การพัฒนาเว็บแอปพลิเคชันเพื่อตรวจจับตัวอักษรภาษาไทยผ่านการแสดงท่าทางของมือ

Development of Web Application to Detect ThSL Alphabet via Hand Gesture

ภูมินทร์ สุขสุวรรณ, ชยุตม์ แซ่อึ้ง, นบนอบ หงส์สุวรรณ, บุญชู จิตนุพงศ์*

Pumin Suksuwan, Chayut Saeeng, Nopnob Hongsuwan, Boonchoo Jitnupong*

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ ศรีราชา มหาวิทยาลัยเกษตรศาสตร์

Department of Computer Science and Information, Faculty of Science at Sriracha, Kasetsart University

บทคัดย่อ

ระบบการตรวจจับอักษรภาษาไทยผ่านการแสดงท่าทางของมือในบทความนี้ได้รับการพัฒนาขึ้นเพื่อช่วยเหลือผู้ที่มีความบกพร่องทางการได้ยินและการพูดหรือบุคคลที่ใช้สื่อสารกับผู้อื่นด้วยสัญญาณมือ ให้มีเครื่องมือสำหรับการเรียนรู้การแสดงออกของมือและนิ้วที่เพื่อการสะกดตัวอักษรไทย วัตถุประสงค์เพื่อพัฒนาเว็บแอปพลิเคชันบนฟลาสก์เฟรมเวิร์คที่สามารถตรวจจับและแปลสัญญาณมือผ่านกล้องเว็บแคมเป็นตัวอักษรภาษาไทย (Thai Sign Language: ThSL) ทั้งนี้เพื่อให้ระบบสามารถรับรู้การเคลื่อนไหวของมือเป็นอักขระภาษาไทยได้ ในงานนี้จึงมีการสร้างตัวแบบอักษรมือไทยที่มีการใช้เทคนิคการเรียนรู้ของเครื่องจักร การเรียนรู้เชิงลึก โดยเฉพาะตัวแบบเครือข่ายประสาทความจำระยะสั้นแบบยาว ที่มีการบูรณาการเฟรมเวิร์คต่าง ๆ เช่น เครื่องมือมีเดียไปป์ เทนเซอร์โฟลว์ โดยมีเป้าหมายที่จะปรับปรุงความแม่นยำผ่านการฝึกฝนแบบวนซ้ำและป้องกันการโอเวอร์ฟิตติ้งโดยการจัดการกระบวนการฝึกฝนอย่างรอบคอบ ในตอนท้ายของงานนี้มีการนำเสนอการประเมินคุณภาพของตัวแบบการเรียนรู้ของเครื่องจักรที่แสดงให้เห็นค่าเฉลี่ยผลการทดสอบความถูกต้องแม่นยำของตัวแบบการเรียนรู้ของเครื่องจักรในแต่ละตัวอักษรไทยทั้ง 42 ตัวในงานนี้ได้ผลลัพธ์ในเกณฑ์ดีมาก (0.98)

คำสำคัญ: ภาษามือไทย การเรียนรู้ของเครื่อง การเรียนรู้เชิงลึก หน่วยความจำระยะยาว-ระยะสั้น

Abstract

This paper presents a hand gesture recognition system for Thai Sign Language (ThSL) to assist people with hearing and speech impairments, as well as those who communicate with others using hand gestures. The system aims to provide a tool for learning Thai letter gestures and fingerspelling. The objective is to develop a web application using the Flask framework that can detect and translate hand gestures from a webcam into Thai characters. To achieve this, the system utilizes machine learning and deep learning techniques, specifically a Long Short-Term Memory (LSTM) neural network model, integrated with various frameworks such as MediaPipe

* *Corresponding author : jitnupong.b@ku.th*

Received: 20 พ.ค. 67; Revised: 25 ธ.ค. 67; Accepted: 27 ธ.ค. 67

and TensorFlow. The goal is to improve accuracy through iterative training and prevent overfitting by carefully managing the training process. The paper concludes with an evaluation of the machine learning model, demonstrating an average accuracy of 0.98 for all 42 Thai letters, indicating a very good performance (0.98).

Keywords: Thai Sign Language, Machine Learning, Deep Learning, Long Short-Term Memory

1. บทนำ

จากสถิติรายงานข้อมูลสถานการณ์ด้านคนพิการในประเทศไทย ของกรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ กระทรวงการพัฒนาสังคมและความมั่นคงของมนุษย์ เมื่อปี พ.ศ. 2566 ได้สำรวจประเภทความพิการของผู้พิการในไทยพบว่าผู้พิการทางการได้ยินหรือสื่อความหมายมีจำนวน 415,999 คน หรือคิดเป็นร้อยละ 18.57 ของจำนวนผู้พิการไทยทั้งหมด (กรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ, 2567) ผู้พิการเหล่านี้คือผู้ที่สื่อสารกับผู้อื่นโดยการใช้ท่าทางของภาษามือเป็นหลักโดยการใช้การกระทำต่าง ๆ ของร่างกาย รวมทั้งมือ นิ้วมือ สำหรับภาษาไทย ชุดข้อมูลที่มีอยู่ส่วนใหญ่มักจะมีข้อจำกัดในการใช้งานโดยเฉพาะตัวอักษรที่มีมากและประกอบไปด้วยการกระทำมากมายจึงทำให้การทำท่าทางผ่านมือนั้นเป็นไปได้ยาก เนื่องจากเป็นรูปแบบเดียวในการสื่อสาร การพัฒนาระบบที่สามารถช่วยตรวจจับและแปลงภาษามือเป็นภาษาไทยได้อย่างแม่นยำให้กับผู้พิการจึงเป็นสิ่งที่มีความสำคัญ

ในงานวิจัยนี้ นำเสนอระบบการตรวจจับท่าทางภาษามือแปลงให้เป็นตัวอักษรภาษาไทยเพื่อช่วยเหลือผู้ที่มีความบกพร่องทางการได้ยินและการพูด เพื่อให้ผู้พิการมีเครื่องมือเพื่อการเรียนรู้การแสดงออกทางมือและนิ้วของแต่ละตัวอักษรในภาษาไทย ระบบจะรับรู้ท่าทางผ่านกล้องตรวจจับการเคลื่อนไหวของมือและนิ้วจากผู้ใช้และแปลออกมาเป็นตัวอักษรภาษาไทยออกทางหน้าจอ โดยพัฒนาออกมาเป็นเว็บแอปพลิเคชันที่มีการประยุกต์ใช้การเรียนรู้ของเครื่องจักร (Machine Learning) การเรียนรู้เชิงลึก (Deep Learning) ผ่านชุดที่ข้อมูลที่ได้มีการจัดเตรียม โดยในบทความนี้เลือกใช้เทคนิควิธีและเครื่องมือหลายส่วนยกตัวอย่างเช่น เครือข่ายหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) มีเดียไปป์ (MediaPipe)

2. วัตถุประสงค์การวิจัย

1) เพื่อให้ได้เว็บแอพพลิเคชั่นเพื่อตรวจจับตัวอักษรภาษาไทยผ่านการแสดงท่าทางของมือ

2) เพื่อให้ได้ผลการทดสอบความถูกต้องแม่นยำของตัวแบบการเรียนรู้ของเครื่องจักรตัวแบบ LSTM

3. การทบทวนทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การรู้จำลักษณะท่าทางของมือภาษาไทย และการสะกดนิ้วมือ

ภาษามือไทยเป็นภาษาที่ใช้ลักษณะท่าทางของมือในการสื่อสารของผู้ที่มีความบกพร่องทางการได้ยิน เพื่อการติดต่อสื่อสารกับผู้อื่นของคนไทย เพื่อใช้สื่อสารของคนหูหนวกหรือผู้ที่มีความบกพร่องทางการได้ยินที่ต่างจากการสื่อสารของบุคคล ทั่วไปที่มีความเป็นปกติซึ่งการเรียนรู้ภาษาของคนปกติจะรับรู้ภาษาโดยอาศัยการพูดและการรับฟังในการสนทนาระหว่างกัน แต่คนหูหนวกหรือผู้ที่มีความบกพร่องทางการได้ยินนั้นไม่สามารถที่จะใช้โสตประสาทเหล่านั้นได้ดังนั้นภาษามือไทยจึงเป็นภาษาใน การสื่อสารที่สำคัญอย่างยิ่งของคนหูหนวกหรือผู้ที่มีความบกพร่องทางการได้ยิน โดยภาษามือจะใช้ลักษณะท่าทางของมือ การเคลื่อนไหวของมือตำแหน่งของท่ามือ และกิริยาท่าทางประกอบในการสื่อความหมายแทนการถ่ายทอดเป็นคำพูด



ภาพที่ 1 ท่าทางการสะกดนิ้วมือไทยของสมาคมคนหูหนวกแห่งประเทศไทย
(ภาควิชาภาษาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, ม.ป.ป.)

