ระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้เชิงลึกบนข้อมูลบางส่วน

ณัฏยา เปลี่ยนวงษ์

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่
วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
ปีการศึกษา 2564

THAI SIGN LANGUAGE TRANSLATION SYSTEM USING FEW SHOT LEARNING

NATTAYA PLIANWONG

An Independent Study Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Big Data Engineering,

College of Innovative Technology and Engineering,

Dhurakij Pundit University

Academic Year 2021

ใบรับรองงานสารนิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อสารนิพนธ์ ระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้เชิงลึกบนข้อมูลบางส่วน

เสนอโดย ณัฏยา เปลี่ยนวงษ์

สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่

อาจารย์ที่ปรึกษาสารนิพนธ์ ดร.ธนภัทร ฆังคะจิตร

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบสารนิพนธ์แล้ว

.....ประธานกรรมการ

(ดร.สรรพฤทธิ์ มฤคทัต)

.....กรรมการและอาจารย์ที่ปรึกษา

(ดร.ธนภัทร ฆังคะจิตร)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ดวงใจ จิตคงชื่น)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

.....

(ดร.ชัยพร เขมะภาตะพันธ์)

คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

วันที่ 7 เดือน ธันวาคม พ.ศ. 2564

หัวข้อสารนิพนธ์ ระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้เชิงลึกบนข้อมูลบางส่วน
ชื่อผู้เขียน ณัฏยา เปลี่ยนวงษ์
อาจารย์ที่ปรึกษา ดร.ธนภัทร ฆังคะจิตร
สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่
ปีการศึกษา 2564

บทคัดย่อ

ผู้พิการทางการได้ยินมีมากเป็นอันดับสองจากจำนวนผู้พิการทั้งหมดในประเทศไทย อุปสรรคหลักในการสื่อสารผ่านทางภาษามือกับผู้พิการทางการได้ยินเกิดจากคำศัพท์บางคำมีท่าทางเฉพาะที่ยากต่อการคาดเดา อย่างไรก็ตามมีงานวิจัยที่นำเสนอการรู้จำคำศัพท์ภาษามือไทยจากวิดีโอตัวอย่างโดยใช้เทคนิคการเรียนรู้เชิงลึก LSTM ซึ่งให้ความแม่นยำสูง แต่ใช้กับคำศัพท์ภาษามือจำนวนน้อยเพียง 5 คำและใช้วิดีโอจำนวนมากถึง 100 วิดีโอในการสอนแบบจำลอง ทำให้เป็นการยากในการสร้างแบบจำลองเพื่อเรียนรู้คำศัพท์ที่มีจำนวนมากขึ้น เนื่องจากต้องใช้ผู้เชี่ยวชาญในการบันทึกวิดีโอตัวอย่าง ดังนั้นงานนี้จึงนำเสนอแนวทางในการสร้างระบบแปลคำศัพท์ภาษามือไทยในชีวิตประจำวัน 47 คำ โดยใช้เทคนิคการเรียนรู้เชิงลึกจากตัวอย่างจำนวนน้อย (Few Shot Learning) สำหรับวิดีโอสอนระบบมาจากการสร้างขึ้นของผู้วิจัยโดยอ้างอิงจากวิดีโอของผู้เชี่ยวชาญ แล้วทำการสกัดคุณลักษณะในแต่ละเฟรมของวิดีโอสอนระบบจากท่าทางจากตำแหน่งของมือและใบหน้า รวมถึงการทำมุมของแขนท่อนบนกับท่อนปลาย เพื่อนำมาให้แบบจำลองเรียนรู้รูปแบบ จากนั้นจึงนำแบบจำลองมาทดสอบด้วยวิดีโอของผู้เชี่ยวชาญซึ่งพบว่าแบบจำลองให้ความแม่นยำสูงถึง 74% โดยมีคำศัพท์ที่ทำนายผิดเกิดจากองค์ประกอบของท่าทางที่คล้ายกันหรือมีท่าทางซ้ำกันในตำแหน่งเดิมหลายครั้ง ดังนั้นในการเพิ่มประสิทธิภาพของแบบจำลองในอนาคตสามารถทำโดยใช้วิดีโอจำนวนน้อยจากผู้เชี่ยวชาญมาเพื่อสอนระบบแล้วจึงนำแบบจำลองไปใช้ในระบบจริง

Independent Study Title	THAI SIGN LANGUAGE TRANSLATION SYSTEM USING FEW SHOT LEARNING
Author	Nattaya Plianwong
Independent Study Advisor	Dr. Thanapat Kangkachit
Department	Big Data Engineering
Academic Year	2564

ABSTRACT

In Thailand, the amount of deaf is the second-largest in total numbers of disabled people. The main obstacle to communicating with the deaf is that some words in sign language have specific gestures that are difficult to predict. The former research in Thai sign language recognition provided high accuracy on a few sign-language words using the deep learning technique. Contrastingly, a hundred sample videos per word were input to the LSTM classifier. An extensive effort from experts to produce training videos is required to build a model to learn a larger number of words. Therefore, this work presents a system to translate everyday-life Thai sign words using the few-shot learning technique. Firstly, the few training videos per word are constructed based on the experts' videos. Then, input features on each frame of the videos are extracted from hand and face positions and the angle of the upper and lower arms. Once the model is constructed, experts' videos are used as testing datasets. The experimental results show that our model produces high accuracy (74%) on experts' videos. The misclassified words seem to have similar postures or repeated gestures in the same position. Furthermore, improving the model's performance requires a small number of accurate training videos from experts.

กิตติกรรมประกาศ

สารนิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ โดยการให้ความช่วยเหลือของ ดร.ธนภัทร ฆังคะจิตร ซึ่งเป็นอาจารย์ที่ปรึกษาสารนิพนธ์ ที่ได้กรุณาให้คำแนะนำ ตรวจสอบ และแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด เพื่อให้สารนิพนธ์ฉบับนี้สมบูรณ์ ผู้เขียนจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

ผู้เขียนขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สรรพฤทธิ์ มฤคทัต ที่กรุณาให้เกียรติเป็นประธานโดยมี ผศ.ดร.ดวงใจ จิตคงชื่น เป็นกรรมการในการสอบสารนิพนธ์ ซึ่งได้กรุณาตรวจ แก้ไขสารนิพนธ์ฉบับ นี้ให้ถูกต้องสมบูรณ์ยิ่งขึ้น และ นางสาวกุลธิดา รอดบุญ รวมถึงเจ้าหน้าที่บัณฑิตมหาวิทยาลัยธุรกิจบัณฑิตย์ ทุกท่านที่ให้ความสะดวกด้านอำนวยการ และประสานงาน ในการทำสารนิพนธ์ให้กับผู้เขียน ทำให้การจัดทำสารนิพนธ์ของผู้เขียนในครั้งนี้สำเร็จลุล่วงไปด้วยดี

ณัฏยา เปลี่ยนวงษ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ฆ

บทคัดย่อภาษาอังกฤษ..... ง

กิตติกรรมประกาศ..... จ

สารบัญตาราง..... ซ

สารบัญภาพ..... ฌ

บทที่

1. บทนำ..... 1
   - 1.1 ที่มาและความสำคัญของปัญหา..... 1
   - 1.2 วัตถุประสงค์ของงานวิจัย..... 2
   - 1.3 ขอบเขตงานวิจัย..... 2
   - 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... 2
   - 1.5 นิยามคำศัพท์..... 2
2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง..... 3
   - 2.1 Meta-Learning..... 3
   - 2.2 Mediapipe..... 5
   - 2.3 OpenCV (Open source Computer Vision)..... 7
   - 2.4 ตัววัดประสิทธิภาพของโมเดล (Confusion Matrix)..... 8
   - 2.5 งานวิจัยที่เกี่ยวข้อง..... 9
3. ระเบียบวิธีวิจัย..... 19
   - 3.1 การเก็บข้อมูล (Input Video)..... 19
   - 3.2 การเตรียมข้อมูลสำหรับสอนระบบ (Preprocess)..... 19
   - 3.3 การสร้างโมเดล (Modeling)..... 23
   - 3.4 การนำโมเดลไปใช้กับระบบ (Implement)..... 25
   - 3.5 เครื่องมือที่ใช้ในงานวิจัย..... 25
4. ผลการศึกษา..... 27
   - 4.1 ผลการเปรียบเทียบประสิทธิภาพของโมเดล Few-Shot Learning..... 27
   - 4.2 การเลือกพารามิเตอร์ และฟีทเจอร์..... 28

สารบัญ (ต่อ)

หน้า

บทที่

4.3 ผลการวัดประสิทธิภาพความถูกต้องของโมเดล..... 32

4.4 ผลจากการนำไปใช้งานกับวิดีโอผู้เชี่ยวชาญ..... 33

5. บทสรุปและข้อเสนอแนะ..... 35

5.1 สรุปผลการศึกษา..... 36

5.2 ข้อสังเกต..... 35

5.3 ข้อเสนอแนะ..... 36

บรรณานุกรม..... 38

ภาคผนวก..... 41

ก รายการคำศัพท์ภาษามือไทย..... 42

ประวัติผู้เขียน..... 44

สารบัญตาราง

ตารางที่		หน้า
2.1	ผลการเปรียบเทียบประสิทธิภาพของโมเดล และAggregation Techniques จากการทดสอบด้วยข้อมูลปกติ และการใช้ข้อมูลที่มีความท้าทาย.....	10
2.2	ผลการเปรียบเทียบประสิทธิภาพของโมเดล.....	11
3.1	ตัวอย่างลำดับการ Concatenate Array ของตำแหน่งด้านขวา.....	23
3.2	การกำหนดตัวแปรของโมเดล.....	24
4.1	ผลการเปรียบเทียบประสิทธิภาพของโมเดล.....	28
4.2	ผลการเปรียบเทียบการใช้ Backbone ต่าง ๆ.....	28
4.3	ตัวอย่างตำแหน่งมือด้านขวา.....	30
4.4	ผลการเปรียบเทียบการใช้ตำแหน่งของร่างกาย (Mediapipe).....	30
4.5	ผลการเปรียบเทียบจำนวนในการ Capture Frame วิดีโอ.....	31
4.6	ผลการเปรียบเทียบจำนวน Task ในการสอน.....	32
4.7	การกำหนดพารามิเตอร์ของโมเดล Prototypical Networks.....	32
5.1	เปรียบเทียบผลการแปลคำศัพท์ที่แปลไม่ถูกต้องจากวิดีโออาสาและผู้เชี่ยวชาญ.....	36

