซต์
Kaggle โดยเกณฑ์ที่ใช้ในการเลือกชุดข้อมูลและขั้นตอนการประมวลผลล่วงหน้าที่ใช้เพื่อให้แน่ใจ
ว่าเหมาะสมกับวัตถุประสงค์การวิจัย

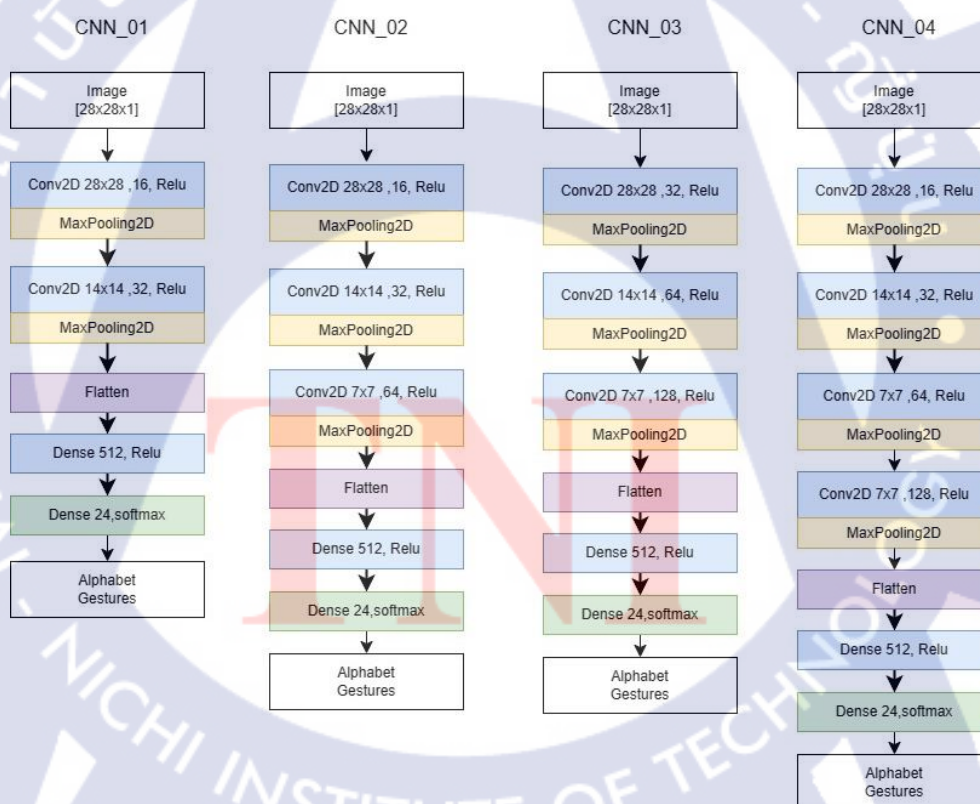
3.4.2 การตั้งค่าการทดลอง

รายละเอียดเกี่ยวกับการตั้งค่าการทดลองที่ใช้ในการฝึกอบรมและประเมินแบบจำลองการจำแนกภาษามือที่พัฒนาขึ้นโดยเป็นส่วนหนึ่งของการวิจัยนี้ ซึ่งรวมถึงข้อมูลจำเพาะของฮาร์ดแวร์ สภาพแวดล้อมซอฟต์แวร์ และทรัพยากรการคำนวณที่ใช้สำหรับการพัฒนาแบบจำลองการฝึกอบรม และการทดลอง

3.4.3 การพัฒนาแบบจำลอง

A. Convolutional Neural Network (CNNs)

จากการศึกษาพบว่า model ที่มีประสิทธิภาพที่สูงในการทำนายผลและมีความนิยมที่สูงคือ คือ โมเดล CNNs โดยจะทำการนำภาพ dataset ที่ได้มาทำการ train กับ model นี้ก่อน โดยเราจะทำการเพิ่มปรับพารามิเตอร์ของโมเดล CNN เพื่อให้ประสิทธิภาพสูงที่สุด โดยทำการออกแบบออกมา 4 รูปแบบ ดังนี้



รูปที่ 3.2 แสดงโมเดล CNN ที่ทำการปรับพารามิเตอร์

- a. โมเดล CNN_01 ที่จะสร้าง 2 convolutional layers, 2 pooling layers, 2 dropout layers และ 2 fully connected layers โดย
- Convolutional layers ทำหน้าที่ดึง features ออกจาก input images,
 - Pooling layers ทำหน้าที่ลดขนาด dimensionality ของ feature maps,
 - Dropout layers ทำหน้าที่ป้องกัน overfitting
 - Fully connected layers ทำหน้าที่ classify images
- b. โมเดล CNN_02 สร้างคล้ายกับโมเดล CNN_01 แต่จะทำการเพิ่ม 1 layer
- c. โมเดล CNN_03 สร้างคล้ายกับโมเดล CNN_02 แต่จะทำการแก้ไขค่า filter โดยเพิ่มขึ้น 2 เท่าทุกๆ layer เมื่อเทียบกับโมเดล CNN_02
- d. โมเดล CNN_04 สร้างคล้ายกับโมเดล CNN_02 แต่จะทำการเพิ่ม 1 layer หลังจากทีนำโมเดลดังกล่าวมาทำการ train แล้ว จะได้ผลลัพธ์ดังนี้

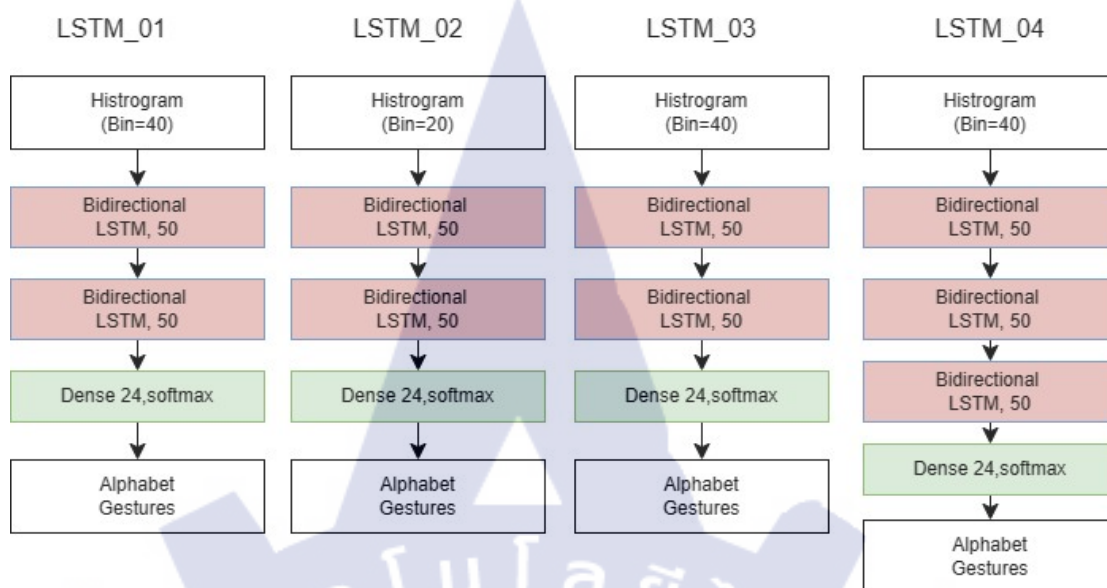
ตารางที่ 3.2 ตารางแสดงประสิทธิภาพของแต่ละโมเดล CNN ที่ epoch = 50

