โปรแกรมแปลภาษามือเป็นข้อความและเสียงพูด โดยการใช้วิธีการระบุพิกัดตำแหน่งด้วยมีเดียไพพ์

ภัทรณัฏฐ์ ศรีบุญเรือง[1], พันธกรณ์ ทะนันชัย[2], คเชนทร์เทพ แขกระโทก[3], วุฒิชัย วิจิตรกุลสวัสดิ์[4], ปราโมทย์ อนันต์วราพงษ์[5]

[1, 2, 3, 4, 5] สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ
E-mail: pramote.a@mail.rmutk.ac.th

Received: Aug 11, 2022
Revised: Nov 11, 2022
Accepted: Dec 21, 2022

บทคัดย่อ

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

คำสำคัญ : ภาษามือไทย, การรู้จำภาษามือ, มีเดียไพพ์, โครงข่ายประสาทเทียม, แอลเอสทีเอ็ม

Software Sign Language Translator to Text and Speech by Using the Landmarks Technique of MediaPipe

Pattaranat Sriboonruang[1], Pantakorn Tananchai[2], Kachanthep Khaggathog[3], Wuttichai Vijitkunsawat[4], Pramote Anunvrapong[5]

[1, 2, 3, 4, 5] Division of Electronic and Telecommunication Engineering Faculty of Engineering
Rajamangala University of Technology Krungthep
E-mail : pramote.a@mail.rmutk.ac.th

Received: Aug 11, 2022
Revised: Nov 11, 2022
Accepted: Dec 21, 2022

Abstract

Many deaf people and people with disabled hearing are commonly seen in the community. Especially in Thailand, there is the second leading disability type among all. This paper studies the performance of Thai Sign Language Recognition (TSLR) with the Long Short-Term Memory (LSTM) algorithm on the landmarks technique of the MediaPipe. We separate the three scenarios for our experiments: the Finger-Spelling for digit numbers (1-9), Natural Sign Language for 20 words, and Software Sign Language Translator (SSLT). The results show that Finger-Spelling and Natural Sign Language accuracy on the SSLT are between 94% - 99% and 80.5%, respectively.

Keywords : Thai Sign Language, Sign Language Recognition, MediaPipe, Artificial Neural Network, LSTM algorithm