สารบัญภาพ

ภาพที่ **หน้า**

2.1 การจัดการข้อมูลของ Meta-dataset…….....…… 5
2.2 Key points ทั้ง 21 บนมือ……..... 6
2.3 Key points ทั้ง 31 จุด บนร่างกาย……..... 6
2.4 แสดงความสัมพันธ์ไลบรารีของ OpenCV……..... 7
2.5 ตัวอย่างเมตริกซ์การวัดประสิทธิภาพสำหรับการจำแนกประเภทข้อมูล 2 กลุ่ม. 8
2.6 Prototypical networks ใน Few-shot และZero-shot……..... 12
2.7 ผลการเปรียบเทียบประสิทธิภาพของโมเดลแบบ Few-shot กับชุดข้อมูล Omniglot…….. 13
2.8 ผลการเปรียบเทียบประสิทธิภาพของโมเดลแบบ Few-shot กับชุดข้อมูล ImageNet……... 13
2.9 Relation Network architecture สำหรับ 5-way 1-shot กับ 1 query example..... 14
2.10 Relation Network architecture สำหรับ few-shot learning (b) และ ส่วนประกอบของแต่ละ convolutional block(a)..... 14
2.11 Relation Network architecture สำหรับ zero-shot learning..... 15
2.12 ผลจากชุดข้อมูล Omniglot สำหรับ few-shot classification..... 15
2.13 ผลจากชุดข้อมูล Imagenet สำหรับ few-shot classification..... 16
2.14 Matching Networks architecture..... 17
2.15 ผลการเปรียบเทียบประสิทธิภาพของโมเดลกับชุดข้อมูล Omniglot..... 18
2.16 ผลการเปรียบเทียบประสิทธิภาพของโมเดลกับชุดข้อมูล ImageNet..... 18
2.17 ตัวอย่างข้อมูล input และsupport set ใน NLP..... 18
3.1 การทำงานของระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้แบบ Few-Shot Learning..... 19
3.2 ตัวอย่างการ Capture Frame..... 20
3.3 ตัวอย่างการ Landmark เฟรมด้วย Mediapipe..... 21
3.4 ตัวอย่างตำแหน่งอ้างอิงของจมูก และตำแหน่ง Pose ของ Mediapipe..... 21
3.5 ตัวอย่างการทำมุมระหว่างแขนท่อนบนกับแขนท่อนปลายของแขนขวา..... 22
3.6 ตัวอย่างระยะห่างระหว่างนิ้วชี้กับจมูก..... 22

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.7 ตัวอย่างโปรแกรมรูปแบบ GUI ในการนำไปใช้จริง..... 25
4.1 ตัวอย่างระยะห่างระหว่างปลายนิ้วโป้งกับปลายนิ้วชี้..... 29
4.2 ตัวอย่างระยะห่างระหว่างปลายนิ้วโป้งกับข้อมือ..... 29
4.3 Confusion Matrix ทดสอบกับวิดีโออาสา..... 33
4.4 ตัวอย่างการทดสอบกับวิดีโอโดยสมาคมคนหูหนวก..... 33
4.5 Confusion Matrix ทดสอบกับวิดีโอโดยสมาคมคนหูหนวก..... 34

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

สถานการณ์ด้านคนพิการในประเทศไทย ปี 2564 จำนวน 2,096,931 คน (ร้อยละ 3.17 ของประชากรทั้งประเทศ) พบว่าเป็นความพิการประเภท ทางการได้ยินหรือสื่อความหมาย 393,826 (ร้อยละ 18.78 ของจำนวนผู้พิการทั้งประเทศ) จัด เป็นอันดับ 2 ของประเภทความพิการทั้งหมด รองจากความพิการทางด้านการเคลื่อนไหว จะเห็นได้ว่ามีผู้พิการจำนวนมากที่ไม่สามารถสื่อสารกับบุคคลทั่วไปได้อย่างปกติ หากบุคคลเหล่านั้นไม่มีความรู้ทางด้านภาษามือ ซึ่งในบางครั้งคำศัพท์ภาษามือสามารถคาดเดาหรือสื่อความหมายได้ แต่ก็มีคำศัพท์จำนวนมากที่ค่อนข้างซับซ้อน จึงต้องอาศัยล่ามภาษามือในการสื่อสารระหว่างคนทั่วไปกับผู้พิการ แต่เนื่องจากทรัพยากรของล่ามภาษามือมีจำนวนไม่เพียงพอและยากต่อการรับบริการ อีกทั้งยังมีปัญหาอุปสรรคในการใช้ล่ามภาษามือ เช่นประเด็นความน่าเชื่อถือ และไว้วางใจในล่ามภาษามือ ในบางครั้งผู้พิการอยากสื่อสารในเรื่องที่ค่อนข้างละเอียดอ่อน หรือเป็นเรื่องสำคัญ ทำให้ไม่ไว้วางใจหากไม่ใช่ ญาติ หรือคนสนิท ดังนั้นหากมีการพัฒนาอุปกรณ์ หรือ ช่องทางในการสื่อสารระหว่างผู้พิการและบุคคลทั่วไปได้ จะช่วยลดช่องว่างของการสื่อสาร และแก้ไขปัญหาเบื้องต้นนี้ลงได้

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบแปลคำศัพท์ภาษามือไทยในชีวิตประจำวัน โดยวิธีการที่สนใจได้แก่การเรียนรู้เชิงลึก เทคนิค Few-Shot Learning เนื่องจากปัจจุบันฐานข้อมูลภาษามือไทยยังไม่มีข้อมูลให้ใช้ได้อย่างแพร่หลาย ส่วนใหญ่จะเป็นภาพวาด หรือรูปถ่าย ที่ใช้ในการสะกดคำ ซึ่งค่อนข้างยากหากจะนำมาเรียนรู้ หลังจากสืบค้นข้อมูลพบว่า สมาคมคนหูหนวกแห่งประเทศไทยได้จัดทำระบบฐานข้อมูลภาษามือไทย รวบรวมคำศัพท์ในหนังสือภาษามือไทย เล่ม 1-6 ของสมาคมคนหูหนวกแห่งประเทศไทย (ไม่น้อยกว่า 1,000 คำ) ในแต่ละคำจะมีตัวอย่างท่าทางเป็นวิดีโอ 6 ตัวอย่าง มีความยาวไม่เกิน 5 วินาที

เนื่องจากข้อมูลที่มีอยู่อย่างจำกัด หากใช้วิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Deep Learning) อาจไม่เหมาะสมเพราะวิธีการ

ดังกล่าวต้องใช้ตัวอย่างข้อมูลเป็นจำนวนมาก นำไปเรียนรู้ซ้ำ ๆ เพื่อหารูปแบบของคำตอบ อีกทั้งยังมีวิธีการซับซ้อน และต้องใช้ทรัพยากรในการประมวลผลมาก จึงไม่เหมาะสมกับงานวิจัยในครั้งนี้ ดังนั้นผู้วิจัยจึงเลือกวิธีการเรียนรู้แบบไม่กี่ตัวอย่าง (Few-Shot Learning) ในการสร้างระบบแปลคำศัพท์ภาษามือไทยในชีวิตประจำวัน โดยทำการสกัดคุณลักษณะในแต่ละเฟรมของวิดีโอสอนระบบจากท่าทางจากตำแหน่งของมือและใบหน้า รวมถึงการทำมุมของแขนท่อนบนกับท่อนปลาย เพื่อนำมาให้แบบจำลองเรียนรู้รูปแบบ

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อแปลคำศัพท์ภาษามือไทยในชีวิตประจำวันได้เบื้องต้น
2. เพื่อสร้างโมเดลโดยใช้ตัวอย่างข้อมูลจำนวนน้อย

1.3 ขอบเขตงานวิจัย

1. สร้างโมเดลที่เหมาะสมเพื่อแปลคำศัพท์ภาษามือไทยโดยแบ่งออกเป็น 3 หมวด ได้แก่ หมวดความรู้สึกจำนวน 21 คำ, หมวดรสชาติจำนวน 9 คำ และหมวดคำกริยาจำนวน 17 คำ รวมทั้งหมด 47 คำ
2. แต่ละคำมีตัวอย่างท่าทางเป็น วิดีโอ คำละ 6 ตัวอย่าง
3. คำศัพท์จากฐานข้อมูลภาษามือไทย โดยสมาคมคนหูหนวกแห่งประเทศไทย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยลดช่องว่างในการสื่อสารระหว่างผู้พิการ กับบุคคลทั่วไป และคู่สื่อสารไม่จำเป็นต้องมีความรู้ภาษามือ
2. ลดปัญหาความน่าเชื่อถือ และความไว้วางใจในล่าม
3. สามารถเข้าถึงได้ง่าย ลดปัญหาล่ามไม่เพียงพอ

1.5 นิยามคำศัพท์

1. วิธีการเรียนรู้แบบไม่กี่ตัวอย่าง หรือ Few-Shot Learning เป็นวิธีการเรียนรู้ที่ไม่จำเป็นต้องใช้ข้อมูลตัวอย่างเป็นจำนวนมาก และเลียนแบบวิธีการเรียนรู้ของมนุษย์ เพื่อแยกแยะความแตกต่างของสิ่งของได้

บทที่ 2
แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

งานวิจัยเรื่องนี้มีวัตถุประสงค์เพื่อแปลคำศัพท์ภาษามือไทยในชีวิตประจำวัน โดยสร้างโมเดลจากวิดีโอคำศัพท์ ด้วยการใช้เทคนิคการเรียนรู้แบบไม่กี่ตัวอย่าง (Few-Shot Learning) ดังนั้นจึงควรศึกษา ทำความเข้าใจ เกี่ยวกับเอกสาร และงานวิจัยที่เกี่ยวข้องดังรายการต่อไปนี้

2.1 Meta-Learning

2.2 Mediapipe

2.3 OpenCV (Open source Computer Vision)

2.4 ตัววัดประสิทธิภาพของโมเดล (Confusion Matrix)

2.5 งานวิจัยที่เกี่ยวข้อง

2.1 Meta-Learning

Meta-Learning หรือLearning to learn หรือการเรียนรู้เพื่อที่จะเรียนรู้ เป็นรูปแบบการเรียนรู้ที่นิยมในช่วงหลายปีที่ผ่านมา โดยการเรียนรู้แบบนี้ เป็นวิธีการเรียนรู้ที่เลียนแบบการเรียนรู้ของมนุษย์ ที่สามารถเรียนรู้สิ่งใหม่ ๆ ได้จากตัวอย่างไม่กี่ตัวอย่างในขณะที่การเรียนรู้เชิงลึกในตอนนี้ยังเป็นเรื่องที่ต้องใช้ข้อมูล การที่จะทำให้โมเดลมีประสิทธิภาพที่ดี ต้องใช้ตัวอย่างการฝึกอบรมหลายล้านหรือหลายพันล้านแบบ ซึ่งเป็นวิธีที่คลาสสิกในการบรรลุเป้าหมาย การเพิ่มข้อมูลเป็นวิธีการหนึ่งในการสร้างตัวอย่างสังเคราะห์ ยิ่งไปกว่านั้นโครงข่ายใยประสาทมาตรฐานไม่สามารถเรียนรู้ความรู้ใหม่ได้ทันที

Meta-Learning เกิดมาเพื่อจัดการกับปัญหานี้ จากตัวอย่างบางส่วน และสามารถเรียนรู้และปรับตัวเข้ากับโดเมนใหม่ได้อย่างรวดเร็ว Meta-Learning มุ่งเน้นไปที่การสร้างประสบการณ์ที่จะทำให้เกิดประโยชน์กับโมเดลภายในอนาคต อาจจะคล้ายกับมนุษย์ที่ ในแต่ละวันจะมีการสะสมความรู้ต่าง ๆ สิ่งที่มนุษย์ทำคือการปรับตัว แก้ปัญหา ให้เข้ากับปัญหาใหม่ในแต่ละวัน ซึ่งความสามารถเหล่านี้ไม่ได้พบเห็นในวิธีการเรียนรู้แบบเก่า