ในประเทศไทย ได้มีการกำหนดท่าสำหรับสะกดนิ้วมืออักษรไทยขึ้นเป็นครั้งแรกเมื่อปี พ.ศ. 2496 โดยคุณหญิงกมลา ไกรฤกษ์ อดีตอาจารย์ใหญ่โรงเรียนเศรษฐเสถียร (โรงเรียนสอนคนหูหนวกดุสิต) โดยดัดแปลงมาจากแบบตัวอักษรสะกดนิ้วมือของอเมริกัน (American Manal Alphabets) และได้ยึดหลักทางสัทศาสตร์ทั้งของภาษาไทยและภาษาอังกฤษ กล่าวคือ เสียงพยัญชนะหรือเสียงสระตัวใดในภาษาไทยที่ออกเสียงเหมือน หรือคล้ายคลึงกับเสียงพยัญชนะหรือเสียงสระในภาษาอังกฤษ ก็ให้ใช้ท่าสะกดนิ้วมือในลักษณะเดียวกัน

การสะกดนิ้วมือ หมายถึง การที่บุคคลทำท่าด้วยนิ้วมือเป็นรูปต่าง ๆ แทนตัว พยัญชนะ สระ วรรณยุกต์ ตลอดจนสัญลักษณ์อื่น ๆ ของภาษาประจำชาติเพื่อการสื่อภาษา โดยทั่วไปแล้วตัวอักษรที่สะกดด้วยนิ้วมือของภาษาใด จะมีจำนวนเท่ากับตัวอักษรของภาษานั้น ซึ่งเป็นวิธีการสื่อสารภาษาชนิดหนึ่งที่ใช้ในหมู่คนหูหนวก

หรือบุคคลปกติที่ต้องการจะสื่อภาษา สื่อความเข้าใจกับคนหูหนวก โดยใช้ประกอบกับ การทำท่าภาษามือเพื่อบอกชื่อเฉพาะ หรือเน้นย้ำคำศัพท์ที่ต้องการ (ภาควิชาภาษาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, ม.ป.ป) ทั้งนี้ภาพที่ 1 แสดงให้เห็นการสะกดนิ้วมือตัวอักษรในภาษาไทย

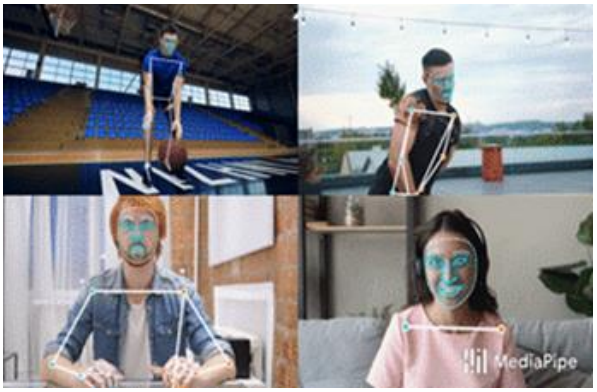
คอมพิวเตอร์วิทัศน์ (Computer Vision) และเครื่องมือมีเดียไปป์ (MediaPipe)

คอมพิวเตอร์วิทัศน์คือเทคโนโลยีในการประมวลผลภาพและวิดีโอเพื่อเข้าใจและวิเคราะห์ภาพเหล่านั้นแบบเดียวกับการมองเห็นของมนุษย์ คอมพิวเตอร์วิทัศน์รวมขั้นตอนทั้งหมดในการประมวลผล การวิเคราะห์และการทำความเข้าใจเพื่อนำข้อมูลออกมาใช้ในการพยากรณ์หรือการตัดสินใจได้ หลักการทำงานของคอมพิวเตอร์วิทัศน์มีหลายแง่ เช่น การแบ่งภาพ (Segmentation) เป็นกระบวนการแบ่งภาพหรือวิดีโอออกเป็นส่วนย่อย เพื่อทำความเข้าใจและประมวลผลในแต่ละส่วนของภาพ การจดจำรูปแบบ (Recognition) คือกระบวนการค้นหาและจดจำรูปแบบหรือภาพที่เหมือนหรือคล้ายกันเป็นการระบุว่าในภาพนี้มีอะไรบ้างที่คล้ายกับภาพอื่นๆ ซึ่งมีการใช้ในการค้นหาสิ่งต่างๆ การจำแนกวัตถุ (Classification) เป็นการระบุหมวดหมู่ของวัตถุในภาพนั้น ๆ ซึ่งเป็นการพยากรณ์ว่าวัตถุที่ปรากฏในภาพคืออะไร ทำให้คอมพิวเตอร์สามารถรู้ว่ามีอะไรอยู่ในภาพ ซึ่งในการจำแนกวัตถุในภาพเป็นรูปแบบของศาสตร์ คอมพิวเตอร์วิทัศน์ที่สำคัญ การค้นหาและติดตามวัตถุ (Tracking) เปรียบเสมือนการจับตาดูวัตถุเป้าหมายในภาพ เปรียบเสมือนสายตาที่ติดตามการเคลื่อนไหวของวัตถุนั้น ๆ คล้ายกับการตามหาตัวผู้ร้ายในกล้องวงจรปิด เทคโนโลยีนี้นำไปใช้เพื่อการเฝ้าระวังและรักษาความปลอดภัยในสถานที่ต่าง ๆ และการจดจำใบหน้า (Face Recognition) เปรียบเสมือนสายตาอัจฉริยะที่ใช้ในการระบุใบหน้ามนุษย์ในภาพ นี้เป็นรูปแบบการตรวจจับวัตถุที่ทันสมัยและใช้ในหลายแนวทาง เช่น การปรับปรุงระบบการเข้าถึงและความปลอดภัย คอมพิวเตอร์วิทัศน์มีความสำคัญในการแก้ไขปัญหาต่าง ๆ และมีการพัฒนาอย่างต่อเนื่องเพื่อเข้าใจและประมวลผลข้อมูลที่เป็นภาพหรือวิดีโอในลักษณะเดียวกันกับระบบการมองเห็นของมนุษย์ในหลายงานและหลายแขนง (Ballard & Brown, 1982; Techhub, 2567)

เพื่อการทำงานกับเทคนิคคอมพิวเตอร์วิทัศน์ให้ได้ประสิทธิภาพ ในงานชิ้นนี้จึงนำเครื่องมือมีเดียไปป์ที่มีความสามารถด้านการตรวจจับวัตถุ มาใช้ในการสะกดนิ้วมือ การรับรู้ท่าทางของมนุษย์ จุดสังเกตใบหน้า และการติดตามมือแบบตลอดเวลาสามารถจับจุดสำคัญบนร่างกายได้ถึง 540 จุด (ท่าทาง 33 จุด มือข้างละ 21 จุด และใบหน้า 468 จุด) มีเดียไปป์เป็นเครื่องมือที่มุ่งเน้นในด้านการประมวลผลภาพโดยใช้ตัวแบบทางปัญญาประดิษฐ์เครือข่ายประสาททางคอมพิวเตอร์ (Neural Network) เพื่อการเรียนรู้และตรวจจับวัตถุที่อยู่ในภาพ ซึ่งเป็นเฟรมเวิร์กโอเพ่นซอร์สที่ออกแบบมาโดยเฉพาะการรับรู้ที่ซับซ้อนของการประมวลผลเพื่อการนำเสนอวิธีคิดที่รวดเร็วและแม่นยำ (Google, 2020) ดังภาพที่ 2 แสดงให้เห็นตัวอย่างความสามารถของเครื่องมือมีเดียไปป์

มีเดียไปป์มีทำงานโดยผสานโมเดลท่าทาง ใบหน้า และมือเข้าด้วยกัน แต่ละโมเดลได้รับการปรับแต่งให้เหมาะกับการใช้งานเฉพาะด้าน แต่ด้วยลักษณะการทำงานที่ต่างกัน ข้อมูลที่เหมาะกับโมเดลหนึ่งอาจไม่เหมาะกับอีกโมเดล ตัวอย่างเช่น โมเดลท่าทางอาจใช้เฟรมวิดีโอความละเอียดต่ำ แต่เมื่อต้องแยกส่วนมือและใบหน้าเพื่อส่งต่อให้โมเดลอื่น ความละเอียดภาพอาจต่ำเกินไป ส่งผลต่อความแม่นยำในการประมวลผล ด้วยเหตุนี้

มีเดียไปป์จึงออกแบบเป็นไปป์ไลน์ที่มีหลายขั้นตอน โดยแต่ละขั้นตอนจะประมวลผลส่วนต่าง ๆ ของภาพด้วยความละเอียดที่เหมาะสม วิธีนี้ช่วยให้นักพัฒนาสร้างแอปพลิเคชันใหม่ ๆ ได้ง่าย และสามารถนำไปใช้งานบนมือถือ เว็บเบราว์เซอร์ และแพลตฟอร์มอื่น ๆ ได้อย่างราบรื่น เป็นประโยชน์ในการสร้างพื้นที่ให้งานวิจัยในอนาคตและเปิดโอกาสให้สามารถสร้างนวัตกรรมใหม่ เช่น การตรวจจับภาษากาย การสั่งการโดยไม่ต้องสัมผัส หรือนวัตกรรมที่ซับซ้อนขึ้นได้ต่อไป (Lugaresi, et al., 2019; Sertis, 2564)



ภาพที่ 2 ตัวอย่างการตรวจจับใบหน้า และอวัยวะส่วนต่าง ๆ ของ MediaPipe (Google, 2020)

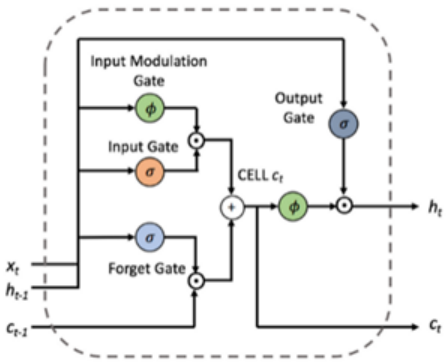
โครงข่ายความจำระยะสั้นแบบยาว (Long Short-Term Memory Networks)

โครงข่ายความจำระยะสั้นแบบยาว (Long Short-Term Memory networks: LSTMs) เป็นชนิดพิเศษของโครงข่ายระบบประสาทเทียมแบบถ่ายทอดซ้ำ (Recurrent Neural Networks: RNNs) ที่สามารถเรียนรู้ความสัมพันธ์ในระยะยาวได้ ถูกนำเสนอโดย Hochreiter และ Schmidhuber (1997) และได้รับการปรับปรุงจนได้รับความนิยม ต่อมามีการทำงานของการเรียนรู้ของเครื่องจักรได้อย่างยอดเยี่ยมในหลายประเภทของปัญหาและจึงได้มีการถูกใช้กันอย่างแพร่หลายในปัจจุบัน (Hochreiter & Schmidhuber, 1997)