บทนำ

ข้อมูลจากกรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการในปี 2563 ที่ได้รับการออกบัตรประจำตัวคนพิการจำนวน 2,076,313 คน คิดเป็นร้อยละ 3.13 ของประชากรทั้งประเทศ พบว่ามีจำนวนผู้พิการทางด้านการได้ยินอยู่ 391,785 คน ซึ่งเป็นอันดับที่ 2 ของจำนวนประชากรผู้พิการในประเทศไทย ปัญหาของการบกพร่องทางการได้ยินทำให้พัฒนาการทางด้านภาษาและการสื่อสารที่ถือว่าเป็นหัวใจสำคัญของพัฒนาการเด็ก ซึ่งความบกพร่องทางการได้ยินและความไม่เข้าใจความต้องการพื้นฐานของเด็กหูหนวกหรือหูตึง เปรียบเสมือนกำแพงขวางกั้นพัฒนาการทางสังคม อารมณ์ และสติปัญญานอกจากนี้ยังพบว่าเด็กที่มีความบกพร่องทางการได้ยินมีแนวโน้มในการต้องอยู่อย่างโดดเดี่ยว อีกทั้งยังมักตกเป็นผู้ถูกรังแกหรือถูกใช้ความรุนแรง มีผลการเรียนที่ต่ำกว่าเกณฑ์ ทั้งนี้ความบกพร่องทางการได้ยินถือเป็นความผิดปกติที่สามารถนำเอาเทคโนโลยีมาช่วยเหลือให้สามารถสื่อสารกับคนปกติได้สะดวกขึ้น ในปัจจุบันได้มีการคิดค้นเครื่องมือหรืออุปกรณ์ในการช่วยแปลความหมายของภาษามือกันมากขึ้น“ภาษามือ” [1] คือ ภาษาสำหรับผู้พิการทางการได้ยิน ใช้มือ สีหน้า และกิริยาท่าทาง ประกอบในการสื่อความหมายและถ่ายทอดอารมณ์แทนการพูดภาษามือของแต่ละชาติมีความแตกต่างกัน เช่นเดียวกับภาษาพูดซึ่งแตกต่างกันตามขนบธรรมเนียมประเพณี วัฒนธรรม และลักษณะภูมิศาสตร์ เช่น ภาษามือจีน ภาษามืออเมริกัน และภาษามือไทย เป็นต้น ภาษามือสามารถแบ่งออกได้เป็น 2 ประเภท (1) ภาษามือประดิษฐ์ (Signed) คือ ภาษามือที่ครู ผู้ปกครอง หรือญาติมิตร ของผู้พิการทางการได้ยินคิดค้นแทนภาษาพูด และภาษาเขียนประจำชาติ เพื่อให้มีคำใช้ให้เพียงพอในการศึกษาและการสื่อสารความหมายโดยเฉพาะเรื่องนามธรรม ภาษามือที่ประดิษฐ์ขึ้นมานี้มักจะนำแบบสะกดนิ้วมือ (Finger-Spelling) มาประสมด้วย เช่น ชื่อเฉพาะ, ตัวอักษร, วรรณยุกต์, สระเดี่ยว, ตัวเลข เป็นต้น [2] ได้ออกแบบพัฒนาอุปกรณ์ช่วยสอนภาษามือแบบสะกดนิ้วมือบนเครื่องโทรศัพท์มือทั้งระบบ iOS และ Android ด้วย Ionic Framework [3] นำเสนอการรู้จำภาษามือโดยใช้วิธี Hand Extraction ด้วย Convolution Neural Network (CNN) อัลกอริทึมซึ่งป้อนอินพุทเป็นภาษามือแบบสะกดนิ้วมือ จำนวน 25 คำ จากวิธีดังกล่าวได้ผลลัพธ์ค่าความแม่นยำ 91.26% อย่างไรก็ตามได้ใช้ภาพนิ่งเป็นข้อมูลอินพุทซึ่งอาจจะไม่สอดคล้องหรือไม่สามารถใช้งานได้จริงในปัจจุบัน (2) ภาษามือธรรมชาติ (Natural Sign Language) คือ ภาษามือที่ผู้พิการทางการได้ยินได้สร้างขึ้นและใช้ร่วมกันในแต่ละชุมชนหรือในแต่ละชาติ ซึ่งส่วนมากเป็นท่าเลียนแบบธรรมชาติที่จะช่วยเหลือผู้พิการการได้ยินให้มีพัฒนาการในภาษาประจำชาติเท่าเทียมกับคนปกติ เช่น ว่ายน้ำ โทรศัพท์ เครื่องบิน รถไฟ ฯลฯ [4-5] ได้นำเสนอการใช้เทคนิคโครงกระดูก (Skeleton) ด้วย MediaPipe โดยใช้อินพุทเป็นภาพเคลื่อนไหว จำนวน 5 และ 15 คำ ซึ่งได้ค่าความแม่นยำ 97% และ 96% ตามลำดับ อย่างไรก็ตามงานวิจัยดังกล่าวขาดการอ้างอิงจำนวนของผู้แสดงท่าทาง (Signers) ซึ่งมีผลอย่างมากกับการรู้จำภาษามือ โดยทั่วไปการออกแบบโมเดลที่ดีต้องได้ข้อมูลอินพุทค่อนข้างหลากหลาย เนื่องจากความช้า-เร็วของท่าทางในแต่ละผู้แสดงท่าทาง และลักษณะการทำภาษามือ เช่น การพลิกมือ ยกมือ ระดับความสูงของมือทั้งสองของแต่ละผู้แสดงท่าทางมีความสำคัญการเป็นอย่างยิ่ง

งานวิจัยนี้ได้ออกแบบสร้างโปรแกรมแปลภาษามือเป็นข้อความโดยการนำโครงข่ายประสาทเทียมแบบประมวลผลลำดับ (LSTM) ให้เรียนรู้และจดจำเพื่อให้สามารถตรวจจับลักษณะท่าทางและการเคลื่อนที่ที่เกิดขึ้นจากมือทั้งสองผ่านกล้องเว็บแคม และนำไปประมวลผลแปลเป็นข้อความและเสียงพูดได้ตามที่กำหนดในมาตรฐานภาษามือ

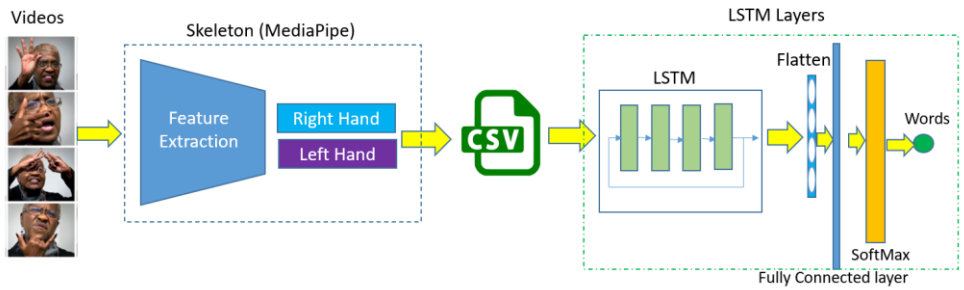
วัตถุประสงค์ของการวิจัย

1. เพื่อออกแบบโปรแกรมการแปลงภาษามือให้คนปกติสามารถเข้าใจความหมายของภาษามือได้ง่าย