Meta-Learning สามารถถูกพิจารณาได้ 2 แนวทางหลักได้แก่

1. Mechanistic view มุ่งเน้นไปที่การพิจารณารูปแบบการทำงาน หรือการเรียนรู้ เช่น

1.1 Algorithm สำหรับอ่านชุดของข้อมูลเรียนรู้ (Support set) เพื่อทำนายบนข้อมูลใหม่ (Query set) ที่ไม่เคยเห็น ทำงานอย่างไร

1.2 Meta-dataset ประกอบด้วย set อะไรบ้าง และ Flow การไหลของข้อมูลเป็นอย่างไร

2. Probabilistic view มุมมองจากทฤษฎีของความน่าจะเป็น ยกตัวอย่าง เช่น

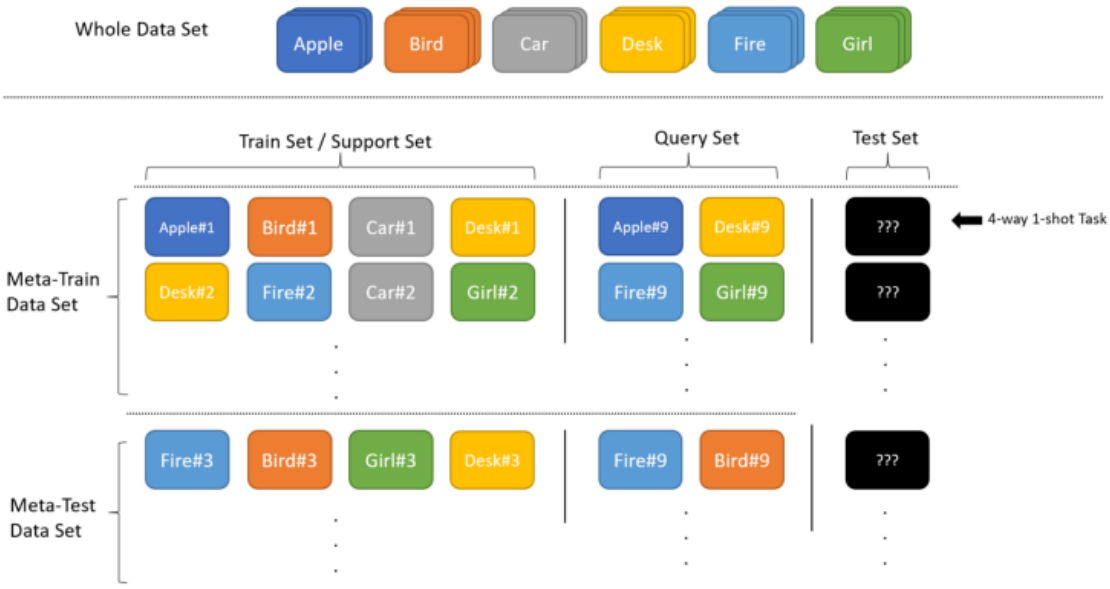
2.1 มองพารามิเตอร์ที่ได้จาก Meta-learning จากหลาย ๆ งาน (Tasks) เป็น Prior เพื่อที่จะทำให้เรียนรู้งานใหม่ ๆ ที่เข้ามาได้อย่างมีประสิทธิภาพ

2.2 Formulate ปัญหา Meta-learning ว่าเป็นการค้นหา posterior parameters ซึ่งน่าจะเป็นมากที่สุดจาก Meta-dataset

โดยทั่วไปชุดข้อมูลที่เกี่ยวกับ Machine Learning จะมีเพียงชุดฝึก (training set), ชุดทดสอบ (testing set) และชุดตรวจสอบความถูกต้อง (validation set) แต่สำหรับการฝึกอบรม, การทดสอบ และการตรวจสอบความถูกต้องใน meta-learning ชื่อเหล่านั้นจะถูกเปลี่ยนชื่อเป็น meta-training set, meta-testing set และmeta-validation set ภายใน meta-training set จะมีจำนวน training

ชุดสนับสนุน (support set) คือชุดคู่ข้อมูลและ ป้ายกำกับ (input และlabel) ในขณะที่ป้ายกำกับ (label) จะแตกต่างกันภายในชุดเดียวกัน และแตกต่างกันในทุกงาน (task) เพื่อค้นหาไปตามชุดสนับสนุนอื่นเพื่อเลือกป้ายกำกับที่เหมาะสมกับข้อมูลที่ตรงกันกับชุดแบบสอบถาม (query set)

N-way K-shot หมายถึงจำนวนป้ายกำกับ (label) N และข้อมูลการฝึกจำนวน K ต่อป้ายกำกับถูกแบ่งออกเป็น Zero-Shot Learning, One-Shot Learning และFew-Shot Learning ตัวอย่างเช่นมีข้อมูล 100 รายการ (อินพุตและป้ายกำกับ) ในขณะที่มีป้ายกำกับที่แตกต่างกัน 6 ป้าย ในทุกชุดข้อมูลจะมีเพียง K ตัวอย่าง ในชุดสนับสนุน (มีป้ายกำกับ N) เท่านั้นที่จะถูกป้อนเข้าไปในโมเดล และไม่จำเป็นต้องจับคู่กับป้ายกำกับที่แตกต่างกันทั้งหมด (เช่น 6 ในกรณีนี้) ป้ายกำกับมีความสอดคล้องกันทั้งชุดการสนับสนุน(support set) และชุดแบบสอบถาม (query set) ของงาน (task) เดียวกัน และแตกต่างกันไปในแต่ละงานดังภาพที่ 2.1



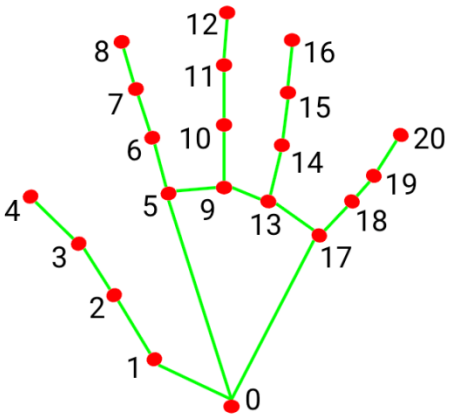
ภาพที่ 2.1 การจัดการข้อมูลของ Meta-dataset

ที่มา: https://ichi.pro/th/kar-naeana-meta-learning-xyang-xxn-yon-156366670479201

2.2 Mediapipe

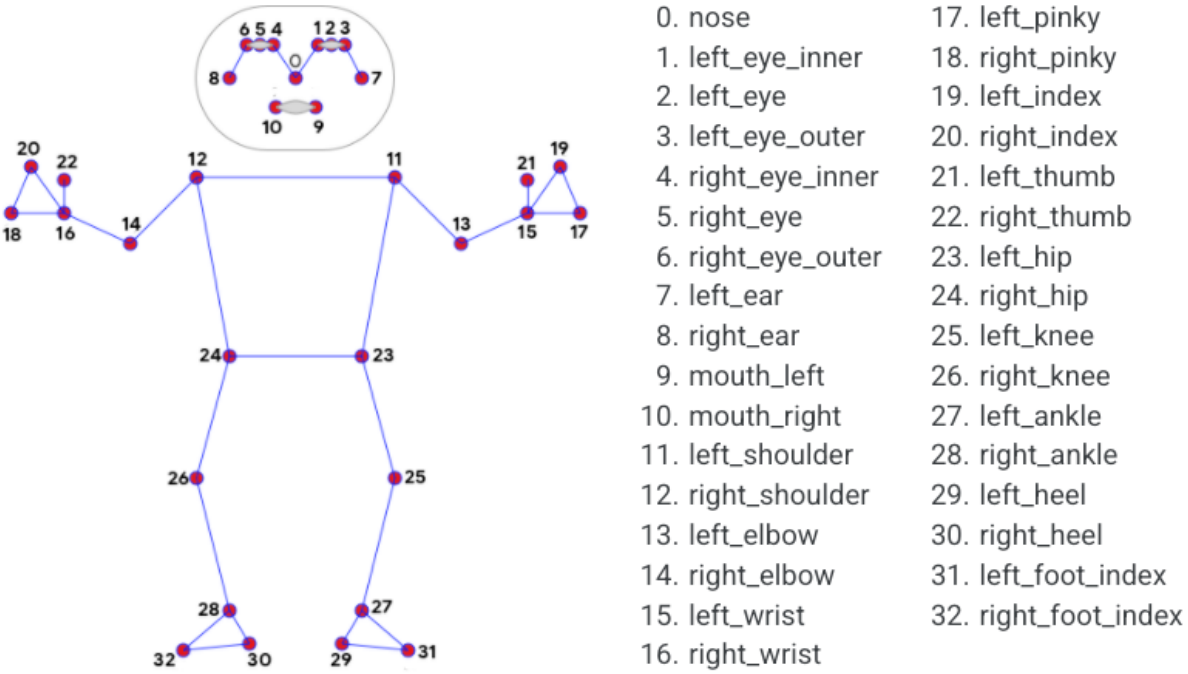
Mediapipe เป็นแพลตฟอร์ม AI แบบ Open source ของ Google ที่สามารถใช้เป็น Pipeline ตรวจจับท่าทาง มือ และใบหน้าของมนุษย์ในเวลาเดียวกัน โดยใช้การโอนถ่ายหน่วยความจำระหว่าง Inteference Backend ซึ่ง Pipeline จะรวมรูปแบบการปฏิบัติการและการประมวลผลที่แตกต่างกันตามการตรวจจับภาพแต่ละส่วนเข้าด้วยกัน และจะได้เป็นโซลูชันแบบครบวงจรที่ใช้งานได้แบบเรียลไทม์และสม่ำเสมอ MediaPipe คือโทโปโลยีล้ำสมัยที่สามารถตรวจจับท่าทาง มือ และใบหน้า

MediaPipe ทำงานแลกเปลี่ยนกันระหว่างการตรวจจับจุดทั้งสามจุด โดยประสิทธิภาพของการทำงานจะขึ้นอยู่กับความรวดเร็ว และคุณภาพของการแลกเปลี่ยนข้อมูล เมื่อรวมการตรวจจับทั้งสามเข้าด้วยกัน จะได้เป็นโทโปโลยีที่ทำงานร่วมกันเป็นหนึ่งเดียว โดยสามารถจับ Key points ของภาพเคลื่อนไหวได้ถึง 540+ จุด (ส่วนของท่าทาง 33 จุด มือข้างละ 21 จุด และส่วนใบหน้า 468 จุด) ซึ่งเป็นระดับที่ไม่เคยทำได้มาก่อน และสามารถประมวลผลได้เกือบจะเรียลไทม์ในการแสดงผลทางโทรศัพท์มือถือ