Model	Model size (Mb)	Total params	Accuracy
CNN_01	1.26	820440	0.97
CNN_02	2.65	331032	1
CNN_03	2.65	695320	1
CNN_04	0.69	175512	0.93

ผลลัพธ์ที่ได้จะเห็นว่า CNN_02 และ CNN_03 มี accuracy สูงถึง 1 แต่โมเดล CNN_02 นั้นมีจำนวนพารามิเตอร์ทั้งหมดน้อยกว่าโมเดล CNN_03 ดังนั้นจึงเลือกโมเดล CNN_02 เป็นโมเดลที่ทำการทดสอบ

B. Long short-term memory

เราจะทำการเพิ่มปรับจูนพารามิเตอร์ของโมเดล LSTM เพื่อให้ประสิทธิภาพสูงที่สุด โดยทำการออกแบบออกมา 4 รูปแบบ ดังนี้



รูปที่ 3.3 แสดงโมเดล LSTM ที่ทำการปรับจูนพารามิเตอร์

ค่าที่ใช้ในการ train model เป็นข้อมูลภาพที่แปลงออกมาในรูป ความถี่(histogram) ของสีที่เป็นภาพแบบ grayscale อยู่ระหว่าง 0-255

a. โมเดล LSTM_01 ทำการ normalize ข้อมูลนำเข้าจาก 255 เป็น 40 และที่จะ สร้างจะ 2 Bidirectional LSTM ,และ 2 Dropout โดย

- Bidirection LSTM ชนิดสองทิศทาง เพื่อเรียนรู้ข้อมูลทั้งจากอดีตไปอนาคต และจากอนาคตกลับสู่อดีต
- Dropout เพื่อสุ่มหยุดการทำงานของหน่วยประมวลผลบางหน่วยใน ระหว่างการฝึกฝน เพื่อป้องกันการจดจำรูปแบบข้อมูลมากเกินไปและช่วยเพิ่มความสามารถทั่วไปของ โมเดล

b. โมเดล LSTM_02 ออกแบบคล้ายกับ LSTM_01 แต่ค่าที่ทำกร normalize เปลี่ยนจาก 40 เป็น 20

c. โมเดล LSTM_03 ออกแบบคล้ายกับ LSTM_01 แต่ค่าที่ทำกร normalize เปลี่ยน batch size จาก 32 เป็น 128

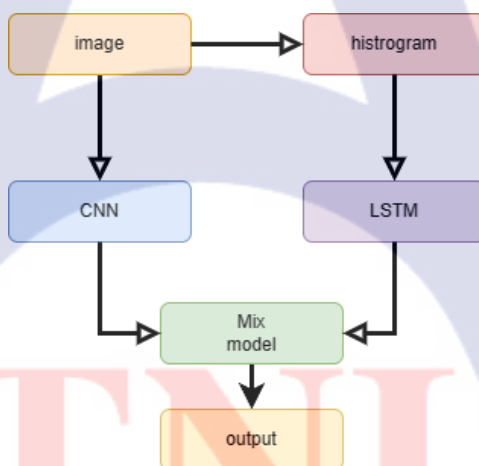
d. โมเดล LSTM_04 ออกแบบคล้ายกับ LSTM_01 แต่ ค่าที่ทำกรเพิ่ม 1 layer

ตารางที่ 3.3 ตารางแสดงประสิทธิภาพของแต่ละโมเดล LSTM ที่ epoch = 50

Model	Model size(KB)	Total param	accuracy
LSTM_01	326.66	83624	0.61
LSTM_02	326.66	83624	0.53
LSTM_03	326.66	83624	0.5
LSTM_04	562.59	144024	0.63

ผลลัพธ์ที่ได้จะเห็นได้ว่า LSTM_04 มี accuracy สูงถึง 0.63 ตามด้วย LSTM_01 อย่างไรก็ตาม ขนาดของโมเดล LSTM_04 มีขนาดใหญ่และจำนวนพารามิเตอร์ที่มากกว่าโมเดลอื่น ดังนั้นงานวิจัยนี้จึงเลือก LSTM_01 ในการทำการ train

C. Mix model



รูปที่ 3.4 แสดงภาพรวมของโมเดลผสม

การจำลองโมเดล Deep Learning ที่ผสมผสาน Convolutional Neural Network (CNN) กับ Long Short-Term Memory (LSTM) โครงสร้างนี้มักใช้ในสถานการณ์ที่ข้อมูลมีทั้งลักษณะเชิงพื้นที่ (คล้ายภาพ) และเชิงเวลา (ลำดับ) และจากทดลองปรับจูนพารามิเตอร์ของแต่ละโมเดลนั้นจะได้โมเดลสุดท้ายดังนี้

a. โมเดล CNN (Convolutional Neural Networks model)

วัตถุประสงค์เพื่อดึงคุณสมบัติจากข้อมูลอินพุต ซึ่งสันนิษฐานว่าเป็นภาพ 2 มิติที่มีช่องสัญญาณเดียว

➤ เลเยอร์:

- Conv2D (3 เลเยอร์):

- ใช้ฟิลเตอร์ Convolutional ดึงรูปแบบเฉพาะที่ในภาพ
- filters: จำนวนฟิลเตอร์ในแต่ละเลเยอร์ (16, 32, 64) กำหนด

ความซับซ้อนของ features ที่ดึงออกมา

- kernel_size: ขนาดของหน้าต่างฟิลเตอร์ (3x3 ในทุกเลเยอร์)
- padding="Same": เติม padding ให้กับอินพุตด้วย zeros เพื่อ

รักษามิติเชิงพื้นที่ให้คงเดิมหลังจาก Convolution

- activation="relu": ใช้ฟังก์ชัน activation ReLU (Rectified

Linear Unit) สำหรับ non-linearity

- MaxPooling2D (3 เลเยอร์):