2. เพื่ออำนวยความสะดวกต่อผู้พิการทางการได้ยินให้สามารถสื่อสารกับบุคคลทั่วไป

ระเบียบวิธีวิจัย

โปรแกรมแปลภาษามือเป็นข้อความถูกสร้างขึ้นโดยการประยุกต์ใช้งานมีเดียไพพ์ ซึ่งมีภาพรวมของระบบดังรูปที่ 1



รูปที่ 1 ภาพรวมของระบบ

จากรูปที่ 1 สามารถแบ่งออกได้ 4 ส่วนหลัก คือ (1) ส่วนข้อมูลอินพุท (Videos), (2) มีเดียไพพ์, (3) การเก็บพิกัดตำแหน่งของ x, y, z และ visibility (CSV) และ (4) ส่วนของ LSTM อัลกอริทึม

ในส่วนของข้อมูลอินพุทประกอบไปด้วยภาษามือแบบการสะกดนิ้วมือชนิดตัวเลข 1-9 ทั้งหมด 9 คำ และภาษามือแบบธรรมชาติทั้งหมด 20 คำ ตามตารางที่ 1 รวมทั้งหมด 29 คำ ซึ่งมีผู้แสดงท่าทางทั้ง 10 คน โดยมีลักษณะแตกต่างกัน เช่น อ้วน, ผอม, ผิวคล้ำ, ผิวขาว, คนหนุ่มและคนสูงวัย ตามตารางที่ 2 เพื่อให้เกิดความหลากหลายของตัวอย่างภาพ ภาพเคลื่อนไหว (Video) มีอัตราความเร็วที่ 20 เฟรมต่อวินาที โดยแต่ละคำจะมีทั้งหมด 100 ไฟล์ และในแต่ละไฟล์จะมีจำนวนภาพระหว่าง 35 ถึง 50 ภาพ ขึ้นอยู่กับระยะเวลาในการทำแต่ละท่าทาง

ตารางที่ 1 คำที่ใช้ในการทดลอง (ภาษามือธรรมชาติ)

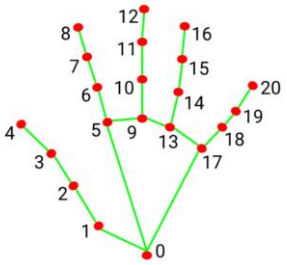
ลำดับ	ท่า	ลำดับ	ท่า
1	โชคดี	11	ชอบ
2	หิว	12	โมเดอร์นา
3	เสียใจ	13	เก้าอี้
4	ขอบคุณ	14	เลขสิบ
5	เมื่อวาน	15	ยัง
6	บ่าย	16	หูตึง
7	ทุกวัน	17	หูดี
8	หูหนวก	18	จมูก
9	กลางคืน	19	แมว
10	เย็นลง	20	อย่าการ์ดตก

สำหรับท่าทางภาษามือแบบสะกดนิ้วมือชนิดตัวเลขและภาษามือแบบธรรมชาติอ้างอิงมาจากสมาคมคนหูหนวกแห่งประเทศไทย

ตารางที่ 2 ลักษณะเด่นของบุคคลทดลองและสภาพแวดล้อม

บุคคล	ลักษณะเด่น	สภาพแวดล้อม
1	ผอม ผิวเหลือง	แสงธรรมชาติ มืด
2	สมส่วน ผิวเหลือง	แสงธรรมชาติ สว่าง
3	อ้วน ผิวคล้ำ	หลอดแอลอีดี 3000K 9 วัตต์
4	ผอม ผิวเหลืองขาว	หลอดแอลอีดี 3000K 9 วัตต์
5	อ้วน ผิวทองแดง	หลอดแอลอีดี 3000K 9 วัตต์
6	ผอม ผิวเหลือง	หลอดแอลอีดี 3000K 9 วัตต์
7	อ้วน ผิวคล้ำ	หลอดแอลอีดี 6500K 40 วัตต์
8	ท้วม ผิวเหลือง	หลอดแอลอีดี 6500K 40 วัตต์
9	ผอม ผิวเหลืองขาว	หลอดแอลอีดี 6500K 40 วัตต์
10	อ้วน ผิวทองแดง	หลอดแอลอีดี 6500K 40 วัตต์

มีเดียไพพ์ [6-7] เป็นแพลตฟอร์มสำหรับการทำ การเรียนรู้ของเครื่อง (Machine Learning) โดยการใช้เทคนิคการกำหนดจุด (Landmark) เพื่อหาค่าฟีเจอร์ที่สำคัญของภาษามือ (Feature Extracture) จากรูปที่ 1 MediaPipe จะแสดงตำแหน่งของจุดพิกัดแกน x, y และ z บนมือทั้งสอง โดยมีตำแหน่งทั้งหมด 21 จุดต่อมือ 1 ข้าง ดังนั้นจะได้จุดทั้งหมด 42 ตำแหน่ง (มือซ้าย-ขวา) ดังรูปที่ 2 และตารางที่ 3