ในภาพที่ 3 แสดงกระบวนการทำงานของโครงสร้างของ LSTM มีกระบวนการโดยประกอบด้วย Cell State (c_t) เป็นเสมือนหน่วยความจำหลักที่เก็บข้อมูลสำคัญในแต่ละขั้นตอนของการประมวลผล ข้อมูลใน Cell State จะไหลผ่านไปยังขั้นตอนถัดไป โดยมีการควบคุมจากประตูทั้ง 3 ประเภท ได้แก่ ประตูลืม (Forget Gate) จะตัดสินใจว่าข้อมูลใดใน Cell State (c_{t-1}) ที่ไม่จำเป็นจะถูกลบออก โดยใช้ข้อมูลจาก Hidden State ก่อนหน้า (h_{t-1}) และ Input ปัจจุบัน (x_t) เพื่อคำนวณผลผ่านฟังก์ชัน Activation (เช่น Sigmoid) และสร้างค่าที่ใช้กำหนดว่าจะ "ลืม" ข้อมูลใด ประตูเข้า (Input Gate) ทำหน้าที่ควบคุมว่าข้อมูลใหม่ (x_t) ที่ได้รับเข้ามาจะถูกอัปเดตใน Cell State (c_t) อย่างไร โดยมีการทำงานร่วมกับ Input Modulation Gate ที่สร้างค่าที่เหมาะสมจากข้อมูล Input และ Hidden State ก่อนหน้า และประตูออก (Output Gate) กำหนดว่าข้อมูลใดจาก Cell State (c_t) จะถูกส่งออกไปยัง Hidden State (h_t) เพื่อใช้ในขั้นตอนถัดไปของกระบวนการสรุปได้ คือ ประตูลืม (Forget Gate) จะลบข้อมูลที่ไม่จำเป็นออกจาก Cell State (c_t) ในขณะที่ประตูเข้า (Input Gate) และ Input Modulation Gate

จะอัปเดตข้อมูลใหม่ลงใน Cell State (c_t) และ ประตูออก (Output Gate) จะกำหนดว่าข้อมูลใดจาก Cell State (c_t) จะถูกส่งออกไปยัง Hidden State (h_t) เพื่อใช้ในขั้นตอนถัดไปของกระบวนการ

ในการทำงานของ LSTM กับภาษามือจึงมีความเหมาะสมกันโดยที่ภาษามือเป็นภาษาที่ต้องใช้การเคลื่อนไหวอย่างต่อเนื่องในรูปแบบ Sequential Data โดยสามารถเรียนรู้ความสัมพันธ์ระหว่างท่าทางที่เกี่ยวข้องกับตัวอักษรไทยแต่ละตัว จะเริ่มจากทำการป้อนข้อมูลเข้าไป โดยในงานนี้ข้อมูลดังกล่าวจะเป็นข้อมูลในรูปแบบภาพที่ถูกแปลงเป็น Feature Vectors ซึ่งเป็นชุดของค่าของตัวเลขที่แทนคุณลักษณะสำคัญของภาพในแต่ละ Vector ประกอบด้วยค่าที่เป็นจุดสำคัญ (Keypoints) บนมือถูกจับตามตำแหน่ง x และ y จากนั้นข้อมูลจะถูกส่งไปที่ประตูเข้าเพื่อคำนวณว่าข้อมูลใดจะไหลเข้า Cell State (c_t) หลังจากนั้นจะเก็บข้อมูลที่สำคัญจากข้อมูลใหม่ (x_t) ที่ถูกส่งเข้าไปและข้อมูลเดิม (c_{t-1}) จะถูกรวมกันจากนั้นประตูลืมจะทำการคำนวณว่าข้อมูลใดใน Cell State ที่ไม่จำเป็นก็จะถูกลบออกจากนั้น ประตูออกจะทำการคำนวณว่าข้อมูลใดจาก Cell State จะถูกส่งต่อไปยัง Hidden State เพื่อนำข้อมูลที่ถูกส่งมาจัดเก็บและทำการแสดงผลออกมาผ่านในตัวแบบโดยในที่นี้คือ ตัวแบบของภาษามือตัวอักษรไทย (Hochreiter & Schmidhuber, 1997)



ภาพที่ 3 แผนภาพโครงสร้างโครงข่ายความจำระยะสั้นแบบยาว

ในงานของ Jintanachaiwat et al. (2024) ได้มีการพัฒนาและทดลองโมเดลเพื่อค้นหาโมเดลที่เหมาะสมที่สุดสำหรับการแปลภาษามือไทยเป็นข้อความแบบเรียลไทม์ โดยการวิจัยนี้ใช้ข้อมูลจากจุดสำคัญของท่าทางมือ ใบหน้า และร่างกาย ที่ถูกดึงออกมาด้วย MediaPipe Holistic เพื่อวิเคราะห์และแปลผล โดยทดลองกับตัวแบบทั้งหมด 5 ตัวแบบ ได้แก่ RNN, Bidirectional-RNN, LSTM, Bidirectional-LSTM และ FNN-LSTM ผลลัพธ์ที่ดีที่สุดซึ่งได้มาจากการทดลองแบบลองผิดลองถูก (Trial-and-Error) ตัวแบบหน่วยความจำระยะสั้นแบบยาวเพื่อระบุตัวแบบที่เหมาะสมที่สุดสำหรับภาษามือที่มีการเคลื่อนไหวแบบต่อเนื่อง ในงานดังกล่าวได้ใช้ข้อมูลจำนวน 30 เฟรมที่ป้อนเข้าไปและเรียงลำดับกันสำหรับการทำนายการแปลแต่ละครั้ง

ในงานวิจัยดังกล่าวเลือกใช้ MediaPipe Holistic สำหรับการดึงคุณลักษณะ (Feature Extraction) ในบริบทนี้หมายถึงการใช้เทคโนโลยีที่สามารถวิเคราะห์และระบุตำแหน่งของ จุดสำคัญ (Keypoints) บนร่างกายได้

กระบวนการนี้จึงช่วยให้แปลงข้อมูลดิบ (Raw Data) ให้กลายเป็นข้อมูลเชิงตัวเลข (Numerical Data) และนำไปวิเคราะห์ต่อโดยใช้ตัวแบบหลัก 3 ประเภท ได้แก่ RNN/Bidirectional RNN, LSTM/Bidirectional LSTM และ FNN-LSTM เนื่องจากมีความสามารถในการจัดลำดับหรือเรียงลำดับของการเคลื่อนไหวท่าทางแบบต่อเนื่องได้ โดยทางผู้วิจัยได้เลือกคำศัพท์ 5 หมวดจากทั้งหมด 13 คำในภาษาไทย ซึ่งใช้บ่อยในการประชุมช่วงสถานการณ์โควิด-19 และในงานดังกล่าวได้ทำทดสอบแล้วพบว่า ตัวแบบ LSTM ให้ผลลัพธ์ดีที่สุดในชุดข้อมูลทดสอบด้วยความแม่นยำร้อยละ 100 แต่เมื่อทดสอบแบบเรียลไทม์ ความแม่นยำลดลงเหลือร้อยละ 86 ในขณะที่ RNN ให้ความแม่นยำร้อยละ 100 บนชุดข้อมูลทดสอบ แต่แสดงผลด้อยลงเมื่อทดสอบแบบเรียลไทม์ โดยได้ผลลัพธ์ร้อยละ 64 ในทางกลับกัน Bi-LSTM ไม่เหมาะสมสำหรับภาษามือไทยแบบต่อเนื่องเนื่องจากความซับซ้อนของการจำแนกลำดับการเคลื่อนไหวด้วยเหตุนี้ตัวแบบ LSTM จึงเหมาะสมที่สุดสำหรับการแปลภาษามือไทยแบบต่อเนื่อง (Jintanachaiwat et al., 2024)

เทนเซอร์โฟลว์

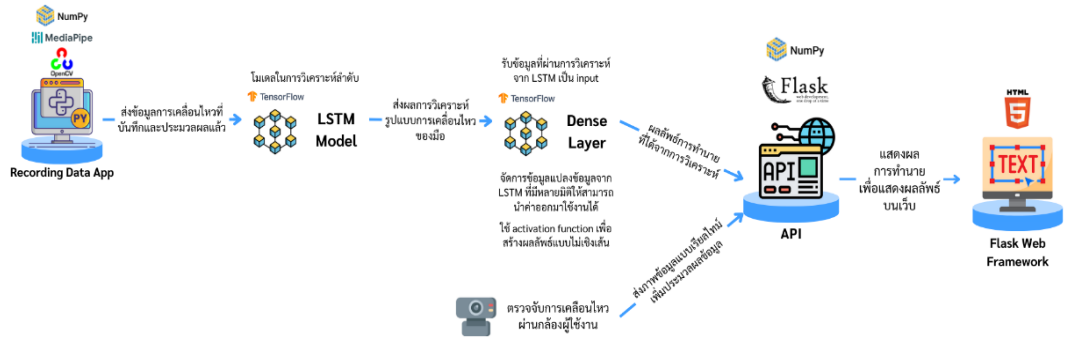
เทนเซอร์โฟลว์ (TensorFlow) เป็นไลบรารีโอเพนซอร์สสำหรับการสร้างโมเดลปัญญาประดิษฐ์ (AI) พัฒนาโดย Google ซึ่งเทนเซอร์โฟลว์เป็นสิ่งที่ช่วยให้นักพัฒนาซอฟต์แวร์ นักวิจัย และนักเรียน นักศึกษา สามารถสร้างโมเดล AI ที่หลากหลาย เช่น โมเดลการเรียนรู้ของเครื่องจักร (Machine Learning Model) โมเดลประสาทเทียม (Neural Network Model) และ โมเดลการเรียนรู้เชิงลึก (Deep Learning Model) โดยเพิ่มประสิทธิภาพกับผลิตภัณฑ์มากมาย ไม่ว่าจะเป็น เครื่องมือค้นหา (Search Engine) การแปลภาษา (Translation) คำบรรยายภาพ (Image Captioning) และเครื่องมือช่วยการเสนอแนะ (Recommendations) กล่าวคือ Google นำปัญญาประดิษฐ์มาช่วยให้พัฒนาประสบการณ์ของผู้ใช้ ทั้งในแง่ความเร็วของผลลัพธ์และในแง่ของผลลัพธ์ ที่ถูกต้องแม่นยำมากขึ้น ส่วนในงานวิจัยครั้งนี้จะใช้เทนเซอร์โฟลว์ในการนำมาคำนวณและพัฒนาในการเรียนรู้ของโมเดลในตัวอักษรภาษาไทยที่มาจากภาษามือ โดยสถาปัตยกรรมของเทนเซอร์โฟลว์ สามารถแบ่งออกด้วยกันเป็น 3 ส่วน ได้แก่ การเตรียมประมวลผลข้อมูล การสร้างแบบจำลอง การฝึกฝนและประเมินแบบจำลอง