ภาพที่ 2.2 Key points ทั้ง 21 บนมือ

ที่มา: https://google.github.io/mediapipe/solutions/hands.html

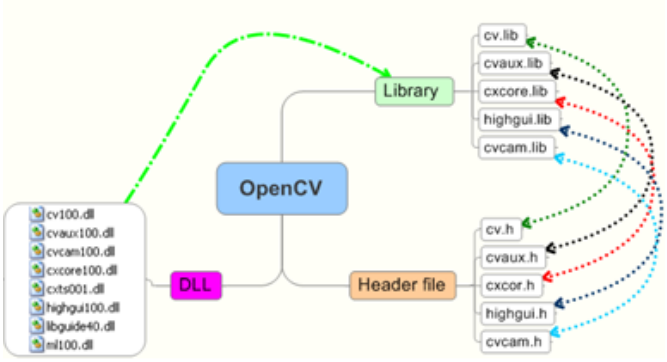


ภาพที่ 2.3 Key points ทั้ง 31 จุด บนร่างกาย

ที่มา: https://google.github.io/mediapipe/solutions/pose.html

2.3 OpenCV (Open source Computer Vision)

OpenCV ย่อมาจาก Open source Computer Vision เป็นไลบรารีสำหรับใช้ในการประมวลผลภาพ (Image Processing) ซึ่งเป็นไลบรารีโอเพนซอร์ส (Open Source) สามารถดาวน์โหลดใช้งานได้ฟรี ไลบรารีต่าง ๆ ของ OpenCV ได้พัฒนาขึ้นโดย บริษัทอินเทล (Intel) จุดเด่นในด้านความสามารถของไลบรารี OpenCV คือสามารถประมวลผลภาพดิจิตอลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องวิดีโอ หรือไฟล์วิดีโอ เป็นต้น โดยไม่ยึดติดทางด้านฮาร์ดแวร์ทำให้ OpenCV สามารถนำไปพัฒนาโปรแกรมร่วมกับภาษาอื่น ๆ รวมถึงมีฟังก์ชันที่ใช้สำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐาน โดยฟังก์ชันต่าง ๆ ของ OpenCV จะสามารถเรียกใช้งานได้จะต้องเรียกใช้ผ่านไฟล์ส่วนหัว (Header file) และลิงค์ (Link)ไลบรารีต่าง ๆ รวมถึง DLL (Dynamic Link Library) โดยมีความสัมพันธ์ดังภาพที่ 2.4 ส่วนใหญ่จะถูกนำไปใช้พัฒนาการแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) อีกทั้งยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และCaffe



ภาพที่ 2.4 แสดงความสัมพันธ์ไลบรารีของ OpenCV

ที่มา: มหาวิทยาลัยบูรพา ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

OpenCV ประกอบด้วยไลบรารี 4 ส่วน ดังนี้

1. CXCORE เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุด ขนาด อาร์เรย์ หน่วยความจำคำสั่งในการวาดภาพ และการประกาศตัวแปรภาพ

2. CV ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ

3. Machine Learning เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติ (Statistical) การแยกคลาสและการแบ่งกลุ่มของข้อมูล (Clustering)

4. HighGUI เป็นไลบรารีที่ใช้ในการดึงภาพ การบันทึกภาพ การเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่าง รวมไปถึงการ ตรวจสอบเมาส์ และแป้นพิมพ์

2.4 ตัววัดประสิทธิภาพของโมเดล (Confusion Matrix)

Confusion Matrix ถือเป็นเครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย หรือ Prediction ที่ทำนายจากโมเดลที่สร้างขึ้นใน Machine learning เทียบกับผลลัพธ์จริง

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ภาพที่ 2.5 ตัวอย่างเมตริกซ์การวัดประสิทธิภาพสำหรับการจำแนกประเภทข้อมูล 2 กลุ่ม

ที่มา: https://medium.com/@pagongatchalee/confusion-matrix-learning-fba6e3f9508c

จากภาพที่ 2.5 สามารถอธิบายเพิ่มเติมได้ดังนี้

1. True Positive (TP) คือสิ่งที่โมเดลทำนายตรงกับสิ่งที่เกิดขึ้นจริง ในกรณี ทำนายว่าจริง และสิ่งที่เกิดขึ้น ก็คือ จริง

2. True Negative (TN) คือสิ่งที่โมเดลทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณี ทำนายว่า ไม่จริง และสิ่งที่เกิดขึ้น ก็คือ ไม่จริง

3. False Positive (FP) คือสิ่งที่โมเดลทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง

4. False Negative (FN) คือสิ่งที่โมเดลทำนายไม่ตรงกับที่เกิดขึ้นจริง คือทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้น คือ จริง

โดยที่ TP, TN, FP และFN ในตารางจะแทนด้วยค่าความถี่ สามารถใช้ Confusion Matrix มาคำนวณ การประเมินประสิทธิภาพการทำนายของโมเดล ในรูปแบบค่าต่าง ๆ ได้หลายค่า ได้แก่ Accuracy (ค่าความถูกต้องที่โมเดลทำนายได้ตรงกับสิ่งที่เกิดขึ้นจริง) ซึ่งหาได้จาก

$$\text{Accuracy (ค่าความถูกต้อง)} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

หรือกล่าวได้ว่า Accuracy เท่ากับ ผลรวมของตัวเลขบนเส้นทแยงมุมในตาราง Confusion Matrix / จำนวน observations ทั้งหมด โดยความเป็นจริงแล้ว Confusion matrix ไม่จำเป็นต้องเป็นแบบ 2x2 หรือมีผลลัพธ์แค่ 2 แบบเสมอไป โดยอาจเป็น 3x3, 4x4 หรือ nxn ก็ได้ซึ่งสามารถหา Accuracy แบบเดิม คือผลรวมของตัวเลขบนเส้นทแยงมุมในตาราง Confusion Matrix / จำนวน observations ทั้งหมด

2.5 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวกับการรู้จำภาษามือไทย และวิธีการเรียนรู้แบบไม่กี่ตัวอย่าง นั้นมีจำนวนไม่มาก ดังนั้นผู้วิจัยจึงได้ศึกษาการรู้จำภาษามือโดยวิธีอื่น ๆ และการนำวิธีการเรียนรู้แบบไม่กี่ตัวอย่างไปประยุกต์ใช้กับงานต่าง ๆ อีกทั้งการเรียนรู้แบบไม่กี่ตัวอย่างยังแบ่งออกเป็นหลายวิธี ผู้วิจัยได้ศึกษาและรวบรวมงานวิจัยที่ได้รับการตีพิมพ์ดังกล่าวไว้ และสรุปได้ดังนี้

Careaga, C., Hutchinson, B., Hodas, N. & Phillips, L., (2019). Metric-Based Few-Shot Learning for Video Action Recognition งานวิจัยนี้กล่าวถึงการรู้จำท่าทางต่าง ๆ โดยใช้ชุดข้อมูล Kinetics 600 โดยเริ่มต้นนำวิดีโอท่าทางมาทำ RGB และOptical Flow หลังจากนั้นทำ pre-trained CNN ด้วย RestNet-18 และ Alext ตามลำดับ งานวิจัยนี้ได้ทดลองใช้ Aggregation Techniques 4 วิธี เพื่อนำแต่ละเฟรมมารวมกัน ได้แก่ Pooling, LSTM, ConvLSTM และ 3D Convolutions และในการทำ Few-Shot Learning ได้ใช้ โมเดลที่ได้รับความนิยม 3 โมเดลมา

เปรียบเทียบกัน ได้แก่ Matching Networks, Prototypical Networks และ Learned distance metric ได้ผลดังตารางที่ 2.1

ตารางที่ 2.1 ผลการเปรียบเทียบประสิทธิภาพของโมเดล และ Aggregation Techniques จากการทดสอบด้วยข้อมูลปกติ และการใช้ข้อมูลที่มีความท้าทาย

General Test Set				
Method	**Averaging**	**LSTM**	**ConvLSTM**	**3dConv**
Prototypical	83.5 ± 0.46	**84.2 ± 0.44**	77.9 ± 0.53	78.8 ± 0.51
Matching	79.1 ± 0.55	81.1 ± 0.50	75.7 ± 0.56	75.7 ± 0.56
Learned	77.9 ± 0.51	-	78.1 ± 0.51	74.1 ± 0.55
Challenge Test Set				
Method	**Averaging**	**LSTM**	**ConvLSTM**	**3dConv**
Prototypical	58.5 ± 0.58	**59.4 ± 0.59**	53.6 ± 0.60	54.4 ± 0.60
Matching	52.3 ± 0.57	54.6 ± 0.60	51.0 ± 0.60	49.2 ± 0.58
Learned	51.5 ± 0.61	-	52.3 ± 0.61	50.3 ± 0.61

ที่มา: Careaga, C., Hutchinson, B., Hodas, N. & Phillips, L., (2019)

จากตารางที่ 2.1 แสดงให้เห็นว่าการใช้โมเดล Prototypical Networks และAggregation Techniques LSTM เพื่อรวมแต่ละเฟรมเข้าด้วยกัน และจัดการความยาวของแต่ละวิดีโอ ให้ผล Accuracy ที่ 84.2 % ในการทดสอบด้วยข้อมูลปกติ และ 59.4 % ในชุดข้อมูลที่มีความท้าทาย ความท้าทายในกรณีนี้หมายถึงข้อมูลที่มีความคล้ายกันมาก ๆ

Chaikaew, A., Somkuan, K., Yuyen, T. (2021). Thai Sign Language Recognition: an Application of Deep Neural Network สำหรับงานวิจัยนี้ได้นำเสนอการรู้จำคำศัพท์ภาษามือไทยโดยสร้างชุดข้อมูลด้วยตนเอง จำนวน 5 ท่าทาง ท่าทางละ 100 วิดีโอ รวมทั้งสิ้น 500 วิดีโอ และทำการมาร์คจุดสำคัญบนมือด้วย Mediapipe ได้ผลลัพธ์ออกมาเป็น ตำแหน่งพื้นที่แต่ละจุด ซึ่งในวิจัยนี้มาร์คเฉพาะจุดบนฝ่ามือจำนวนข้างละ 21 จุด หลังจากนั้น แคปเจอร์เฟรมจากวิดีโอ นำออกเป็นไฟล์

CSV แล้วนำไปเทรนโมเดล โดยใช้ LSTM, BLSTM และGRU นำผลลัพธ์มาเปรียบเทียบกันดังตารางที่ 2.2 พบว่า LSTM ให้ประสิทธิภาพดีที่สุด มี Accuracy ที่ 97 % และ Loss ที่ 6 %

ตารางที่ 2.2 ผลการเปรียบเทียบประสิทธิภาพของโมเดล

Model	Accuracy	Loss
LSTM	**0.97**	**0.06**
BLSTM	0.94	0.23
GRU	0.94	0.14

ที่มา: Chaikaew, A., Somkuan, K., Yuyen, T. (2021)