- ลดขนาด feature maps เพื่อลดความซับซ้อนเชิงพื้นที่และ

ค่าใช้จ่ายในการคำนวณ

- pool_size: ขนาดของหน้าต่าง pooling (2x2 ในทั้งสองเลเยอร์)
- strides=(2, 2) (เฉพาะในเลเยอร์ที่สอง):

- ใช้ stride 2 หมายความว่า output มีขนาดครึ่งหนึ่งของ

input

- Dropout (3 เลเยอร์):

- สุ่มลบ activations บางส่วน (25%) เพื่อป้องกัน overfitting

- Flatten:

- เปลี่ยน feature maps 3 มิติเป็นเวกเตอร์ 1 มิติสำหรับป้อนเข้าสู่

fully connected layers

- Dense (2 เลเยอร์):

- Fully connected layers ที่เรียนรู้ features ระดับสูงจาก

output ที่ถูก flatten

- units: จำนวน neurons ในแต่ละเลเยอร์ (512, 24)
- activation="relu" (เลเยอร์แรก): ใช้ ReLU activation
- activation="softmax" (เลเยอร์ที่สอง):

- ใช้ softmax activation สำหรับสร้าง probabilities
- บนหลาย classes (24 ในกรณีนี้)

b. โมเดล LSTM (Long Short-Term Memory model)

วัตถุประสงค์ เพื่อจับ dependencies เชิงเวลาและข้อมูลลำดับจาก input data คือ สันนิษฐานว่าเป็น time series, 40 timesteps, และ 1 feature ต่อ timestep

➢ เลเยอร์:

สอง)

- LSTM (2 Bi-Directional layers):
 - units: จำนวน memory cells ในแต่ละ LSTM layer (50 ในทั้ง
 - return_sequences=True (ในทั้งสองเลเยอร์):
 - ส่งคืน output sequence ทั้งหมดสำหรับการประมวลผล

เพิ่มเติม

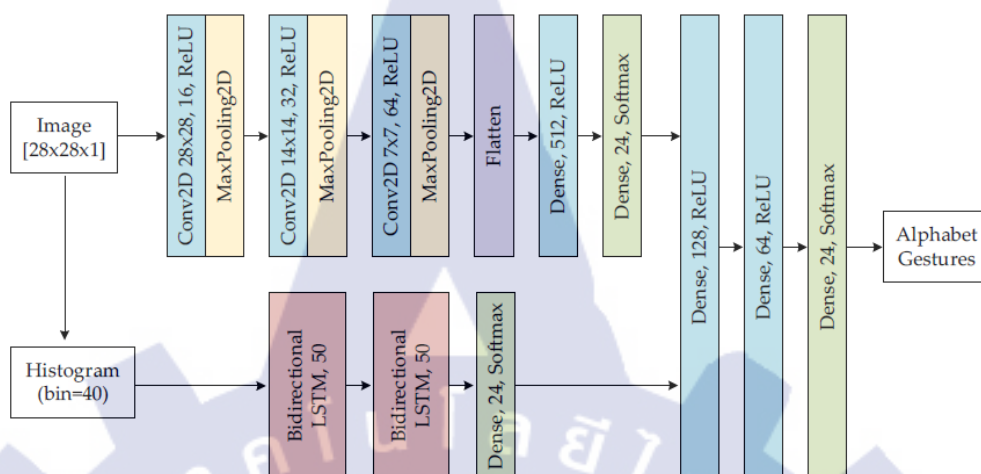
- Dropout (2 เลเยอร์):
 - สุ่มลบ activations บางส่วน (20%) เพื่อป้องกัน overfitting

c. โมเดลรวม(Mix model)

LSTM

- Concatenate:
 - รวม flattened output ของ CNN กับ output สุดท้ายของ LSTM
 - ผสมผสาน features ที่ดึงออกมา
- Dense (2 เลเยอร์):
 - Fully connected layers ที่เรียนรู้จาก features
 - ที่ถูกผสมผสาน
 - units: จำนวน neurons ในแต่ละเลเยอร์ (128, 64)
 - activation="relu":
 - ใช้ ReLU activation
 - สำหรับ non-linearity
- Fully connected layer สุดท้าย
 - units: จำนวน neurons เท่ากับจำนวน classes (24)
 - activation="softmax":
 - สร้าง probabilities

- สำหรับแต่ละ class



รูปที่ 3.4 แสดง model ของงานวิจัยนี้

การคำนวณค่าพารามิเตอร์ส่วนหน้า งานวิจัยนี้เลือกฟังก์ชันความเป็นสมาชิกเกาส์เซียน (Gaussian Membership Function) เนื่องจากสอดคล้องกับพฤติกรรมค่าความหนาแน่นที่เปลี่ยนแปลงไปตามตัวแปรอินพุต ซึ่งฟังก์ชันประกอบไปด้วย 2 พารามิเตอร์ ได้แก่ $\{c, \sigma\}$ ดังแสดงความสัมพันธ์ในสมการ (3.1) และในการศึกษานี้แบ่งฟังก์ชันความเป็นสมาชิกของแต่ละตัวแปรอินพุตออกเป็น 3 ระดับ ได้แก่ ต่ำ กลาง และสูง เพื่อให้ตัวแปรอินพุตมีระดับความเป็นสมาชิกที่คลุมเครือมากกว่าการมีระดับสูงหรือต่ำไปเลย ดังที่แสดงในรูปที่ 3.4

บทที่ 4

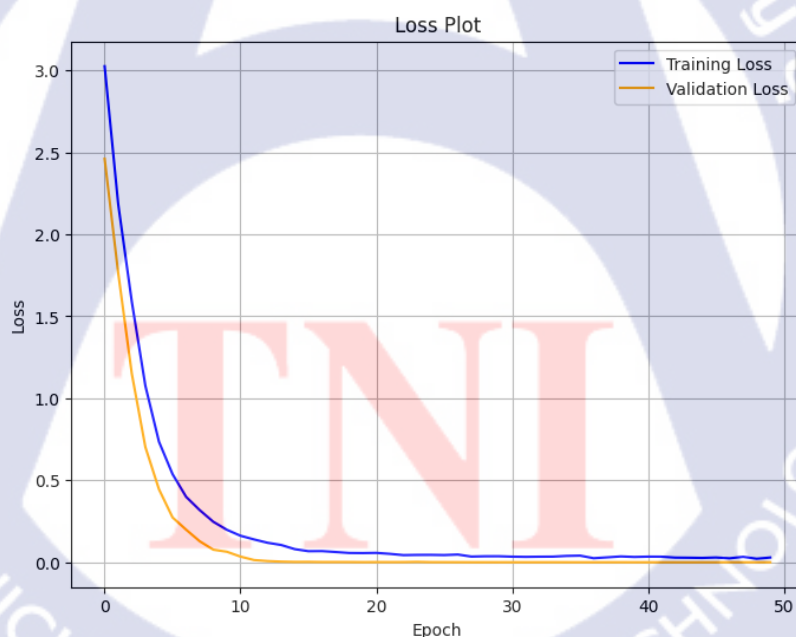
ผลการศึกษาและการอภิปรายผล

ในบทนี้ผู้ทำการวิจัยจะแสดงผลลัพธ์ของงานวิจัยที่ได้ดำเนินการตามขั้นตอนในบทที่ 3 และอภิปรายผลลัพธ์ของงานวิจัยในหัวข้อ ความแม่นยำของแต่ละอัลกอริทึม, ประสิทธิภาพในด้านเวลาของแต่ละอัลกอริทึม, ประสิทธิภาพในด้านการใช้ทรัพยากร CPU ในแต่ละอัลกอริทึม และการเปรียบเทียบความแม่นยำระหว่างอัลกอริทึม