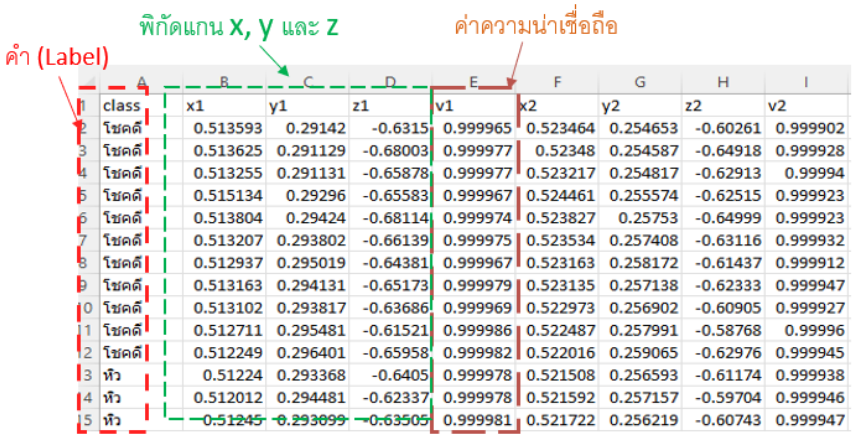
รูปที่ 2 Hand Landmark Model

ตารางที่ 3 ชื่อตำแหน่ง (Landmark Names)

0. Wrist	11. Middle Finger DIP
1. Thumb CMC	12. Middle Finger TIP
2. Thumb MCP	13. Ring Finger MCP
3. Thumb IP	14. Ring Finger PIP
4. Thumb TIP	15. Ring Finger DIP
5. Index Finger MCP	16. Ring Finger TIP
6. Index Finger PIP	17. Pinky MCP
7. Index Finger DIP	18. Pinky PIP
8. Index Finger TIP	19. Pinky DIP
9. Middle finger MCP	20. Pinky TIP
10. Middle Finger PIP	

หลังจากได้ตำแหน่งทั้งหมด 42 จุดก็จะถูกส่งต่อและเก็บไว้ที่ CSV ไฟล์

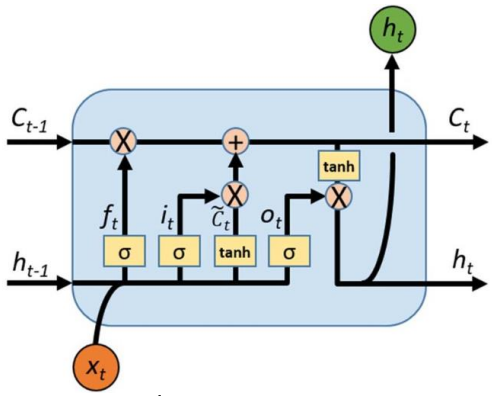
ในส่วนของ CSV ไฟล์มีหน้าที่เก็บค่าตำแหน่งของ 42 จุด ซึ่งในแต่ละจุดประกอบไปด้วยพิกัดแกน x, y, z และค่าความน่าเชื่อถือ (Visibility) เพื่อเป็นค่าอินพุตให้กับ LSTM อัลกอริทึมต่อไป



	class	x1	y1	z1	v1	x2	y2	z2	v2
2	โชคดี	0.513593	0.29142	-0.6315	0.999965	0.523464	0.254653	-0.60261	0.999902
3	โชคดี	0.513625	0.291129	-0.68003	0.999977	0.52348	0.254587	-0.64918	0.999928
4	โชคดี	0.513255	0.291131	-0.65878	0.999977	0.523217	0.254817	-0.62913	0.99994
5	โชคดี	0.515134	0.29296	-0.65583	0.999967	0.524461	0.255574	-0.62515	0.999923
6	โชคดี	0.513804	0.29424	-0.68114	0.999974	0.523827	0.25753	-0.64999	0.999923
7	โชคดี	0.513207	0.293802	-0.66139	0.999975	0.523534	0.257408	-0.63116	0.999932
8	โชคดี	0.512937	0.295019	-0.64381	0.999967	0.523163	0.258172	-0.61437	0.999912
9	โชคดี	0.513163	0.294131	-0.65173	0.999979	0.523135	0.257138	-0.62333	0.999947
10	โชคดี	0.513102	0.293817	-0.63686	0.999969	0.522973	0.256902	-0.60905	0.999927
11	โชคดี	0.512711	0.295481	-0.61521	0.999986	0.522487	0.257991	-0.58768	0.99996
12	โชคดี	0.512249	0.296401	-0.65958	0.999982	0.522016	0.259065	-0.62976	0.999945
13	หิว	0.51224	0.293368	-0.6405	0.999978	0.521508	0.256593	-0.61174	0.999938
14	หิว	0.512012	0.294481	-0.62337	0.999978	0.521592	0.257157	-0.59704	0.999946
15	หิว	0.51245	0.293699	-0.63505	0.999981	0.521722	0.256219	-0.60743	0.999947