Janeera.D.A, K.Mukilan Raja, Pravin U K R & Krishor Kumar.M. (2021).Neural Network based Real Time Sign Language Interpreter for Virtual Meet งานวิจัยนี้กล่าวถึงการรู้จำภาษามือบน Virtual Meet โดยขอบเขตงานครอบคลุมตั้งแต่การ Set Up เครื่องข่าย, การสร้าง Server, การสร้างโมเดล ในส่วนของการสร้างโมเดลผู้วิจัยได้ใช้ โมเดล CNN ในการเทรน และ Capture รูปภาพขณะประชุม แบ่งเป็นท่าทางทั้งหมด 16 ท่าทาง โดยโมเดลมีประสิทธิภาพที่ Accuracy 97% และมี error ที่ 3% ต้นเหตุมาจากแสงไม่เพียงพอ และพื้นหลังที่เปลี่ยนแปลงอยู่เสมอ ใช้เวลาในการประมวลผลประมาณ 1-2 วินาที

Lu, J., Nguyen, M. & Yan, W. (2021). Sign Language Recognition from Digital Videos Using Deep Learning Methods งานวิจัยนี้กล่าวถึงการรู้จำท่าทางของคำศัพท์ 4 คำได้แก่ Hello, Nice, Meet และ You โดยใช้ Capsule Network และ LSTM .ในการสร้างโมเดลทำนาย class ของแต่ละท่าทางหลังจากนั้นนำผลที่ได้มาโหวต เพื่อเลือกผลลัพธ์สุดท้าย ซึ่งวัดประสิทธิภาพได้ Accuracy ที่ 98.96%

Rivera, A. M., Ruiz-Varela, J., Ortega-Cisneros, S., Rivera, J., Parra-Michel, R., & Mejia-Alvarez, P. (2021). Spelling Correction Real-Time American Sign Language Alphabet Translation System Based on YOLO Network and LSTM งานวิจัยดังกว่านำเสนอการสะกดคำภาษามือพร้อมกับการแก้ไขคำผิด งานวิจัยถูกแบ่งออกเป็น 3 ส่วนที่สำคัญ ได้แก่

1. ปรับขนาดรูปและobject detection YOLO ซึ่งรับ input มาจาก กล้องแบบ Real time เนื่องจากสามารถตรวจจับได้เร็วที่สุด 45 เฟรมต่อวินาที การแปลงรูปที่ได้ให้ออกมาเป็นอักษรหรือสระ แต่ละ time series

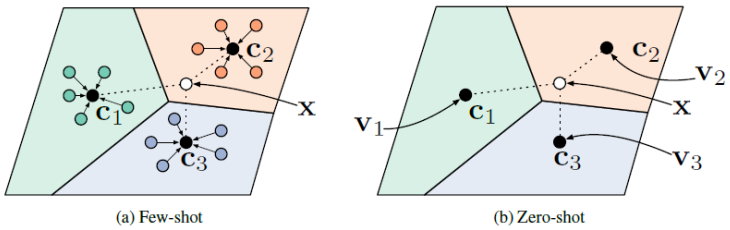
2. นำตัวอักษรที่ได้ในขั้นแรกมาแปลงเป็นคำ หากมีตัวอักษรที่เหมือนกันติดกันจะลบออก

3. ปรับปรุงคำศัพท์จากขั้นตอนที่ 2 นำเข้าโมเดล BiLSTM แก้ไขคำผิด และทำนายประโยคจากคลังรวบรวมประโยคจำนวน 235 ประโยค ที่ใช้ในชีวิตประจำวัน

งานวิจัยนี้ได้เปรียบเทียบโมเดล YOLOv3-tiny กับ YOLOv3 และมีการทดลองปรับจูนตัวแปรในแต่ละชั้น ได้โมเดลที่มีประสิทธิภาพดีที่สุดคือ YOLOv3 ได้ mAP @ 50 ที่ 99.81 %/81.74% และ 99.85%/81.76% สำหรับภาพขนาด 352×352 และ 416×416 ตามลำดับ และสำหรับผลการทดลองประสิทธิภาพของโมเดล BiLSTM ในการทำนายประโยคได้ Accuracy 98.07%

Snell, J., Swersky, K. & Zemel, R. (2017). Prototypical Networks for Few-shot Learning สำหรับงานวิจัยนี้ นำเสนอแนวคิดแบบจัดกลุ่ม และถูกนำไปใช้เพื่อทำนายป้ายกำกับของข้อมูลในขณะที่โมเดลการทำคลัสเตอร์นี้จะได้รับการฝึกอบรม (train) สำหรับแต่ละตอน (episode) และคำนวณการสูญเสีย (loss)

ในทุกตอน โมเดลจะถ่ายโอนทั้ง support set และquery set ไปยังเลเยอร์การ embedded เซนทรอยด์หรือจุดศูนย์กลางของกลุ่ม (เช่น c_1, c_2 และ c_3 ในภาพ) คือค่าเฉลี่ยของ label ที่เกี่ยวข้อง ในขณะที่ query set (เช่น x ในภาพ) จะถูกจัดให้อยู่ในประเภทเดียวกันกับคลัสเตอร์ที่ใกล้ที่สุด เช่น c_2 ในภาพ แทนที่จะใช้การหาระยะห่างแบบโคไซน์ (Cosine distance) Prototypical Networks ใช้ระยะห่างแบบยุคลิด (Euclidean distance) เพื่อคำนวณหาความแตกต่างของเซนทรอยด์ระหว่าง support set และquery set



ภาพที่ 2.6 Prototypical networks ใน Few-shot และZero-shot

ที่มา: Snell, J., Swersky, K. & Zemel, R. (2017)

วิจัยนี้ใช้ชุดข้อมูลเดียวกันกับ Matching Networks (Omniglot และImageNet) เพื่อเปรียบเทียบผลลัพธ์กับ Matching Networks ผลการทดลองแสดงให้เห็นว่า Prototypical Networks มีประสิทธิภาพดีกว่า Matching Networks

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot	20-way Acc. 1-shot	20-way Acc. 5-shot
MATCHING NETWORKS [29]	Cosine	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETWORKS [29]	Cosine	Y	97.9%	98.7%	93.5%	98.7%
NEURAL STATISTICIAN [6]	-	N	98.1%	99.5%	93.2%	98.1%
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	**98.8%**	**99.7%**	**96.0%**	**98.9%**

ภาพที่ 2.7 ผลการเปรียบเทียบประสิทธิภาพของโมเดลแบบ Few-shot กับชุดข้อมูล Omniglot

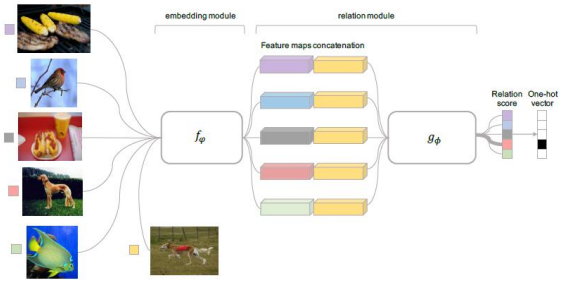
ที่มา: Snell, J., Swersky, K. & Zemel, R. (2017)

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot
BASELINE NEAREST NEIGHBORS*	Cosine	N	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
MATCHING NETWORKS [29]*	Cosine	N	$43.40 \pm 0.78\%$	$51.09 \pm 0.71\%$
MATCHING NETWORKS FCE [29]*	Cosine	N	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
META-LEARNER LSTM [22]*	-	N	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	$\mathbf{49.42 \pm 0.78\%}$	$\mathbf{68.20 \pm 0.66\%}$

ภาพที่ 2.8 ผลการเปรียบเทียบประสิทธิภาพของโมเดลแบบ Few-shot กับชุดข้อมูล ImageNet

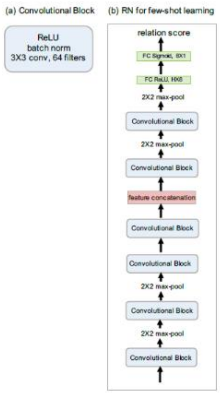
ที่มา: Snell, J., Swersky, K. & Zemel, R. (2017)

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T.(2018). Learning to Compare: Relation Network for Few-Shot Learning งานวิจัยนี้นำเสนอวิธีการ Relation Network ประกอบไปด้วย embedding module ซึ่งทำการ embedding ระหว่าง query และ training หลังจากนั้นนำมาเปรียบเทียบโดยใช้ relation module ว่าจัดอยู่ในประเภทเดียวกันหรือไม่ ซึ่งผลที่ได้จาก relation module จะได้ผลออกมาเป็น relation score ดังภาพที่ 2.9



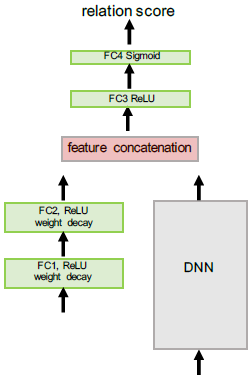
ภาพที่ 2.9 Relation Network architecture สำหรับ 5-way 1-shot กับ 1 query example

ที่มา: Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T. (2018)



ภาพที่ 2.10 Relation Network architecture สำหรับ few-shot learning (b) และส่วนประกอบของแต่ละ convolutional block (a).

ที่มา: Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T. (2018)



ภาพที่ 2.11 Relation Network architecture สำหรับ zero-shot learning

ที่มา: Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T. (2018)

ผลการทดลองประเมินจาก 2 งานคือแบบ few-shot Learning บนชุดข้อมูล Omniglot และ ImageNet และ zero-shot Learning บนชุดข้อมูล Animals with Attributes (AwA) และCaltech-UCSD Birds-200-2011 (CUB)

Model	Fine Tune	5-way Acc.		20-way Acc.	
		1-shot	5-shot	1-shot	5-shot
MANN [32]	N	82.8%	94.9%	-	-
CONVOLUTIONAL SIAMESE NETS [20]	N	96.7%	98.4%	88.0%	96.5%
CONVOLUTIONAL SIAMESE NETS [20]	Y	97.3%	98.4%	88.1%	97.0%
MATCHING NETS [39]	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETS [39]	Y	97.9%	98.7%	93.5%	98.7%
SIAMESE NETS WITH MEMORY [18]	N	98.4%	99.6%	95.0%	98.6%
NEURAL STATISTICIAN [8]	N	98.1%	99.5%	93.2%	98.1%
META NETS [27]	N	99.0%	-	97.0%	-
PROTOTYPICAL NETS [36]	N	98.8%	99.7%	96.0%	98.9%
MAML [10]	Y	98.7 ± 0.4%	**99.9 ± 0.1%**	95.8 ± 0.3%	98.9 ± 0.2%
RELATION NET	N	**99.6 ± 0.2%**	**99.8± 0.1%**	**97.6 ± 0.2%**	**99.1± 0.1%**

ภาพที่ 2.12 ผลจากชุดข้อมูล Omniglot สำหรับ few-shot classification

ที่มา: Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T. (2018)

จากภาพที่ 2.12 แสดงให้เห็นว่า Relation Network ให้ผลที่ดีที่สุดเมื่อเทียบกับทุกโมเดล หากไม่ ปรับจูน network ที่ 5-way 1 shot มีค่า Accuracy อยู่ที่ 99.6 ± 0.2% เมื่อเพิ่มจำนวน

shot ทำให้มีประสิทธิภาพเพิ่มขึ้น 0.2 % ที่ 5-way 5 shot มีค่า Accuracy อยู่ที่ 99.8 ± 0.1% และที่ 20-way 1 shot มีค่า Accuracy อยู่ที่ 97.6 ± 0.2% หากเพิ่ม shot เป็น 5 shot มีค่า Accuracy อยู่ที่ 99.1 ± 0.1%