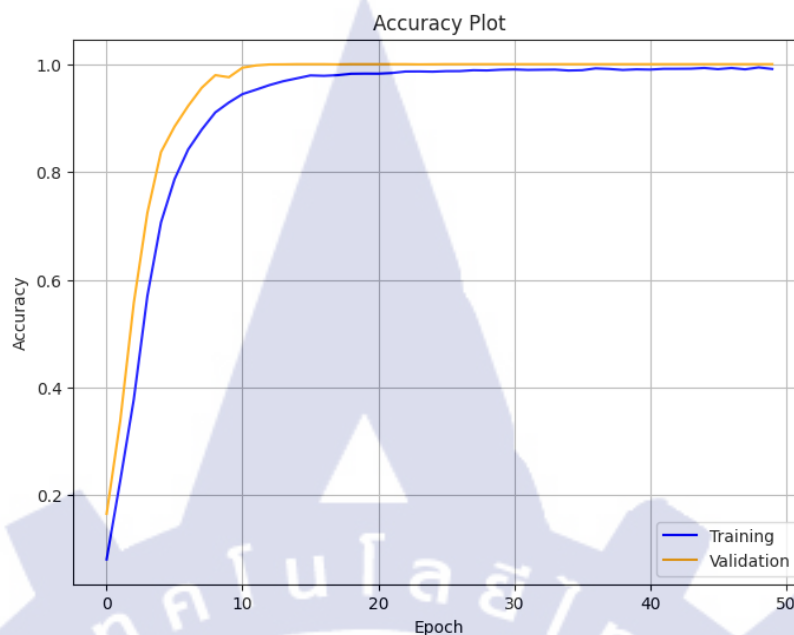
การทดลองและการประเมินผลแบบจำลองการจำแนกภาษามือที่พัฒนาขึ้นโดยเป็นส่วนหนึ่งของการวิจัยนี้ ผลลัพธ์ครอบคลุมการวัดประสิทธิภาพ การคาดการณ์แบบจำลอง และการวิเคราะห์เชิงคุณภาพของผลลัพธ์การจำแนกประเภท

4.1 ตัวชี้วัดประสิทธิภาพ

บทความนี้รายงานตัวชี้วัดประสิทธิภาพเชิงปริมาณของแบบจำลองการจำแนกภาษามือ



รูปที่ 4.1 กราฟแสดงค่า Loss ในการของโมเดลผสม



รูปที่ 4.2 กราฟแสดงค่า Accuracy ในการของโมเดลผสม

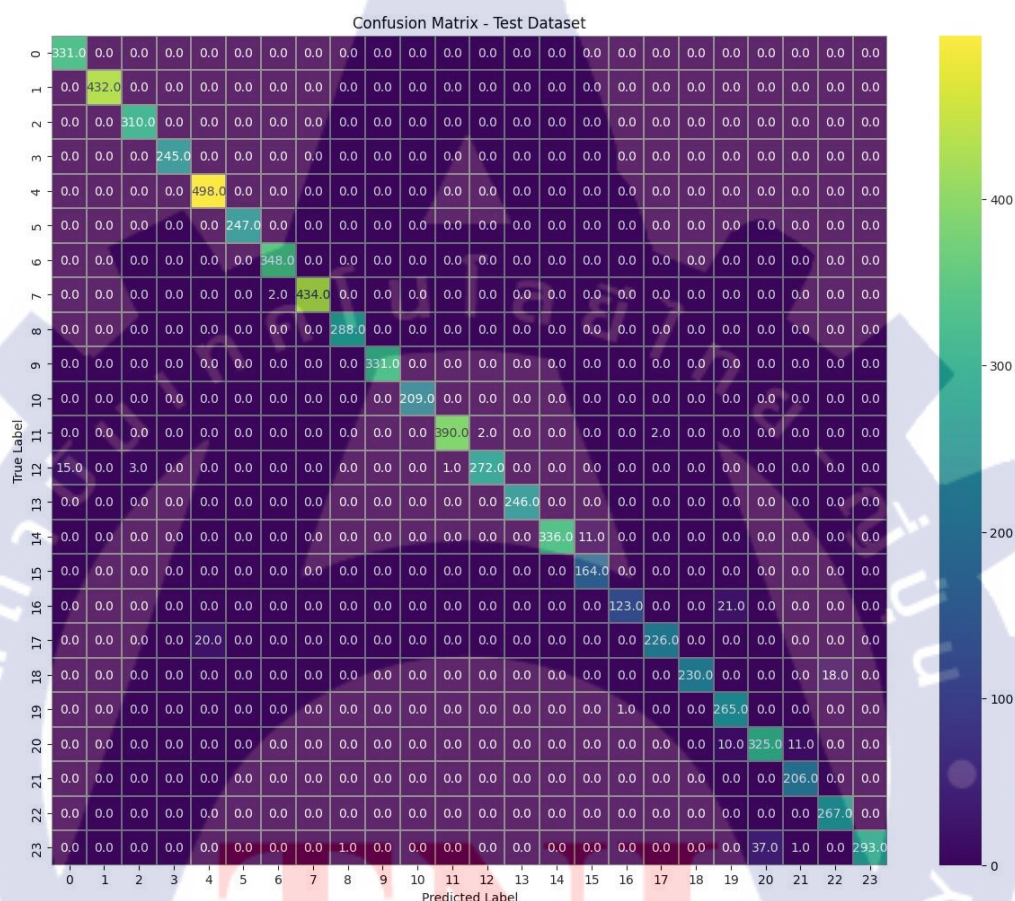
หลังจากที่นำโมเดลที่ได้นำไป train จำนวน 50 รอบจะได้กราฟ loss(ภาพที่ 4.1) และ Accuracy(ภาพที่ 4.2) ซึ่งพบว่าค่า loss มีค่าลดลงทั้ง Training และ Validation ซึ่งมีแนวโน้มในทิศทางเดียวกัน และใช้เวลาเพียง 20 ครั้ง ค่า Loss เข้าใกล้ 0 และ Accuracy มีค่าเข้าใกล้ 1