รูปที่ 3 ข้อมูลพิกัดใน CSV ไฟล์

LSTM Layer [8] เป็นอัลกอริทึมอยู่ในหมวดของการจัดการด้าน Time Sequence ซึ่งได้รับการพัฒนามาจาก Recurrent Neural Network (RNN) และ (Gated Recurrent Unit (GRU) ส่วนหลักๆ ของ LSTM อัลกอริทึม ประกอบด้วย Forget gate (f_t), Input gate (i_t), Cell update ($\tilde{C}_t$), Cell State (C_t) , Output gate (o_t) และ Output (h_t) โดยมีลักษณะภายในดังรูปที่ 4



รูปที่ 4 LSTM อัลกอริทึม

และมีสมการตาม (1) – (6)

$$f_t = \sigma(X_t U^f + h_{t-1} W^f) \quad (1)$$

$$i_t = \sigma\left(X_t U^i + h_{t-1} W^i\right) \quad (2)$$

$$o_t = \sigma(X_t U^o + h_{t-1} W^o) \quad (3)$$

$$\tilde{C}_t = tanh(X_t U^g + h_{t-1} W^g) \quad (4)$$

$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right) \quad (5)$$

$$h_t = tanh(C_t) * o_t \quad (6)$$

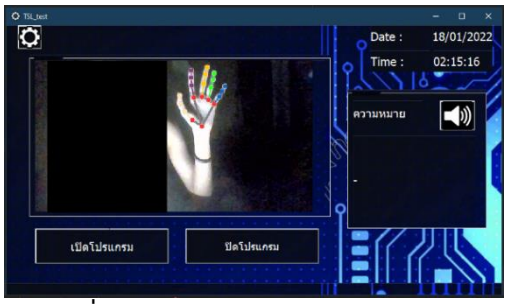
เมื่อ LSTM อัลกอริทึมได้รับค่าตำแหน่งจาก CSV ไฟล์เป็นอินพุทข้อมูล ข้อมูลดังกล่าวจะถูกแบ่งออกเป็น 2 ส่วน คือ Training data (80%) และ Test data (20%) โดยมีค่า Hyperparameter ของอัลกอริทึมดังนี้ LSTM Cell ในแต่ละชั้น คือ 256, 128, 64 และ 16, ค่า Batch size = 64, Optimizer = Adam และ Epochs = 100 โดย LSTM อัลกอริทึมได้ออกแบบบนแพลตฟอร์ม Tensorflow และ Keras

การทดลองเพื่อทำการทดสอบประสิทธิภาพของโปรแกรมแปลภาษามือเป็นข้อความแบ่งออกเป็นกรณีศึกษาดังต่อไปนี้

กรณีศึกษาที่ 1: การทดลองด้วยฐานข้อมูลของภาษามือแบบการสะกดนิ้วมือชนิดตัวเลข ตั้งแต่เลข 1 -9 ทั้งหมด 9 คำ โดยเปรียบเทียบผลกับงานวิจัย [3]

กรณีศึกษาที่ 2: การทดลองด้วยการนำฐานข้อมูลของภาษามือแบบธรรมชาติ จำนวน 20 คำ ตามข้อมูลในตารางที่ 1

กรณีศึกษาที่ 3: การทดลองด้วยการนำฐานข้อมูลของภาษามือธรรมชาติจำนวน 20 คำ มาทำการใช้งานโปรแกรมแปลภาษามือเป็นข้อความ ซึ่งโปรแกรมดังกล่าวได้พัฒนาด้วยภาษา Python เวอร์ชั่น 3.7.9บนโปรแกรม PyQt5 ซึ่งใช้ library ของ Google Text-to-Speech (gTTS) [9] โดยทำการกำหนดข้อมูลไฟล์เสียงด้วย library ในตัว gTTS เพื่อแปลงข้อความให้เป็นไฟล์เสียง พร้อมทำการบันทึกไฟล์เสียงไปยังตำแหน่งที่ต้องการบันทึกเพื่อนำไปใช้ในการเล่นไฟล์เสียงต่อไป โดยทั้ง 3 กรณีจะใช้บุคคลที่ทำการทดลองที่มีลักษณะทางกายภาพแตกต่างกัน 10 คน ซึ่งมีลักษณะดังที่กล่าวตามตารางที่ 2



รูปที่ 5 หน้าต่างอินเตอร์เฟสของโปรแกรม

ผลการวิจัย

กรณีศึกษาที่ 1: การทดลองด้วยฐานข้อมูลของภาษามือแบบสะกดนิ้วมือชนิดตัวเลข จำนวน 9 ท่า คือท่าเลข 1 ถึงเลข 9

จะเห็นได้ว่าความถูกต้องของท่าเลข 1 ถึงเลข 9 ของบุคคลทดลองทั้ง 10 คนอยู่ในช่วง 80% - 100% และเมื่อนำค่าความถูกต้องมาหาค่าเฉลี่ยในแต่ละท่าจะได้ดังกราฟในรูปที่ 6