Model	FT	5-way Acc.	
		1-shot	5-shot
MATCHING NETS [39]	N	43.56 ± 0.84%	55.31 ± 0.73%
META NETS [27]	N	49.21 ± 0.96%	-
META-LEARN LSTM [29]	N	43.44 ± 0.77%	60.60 ± 0.71%
MAML [10]	Y	48.70 ± 1.84%	63.11 ± 0.92%
PROTOTYPICAL NETS [36]	N	49.42 ± 0.78%	**68.20 ± 0.66%**
RELATION NET	N	**50.44 ± 0.82%**	65.32 ± 0.70%

ภาพที่ 2.13 ผลจากชุดข้อมูล Imagenet สำหรับ few-shot classification

ที่มา: Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T. (2018)

จากภาพที่ 2.13 แสดงให้เห็นว่า Relation Network ให้ผลที่ดีที่สุดโดยไม่ปรับจูน network ที่ 5-way 1 shot มีค่า Accuracy อยู่ที่ 50.44 ± 0.82% เมื่อเพิ่มจำนวน shot ทำให้มีประสิทธิภาพเพิ่มขึ้น ที่ 5-way 5 shot มีค่า Accuracy อยู่ที่ 65.32 ± 0.70%

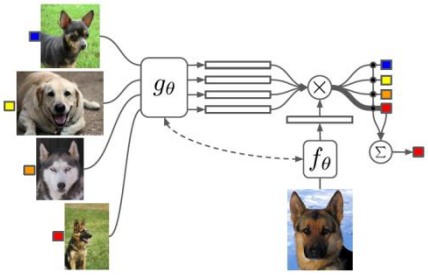
สำหรับ Zero-shot classification โดยใช้ชุดข้อมูล AwA และCUB ซึ่ง Relation Network ให้ผลที่ดีที่สุดที่ 84.5% และ62% ตามลำดับ ในชุดข้อมูล AwA โมเดล Learning a deep embedding model ให้ผลที่ดีกว่าที่ Accuracy ที่ 86.7/78.8

Vinyals, O., Blundell, C., & Lillicrap, T. (2017). Matching Networks for One Shot Learning งานวิจัยนี้นำเสนอวิธีการใหม่ที่เรียกว่า Matching Networks สร้างขึ้นเพื่อแก้ไขปัญหา One-Shot Learning หรือ Few-Shot Learning เป็นโมเดลที่ผสมผสานข้อดีระหว่างแบบ non-parametric และ parametric เข้าด้วยกัน ทำให้สามารถเรียนรู้ได้อย่างรวดเร็ว และกระบวนการในการเรียนรู้แบบง่าย ๆ ไม่ซับซ้อน ใช้ตัวอย่างในการเทรนเพียงไม่กี่ตัวอย่าง ต่อ class และยังเปลี่ยน task แบบ minibatch ต่อ minibatch แนวคิดของวิธีการนี้คือ หลังจากได้ support set และquery set แล้ว นำไปทำ Embedding แล้วใช้ Cosine distance ในการคำนวณหาความคล้ายของข้อมูล มีขั้นตอนดังต่อไปนี้

1. เลือก N คือจำนวนคู่ข้อมูล และlabel ที่แตกต่างกัน (input และlabel) เป็นข้อมูล support set

2. เลือก K คือจำนวนข้อมูลการฝึก K ต่อ label ในขณะที่ label ของ K เป็นหนึ่งใน label ของ support set ขั้นตอนที่ 1

3. คำนวณหาความคล้าย (เช่น Cosine distance) ระหว่างผลลัพธ์ผลลัพธ์ จากขั้นตอนที่ 2 และผลลัพธ์จาก support set



ภาพที่ 2.14 Matching Networks architecture

ที่มา: Vinyals, O., Blundell, C., & Lillicrap, T. (2017)

งานวิจัยนี้ได้ทดสอบกับงานประเภท Computer Vision (CV) และ Natural Language Processing (NLP) เพื่อยืนยันว่าโมเดลสามารถนำไปใช้กับปัญหาโดเมนต่าง ๆ ได้ ในงานโดเมน CV จะใช้ชุดข้อมูล Omniglot และImageNet สำหรับการทดสอบ ได้ผลลัพธ์ดังภาพที่ 2.15 และภาพที่ 2.16 ตามลำดับ

Model	Matching Fn	Fine Tune	5-way Acc		20-way Acc	
			1-shot	5-shot	1-shot	5-shot
PIXELS	Cosine	N	41.7%	63.2%	26.7%	42.6%
BASELINE CLASSIFIER	Cosine	N	80.0%	95.0%	69.5%	89.1%
BASELINE CLASSIFIER	Cosine	Y	82.3%	98.4%	70.6%	92.0%
BASELINE CLASSIFIER	Softmax	Y	86.0%	97.6%	72.9%	92.3%
MANN (NO CONV) [21]	Cosine	N	82.8%	94.9%	–	–
CONVOLUTIONAL SIAMESE NET [11]	Cosine	N	96.7%	98.4%	88.0%	96.5%
CONVOLUTIONAL SIAMESE NET [11]	Cosine	Y	97.3%	98.4%	88.1%	97.0%
MATCHING NETS (OURS)	Cosine	N	**98.1%**	**98.9%**	**93.8%**	98.5%
MATCHING NETS (OURS)	Cosine	Y	97.9%	98.7%	93.5%	**98.7%**

ภาพที่ 2.15 ผลการเปรียบเทียบประสิทธิภาพของโมเดลกับชุดข้อมูล Omniglot

ที่มา: Vinyals, O., Blundell, C., & Lillicrap, T. (2017)

Model	Matching Fn	Fine Tune	5-way Acc 1-shot	5-shot
PIXELS	Cosine	N	23.0%	26.6%
BASELINE CLASSIFIER	Cosine	N	36.6%	46.0%
BASELINE CLASSIFIER	Cosine	Y	36.2%	52.2%
BASELINE CLASSIFIER	Softmax	Y	38.4%	51.2%
MATCHING NETS (OURS)	Cosine	N	41.2%	56.2%
MATCHING NETS (OURS)	Cosine	Y	42.4%	58.0%
MATCHING NETS (OURS)	Cosine (FCE)	N	44.2%	57.0%
MATCHING NETS (OURS)	Cosine (FCE)	Y	**46.6%**	**60.0%**

ภาพที่ 2.16 ผลการเปรียบเทียบประสิทธิภาพของโมเดลกับชุดข้อมูล ImageNet

ที่มา: Vinyals, O., Blundell, C., & Lillicrap, T. (2017)

สำหรับโดเมน NLP ใช้ชุดข้อมูล Penn Treebank ประสิทธิภาพของ Matching Networks ยังไม่โดดเด่นเมื่อเปรียบเทียบกับโมเดลอื่น ๆ

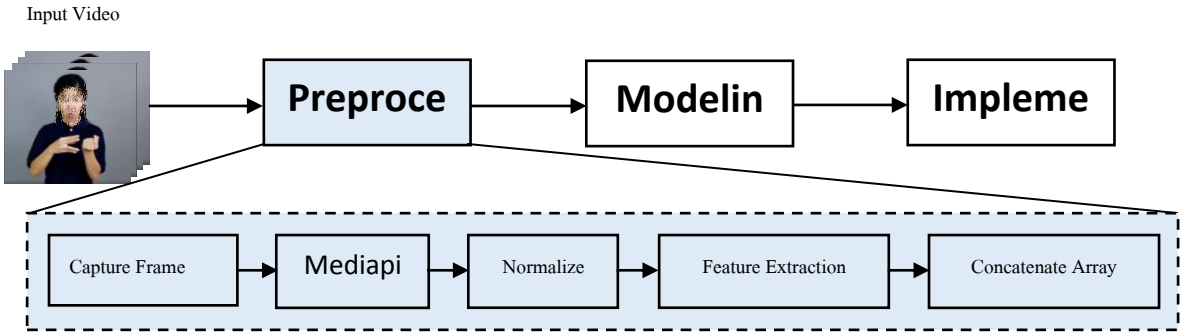
1. an experimental vaccine can alter the immune response of people infected with the aids virus a <blank_token> u.s. scientist said.	prominent
2. the show one of five new nbc <blank_token> is the second casualty of the three networks so far this fall.	series
3. however since eastern first filed for chapter N protection march N it has consistently promised to pay creditors N cents on the <blank_token>.	dollar
4. we had a lot of people who threw in the <blank_token> today said <unk> ellis a partner in benjamin jacobson & sons a specialist in trading ual stock on the big board.	towel
5. it's not easy to roll out something that <blank_token> and make it pay mr. jacob says.	comprehensive
Query: in late new york trading yesterday the <blank_token> was quoted at N marks down from N marks late friday and at N yen down from N yen late friday.	dollar

ภาพที่ 2.17 ตัวอย่างข้อมูล input และsupport set ใน NLP

ที่มา: Vinyals, O., Blundell, C., & Lillicrap, T. (2017)

บทที่ 3
ระเบียบวิธีวิจัย

การศึกษาวิจัยครั้งนี้เป็นการนำเสนอระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้เชิงลึกบนข้อมูลบางส่วนด้วยเทคนิค Few-Shot Learning จากวิดีโอโดยมีแนวทางการวิจัยดังนี้



ภาพที่ 3.1 การทำงานของระบบแปลคำศัพท์ภาษามือไทยโดยการเรียนรู้แบบ Few-Shot Learning

3.1 การเก็บข้อมูล (Input Video)

3.1.1 ข้อมูลที่ใช้ในการศึกษา

ในการเก็บข้อมูล ทำการบันทึกคลิปวิดีโอโดยใช้ Resolution หรือความละเอียดคมชัดของวิดีโอ 1080p HD ที่ 30 fps (Frame Per Second) โดยตั้งกล้องถ่ายในพื้นหลังแบบหยุดนิ่ง และถ่ายเพียงแค่ท่อนบนของร่างกายตั้งแต่เอวขึ้นไป ซึ่งบันทึกทั้งหมด 47 คำ คำละ 6 วิดีโอ โดยเฉลี่ยแล้วความยาวต่อคำอยู่ที่ 4 วินาที

3.2 การเตรียมข้อมูลสำหรับสอนระบบ (Preprocess)

3.2.1 Capture Frame

เมื่อได้รับ Input Video เข้ามาจะใช้ OpenCV ในการ Capture Frame ในแต่ละ Video ให้เป็นเฟรมภาพ และเก็บไว้ในโฟลเดอร์ โดยเฉลี่ยแล้วความยาวต่อคำอยู่ที่ 4 วินาที ดังนั้นจึงกำหนดให้วิดีโอสูงสุดมีความยาว 4 วินาที ใน 1 วินาทีสามารถ Capture Frame ได้ 30 Frame ดังนั้น