ส่วนประกอบของเทนเซอร์โฟลว์ ส่วนที่ 1 คือ Tensor การคำนวณทั้งหมดที่เกี่ยวข้องเวกเตอร์และเมทริกซ์หลายมิติ (Multidimensional Matrix) ที่มีข้อมูลอยู่อยากหลายชนิด ค่าทั้งหมดในหนึ่ง Tensor จะมีขนาดของข้อมูลแตกต่างกันไปที่เรียกว่ารูปร่าง (Shape) ส่วนต่อมา คือ Graphs โดยจะเป็นตัวรวบรวมและอธิบายชุดการคำนวณทั้งหมดในระหว่างการฝึกสามารถทำงานผ่าน CPUs และ GPUs ได้หลายตัว ทั้งยังทำงานผ่านมือถือได้ ความสามารถในเรื่องการพกพา (Portability) ในความสามารถนี้จึงทำให้ เทนเซอร์โฟลว์มีความสามารถในการทำงานข้ามแพลตฟอร์ม (Cross-platform) ทำให้โมเดลที่สร้างสามารถนำไปใช้งานได้บนหลากหลายอุปกรณ์ ทั้ง Desktop, Mobile และ Cloud platforms โดยไม่ต้องเขียนโค้ดใหม่ ช่วยให้การนำโมเดลไปใช้งานจริงทำได้สะดวกและรวดเร็วและสามารถบันทึกกราฟเพื่อดำเนินการต่อในอนาคต การคำนวณทั้งหมดในกราฟเกิดจาก Tensor ที่เชื่อมไว้ด้วยกัน (Goldsborough, 2016)

เคราส (Keras) เป็นวิธีเรียกใช้โปรแกรม (Application Programming Interface) ด้านโครงข่ายประสาทระดับสูงที่เขียนด้วยภาษา Python และสามารถทำงานบนเทนเซอร์โฟลว์ได้ โดย เคราส ช่วยให้สามารถสร้างโครงข่ายประสาทที่รองรับทั้งแบบโครงข่ายประสาทแบบคอนโวลูชัน (CNNs) โครงข่ายประสาทเทียมแบบวนซ้ำ (RNNs) และโครงข่ายความจำระยะสั้นแบบยาว (LSTMs) ในการสร้างโมเดลการเรียนรู้ของเครื่องจักรด้วยเคราส ในงานวิจัยครั้งนี้ใช้โมเดลแบบ Sequential Model บน API ของเคราชเพื่อเพิ่มชั้น LSTM Layers และ Dense Layers ทำการเชื่อมชั้นเข้าด้วยกัน (Joseph, Nonsiri, & Monsakul, 2021)

4. วิธีดำเนินการวิจัย

ในงานวิจัยนี้ ได้ใช้ โมเดล LSTM สำหรับการรู้จำตัวอักษรไทยในภาษามือ โดยมีการใช้ ชุดข้อมูลภาพมือ ที่แสดงไว้ในตารางที่ 1 ซึ่งแต่ละภาพประกอบด้วยตำแหน่งจุดสำคัญของมือ โดยกระบวนการทำงานที่เชื่อมโยงระหว่างชุดข้อมูลภาพมือกับโมเดล LSTM โดยเป็นการการพัฒนาเว็บแอปพลิเคชันที่มีจุดประสงค์เพื่อสร้างระบบตรวจจับท่าทางของสัญลักษณ์ที่แสดงออกมาจากท่าทางของมือเป็นตัวอักษรในภาษาไทยที่กำหนดแบบเรียลไทม์ โดยใช้ฐานข้อมูลสำหรับภาษามือไทย (Thai Sign Language: ThSL) ในการดำเนินงานให้ได้ผลลัพธ์ตามจุดประสงค์จะมีกระบวนการคือ การเตรียมข้อมูล, การฝึกฝนและทดสอบ, การพัฒนาเว็บแอพพลิเคชั่น ภาพที่ 4 แสดงกรอบแนวคิดของทั้ง 3 กระบวนการ



ภาพที่ 4 กรอบแนวคิดการดำเนินงาน

การเตรียมข้อมูลเพื่อใช้การฝึกฝนการเรียนรู้ของเครื่องจักรและการทดสอบ

จากภาพที่ 4 ที่แสดงให้เห็นถึงแผนภาพโครงสร้างของระบบ ในกระบวนการที่ 1 องค์ประกอบย่อยแรก (Recording Data App) ทำหน้าที่ในการเก็บข้อมูลเพื่อนำมาวิเคราะห์กับตัวแบบการเรียนรู้ของเครื่องจักร

การเก็บข้อมูลในกระบวนการที่ 1 นี้จะใช้เครื่องมือการเรียนรู้ของเครื่องจักรที่ชื่อว่ามีเดียไปป์ (MediaPipe) ในการตรวจจับท่าทางของมือ โดยการสร้างจุดสำคัญบนมือทั้งหมด 126 จุดเพื่อเก็บข้อมูล ประกอบไปด้วยการกำหนดจุดสำคัญของมือซ้ายและมือขวาจำนวน 21 จุด ทำการกำหนดค่าสี 3 สี ในการเก็บข้อมูลที่แสดงผ่านออกทางหน้าจอซึ่งกำหนดจุดสำคัญของมือซ้าย จำนวน 21 * 3 จุดและมือขวา จำนวน 21 * 3 จุด

มีการกำหนดค่าสีในการเก็บข้อมูลที่แสดงผ่านออกทางหน้าจอ สร้างจุดสำคัญต่าง ๆ บนมือและส่งค่าการเก็บข้อมูลแสดงออกผ่านทางหน้าจอ โดยทำการสร้างโฟลเดอร์ข้อมูลเก็บไว้ใน DATA_PATH ที่กำหนดไว้เป็นในรูปแบบของอาร์เรย์ โดยภายในชุดข้อมูลมีการเก็บท่าทางอาร์เรย์เป็นโฟลเดอร์แยกกันตามจำนวนชุดข้อมูลที่กำหนด โดยการเก็บไฟล์ข้อมูลของวีดีโอนำมาเป็นภาพโฟลเดอร์ละ 30 เฟรม ภายในอาเรย์มีทั้งหมด 30 โฟลเดอร์ และความยาวของ 1 เฟรมภาพ มีทั้งหมดเท่ากับ 30 วินาทีจะเรียงและสรุปให้เข้าใจ ภาพมือจากชุดข้อมูลจะถูกแปลงเป็น Feature Vectors ซึ่งประกอบด้วยตำแหน่งสำคัญของจุดต่าง ๆ จะแทนความสัมพันธ์เชิงโครงสร้างบนมือในรูปแบบพิกัด x และ y ดังภาพตัวอย่างในตารางที่ 1 และเมื่อเตรียมชุดข้อมูลเรียบร้อยแล้วจะใช้ข้อมูลดังกล่าวเพื่อป้อนข้อมูลเข้าสู่โมเดลการฝึกฝนเครื่องจักรด้วยตัวแบบ LSTM ต่อไป Feature Vectors ที่ได้จะถูกป้อนเข้าสู่โมเดล LSTM ทีละลำดับ (Sequence) เพื่อให้โมเดลสามารถวิเคราะห์ลักษณะเชิงลำดับของท่าทางมือได้

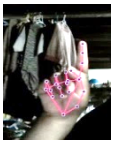
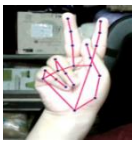
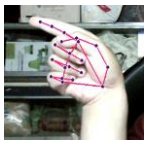
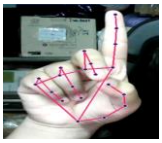
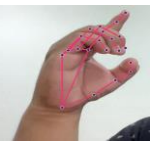
การฝึกและทดสอบตัวแบบการเรียนรู้ของเครื่องจักรด้วยตัวแบบ LSTM

ภาพที่ 4 แสดงให้เห็นถึงแผนภาพโครงสร้างของระบบ ในกระบวนการที่ 2 เป็นขั้นตอนการแบ่งข้อมูลเพื่อการฝึกและทดสอบ โดยใช้ร้อยละ 80 ของข้อมูลสำหรับการฝึก และร้อยละ 20 สำหรับการทดสอบ ซึ่งการแบ่งข้อมูลนี้ช่วยให้การฝึกฝนมีความน่าเชื่อถือและสามารถประเมินความแม่นยำของโมเดลได้ โดยทำการแบ่งข้อมูลออกเป็นส่วน ๆ และแสดงผลข้อความออกไปยังเทอร์มินอลเพื่อตรวจสอบความครบถ้วนของการแบ่งข้อมูล

ในกระบวนที่ 2 หลังจากที่ชุดข้อมูลพร้อมแล้ว จะใช้ TensorFlow และ Keras สร้างโมเดล LSTM ซึ่งเป็นโครงข่ายประสาทเทียมที่เหมาะสมสำหรับข้อมูลแบบลำดับ (Sequential Data) ซึ่งเป็น API สำหรับ Neural Networks ระดับสูงที่เขียนขึ้นด้วยภาษา Python ซึ่งเป็น Open-Source โดย Keras ทำหน้าที่เป็นอินเตอร์เฟซสำหรับ Artificial Neural Networks ที่ทำให้การทำงานง่ายขึ้นในการสร้างโมเดลทดลอง และการใช้งานเพื่อใช้ในการฝึกและทดสอบโมเดลแบบสองชั้นของ LSTM Layer จะทำหน้าที่วิเคราะห์ความสัมพันธ์เชิงเวลาในชุดข้อมูลในการผสานกัน Dense Layer ซึ่งเป็น Fully Connected Layers จะรับข้อมูลที่ถูกประมวลผลจาก LSTM Layer เพื่อทำการจำแนกประเภท หรือ ทำนายผลลัพธ์ในกรณีนี้คือการทำนายท่าทางของมือ โมเดลจะเรียนรู้ลักษณะเฉพาะของลำดับการเปลี่ยนแปลงของจุดสำคัญที่สัมพันธ์กับตัวอักษรแต่ละตัวในภาษามือไทย สุดท้ายจะใช้ Softmax เป็นฟังก์ชันทางคณิตศาสตร์ที่รับเวกเตอร์ของคะแนนตัวเลขเป็น Input และสร้างการกระจายความน่าจะเป็นบนหลาย ๆ หมวดหมู่ ซึ่งมักใช้เป็น Activation Function ในชั้น Output ของ Neural Network