ค่าเฉลี่ยความถูกต้องของทั้ง 9 ท่าจะเป็น 98% เมื่อเปรียบเทียบประสิทธิภาพกับงานวิจัยอื่น ๆ ถือว่ามีค่าความถูกต้องที่ดีมาก เหมาะที่จะนำไปทดลองท่าทางอื่น ๆ ต่อไป

รูปที่ 6 ค่าเฉลี่ยผลการทดลองกรณีศึกษาที่ 1

กรณีศึกษาที่ 2: การทดลองด้วยการนำฐานข้อมูลของภาษามือธรรมชาติ จำนวน 20 คำ ตามตารางที่ 1 ซึ่งมีการสร้างมาจากบุคคลทดลอง 10 คน ผลการทดลองดังแสดงในรูปที่ 7 โดยใช้หลักการของการเรียนรู้ของระบบ และหาค่า F1-score ด้วยสมการที่ (7) (8) และ (9) เพื่อใช้สำหรับการตรวจสอบความแม่นยำของชุดฐานข้อมูลที่สร้างขึ้นมา

$$\text{Precision} = \frac{tp}{tp+fp} \qquad (7)$$

$$Recall = \frac{tp}{tp+fn} \qquad (8)$$

$$F1 - score = 2 \times \frac{\mathrm{Pr} \times Re}{Pr + Re} \qquad (9)$$

เมื่อ

Precision หมายถึง ค่าความแม่นยำที่เกิดจากการนำค่า tp มาเทียบกับ fn

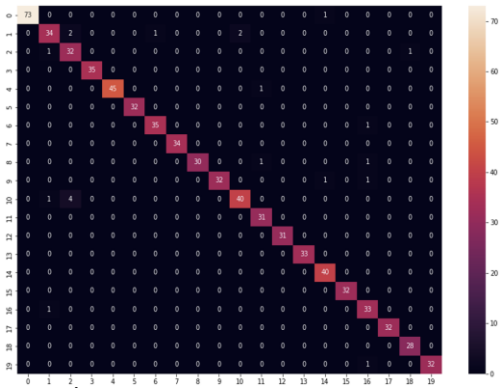
Recall หมายถึง ค่าความถูกต้องที่เกิดจากการนำค่า tp มาเทียบกับ fn

F1-Score หมายถึงค่าเฉลี่ยของค่าความแม่นยำและค่าความถูกต้อง

tp หมายถึงข้อมูลที่ทายแล้วถูกต้องเมื่อเทียบกับเฉลย

fp หมายถึงข้อมูลที่อยู่ในเฉลยแต่ไม่มีในการทำนาย (ตรงข้ามกับ fn)

fn หมายถึงข้อมูลที่ทำนายแล้วไม่ถูกต้องเมื่อเทียบกับเฉลย



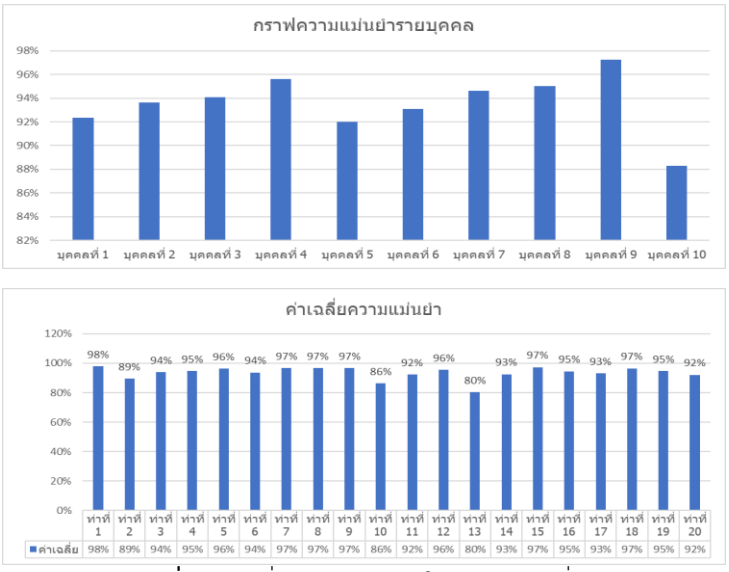
รูปที่ 7 Confusion Matrix ของของภาษามือธรรมชาติ จำนวน 20 คำ

Classification Report

	precision	recall	f1-score	support
0	1.00	0.99	0.99	74
1	0.92	0.87	0.89	39
2	0.84	0.94	0.89	34
3	1.00	1.00	1.00	35
4	1.00	0.98	0.99	46
5	1.00	1.00	1.00	32
6	0.97	0.97	0.97	36
7	1.00	1.00	1.00	34
8	1.00	0.94	0.97	32
9	1.00	0.94	0.97	34
10	0.95	0.89	0.92	45
11	0.94	1.00	0.97	31
12	1.00	1.00	1.00	31
13	1.00	1.00	1.00	33
14	0.95	1.00	0.98	40
15	1.00	1.00	1.00	32
16	0.89	0.97	0.93	34
17	1.00	1.00	1.00	32
18	0.97	1.00	0.98	28
19	1.00	0.97	0.98	33
accuracy			0.97	735
macro avg	0.97	0.97	0.97	735
weighted avg	0.97	0.97	0.97	735