นอกจากนี้เมตริกซ์ accuracy, precision, recall และ F1 score และค่า confusion matrix หน่วยวัดเหล่านี้ยังให้ข้อมูลเชิงลึกเกี่ยวกับความแม่นยำในการจำแนกประเภท ความคงทน และความสามารถในการวางนัยทั่วไปของแบบจำลองผ่านท่าทางสัญลักษณ์ต่างๆ

ตารางที่ 4.1 สรุปตัวชี้วัดประสิทธิภาพ

Alphabet	Accuracy	Precision	Recall	F1-score
A	1.00	0.96	1.00	0.98
B	1.00	1.00	1.00	1.00
C	1.00	0.99	1.00	1.00
D	1.00	1.00	1.00	1.00
E	1.00	0.96	1.00	0.98
F	1.00	1.00	1.00	1.00
G	1.00	0.99	1.00	1.00
H	1.00	1.00	1.00	1.00
I	1.00	1.00	1.00	1.00
K	1.00	1.00	1.00	1.00
L	1.00	1.00	1.00	1.00
M	0.99	1.00	0.99	0.99
N	0.93	0.99	0.93	0.96
O	1.00	1.00	1.00	1.00
P	0.97	1.00	0.97	0.98
Q	1.00	0.94	1.00	0.97
R	0.85	0.99	0.85	0.92
S	0.92	0.99	0.92	0.95
T	0.93	1.00	0.93	0.96
U	1.00	0.90	1.00	0.94
V	0.94	0.90	0.94	0.92
W	1.00	0.94	1.00	0.97
X	1.00	0.94	1.00	0.97
Y	0.88	1.00	0.88	0.94

จากตารางที่ 4.1 เป็นการตารางสรุปประสิทธิภาพของการทำนาย class ตั้งแต่ A-Y จำนวน 24 class และแสดงความแม่นยำ 4 ประเภทได้แก่ Accuracy , Precision, Recall, และ F1-score โดยค่ามีค่า Accuracy สูงสุดที่ 1 และต่ำสุดคือ Y ที่ 0.88, Precision สูงสุดที่ 1 และต่ำสุดคือ U,V ที่ 0.90, Recall สูงสุดที่ 1 ต่ำสุดคือ Y ที่ 0.88, และ F1-score สูงสุดที่ 1 และต่ำสุดคือ R,V ที่ 0.92



รูปที่ 4.3 แสดง confusion matrix ของ 24 ตัวอักษร

Confusion matrix อีกหนึ่งเครื่องมือสำคัญการประเมินผลลัพธ์ของการ Train AI โดยใช้โมเดล Machine Learning เพื่อบอกความถูกต้องของโมเดลในการทำนาย และวิเคราะห์ประเภทของ Error ที่เกิดขึ้น

ตารางที่ 4.2 เปรียบเทียบประสิทธิภาพของการศึกษาก่อน

Authors	Works	Accuracy(%)
[10]	Transfer learning with data augmented	93.30
[11]	2S-CNN model on ASL dataset	92.08
[12]	CapsNet with augmented	95.08
[13]	Inflated 3D ConvNet with BLSTM	89.74
This Work	Mixed-Features DL Model	97.54

4.2 การคาดการณ์แบบจำลอง

บทความนี้จะวิเคราะห์และอภิปรายเกี่ยวกับการทำนายแบบจำลองที่สร้างขึ้นโดยแบบจำลองการจำแนกภาษามือที่ได้รับการฝึกอบรม มีตัวอย่างการคาดการณ์ที่ถูกต้องและไม่ถูกต้องโดยเน้นจุดแข็งและจุดอ่อนของแบบจำลองในการจดจำท่าทางสัญลักษณ์เฉพาะ

4.3 การวิเคราะห์เชิงคุณภาพ

บทความนี้ดำเนินการวิเคราะห์เชิงคุณภาพของผลการจำแนกประเภทเพื่อระบุรูปแบบแนวโน้ม และความท้าทายที่พบในระหว่างการประชุมแบบจำลอง จะมีการหารือเกี่ยวกับการค้นพบที่โดดเด่น ความไม่สอดคล้องกัน หรือความคลุมเครือที่พบในผลลัพธ์การจำแนกประเภท โดยให้ข้อมูลเชิงลึกเกี่ยวกับปัจจัยพื้นฐานที่ขับเคลื่อนการคาดการณ์แบบจำลอง

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

บทความนี้นำเสนอโมเดลการเรียนรู้เชิงลึกที่มีพีเจอร์ผสมสำหรับการจำแนกท่าทางภาษามือ ประเภทข้อมูลแบบผสม เช่น รูปภาพและฮิสโตแกรม ถูกนำมาใช้เป็นอินพุตของโมเดลการเรียนรู้เชิงลึก กล่าวอีกนัยหนึ่ง โมเดลของ CNN ถูกใช้เพื่อแยกแยะความสัมพันธ์เชิงพื้นที่และคุณลักษณะภายในข้อมูลภาพ ทำให้เกิดการตรวจสอบลักษณะทางกายภาพของท่าทางสัญญาณอย่างชัดเจน ในขณะเดียวกัน โมเดล LSTM จะตีความข้อมูลชั่วคราวที่ได้มาจากฮิสโตแกรม โดยห่อหุ้มการเคลื่อนไหวแบบไดนามิกที่เป็นพื้นฐานของการสื่อสารด้วยภาษามือ ผลการทดลองแสดงประสิทธิภาพที่เหมาะสมในเมตริกซ์ความสับสนและตัวชี้วัดประสิทธิภาพทั้งหมด ประสิทธิภาพนี้บ่งชี้ถึงระบบการจำแนกประเภทที่มีความน่าเชื่อถือสูงด้วยขนาดการจัดเก็บข้อมูลที่กะทัดรัดสำหรับท่าทางภาษามือ งานนี้ไม่เพียงแต่ให้ความมั่นใจอย่างสูงสำหรับแบบจำลองการจำแนกภาษามือเท่านั้น แต่ยังนำเสนอโซลูชันที่ไม่ซับซ้อนและมีประสิทธิภาพ ซึ่งสามารถบูรณาการเข้ากับแอปพลิเคชันต่างๆ ได้อย่างง่ายดาย