จะได้ Frame ทั้งหมด 120 Frame แต่จากการทดลองปรับจำนวนการ Capture Frame ลงพบว่าหากลดลง 50 % หรือ 15 Frame per second จะได้จำนวนเฟรมทั้งหมด 60 Frame ทำให้โมเดลมีประสิทธิภาพมากที่สุด เนื่องจาก Frame ที่ติดกันอาจจะมีท่าทางที่คล้ายกันมาก หรืออาจจะยังอยู่ในตำแหน่งเดียวกัน ทำให้ไม่สามารถช่วยในการแยกแยะท่าทางได้

ภาพที่ 3.2 ตัวอย่างการ Capture Frame

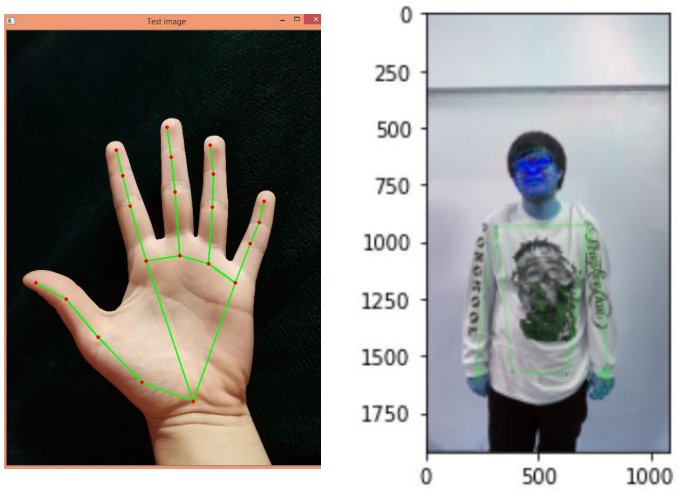
3.2.2 การแปลงระบบสี

สำหรับข้อมูลจาก Webcam จะมีระบบสี BGR แต่ Mediapipe ทำงานกับระบบสี RGB ดังนั้นจะต้องมีการแปลงค่าสีด้วยคำสั่ง cv2.cvtColor() ที่รับค่ารูปจากระบบสี BGR หรือรูปจาก Webcam และ ตัวแปลงค่าสีจาก BGR ไป RGB ที่อยู่ในคำสั่ง cv2.COLOR_BGR2RGB

3.2.3 Mediapipe

หลังจากดำเนินการขั้นตอนที่ 3.2.2 เรียบร้อยแล้ว และทำการระบุตำแหน่งของ Landmark หรือจุดสำคัญต่าง ๆ ของมือ และท่าทางโดยใช้ Mediapipe เก็บตำแหน่งของแต่ละจุด ค่าที่ได้เป็นพิกัดจุด 3 มิติ (x, y, z) บอกพิกัดตาม % ของขนาดภาพ สำหรับ x จะเทียบกับความกว้าง และ y จะเทียบกับความสูง เช่นภาพขนาด 1080 x 1080 หากได้ x: 0.50 และ y 0.10 หมายถึงพิกัดที่ x = 0.5 * 1080 และ y = 0.10 * 1080 หรือ (540, 108) ในพิกัด (x, y) ซึ่งจุด 0,0 เริ่มต้นที่มุมซ้ายบน และมีค่า x, y ที่มากที่สุดที่มุมขวาล่าง ซึ่งในงานวิจัยนี้จะพิจารณาตำแหน่งใน 2 มิติเท่านั้น ได้แก่ ค่าในแนวแกน x และค่าในแนวแกน y

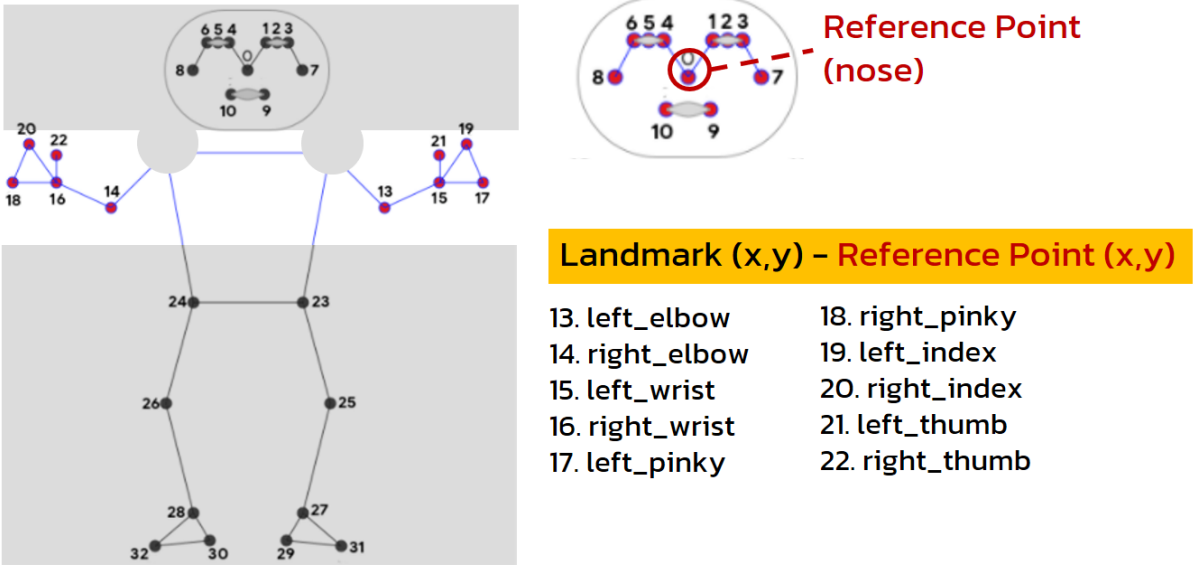
เริ่มนับเฟรมแรกของวิดีโอเมื่อมีการรับค่าตำแหน่งของมือซ้ายหรือขวาในเฟรมนั้น ๆ เพื่อนำค่าไปทำ Normalize และFeature Extraction ต่อไป หลังจากเริ่มนับเฟรมแรกไปแล้ว เมื่อทำ Landmark แต่ละเฟรม แล้วไม่พบตำแหน่งที่ต้องการจะแทนค่าด้วยเมตริก 0 ตามขนาดเดิมในตำแหน่งนั้นแทน



ภาพที่ 3.3 ตัวอย่างการ Landmark เฟรมด้วย Mediapipe

3.2.4 Normalize Landmark

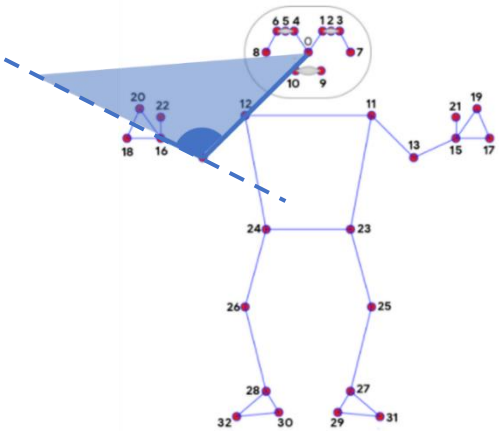
เนื่องจากสรีระร่างกาย หรือส่วนสูงของผู้ใช้งาน ไม่เท่ากันดังนั้นจึงจำเป็นต้องทำการ Normalize จุดต่าง ๆ ที่ได้จาก Landmark โดยผู้วิจัยจะถือให้ Pose [0] หรือตำแหน่งของจมูก เป็นจุดอ้างอิงของแต่ละบุคคล หลังจากนั้นจะนำจุดต่าง ๆ มาลบด้วยตำแหน่งจมูก โดยตำแหน่งของร่างกายที่นำมาใช้ทั้งหมด 10 จุดดังภาพที่ 3.4



ภาพที่ 3.4 ตัวอย่างตำแหน่งอ้างอิงของจมูก และตำแหน่ง Pose ของ Mediapipe

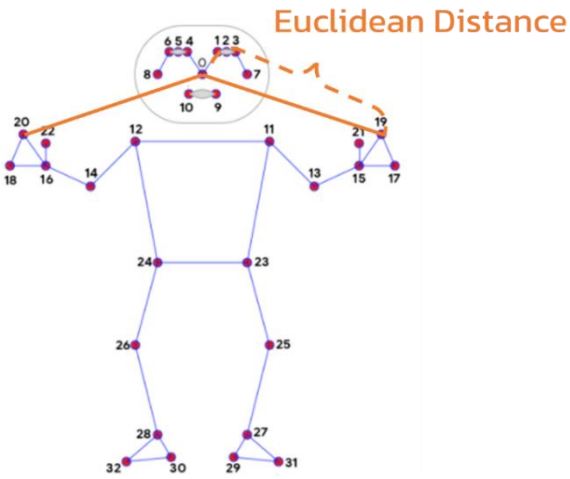
3.2.5 Feature Extraction

การทำมุมระหว่างแขนท่อนบนกับแขนท่อนปลาย เช่นการทำมุมของแขนด้านขวาใช้สร้างเวกเตอร์ระหว่างตำแหน่งที่ 12 หรือไหล่ ไปจนถึงตำแหน่งที่ 14 หรือศอก และสร้างเวกเตอร์ระหว่างตำแหน่งที่ 16 หรือข้อมือ ไปจนถึงตำแหน่งที่ 14 หรือศอก เป็นแขนของมุม กำหนดให้ตำแหน่งของศอกเป็นจุดยอดมุม หามุมที่กระทำระหว่างแขนด้านซ้ายเช่นเดียวกันกับด้านขวา



ภาพที่ 3.5 ตัวอย่างการทำมุมระหว่างแขนท่อนบนกับแขนท่อนปลายของแขนขวา

การหาระยะห่างระหว่างนิ้วชี้ของมือทั้งสองข้างกับจมูก โดยใช้การหาระยะทางแบบ Euclidean Distance จากตำแหน่งจมูกไปยังตำแหน่งปลายนิ้วชี้โดยใช้พิกัดจาก Pose



ภาพที่ 3.6 ตัวอย่างระยะห่างระหว่างนิ้วชี้กับจมูก

3.2.6 Concatenate Array

เมื่อดำเนินการ Normalize Landmark และ Feature Extraction เรียบร้อยแล้วจะได้ Array ของแต่ละตำแหน่ง กรณีเป็นพิกัดจุดจะแปลงให้เป็น Array มิติเดียวโดยใช้ Flatten หลังจากนั้นนำ Array มา Concatenate กันตามลำดับดังตารางที่ 3.1 จะได้ Array สำหรับ 1 เฟรมได้แก่ Array 1 มิติ มีสมาชิกจำนวน 39 Array

ตารางที่ 3.1 ตัวอย่างลำดับการ Concatenate Array ของตำแหน่งด้านขวา

ลำดับ	ตำแหน่ง	พิกัด	Shape (Array)
1	การทำมุมระหว่างแขนท่อนบนกับแขนท่อนปลาย	Pose(12,14) ทำมุมกับ Pose(14,16)	4
2	ระยะห่างระหว่างนิ้วชี้กับจมูก	ระยะห่างระหว่างPose(0) กับPose(20)	2
3	ตำแหน่งของท่าทาง	Pose(14,16,18,20,22)	33
รวม		Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก)	39