ภายในกระบวนการฝึกฝน ในงานนี้จะมีการนำอัลกอริทึมมาช่วยเพิ่มประสิทธิในการทำงานของตัวแบบ เรียกว่า “Adam Optimizer” หรือ Adaptive Moment Estimation Optimizer เพื่อปรับปรุงค่าน้ำหนักของตัวแบบในขณะฝึกฝนเพื่อการเรียนรู้ของเครื่องจักร LSTM โดยลดค่าฟังก์ชันสูญเสีย (Loss Function) และปรับค่าน้ำหนักให้เหมาะสม จากผลการฝึกโมเดลจำนวน 500 รอบ (Epochs) ในการฝึกโมเดล 1 ครั้ง โดยฝึกฝนให้ค่าฟังก์ชันสูญเสียลดลงและให้ค่าความแม่นยำเพิ่มขึ้น (Accuracy) ให้เข้าใกล้ค่า 1 มากที่สุด การฝึกโมเดลคือกระบวนการปรับน้ำหนักของโมเดลเพื่อให้มันให้ผลลัพธ์ที่ดีขึ้นตามข้อมูลการฝึก

ตารางที่ 1 ภาพตัวอย่างชุดข้อมูลรูปภาพ และจุดสำคัญที่ใช้ตัวแบบ

ร อ	ก เอ๋ย ก ไก่	ข ไข่ อยู่ในเล้า	ค ควาย เข้านา	ฆ ระฆัง ข้างฝา	จ จาน ใช้ดี
					
ต เต่า หลังตุง	ถ ถุง แบกขน	ฐ สันฐาน เข้ามารอง	ฒ ผู้เฒ่า เดินย่อง	ฑ นางมนโฑ หน้าขาว	ญ หญิง โสภา
					
ฏ ปฏัก หุนหัน	ส เสือ ดาวคะนอง	ศ ศาลา เงียบเหงา	ษ ฤๅษี หนวดยาว	ช โช่ ล่ามที	ม ม้า คึกคัก
					
พ พาน วางตั้ง	ป ปลา ตากลม	ผ ผึ้ง ทำรัง	ภ สำเภา กางใบ	ห หีบ ใส่ผ้า	น หนู ขวักไขว่
					
ฮ นกฮูก ตาโต	บ ใบไม้ ทับถม	ร เรือ พายไป	ว แหวน ลงยา	ด เด็ก ต้องนิมนต์	ง งู ใจกล้า
					
ฎ ชะฎา สวมพลัน	ฟ ฟัน สะอาดจัง	ฝ ฝา ทนทาน	ล ลิง ไต่ราว	ย ยักษ์ เขี้ยวใหญ่	ท ทหาร อดทน
					
ธ ธง คนนิยม	ฉ ฉิ่ง ตีดัง	ช ช้าง วิ่งหนี	ฌ กะเฌอ คู่กัน	อ อ่าง เนืองนอง	ฬ จุฬา ท่าผยอง
					

การประมวลผลข้อมูลภาพในโมเดล LSTM ผ่านกระบวนการ Gate Mechanisms (Input Gate, Forget Gate และ Output Gate)

ประตูลืม Forget Gate ทำหน้าที่ตัดสินใจว่าข้อมูลจุดสำคัญใดที่ไม่สำคัญและควรถูกลบออกจาก Cell State ในขณะที่ประตูเข้า (Input Gate) เพิ่มข้อมูลจุดสำคัญใหม่จากภาพที่ป้อนเข้าสู่ Cell State โดยประตูออก (Output Gate) ส่งผลลัพธ์ของข้อมูลที่สำคัญไปยัง Hidden State เพื่อการจำแนกตัวอักษร หลังจากนั้นผลลัพธ์จาก Hidden State จะถูกส่งผ่านชั้นสุดท้าย (Output Layer) หรือ Activation Function เพื่อจำแนกตัวอักษรที่สอดคล้องกับท่าทางในภาพมือดังตารางที่ 1 โดยที่โมเดล LSTM นี้ สามารถเชื่อมโยงท่าทางมือที่แตกต่างกันเข้ากับตัวอักษรไทยได้อย่างแม่นยำ

การแสดงผลบนเว็บแอปพลิเคชันผ่านฟลาสก์เฟรมเวิร์ค

จากภาพที่ 4 ซึ่งเป็นแผนภาพโครงสร้างของระบบ และในกระบวนการที่ 3 นำเข้าโมดูลฟลาสก์หรือ API ใน Python เพื่อนำเข้ามาใช้เป็นตัวรันเว็บการกำหนดชื่อโมดูลฟราสก์เพื่อสร้างอ็อบเจ็กต์ของฟราสก์ใน Python ส่วนนี้ใช้ในการสร้างเซิร์ฟเวอร์ฟลาสก์ในโปรแกรม แสดงไปยังไคลเอ็นต์ผู้ใช้โดยแสดงเป็น HTML บนหน้าเว็บไซต์ดังตัวอย่างส่วนประสานงานผู้ใช้ที่แสดงไว้ในภาพที่ 5

การส่งข้อมูลภาพที่ทำการวิเคราะห์ขึ้นไปยังเว็บไซต์และส่งค่ากลับข้อมูลวิดีโอเป็น multipart/x-mixed-replace คือต้องเป็นเนื้อหาที่เปลี่ยนแปลงอยู่เสมอ และส่งไปยังเบราวเซอร์ในรูปแบบที่ถูกต้องเพื่อให้เกิดการอัปเดตแบบต่อเนื่อง ซึ่งจะอนุญาตให้สตรีมวิดีโอไปยังเบราว์เซอร์ของไคลเอ็นต์ผู้ใช้



ภาพที่ 5 ตัวอย่างส่วนประสานงานผู้ใช้ของเว็บแอพพลิเคชั่น

4. ผลการวิจัย

ความเหมาะสมของ LSTM กับภาษามือไทย

ความสามารถในการจัดการข้อมูลแบบลำดับภาษามือเป็นข้อมูลที่ต้องอาศัยลำดับการเคลื่อนไหวของมือและนิ้ว เช่น การเปลี่ยนตำแหน่งมือในแต่ละเฟรมเพื่อสร้างความหมายของตัวอักษร (เช่น ก เอ๋ย ก ไก่ หรือ ค ควาย เข้านา) ซึ่ง LSTM ถูกออกแบบมาเพื่อจดจำและเข้าใจความสัมพันธ์ของข้อมูลที่มีลำดับได้อย่างมีประสิทธิภาพ

การเชื่อมโยงกันในการเคลื่อนไหวในแต่ละท่าทางเช่น การแสดงตัวอักษร ก เอ๋ย ก ไก่ เปรียบเทียบกับ ค ควาย เข้านา จะมีลักษณะคล้ายกันในบางเฟรม แต่การเปลี่ยนแปลงลำดับของจุดสำคัญในช่วงเวลา (Time Steps) จึงทำให้จดจำและเรียนรู้ความสัมพันธ์ระยะยาว (Long-Term Dependencies) ระหว่างเฟรมต่าง ๆ ทำให้สามารถเชื่อมโยงการเคลื่อนไหวในลำดับต่าง ๆ และแยกแยะตัวอักษรที่คล้ายกันได้อย่างมีประสิทธิภาพ

ตัววัดที่ใช้ในการวิเคราะห์คุณภาพเชิงปริมาณของการฝึกฝนตัวแบบ LSTM

ในการวิเคราะห์เชิงปริมาณของชุดข้อมูลฝึกฝนและทดสอบ เพื่อสร้างผลการประเมินแบบโมเดล และมีการสร้างเมทริกซ์ออกมาเป็นแผนภาพ จนมีการประมวลผลออกมาเป็นค่าให้ได้เข้าใจง่ายมากขึ้นโดยมีค่า 4 ค่า ดังต่อไปนี้

ค่าที่หนึ่งคือค่า True Positive (TP) คือจำนวนของกรณีที่ระบบหรือโมเดลทำนายว่าเป็นคลาสบวก (Positive Class) สอดคล้องกับข้อมูลจริงที่เป็นคลาสบวก เช่น สรุปได้ว่าระบบทำนายถูกต้องว่าเป็นคลาสบวก ค่า ค่าที่ 2 คือ ค่า True Negative (TN) คือ จำนวนของกรณีที่ระบบหรือโมเดลทำนายว่าเป็นคลาสลบ (Negative Class) สอดคล้องกับข้อมูลจริงที่เป็นคลาสลบเช่นกัน สรุปได้ว่าระบบทำนายถูกต้องว่าเป็นคลาสลบ ค่า ค่าที่ 3 คือ ค่า False Positive (FP) คือ จำนวนของกรณีที่ระบบหรือโมเดลทำนายว่าเป็นคลาสบวก แต่ข้อมูลจริงเป็นคลาสลบ นั่นคือระบบทำนายผิดว่าเป็นคลาสบวก และค่าสุดท้าย ค่า False Negative (FN) คือ จำนวนของกรณีที่ระบบหรือโมเดลทำนายว่าเป็นคลาสลบ แต่ข้อมูลจริงเป็นคลาสบวก นั่นคือระบบทำนายผิดว่าเป็นคลาสลบทั้ง 4 ค่าจะถูกนำไปใช้ในการคำนวณเพื่อการทดสอบคุณภาพของตัวแบบฝึกฝน ซึ่งจะกล่าวในหัวข้อถัดไป

ค่าความแม่นยำรวม (Accuracy)

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

สมการที่ 1 หาความแม่นยำรวมโดยนับจำนวนครั้งที่ระบบทำนายถูกต้องและนับครั้งที่ระบบทำนายผิด จากนั้นนำจำนวนครั้งที่ระบบทำนายถูกต้อง (TP + TN) มาหารด้วยจำนวนครั้งที่ทั้งหมดที่ระบบทำนายทั้งหมด (TP + TN + FP + FN) ที่แสดงสัดส่วนความสัมพันธ์ระหว่างการทำนายที่ถูกต้องกับการทำนายทั้งหมด

ค่าความแม่นยำ (Precision)

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

สมการที่ 2 หาความแม่นยำ บอกถึงความแม่นยำในการทำนายคลาสบวก (Positive Class) หรือความสามารถในการตรวจจับคลาสบวกโดยเฉพาะ คำนวณจาก True Positives (TP) และ False Positives (FP) มีค่าระหว่าง 0 ถึง 1 โดยค่า 1 หมายถึงระบบทำนายคลาสบวกที่มีความแม่นยำสมบูรณ์ และค่า 0 หมายถึงระบบทำนายคลาสบวกที่ไม่มีความแม่นยำเลย