5.2 ปัญหาและอุปสรรค

5.2.1 การเชื่อมต่อระหว่าง 2 อัลกอริทึม บ่อยครั้งจะเข้ากันไม่ได้และทำให้ความแม่นยำของ model นั้นมีค่าต่ำกว่าอัลกอริทึมดั้งเดิม

5.2.2 ในการ train model นั้น เมื่อนำภาพเข้า train บน google colab จะมีปัญหาในการดึงข้อมูลภาพที่มีจำนวนมากและขนาดใหญ่ ทำให้การ train แต่ละครั้งใช้ระยะเวลาที่นาน

5.3 ข้อเสนอแนะ

5.3.1 ตรวจสอบเพิ่มเติมว่าค่าความเข้มแสงของแต่ละภาพที่แปลงออกเป็น histogram นั้นส่งผลต่อการ ทำนายหรือไม่ มากน้อยแค่ไหน

บรรณานุกรม

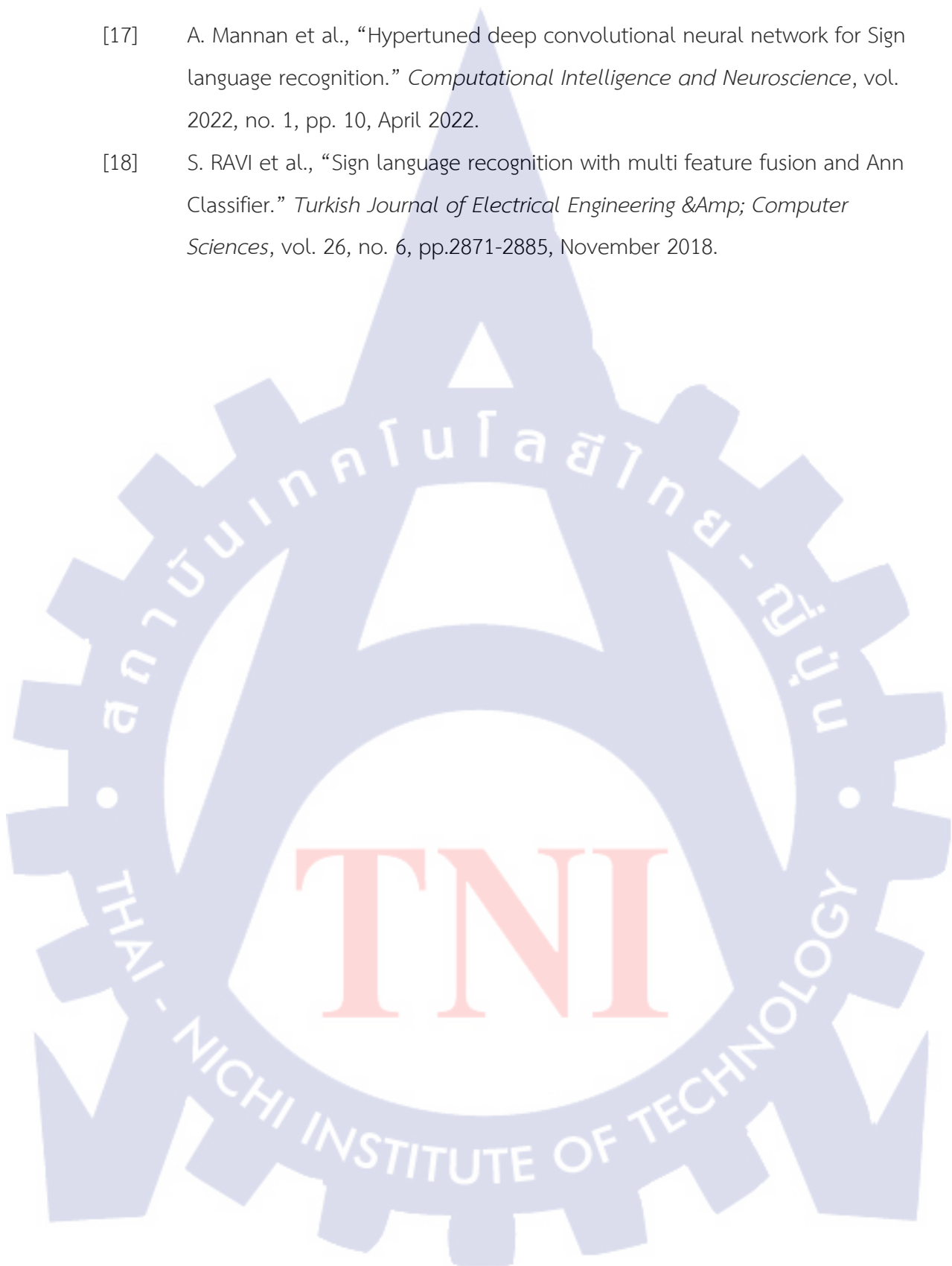
TNI

บรรณานุกรม

- [1] Graceza, “มารู้จัก ASL กันดีกว่า,” [Online]. Available: <https://yodiagrace.wordpress.com/2012/05/06/%e0%b8%a1%e0%b8%b2%e0%b8%a3%e0%b8%b9%e0%b9%89%e0%b8%88%e0%b8%b1%e0%b8%81%e0%b8%81%e0%b8%b1%e0%b8%99%e0%b8%94%e0%b8%b5%e0%b8%81%e0%b8%a7%e0%b9%88%e0%b8%b2/> [Accessed : 24 เมษายน 2567].
- [2] Wikipedia Contributors, “*American Sign Language*,” [Online]. Available: https://en.wikipedia.org/wiki/American_Sign_Language [Accessed: April 24, 2024].
- [3] ResearchGate, “*Figure 1. The 26 letters and 10 digits of American Sign Language (ASL)*,” [Online]. Available: https://www.researchgate.net/figure/The-26-letters-and-10-digits-of-American-Sign-Language-ASL_fig1_328396430 [Accessed: April 24, 2024].
- [4] DEMETER ICT, “*Machine Learning VS Deep Learning แท้จริงแล้วคืออะไรกันแน่?*,” [Online]. Available: <https://www.dmit.co.th/th/zendesk-updates-th/machine-learning-vs-deep-learning/> [Accessed: 24 เมษายน 2567].
- [5] Metaverse Post, “*การเรียนรู้ของเครื่อง — อธิบายแล้ว Definition และตัวอย่าง*,” [Online]. Available: <https://mpost.io/th/glossary/machine-learning/> [Accessed: 24 เมษายน 2567].
- [6] P. I. / P. OÜ, “*The 3 Main Types of Machine Learning*,” [Online]. Available: <https://www.pipedrive.com/en/blog/types-of-machine-learning> [Accessed: April 24, 2024].
- [7] A. A. Sulaiman, “*Basic Introduction to Reinforcement Learning*,” [Online]. Available: <https://medium.com/@Khalboy10/basic-introduction-to-reinforcement-learning-c37c12924dd7> [Accessed: April 24, 2024].
- [8] “*Machine Learning and Deep Learning: A Review of Methods and Applications*,” [Online]. Available : https://www.researchgate.net/publication/371011515_Machine_Learning_and_Deep_Learning_A_Review_of_Methods_and_Applications [Accessed: April 24, 2024].