รูปที่ 8 ผลการคำนวณกรณีศึกษาที่ 2

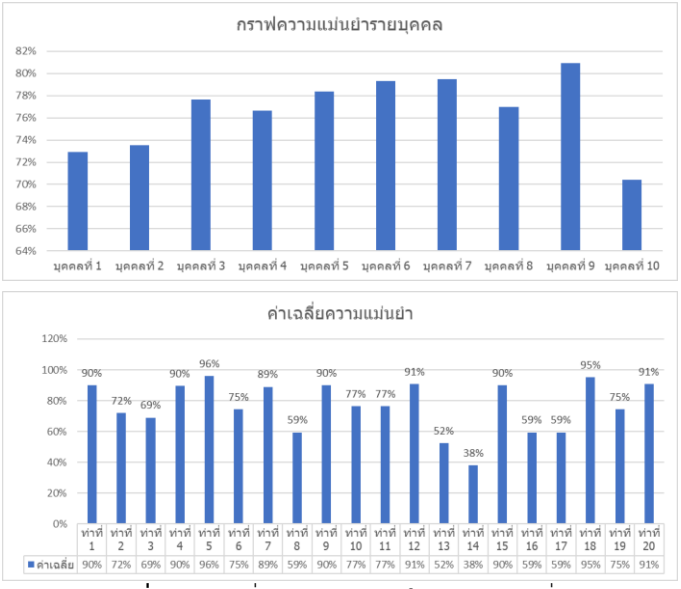
โดยการใช้โปรแกรมวิเคราะห์ตามสมการที่ (7) (8) และ (9) จะได้ค่าความถูกต้องของท่าที่ 1-20 ของชุดข้อมูลอยู่ในช่วง 89% - 100% โดยข้อมูลที่โปรแกรมจัดสร้างแบ่งขึ้นมา 735 ชุดที่เป็นค่า F1-score ซึ่งจากค่าที่บ่งบอกมานั้นถือว่าความผิดเพี้ยนน้อยมาก ค่าความแม่นยำเฉลี่ยที่ 97.14% และมีค่าความสูญเสียที่ 11.3% แต่เมื่อนำข้อมูลที่ทำการจัดเก็บมาแยกโดยแบ่งตามรายบุคคล จะมีค่าอยู่ในช่วง 67% - 100% ดังแสดงในรูปที่ 8 ซึ่งใช้อุปกรณ์การเก็บข้อมูลที่ไม่ใช่ชุดเดียวกับที่ทำฐานข้อมูล และมีสภาพแวดล้อมที่เปลี่ยนไปแต่เมื่อนำมาเฉลี่ยในรายบุคคลจะได้กราฟดังแสดงในรูปที่ 9



รูปที่ 9 ค่าเฉลี่ยผลการทดลองในกรณีศึกษาที่ 2

ค่าเฉลี่ยทั้งหมดของกรณีที่ 2 มีค่า 94% เมื่อนำมาเทียบกับค่าความถูกต้องที่ 97% ที่ได้จากการวัดผลในขั้นตอนการสร้างฐานข้อมูลซึ่งนับว่าใกล้เคียงกัน แต่เมื่อเปรียบเทียบเป็นรายบุคคลจะเห็นได้ว่าบุคคลที่มีผิวสีเข้มหรือในสภาพแวดล้อมที่มืดมีผลต่อการเก็บข้อมูลของกล้องทำให้ค่าความถูกต้องลดลง

กรณีศึกษาที่ 3: การทดลองด้วยการนำฐานข้อมูลของภาษามือจำนวน 20 ท่าทางมาทำการใช้งานจริงโปรแกรมแปลภาษามือเป็นข้อความ



รูปที่ 10 ค่าเฉลี่ยผลการทดลองในกรณีศึกษาที่ 3

จากรูปที่ 10 ค่าเฉลี่ยของทั้ง 20 คำจะได้ 77.7% เมื่อนำมาเทียบกับค่าความถูกต้องที่ 97% ที่ได้จากการวัดผลถือว่าห่างกันค่อนข้างมาก ทั้งนี้เกิดจากการใช้ทรัพยากรในการคิดคำนวณค่าที่ต่างกัน และการแสดงท่าที่มีการเคลื่อนไหวที่ค่อนข้างเร็วจะมีค่าความถูกต้องที่ต่ำเช่น คำที่ 14 ส่วนคำที่ 8, 16 และ 17 เกิดจากมีลักษณะท่าที่มีการกระทำที่ใกล้เคียงกันทำให้การทำนายของโปรแกรมเกิดการผิดพลาดได้ง่าย สำหรับคำที่13 เกิดจากการที่โปรแกรมเกิดการตรวจจับผิดพลาดในส่วนของนิ้วมือที่วางซ้อนทับกัน