ค่าการจดจำ (Recall)

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

สมการที่ 3 คือการวัดความสามารถในการตรวจจับคลาสบวกและความเป็นคลาสบวกในข้อมูลจริง หมายความว่าข้อมูลเกี่ยวกับความสามารถที่จะไม่มีการพลาดในการตรวจจับคลาสบวกเดิมและประสิทธิภาพของระบบ

ค่าคะแนน F1 (F1 Score)

$$\text{F1 Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

สมการที่ 4 คือประเมินประสิทธิภาพของระบบหรือโมเดลในการจำแนกข้อมูลโดยรวมความแม่นยำและความเร็วเข้าด้วยกันค่า F1 Score มีค่าระหว่าง 0 ถึง 1 โดยค่า 1 หมายถึงระบบที่มีความแม่นยำและความเร็วสูงสุดและค่า 0 หมายถึงระบบที่มีความแม่นยำและความเร็วต่ำสุด

ค่า Support

จำนวนข้อความในแต่ละตัวอักษรภาษาไทยจากแหล่งข้อมูลที่ใช้ฝึกฝนโมเดล ค่านี้ไม่มีสูตรการหาจำนวนตัวอย่างได้โดยตรง สรุปแล้วช่วยให้ระบุคลาสที่โมเดลทำนายได้ดีหรือไม่ดี ขึ้นอยู่กับจำนวนมากและน้อยของชุดข้อมูลโดยค่านี้ไม่ได้บ่งบอกถึงประสิทธิภาพของโมเดล จึงควรพิจารณา ค่า Support ร่วมกับตัวชี้วัดอื่น ๆ เช่น Precision, Recall, F1-Score

ผลลัพธ์การฝึกตัวแบบด้วยโครงข่ายประสาทเทียม (LSTM)

ในการฝึกฝนตัวแบบทั้งหมดจำนวน 500 รอบ โดยการฝึกข้อมูลชุดนี้ มีข้อมูลทั้ง 42 โฟลเดอร์ (ตัวอักษร) จะประกอบไปด้วย actions ที่เป็น อาเรย์ทั้งหมด 0-29 NumPy ซึ่งใช้ในการคำนวณทางคณิตศาสตร์ อธิบายได้ดังนี้คือ การเก็บข้อมูล actions แต่ละตัวอักษรภาษามือ (เช่น ก เอ๋ย ก ไก่, ข ไข่ อยู่ในเล้า) จะถูกจัดเก็บในรูปแบบของ อาเรย์ (Array) ซึ่งในแต่ละอาเรย์จะประกอบด้วยข้อมูล 30 เฟรม โดยในแต่ละเฟรมจะมีค่าของตำแหน่งจุดสำคัญของมือทั้งหมด 126 จุด (ตำแหน่ง x, y, z) ที่ตรวจจับได้จาก MediaPipe โดยมีการจัดการข้อมูลขนาดใหญ่และเมทริกซ์ ในตัวอักษรแต่ละตัวโดยใช้ NumPy ที่เป็นไลบรารีใน Python ที่ใช้ในการคำนวณทางคณิตศาสตร์ ที่จะเสนอเป็นการใช้โมเดล LSTM ที่เป็นลำดับ จากการทดลองตามผลรวมของโมเดลที่เรียงลำดับกัน (Sequential Model) จะได้โมเดลที่มีประสิทธิภาพด้วยกันทั้งหมด 6 ชั้นในโมเดล LSTM เป็นการจดจำและทำนายข้อมูลในแต่ละช่วงเวลา โดยเพิ่มชั้น Dense เป็นชั้นที่ทำหน้าที่ให้เซลล์ทั้งหมดก่อนหน้าเชื่อมต่อกัน หรือ ที่เรียกว่า Fully Connected Layer คือ

1. LSTM 64 ชั้นแรกมี 64 Memory Cells ซึ่งทำหน้าที่ในการ ดึงข้อมูลลำดับพื้นฐาน (Low-level features) ของการเคลื่อนไหว เช่น การเปลี่ยนแปลงของตำแหน่งจุดสำคัญในแต่ละเฟรม ใช้จำนวน Memory Cells ไม่มากเกินไปในชั้นแรกเพื่อให้ได้การประมวลผลเบื้องต้นที่รวดเร็ว โดยที่ เซลล์ทั้งหมด 64 เซลล์ จะทำหน้าที่จดจำข้อมูลสำคัญ และลืมข้อมูลที่ไม่สำคัญ ผ่านกระบวนการที่เรียกว่า Gate Mechanisms (Input Gate, Forget Gate และ Output Gate) โดยที่หน่วยประมวลผล 64 หน่วยนั้นจะทำงานอย่างอิสระ โดยจะพยายามเรียนรู้รูปแบบ หรือคุณลักษณะ ของข้อมูลที่ป้อนเข้ามา

2. LSTM_1 128 เพิ่มจำนวน Memory Cells เป็น 128 เพื่อเพิ่มความสามารถในการจดจำลำดับที่ซับซ้อนมากขึ้น (High-level features) เช่น การจับลำดับการเคลื่อนไหวที่เกี่ยวข้องกับรูปแบบเฉพาะของตัวอักษร เช่น "ก เอ๋ย ก ไก่" หรือ "ข ไข่ อยู่ในเล้า" ชั้นนี้ช่วยให้โมเดลเรียนรู้ ความสัมพันธ์ระยะยาวระหว่างเฟรมได้ เมื่อ Memory Cells มากขึ้นช่วยให้ LSTM สามารถจับความสัมพันธ์ของตัวอักษรไทยได้อย่างมีประสิทธิภาพ

3. LSTM_2 64 ลดจำนวน Memory Cells กลับมาเป็น 64 เพื่อ ลดการใช้ทรัพยากรและป้องกันการโอเวอร์ฟิตติ้ง (Overfitting) ของข้อมูล ชั้นนี้เป็นการรวบรวมและสรุปข้อมูล กรองคุณลักษณะที่สำคัญที่สุด (Feature abstraction) ทำให้โมเดล LSTM จดจำและสรุปผลลัพธ์จากการเคลื่อนไหวของมือได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

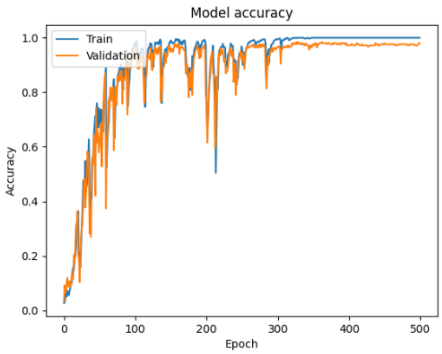
4. Dense 64 และ Dense_1 32 ทำหน้าที่เป็น Fully Connected Layers เพื่อเชื่อมต่อข้อมูลจากชั้น LSTM ทั้งหมดและประมวลผลในลำดับขั้นสุดท้าย

5. Dense_2 27 ซึ่งเป็น Output Layer ที่ใช้ Softmax Activation Function เพื่อจำแนกผลลัพธ์ ออกเป็น 42 หมวดหมู่ (ตัวอักษรไทยในภาษามือที่กำหนด)

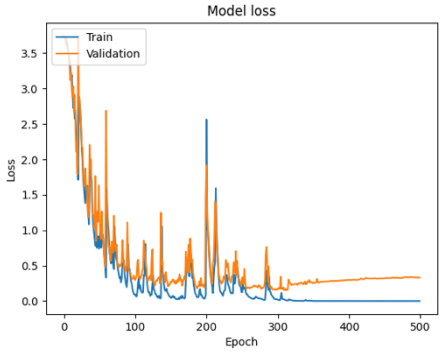
ในการทดสอบครั้งนี้จะทำการคำนวณค่าความแม่นยำของข้อมูลที่ทดสอบ และ ค่าความแม่นยำของข้อมูลที่ฝึก จะใช้ข้อมูล 42 ตัวอักษรใน ตารางที่ 1 และผลการทดสอบประสิทธิภาพ ตามตารางที่ 2 ในการฝึกฝนชุดข้อมูลและทำการทดสอบชุดข้อมูลแล้วมีการวัดผลประสิทธิภาพของโมเดลด้วยวิธีต่าง ๆ โดยขอเสนอผลความแม่นยำของโมเดลและโมเดลการฝึก โดยได้ค่าความแม่นยำของโมเดลเท่ากับ 0.98 คิดเป็นร้อยละ 98.41 และ ค่าความแม่นยำของโมเดลการฝึก เท่ากับ 0.995 คิดเป็นร้อยละ 99.60

ในการสรุปค่าที่ใช้ในการวัดและประเมินประสิทธิภาพของโมเดลการเรียนรู้เชิงลึก (Deep Learning) ระหว่างการฝึกฝนตัวแบบที่ใช้ในเทนเซอร์โฟลว์ ร่วมกับ Keras โดยใช้ epoch_categorical_accuracy และ epoch_loss แสดงในภาพที่ 6 และ ภาพที่ 7 เมื่อพิจารณาจากกราฟในภาพที่ 6 ที่แสดงความสัมพันธ์ระหว่าง จำนวนรอบการฝึกฝนกับความแม่นยำ จะพบว่า เส้นความแม่นยำบนชุดข้อมูลการฝึก (Train) อยู่เหนือเส้นความแม่นยำบนชุดข้อมูลที่ยังไม่เคยพบมาก่อนหรือชุดข้อมูลทดสอบ (Validation) เสมอ และมีความสัมพันธ์สอดคล้องไปในทิศทางเดียวกันตามจำนวนรอบการทดสอบ ค่าความแม่นยำบนชุดข้อมูลทดสอบ เพิ่มสูงขึ้นในช่วงแรก แต่หลังจากรอบการฝึกฝนประมาณ 200 ค่าเริ่มคงที่ มีค่าเข้าใกล้ 1 ที่แสดงว่าชุดข้อมูลทดสอบกับชุดข้อมูลการฝึกมีความแม่นยำอยู่ในระดับดี และจากภาพที่ 7 ที่แสดงค่าความสูญเสียกับจำนวนรอบการฝึกฝนที่ยิ่งมีรอบของการฝึกฝนมากขึ้น จำนวนการสูญเสียของโมเดลก็ลดลงจนเข้าใกล้ 0 โดยที่ค่าความสูญเสียจะลดลงเมื่อโมเดลมีประสิทธิภาพดีขึ้นมีและมีความสัมพันธ์สอดคล้องไปในทิศทางเดียวกัน