- [9] Medium, “*Medium*,” [Online]. Available: <https://medium.com/@natthawatphongchit/%E0%B8%A1%E0%B8%B2%E0%B8%A5%E0%B8%AD%E0%B8%87%E0%B8%94%E0%B8%B9%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%84%E0%B8%B4%E0%B8%94%E0%B8%82%E0%B8%AD%E0%B8%87-cnn-%E0%B8%81%E0%B8%B1%E0%B8%99-e3f5d73eebaa> [Accessed: April 24, 2024]
- [10] SlideShare, “*Convolutional Neural Networks*,” [Online]. Available: <https://www.slideshare.net/slideshow/convolutional-neural-networks-249580831/249580831> [Accessed: April 24, 2024].
- [11] IndianTechWarrior, “*Fully Connected Layers in Convolutional Neural Networks*,” [Online]. Available: <https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/> [Accessed: April 24, 2024].
- [12] I. Faridmehr et al., “Predicting embodied carbon and cost effectiveness of post-tensioned slabs using novel hybrid firefly ANN,” *Sustainability*, vol. 13, no. 21, pp. 12319, November 2021.
- [13] d2l.ai, “9.2. Long Short Term Memory (LSTM) — Dive into Deep Learning 0.14.4 documentation,” [Online]. Available: https://d2l.ai/chapter_recurrent-modern/lstm.html [Accessed: April 24, 2024].
- [14] W. Li and J. Gao, “Automatic Sleep Staging by a Hybrid Model Based on Deep 1D-ResNet-SE and LSTM with Single-Channel Raw EEG Signals,” [Online]. Available: <https://peerj.com/articles/cs-1561/> [Accessed: April 24, 2024].
- [15] M. M. Hasan et al., “Classification of Sign Language Characters by Applying a Deep Convolutional Neural Network,” [Online]. Available: <https://doi.org/10.1109/ICAICT51780.2020.9333456> [Accessed: April 24, 2024].
- [16] B. P. P. Kumar and M. B. Manjunatha, “A hybrid gesture recognition method for American Sign Language.” *Indian Journal of Science and Technology*, vol. 10, no. 1, pp. 1-12, January 2017.

- [17] A. Mannan et al., “Hypertuned deep convolutional neural network for Sign language recognition.” *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 10, April 2022.
- [18] S. RAVI et al., “Sign language recognition with multi feature fusion and Ann Classifier.” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, no. 6, pp.2871-2885, November 2018.





ภาคผนวก

ภาคผนวก ก.
โค้ดโปรแกรม

TNII

```
from google.colab import drive
drive.mount('/content/drive')

import os

names = os.listdir('/content/drive/MyDrive/TNI/new')

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")

import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/TNI/new'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# read training data
train=pd.read_csv("/content/drive/MyDrive/TNI/new/train_new.csv")
print(train.shape)
train.head()

# read testing data
val=pd.read_csv("/content/drive/MyDrive/TNI/new/val_new.csv")
print(val.shape)
val.head()

# read testing data
```

```

test=pd.read_csv("/content/drive/MyDrive/TNI/new/test_new.csv")
print(test.shape)
test.head()

#-----CNN
Y_train_pic=train["label"]
Y_val_pic=val["label"]
Y_test_pic=test["label"]
X_train_pic=train.drop(labels=["label"],axis=1)
X_val_pic=val.drop(labels=["label"],axis=1)
X_test_pic =test.drop(labels=["label"],axis=1)

#-----histogram
Y_train_his=train["label"].to_numpy()
Y_val_his=val["label"].to_numpy()
Y_test_his=test["label"].to_numpy()
X_train_his=train.drop(labels=["label"],axis=1).to_numpy()
X_val_his=val.drop(labels=["label"],axis=1).to_numpy()
X_test_his=test.drop(labels=["label"],axis=1).to_numpy()

from sklearn.preprocessing import MinMaxScaler

def images_to_normalized_histograms(images, num_bins=40):
    normalized_histograms = []
    scaler = MinMaxScaler()

    for image in images:
        hist, _ = np.histogram(image, bins=num_bins, range=(0, 256))
        normalized_hist = scaler.fit_transform(hist.reshape(-1, 1)).flatten()
        normalized_histograms.append(normalized_hist)

```

```

return np.array(normalized_histograms)

norhis_train = images_to_normalized_histograms(X_train_his)
norhis_val = images_to_normalized_histograms(X_val_his)
norhis_test = images_to_normalized_histograms(X_test_his)

print(norhis_train.shape)

X_train_pic=X_train_pic/255.0
X_val_pic=X_val_pic/255.0
X_test_pic=X_test_pic/255.0

print("x_train_pic shape: ",X_train_pic.shape)
print("x_val_pic shape: ",X_val_pic.shape)
print("test shape: ",X_test_pic.shape)

num_images = 5

fig, axes = plt.subplots(1, 5, figsize=(20, 4))

for i in range(5):
    axes[i].bar(range(40), norhis_train[i])
    axes[i].set_title(f'Image {i+1} Normalized Histogram')
    axes[i].set_xlabel('Bin')
    axes[i].set_ylabel('Normalized Frequency')

plt.tight_layout()
plt.show()

from sklearn.preprocessing import LabelBinarizer