สรุปและอภิปรายผล

โปรแกรมแปลภาษามือเป็นข้อความที่ออกแบบสามารถทำการสร้างฐานข้อมูลภาษามือทั้งการสะกดนิ้วมือ และแบบภาษามือธรรมชาติ นำมาใช้งานได้โดยประสิทธิภาพอยู่ในเกณฑ์ระดับดี

ในกรณีศึกษาที่ 1 มีความแม่นยำสูงเพราะเป็นท่าที่ไม่มีการเปลี่ยนแปลงของท่าทาง (Single Posture) ขณะที่กรณีศึกษาที่ 2 เป็นท่าทางที่ซับซ้อน (Multiple Posture) แต่เป็นการวัดผลด้วยโปรแกรมจูปิเตอร์แลปและทำการเทียบหาค่าโดยการดูค่าความใกล้เคียงจากค่าชุดข้อมูลที่ทางบุคคลทดลองได้ทำการบันทึกขึ้นมาใหม่ความแม่นยำก็ยังค่อนข้างสูง ส่วนกรณีศึกษาที่ 3 เป็นการทดลองใช้งานจริงบนโปรแกรมแปลภาษามือเป็นข้อความ ซึ่งใช้ชุดคำสั่งในการเก็บค่ามาคำนวณต่างกันกับการทดลองในกรณีศึกษาที่ 2 จะเห็นได้ว่าค่าความถูกต้องของการแปลมีค่าต่ำกว่าอยู่ถึง 16.3% และมีค่าเบี่ยงเบนของข้อมูลระหว่างบุคคลทดลองที่สูง ทั้งนี้เกิดจากโปรแกรมแปลภาษามือเป็นข้อความมีระบบการตรวจจับมือได้น้อยกว่าการนำชุดข้อมูลไปเปรียบเทียบในโปรแกรมจูปิเตอร์แลปที่ระหว่างการเก็บค่าข้อมูล

ปัจจัยที่สำคัญอย่างหนึ่งต่อคุณภาพของการแปลภาษามือคือความถูกต้องของฐานข้อมูล ซึ่งสามารถเพิ่มความถูกต้องโดยการใช้ผู้ที่เป็นผู้เชี่ยวชาญภาษามือมาเป็นแบบในการจัดเก็บข้อมูล ซึ่งงานวิจัยนี้ใช้บุคคลปกติมาเป็นบุคคลทดลอง นอกจากนี้ยังมีปัจจัยที่มาจากเทคโนโลยี เช่น คุณภาพของกล้องรับภาพ ตลอดจนในความเป็นจริงของภาษามือนั้นจะประกอบไปด้วยท่าทางของอวัยวะส่วนอื่นของร่างกายและสีหน้าของผู้ใช้ภาษามือจึงจะสามารถแปลค่าได้ถูกต้องมากยิ่งขึ้นได้

References

[1] กมลา ไกรฤกษ์, *“หนังสือคู่มือภาษามือ”* พิมพ์ครั้งที่ 1, กรุงเทพฯ: สำนักพิมพ์คุรุสภาลาดพร้าว, 2544.

[2] A. Chaikaew, K. Somkuan and P. Sarapee, “Mobile Application for Thai Sign Language”, in 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, pp. 1-4.

[3] P. Nakjai and T. Katanyukul, “Hand sign recognition for thai finger-spelling: An application of convolution neural network”, Journal of Signal Processing Systems (2019), pp. 131-146.

[4] A. Chaikaew, K. Sornkuan, and T. Yuyen, "Thai Sign Language Recognition: an Application of Deep Neural Network" in International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-CON), Cha-am, Thailand, 2021, pp. 128-131.

[5] A. Chaikeaw, "An applied holistic landmark with deep learning for Thai sign language recognition" in International Technical Conference on Circuits/Systems, Computers and Communication (ITC-CSCC), Phuket, Thailand, pp. 1046-1049.

[6] F. Zhang, V. Bazaresky, A. Vakunov et al., "MediaPipe Hand: On-device Real-time Hand Tracking", in CVPR Workshop on Computer Vision for Augmented and Virtual Reality (CVPR), Seattle, USA, 2020.

[7] G. Anikumar, Fouzia M S, Anisha GS, "Imperative Methodology to Detech the Palm Gestures (American Sign Language) using YOLOv5 and MediaPipe" in 2nd International Conference on Intelligent Technology (CONIT), Hubli, India, 2022, pp. 1-4.

[8] J. Brownlee, "*Long Short-Term Memory Networks with Python*", Edition 1.0, Published eBook 2017.

[9] R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python", 2nd International Conference on Intelligent Computing, Instrumentation and Control Technology (ICICICT), Kannur, India, 2019, pp. 486-489.