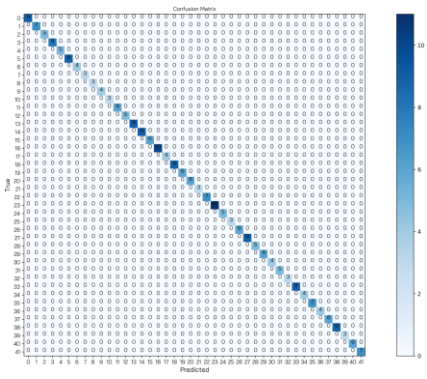
การวัดประสิทธิภาพโมเดลจาก Confusion Matrix จากชุดข้อมูลที่เก็บตัวอักษรภาษาไทย 42 ตัว จากตารางที่ 1 ใช้ชุดข้อมูลรูปภาพ Confusion Matrix ในการคำนวณประสิทธิภาพของโมเดลใช้สำหรับการวิเคราะห์ประสิทธิภาพแบบเรียลไทม์ของโมเดลนอกจากนี้ยังช่วยให้เห็นภาพของการทำนายในกรณีที่การคาดการณ์ผิดพลาดและแสดงค่าความแม่นยำตามสี แสดงดังภาพที่ 8 แสดง Confusion Matrix ของโมเดล ที่สะท้อนให้เห็นถึงแกน X ที่เป็นค่าที่ระบบคาดการณ์ได้จากตัวอักษรภาษาไทย 42 ตัว และแกน Y แทนค่าความเป็นจริงที่ระบบคาดการณ์ สามารถเห็นได้ว่าจากภาพมีตัวอักษรที่มีการทำนายที่แม่นยำสูง (ช่องที่มีสีเข้มบนแนวทแยงมุม) ในขณะที่ตัวอักษรอื่น ๆ อาจมีการทำนายที่ผิดพลาด (ช่องที่มีสีเข้มนอกแนวทแยงมุม) ซึ่งทำให้สามารถระบุความสับสนของการทำนายโมเดลได้และการวิเคราะห์นี้ยังช่วยในการทำความเข้าใจว่าโมเดลว่ามีประสิทธิภาพอย่างไรกับตัวอักษรภาษาไทย โดย Confusion Matrix จะเป็นข้อมูลที่มีค่ามากในการปรับปรุงโมเดลให้ดีขึ้น



ภาพที่ 6 ค่า epoch_categorical_accuracy ในการฝึกฝน 500 รอบที่โมเดลเรียนรู้จากชุดข้อมูล



ภาพที่ 7 ค่า epoch_loss ในการฝึกฝน 500 รอบที่โมเดลเรียนรู้จากชุดข้อมูล



ภาพที่ 8 แสดงประสิทธิภาพโมเดลจาก Confusion Matrix

ตารางที่ 2 ผลทดสอบในชุดข้อมูล

ชุดข้อมูลทดสอบ	Precision	Recall	F1-Score	Support
0: รอ	1.0000	1.0000	1.0000	5
1: ก เอ๋ย ก ไก่	1.0000	1.0000	1.0000	6
2: ข ไข่ อยู่ในเล้า	1.0000	1.0000	1.0000	10
3: ค ควาย เข้านา	1.0000	1.0000	1.0000	5
4: ฆ ระฆัง ข้างฝา	1.0000	1.0000	1.0000	4
5: ต เต่า หลังตุง	1.0000	1.0000	1.0000	7
6: ถ ถุง แบกขน	1.0000	1.0000	1.0000	2
7: ฐ สันฐาน เข้ามารอง	1.0000	1.0000	1.0000	5
8: ฒ ผู้เฒ่า เดินย่อง	1.0000	1.0000	1.0000	5
9: ท นางมนโท หน้าขาว	1.0000	1.0000	1.0000	7
10: ฏ ปฏัก หุนหัน	1.0000	1.0000	1.0000	4
11: ส เสือ ดาวคะนอง	0.8333	1.0000	0.9091	5
12: ศ ศาลา เงียบเหงา	1.0000	1.0000	1.0000	10
13: ษ ฤษี หนวดยาว	1.0000	1.0000	1.0000	8
14: ซ โซ่ ล่ามที	1.0000	1.0000	1.0000	10
15: พ พาน วางตั้ง	0.8333	1.0000	0.9091	5
16: ป ปลา ตากลม	1.0000	1.0000	1.0000	7
17: ผ ผึ้ง ทำรัง	1.0000	1.0000	1.0000	4
18: ภ สำเภา กางใบ	1.0000	1.0000	1.0000	2
19: ห หีบ ใส่ผ้า	1.0000	0.8000	0.8889	5
20: ฮ นกฮูก ตาโต	1.0000	1.0000	1.0000	9
21: บ ใบไม้ ทับถม	1.0000	1.0000	1.0000	4
22: ร เรือ พายไป	1.0000	1.0000	1.0000	5
23: ว แหวน ลงยา	1.0000	1.0000	1.0000	9
24: ด เด็ก ต้องนิมนต์	1.0000	0.7143	0.8333	7
25: ฎ ชะฎา สวมพลัน	0.7778	1.0000	0.8750	7
26: ฟ ฟัน สะอาดจัง	1.0000	1.0000	1.0000	8
27: ฝ ฝา ทนทาน	1.0000	1.0000	1.0000	6
28: ล ลิง ไต่ราว	1.0000	1.0000	1.0000	6
29: ย ยักษ์ เขี้ยวใหญ่	1.0000	1.0000	1.0000	9
30: จ จาน ใช้ดี	1.0000	1.0000	1.0000	5
31: ญ หญิง โสภา	1.0000	1.0000	1.0000	7
32: ม ม้า คึกคัก	1.0000	1.0000	1.0000	5
33: น หนู ขวักไขว่	1.0000	1.0000	1.0000	2
34: ง งู ใจกล้า	1.0000	1.0000	1.0000	6

ตารางที่ 2 ผลทดสอบในชุดข้อมูล (ต่อ)

ชุดข้อมูลทดสอบ	Precision	Recall	F1-Score	Support
35: ท ทหาร อดทน	1.0000	1.0000	1.0000	6
36: ธ ธง คนนิยม	1.0000	1.0000	1.0000	4
37: ฉ ฉิ่ง ตีดัง	1.0000	1.0000	1.0000	3
38: ช ช้าง วิ่งหนี	1.0000	1.0000	1.0000	7
39: ฌ กะเฌอ คู่กัน	1.0000	1.0000	1.0000	5
40: อ อ่าง เนืองนอง	1.0000	0.8889	0.9412	9
41: ฬ จุฬา ท่าผยอง	1.0000	1.0000	1.0000	7
accuracy			0.9841	252
macro avg	0.9868	0.9858	0.9847	252
weighted avg	0.9872	0.9841	0.9840	252

5. อภิปรายผลการวิจัยและข้อเสนอแนะ

ระบบที่นำเสนอนี้สามารถใช้ตรวจจับตัวอักษรภาษาไทย และแปลงภาพจากกล้องเว็บแคมเป็นตัวอักษรได้ มีการใช้หน่วยความจำน้อย และใช้ทรัพยากรในการประมวลผลต่ำและสามารถนำไปใช้กับอุปกรณ์คอมพิวเตอร์โดยทั่วไปได้ นอกจากนั้นตัวแบบ LSTM ที่ใช้ในการฝึกฝนการรู้จำมีประสิทธิภาพในการจดจำท่าทางได้ดี โดยจากการได้รับการฝึกฝนผ่านชุดข้อมูลที่สร้างขึ้นทั้งหมดจำนวน 600 ภาพสำหรับตัวอักษรภาษาไทยแต่ละตัวและนำมาวิเคราะห์จึงได้ดังนี้ ค่าความแม่นยำ ความถูกต้องในการตรวจจับและแปลงภาษามือไทย ค่ามาโครเฉลี่ย (Macro Average) ค่าเฉลี่ยของความแม่นยำของแต่ละตัวอักษรภาษาไทย ค่าน้ำหนักเฉลี่ย (Weighted Average) ค่าเฉลี่ยของความแม่นยำที่คำนวณตามน้ำหนักของแต่ละตัวอักษร ซึ่งพบว่าอยู่ในเกณฑ์ดีมาก (มากกว่าร้อยละ 95)

สำหรับชุดข้อมูลที่ได้สร้างขึ้นหรือกำหนดขึ้นเอง และเป็นการยืนยันว่าระบบนี้สามารถใช้ได้อย่างมีประสิทธิภาพในการโต้ตอบกับระบบคอมพิวเตอร์ผ่านเว็บแอปพลิเคชันมีค่าความแม่นยำหรือความถูกต้องอยู่ที่ร้อยละ 98.41 ซึ่งทำให้เห็นว่าระบบที่นำเสนอนี้มีความสำเร็จ และงานวิจัยชิ้นนี้สามารถนำไปต่อยอดการทำงานได้กับการเรียนรู้ของผู้พิการทางการได้ยินให้รู้จักตัวอักษรภาษาไทยผ่านการแสดงท่าทางของมือ

อย่างไรก็ตาม ปัญหาที่ผู้วิจัยพบคือในขณะทำการเก็บข้อมูลไฟล์ภาพ ปัจจัยต่อไปนี้มีผลต่อระบบท่าทางของมือที่แปลงออกมาเป็นคำ ได้แก่ ระยะภาพที่ต้องอยู่ในขอบเขตการเก็บข้อมูล อุปกรณ์ต้องมีความคมชัดและมีคุณภาพสูง ความถูกต้องของตัวแบบขึ้นกับความแม่นยำและความเร็วในการเก็บข้อมูล ใช้เวลานานต่อการเก็บชุดข้อมูลภาพในแต่ละตัวอักษา บุคคลธรรมดาต้องสามารถใช้ท่าทางภาษามือไทยได้

เพื่อหลีกเลี่ยงการเกิดปัญหาดังกล่าว ในการเก็บข้อมูลทุกครั้งจึงต้องมีการควบคุมในเรื่องของการโฟกัสของกล้องในการตรวจจับ ระยะในการจับภาพ การใช้อุปกรณ์ในการตรวจจับที่มีคุณภาพสูงโดยความเร็วในการเก็บข้อมูล นอกจากนี้ในการเก็บข้อมูลตัวอักษรทั้งหมดมีความล่าช้าเป็นอย่างมากเนื่องจากการเก็บข้อมูลต่อหนึ่ง