```

```
label_binrizer = LabelBinarizer()
Y_train_images = label_binrizer.fit_transform(Y_train_pic)
Y_val_images = label_binrizer.fit_transform(Y_val_pic)
Y_test_images = label_binrizer.fit_transform(Y_test_pic)
print(Y_train_images.shape)
print(Y_train_images)
```

```
from sklearn.preprocessing import LabelBinarizer
label_binrizer = LabelBinarizer()
Y_train_hist = label_binrizer.fit_transform(Y_train_his)
Y_val_hist = label_binrizer.fit_transform(Y_val_his)
Y_test_hist = label_binrizer.fit_transform(Y_test_his)
print(Y_train_his.shape)
print(Y_train_his)
```

```
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split

X_train_images = X_train_pic.values.reshape(-1, 28, 28, 1)
X_val_images = X_val_pic.values.reshape(-1, 28, 28, 1)
X_test_images = X_test_pic.values.reshape(-1, 28, 28, 1)

X_train_hist = norhis_train
X_val_hist = norhis_val
X_test_hist = norhis_test

print("X_train_hist shape:", X_train_hist.shape)
print("X_val_hist shape:", X_val_hist.shape)
print("X_test_hist shape:", X_test_hist.shape)
print("Y_train_hist shape:", Y_train_hist.shape)
print("Y_val_hist shape:", Y_val_hist.shape)
```

```
print("Y_test_hist shape:", Y_test_hist.shape)
```

```
print("X_train_images shape:", X_train_images.shape)
```

```
print("X_val_images shape:", X_val_images.shape)
```

```
print("X_test_images shape:", X_test_images.shape)
```

```
print("Y_train_images shape:", Y_train_images.shape)
```

```
print("Y_val_images shape:", Y_val_images.shape)
```

```
print("Y_test_images shape:", Y_test_images.shape)
```

```
from keras.models import Sequential, Model
```

```
from keras.layers import Input, Concatenate, Dense, Flatten, Conv2D, MaxPooling2D,
```

```
Dropout, LSTM, Bidirectional
```

```
from keras.optimizers import Adam
```

```
from sklearn.metrics import confusion_matrix, classification_report
```

```
from keras.callbacks import ModelCheckpoint
```

```
import pickle
```

```
cnn_model = Sequential()
```

```
cnn_model.add(Conv2D(filters=16, kernel_size=(3, 3), padding="Same",  
activation="relu", input_shape=(28, 28, 1)))
```

```
cnn_model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
cnn_model.add(Dropout(0.25))
```

```
cnn_model.add(Conv2D(filters=32, kernel_size=(3, 3), padding="Same",  
activation="relu"))
```

```
cnn_model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
```

```
cnn_model.add(Dropout(0.25))
```

```
cnn_model.add(Conv2D(filters=64, kernel_size=(3, 3), padding="Same",  
activation="relu"))
```

```
cnn_model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
```

```
cnn_model.add(Dropout(0.25))
```

```

cnn_model.add(Flatten())
cnn_model.add(Dense(512, activation="relu"))
cnn_model.add(Dropout(0.25))
cnn_model.add(Dense(24, activation="softmax"))

n_timesteps = 40
n_features = 1
lstm_input = Input(shape=(n_timesteps, n_features), name='lstm_input')
x = Bidirectional(LSTM(units=50, return_sequences=True))(lstm_input)
x = Dropout(0.2)(x)
x = Bidirectional(LSTM(units=50))(x)
x = Dropout(0.2)(x)
lstm_output = Dense(24, activation='softmax')(x)

merged = Concatenate()([cnn_model.output, lstm_output])

merged = Dense(128, activation='relu')(merged)
merged = Dropout(0.2)(merged)
merged = Dense(64, activation='relu')(merged)

final_output = Dense(24, activation='softmax')(merged)

combined_model = Model(inputs=[cnn_model.input, lstm_input],
outputs=final_output)

model_checkpoint =
ModelCheckpoint('/content/drive/MyDrive/TNI/new/best_model1.h5',
save_best_only=True, monitor='val_accuracy', mode='max', verbose=1)
model_checkpoint2 =
ModelCheckpoint('/content/drive/MyDrive/TNI/new/latest_model1.h5',
save_best_only=True, monitor='accuracy', mode='max', verbose=1)

```

```

optimizer = Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999)
combined_model.compile(optimizer=optimizer, loss="categorical_crossentropy",
metrics=["accuracy"])

history = combined_model.fit(
    [X_train_images[:X_train_hist.shape[0]], X_train_hist.reshape((X_train_hist.shape[0],
n_timesteps, n_features))),
    Y_train_images[:X_train_hist.shape[0]],
    epochs=50, batch_size=64,
    validation_data=(
        [X_val_images[:X_val_hist.shape[0]], X_val_hist.reshape((X_val_hist.shape[0],
n_timesteps, n_features))),
        Y_val_images[:X_val_hist.shape[0]]
    ),
    steps_per_epoch=X_train_hist.shape[0] // 64,
    callbacks=[model_checkpoint, model_checkpoint2]
)

with open('/content/drive/MyDrive/TNI/new/training_history.pkl', 'wb') as file:
    pickle.dump(history.history, file)

combined_model.summary()

import matplotlib.pyplot as plt
from keras.models import load_model
import pickle

latest_model_path = '/content/drive/MyDrive/TNI/new/latest_model_98%.h5'
combined_model = load_model(latest_model_path)

```

```
with open('/content/drive/MyDrive/TNI/new/training_history.pkl', 'rb') as file:
    load_history = pickle.load(file)

print(load_history.keys())
n_timesteps = 40
n_features = 1

history = load_history

plt.figure(figsize=(8, 6))

plt.plot(history['accuracy'], label='Training', color='blue', linestyle='-')
plt.plot(history['val_accuracy'], label='Validation', color='orange', linestyle='-')

plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Accuracy Plot')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(8, 6))

plt.plot(history['loss'], label='Training', color='blue', linestyle='-')
plt.plot(history['val_loss'], label='Validation', color='orange', linestyle='-')

plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Loss Plot')
plt.legend()
plt.grid(True)
```

```
plt.show()
```

```
Y_pred = combined_model.predict([X_val_images,  
X_val_hist.reshape((X_val_hist.shape[0], n_timesteps, n_features))])
```

```
Y_pred_classes = np.argmax(Y_pred, axis=1)
```

```
Y_true = np.argmax(Y_val_images, axis=1)
```

```
confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)
```

```
f, ax = plt.subplots(figsize=(15, 12))
```

```
sns.heatmap(confusion_mtx, annot=True, linewidths=0.01, cmap="viridis",  
linecolor="gray", fmt=".1f", ax=ax)
```

```
plt.xlabel("Predicted Label")
```

```
plt.ylabel("True Label")
```

```
plt.title("Confusion Matrix")
```

```
plt.show()
```

```
print(classification_report(Y_true, Y_pred_classes))
```

```
Y_test_pred = combined_model.predict([X_test_images,  
X_test_hist.reshape((X_test_hist.shape[0], n_timesteps, n_features))])
```

```
Y_test_pred_classes = np.argmax(Y_test_pred, axis=1)
```

```
Y_test_true = np.argmax(Y_test_images, axis=1)
```

```
confusion_mtx_test = confusion_matrix(Y_test_true, Y_test_pred_classes)
```

```
f, ax = plt.subplots(figsize=(15, 12))
```

```
sns.heatmap(confusion_mtx_test, annot=True, linewidths=0.01, cmap="viridis",  
linecolor="gray", fmt=".1f", ax=ax)  
plt.xlabel("Predicted Label")  
plt.ylabel("True Label")  
plt.title("Confusion Matrix - Test Dataset")  
plt.show()  
  
print(classification_report(Y_test_true, Y_test_pred_classes))
```