ตัวอักษรนั้นต้องใช้เวลาค่อนข้างนาน และบุคคลธรรมดาต้องสามารถใช้ท่าทางภาษามือไทยได้ในแต่ละตัวอักษร ปัญหาที่ตามมาก็คือคนที่ถูกใช้เป็นข้อมูลตัวอย่างต้องถูกฝึกให้มีความแม่นยำในการใช้ท่าทางภาษามือหรือต้องฝึกบุคคลธรรมดาให้สามารถใช้ท่าทางภาษามือของตัวอักษรให้มีความแม่นยำในการเก็บด้วย

การประเมินผลลัพธ์ทั้งหมดจากข้อมูลในตารางที่ 2 และการวิเคราะห์ข้อมูลออกมาเป็นค่าความแม่นยำแสดงว่าโมเดลมีประสิทธิภาพที่ดี มีความแม่นยำสูง ทั้งบนชุดข้อมูลการฝึกและชุดข้อมูลที่ยังไม่เคยพบมาก่อน และ โมเดลมีค่าความสูญเสีย (Loss) ต่ำ อย่างไรก็ตาม โมเดลอาจเกิดปัญหาการโอเวอร์ฟิตติ้งจึงควรหาทางแก้ปัญหาโอเวอร์ฟิตติ้ง ชุดข้อมูลของโมเดลที่ฝึกฝนและทดสอบเป็นส่วนสำคัญในการพัฒนาและประเมินผลของการตรวจจับชุดข้อมูลและแปลงภาษามือไทย (ThSL) และตัวแบบ LSTM ในการจดจำท่าทางของมือสำหรับตัวอักษรภาษาไทย ในการแก้ปัญหาโอเวอร์ฟิตติ้ง บทความนี้ได้ทำการแบ่งข้อมูลชุดฝึกและชุดทดสอบในอัตราส่วน 80:20 มีส่วนประกอบที่ทำให้เกิดปัญหาดังนี้ ความไม่หลากหลายของข้อมูลชุดฝึก คือ ชุดข้อมูลฝึกอาจมีลักษณะเฉพาะที่ไม่ครอบคลุม เช่น รูปแบบของการเคลื่อนไหวในบางตัวอักษรอาจมีลักษณะใกล้เคียงกัน ส่งผลให้โมเดลจดจำเฉพาะข้อมูลในชุดฝึกได้ดีเกินไป แต่ไม่สามารถประมวลผลกับข้อมูลใหม่ (Unseen Data) ได้อย่างแม่นยำ และการแบ่งข้อมูลในสัดส่วนนี้ทำให้ข้อมูลที่ใช้ในการทดสอบมีจำนวนน้อย อาจส่งผลให้การประเมินผลโมเดลไม่สะท้อนประสิทธิภาพจริง ในส่วนนี้จะแก้ไขปัญหาโดย การเพิ่มข้อมูลให้มีจำนวนเฟรมภาพมากขึ้น, ลดความซ้ำซ้อนของโมเดลโดย ลดจำนวน Memory Cells หรือ Units ในแต่ละชั้น LSTM หรือจำนวนชั้น (Layers) ในโมเดล เพื่อป้องกันโมเดลเรียนรู้ข้อมูลที่ไม่จำเป็น หลังจากนั้นตรวจสอบค่าผลลัพธ์และประเมินผลต่อไป

ข้อเสนอแนะต่องานวิจัยในอนาคต

ผลจากการทดสอบระบบนั้นยังพบว่ายังมีประเด็นอีกหลายอย่างที่สามารถนำงานนี้ไปพัฒนาต่อยอดได้ โดยขอเสนอไว้ใน 5 ประเด็น

ในประเด็นแรก การพัฒนาเทคนิคเพื่อปรับปรุงความเร็วในการเก็บข้อมูลและการแปลงภาษามือให้มีประสิทธิภาพมากยิ่งขึ้น รวมไปถึงพัฒนาระบบให้สามารถรองรับการใช้งานบนอุปกรณ์คอมพิวเตอร์ที่มีความแตกต่างกันโดยไม่สูญเสียความเร็วและความแม่นยำ รวมไปถึงการเข้าถึงบุคคลข้อมูลตัวอย่างให้มีความแม่นยำในการใช้ท่าทางภาษามือในการเก็บข้อมูล

ในประเด็นที่ 2 ในการทดสอบควรพิจารณาถึงสภาพแวดล้อมอื่น ๆ ที่เป็นไปได้ เช่น ขนาดของมือที่หลากหลาย แสงหรือสีที่ปรากฏในภาพ ให้ได้ผลการทดสอบที่สามารถรองรับการทำงานในสภาพแวดล้อมที่หลากหลายสอดคล้องกับสถานการณ์การใช้งานจริง

ในประเด็นที่ 3 ในการทดสอบประสิทธิภาพของชุดข้อมูลโดยการผ่านโมเดล LSTM ดังกล่าว ได้มีการเกิดโอเวอร์ฟิตติ้งขึ้น ซึ่งหมายถึง โมเดลเรียนรู้จากข้อมูลการฝึกฝนมากจนจำรายละเอียดของข้อมูลการฝึกฝนมากเกินไป ส่งผลให้ประสิทธิภาพบนข้อมูล Validation หรือ บนข้อมูลทดสอบที่ไม่เคยเห็นมาก่อนลดลง เพื่อพยายามหาทางแก้ปัญหาการเกิดโอเวอร์ฟิตติ้งโดยแนะนำให้เพิ่มชุดข้อมูลเฟรมภาพการฝึกฝนให้มากขึ้น ปรับแต่ง hyperparameters ของโมเดล เช่น ปรับอัตราการเรียนรู้ (Learning Rate), ปรับแต่งจำนวนรอบการ

ฝึกฝน (Number of Epochs), ปรับขนาดแบตช์ (Batch Size) และยังสามารถใช้เทคนิคการปรับลดความซับซ้อน (Regularization) คือ การปรับลดขนาดของพารามิเตอร์ทั้งหมดให้มีค่าน้อยลงจากค่าพารามิเตอร์ที่มีค่าสูง (L2 Regularization), การสุ่มปิดบางหน่วยในชั้นเพื่อป้องกันการพึ่งพามากเกินไป (Dropout)

ในประเด็นที่ 4 ในการแบ่งข้อมูลชุดฝึกและชุดทดสอบในอัตราส่วน 80:20 อาจทำให้เกิดปัญหาโอเวอร์ฟิตติ้งได้ทำให้เกิดปัญหาดังนี้ ความไม่หลากหลายของข้อมูลชุดฝึก คือ ชุดข้อมูลฝึกอาจมีลักษณะเฉพาะที่ไม่ครอบคลุม เช่น รูปแบบของการเคลื่อนไหวในบางตัวอักษรอาจมีลักษณะใกล้เคียงกัน ส่งผลให้โมเดลจดจำเฉพาะข้อมูลในชุดฝึกได้ดีเกินไป แต่ไม่สามารถประมวลผลกับข้อมูลใหม่ได้อย่างแม่นยำ และการแบ่งข้อมูลในสัดส่วนนี้ทำให้ข้อมูลที่ใช้ในการทดสอบมีจำนวนน้อย อาจส่งผลให้การประเมินผลโมเดลไม่สะท้อนประสิทธิภาพจริง ในส่วนนี้จะแก้ไขปัญหาโดย การเพิ่มข้อมูลให้มีจำนวนเฟรมภาพมากขึ้น, ลดความซ้ำซ้อนของโมเดลโดย ลดจำนวน Memory Cells หรือ Units ในแต่ละชั้น LSTM หรือจำนวนชั้นในโมเดล เพื่อป้องกันโมเดลเรียนรู้ข้อมูลที่ไม่จำเป็น

สำหรับประเด็นสุดท้าย คือการพัฒนาแอปพลิเคชันบนแพลตฟอร์มอื่น หรือการพัฒนาอุปกรณ์ที่ใช้เทคนิควิธีทางอินเทอร์เน็ตของสรรพสิ่ง ให้สามารถใช้งานได้ง่ายและสะดวกสำหรับผู้ที่ต้องการเรียนรู้ภาษามือ

6. เอกสารอ้างอิง

กรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ. (2567). สถานการณ์คนพิการ 30 กันยายน 2566. กรุงเทพฯ: กรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ. https://dep.go.th/th/law-academic/knowledge-base/disabled-person-situation/สถานการณ์คนพิการ-30-กันยายน-2565

ภาควิชาภาษาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์. (ม.ป.ป.). แบบสะกดนิ้วมือไทย. https://pirun.ku.ac.th/~fhumalt/THSL/THSL/html/nav_th/THSL_fsp_th.htm

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision*. Prentice Hall.

Goldsborough, P. (2016). A tour of TensorFlow. *arXiv preprint arXiv:1610.01178*. https://doi.org/10.48550/arXiv.1610.01178

Google. (2020, December 14). MediaPipe Holistic: Simultaneous face, hand, and pose prediction. https://blog.research.google/2020/12/mediapipe-holistic-simultaneous-face.html

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jintanachaiwat, W., Jongsathitphaibul, K., Pimsan, N., Sojiphan, M., Tayakee, A., Junthep, T., & Siriborvornratanakul, T. (2024). Using LSTM to translate Thai sign language to text in real time. *Discover Artificial Intelligence*, *4*(1), 17. https://doi.org/10.1007/s44163-024-00113-8

Joseph, F. J., Nonsiri, S., & Monsakul, A. (2021). Keras and TensorFlow: A hands-on experience. In *Advanced deep learning for engineers and scientists: A practical approach* (pp. 85-111).

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J., Chang, W.T., Hua, W., Georg, M., Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*. https://doi.org/10.48550/arXiv.1906.08172

Sertis. (2564). MediaPipe Holistic อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า มือ และท่าทางได้ในเวลาเดียวกัน. https://sertiscorp.medium.com/mediapipe-holistic-อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า-มือ-และท่าทางได้ในเวลาเดียวกัน-e1185469e111

Techhub. (2567). รู้จัก Computer Vision เทคโนโลยีที่ทำให้ AI มองเห็น. https://www.techhub.in.th/computer-vision-technology-for-ai