เนื่องจากข้อมูลวิดีโอคำศัพท์เป็นข้อมูลประเภท Sequence ดังนั้นการจะนำข้อมูลไปใช้ในโมเดลนั้นต้องนำ Array ของเฟรมทั้งหมดมารวมกันเป็น Input สำหรับ 1 วิดีโอ โดยนำ Array มาสร้างเป็น list แล้วจึงนำเข้าไปสอนโมเดล

3.3 การสร้างโมเดล (Few-Shot Learning Modeling)

ข้อมูลที่ได้จากข้อ 3.2.6 จะถูกใช้เป็นข้อมูลเพื่อสอนระบบ Training Data ทั้งหมด 282 วิดีโอ แบ่งการชุดการอบรมในการสอนระบบแต่ละงาน(Task / Episode) ดังนี้ N-Way = 47, K-Shot = 6, Query = 6 โดยสอนทั้งหมด 600 Task และทดสอบ 100 Task

หลังจากศึกษางานวิจัยที่เกี่ยวข้องพบว่ามีโมเดลที่นิยมใช้ในการทำ Few-Shot Learning อยู่ 3 โมเดล ได้แก่ Prototypical Networks, Matching Networks และ Relation Networks ดังนั้นจึงได้ทดลองใช้โมเดลทั้ง 3 โมเดลในการแปลคำศัพท์ เพื่อเลือกโมเดลที่มีประสิทธิภาพมากที่สุด ไปใช้งานจริง พื้นฐานของโมเดล Few-Shot Learning ทั้ง 3 แบบ คือ neural network ที่เป็น CNN แต่

จะแตกต่างกันที่วิธีการจัดกลุ่ม หรือจัดประเภท ส่วนวิธีการสอน หรือหลักการในส่วนอื่น ๆ คล้ายกันเกือบทั้งหมด การสร้างโมเดลในงานวิจัยนี้ประยุกต์มาจาก package ของ sicara ซึ่งมี Library ให้ใช้ตามความต้องการ

สำหรับโมเดลที่เลือกใช้ได้แก่ Prototypical Networks ขั้นแรก เลือก backbone เป็น ResNet50 เพื่อทำ Feature Extraction และเป็นโครงสร้างหลักโดยผ่านการสอนมาจาก ImageNet เนื่องจาก ResNet มีรูปแบบ Input เป็น 3 มิติได้แก่ (height, width, channel) เช่น (n_sample, 224, 224, 3) แต่ Input ที่ได้จากการ Concatenate Array เป็น 2 มิติ ดังนั้นจึงจำเป็นต้องจัดข้อมูลจากเดิม คือ (จำนวนเฟรม, Position ต่าง ๆ) ทำการเพิ่มมิติที่สามเป็นจำนวน วิดีโอ ที่มีอยู่ใน Dataset จะได้เป็น (จำนวนเฟรม, Position ต่าง ๆ, จำนวนวิดีโอ) และตัด layer สุดท้ายของโมเดลออกหรือชั้น output และต่อด้วย Flatten layer จากนั้นนำค่าจาก ResNet50 เข้าสู่โมเดล Prototypical Networks ซึ่งหลักการ Predict class ของ Prototypical Networks มาจากระยะทาง (คำนวณจาก Euclidean distance) ของต้นแบบ(support set) เทียบกับ input (query) ที่ใส่เข้าไป และใช้ Optimizer คือ Adam หลังจากนั้นก็นำมาทดลองปรับเปลี่ยนตัวแปร Backbone และเพิ่ม task ในการสอน และตำแหน่งที่นำเข้าไปเป็น input เพื่อปรับปรุงโมเดลให้ดีขึ้น โดยกำหนดตัวแปรดังตารางที่ 3.2 ซึ่งสามารถวัดประสิทธิภาพของโมเดล Prototypical Networks ได้ Accuracy = 0.781, Loss = 0.27

ตารางที่ 3.2 การกำหนดตัวแปรของโมเดล

Parameter	
Model	Prototypical Networks
Training Task	600 Task
Backbone	ResNet50 (imageNet)
Input	Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก)
FPS	15 fps
Optimizer	Adam
Distance	Euclidean distance
Loss Function	Categorical Cross Entropy

3.4 การนำโมเดลไปใช้กับระบบ (Implement)

นำโมเดล Prototypical Networks ซึ่งเป็นโมเดลที่มีประสิทธิภาพมากที่สุด ไปใช้ในการแปลคำศัพท์ภาษามือไทย โดยผ่าน GUI (Graphical User Interface) อย่างง่ายที่พัฒนาขึ้น ประกอบไปด้วยการทำงาน 3 ส่วนดังนี้

ส่วนที่ 1 จอแสดงผลวิดีโอ ซึ่งสามารถเชื่อมต่อกับกล้องของโน้ตบุ๊กแบบ Real ในกรณีแปลคำศัพท์จากกล้อง

ส่วนที่ 2 แสดงผลลัพธ์ของคำศัพท์ที่แปลได้ โดยอ้างอิงจากคลังคำศัพท์ที่สอนระบบ

ส่วนที่ 3 เมนูการใช้งาน ประกอบด้วย เมนูแปลคำศัพท์จากกล้องแบบ Real time ระบบจะแปลคำศัพท์จากการจับภาพผ่านกล้องของเครื่องโน้ตบุ๊ก เมนูแปลศัพท์จากวิดีโอระบบสามารถแปลคำศัพท์จากวิดีโอที่นำเข้าได้ และเมนูสุดท้ายคือเมนูออกจากระบบ เพื่อปิดการทำงานของระบบ

ภาพที่ 3.7 ตัวอย่างโปรแกรมรูปแบบ GUI ในการนำไปใช้จริง

3.5 เครื่องมือที่ใช้ในงานวิจัย

3.5.1 ภาษาไพธอน (Python)

ภาษา Python เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โครงสร้าง และไวยากรณ์ของภาษาไม่ซับซ้อน ในส่วนของการแปลงชุดคำสั่งมีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด นอกจากนั้นภาษา

โปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายและเหมาะสำหรับการทำ Data Science เนื่องจากมี Package ของชุดคำสั่งที่สามารถเลือกใช้ได้อย่างเหมาะกับงานประเภทนี้

3.5.2 Jupyter Notebook

เป็นเครื่องมือ opensource ที่ใช้ในการสร้าง Reproducible Document ซึ่งก็คือเอกสารที่ มีคำอธิบายและ code ที่สามารถ execute ได้ เพื่อทำการทดลองซ้ำและสามารถดูผลการทดลองได้ ทั้งกับข้อมูลชุดเดิมหรือข้อมูลชุดใหม่ได้ โดย Jupyter สามารถเขียน source code เป็น block สั้นๆ และเขียนอธิบายแต่ละส่วนด้วย markdown ได้ ซึ่ง Jupyter ได้สร้างระบบ kernel ที่ให้นักพัฒนาเขียน configuration เพื่อใช้งานกับภาษาหรือระบบได้หลากหลาย

3.5.3 PyTorch

PyTorch ได้รับความนิยมมาก โดยเฉพาะอย่างยิ่งสนับสนุนการทำงานในระดับโมบายล์ การเขียนโค้ดสำหรับ PyTorch นี้มีความเข้าใจง่าย framework สนับสนุนการทำงานของ python ซึ่งโมเดลนี้เป็นรูปแบบเบสิคของ python อย่างเช่นการ การเพิ่มประสิทธิภาพ,การโหลดข้อมูล,การใช้งานฟังก์ชันที่น้อยลง,การเปลี่ยนรูป และอื่น ๆ สามารถ debug โปรแกรมได้จาก tensorboard หรือใช้เทคนิคของ python โดยทั่วไป ที่มีการสร้าง กราฟจาก stack ตัวอย่าง ทำให้ง่ายต่อการเรียนรู้ deep learning จาก framework อื่น ๆ ของ data science เช่น pandas หรือ Scikit-learn

PyTorch เป็นหนึ่งใน library ที่สำคัญที่สุดของ machine learning ประโยชน์ของ PyTorch นั้นมีมากมายโดยเฉพาะในสาขา Computer Vision และ Natural Language Processing (NLP) ทั้งนี้ซอฟต์แวร์สาย Deep learning อย่างเช่น Autopilot ของ Tesla หรือว่า Pyro ของ Uber ล้วนแต่ถูกพัฒนาต่อยอดจาก PyTorch ทั้งสิ้น

3.5.4 PyQt

PyQt คือ Module ที่ไว้สร้าง GUI ในภาษา Python ที่นิยมใช้กันอย่างกว้างขวาง qt คือเฟรมเวิร์กสร้าง GUI ที่ได้รับความนิยมสูงและถูกใช้สร้างโปรแกรมต่าง ๆ มามากมายแล้ว โดยเดิมมีพื้นฐานมาจากภาษา C++ แต่ก็ถูกพัฒนาขึ้นมาให้ใช้ในภาษาต่าง ๆ เช่น java, php, python, ruby, ฯลฯ ซึ่งในปัจจุบันพัฒนามาถึง qt5 แล้ว Module ของ qt5 ในไพธอนมีชื่อว่า pyqt5 การสร้าง GUI จะทำโดยการเอา widget ต่าง ๆ มาประกอบเข้าด้วยกัน สำหรับใน pyqt นั้น widget ชนิดต่าง ๆ นั้นอยู่ในรูปของ class ของ Python ซึ่งทั้งหมดถูกบรรจุอยู่ใน Module ย่อย PyQt5.QtWidgets เรียกใช้ได้ง่าย และมี widget ให้ใช้หลากหลาย

บทที่ 4
ผลการศึกษา

จากการพัฒนาโมเดลสำหรับการแปลคำศัพท์ภาษามือไทยเพื่อช่วยเป็นตัวกลางในการสื่อสารระหว่างผู้พิการทางการได้ยินหรือสื่อความหมาย กับบุคคลทั่วไป โดยนำเอาการเรียนรู้แบบไม่กี่ตัวอย่าง (Few-Shot Learning) มาประยุกต์ใช้ในการสร้างเครื่องมือที่ใช้งานได้สะดวกผ่าน User Interface ซึ่งผลทดสอบจากการใช้งานมีรายละเอียดดังนี้

4.1 ผลการเปรียบเทียบประสิทธิภาพของโมเดล Few-Shot Learning

จากการศึกษางานวิจัยพบว่าโมเดลที่ได้รับความนิยมเกี่ยวการทำ Few-Shot Learning ได้แก่ Prototypical Networks, Matching Networks และRelation Networks ดังนั้นจึงทดลองใช้โมเดลทั้ง 3 ในการสอนระบบ นำ Training Data แบ่งการชุดการอบรมในการสอนระบบแต่ละงาน (Task / Episode) ดังนี้ N-Way = 5 ,K-Shot = 5, Query = 10 โดยสอนทั้งหมด 400 Task และทดสอบ 100 Task พบว่า Prototypical Networks มีประสิทธิภาพมากที่สุด มี Accuracy = 0.633 Loss = 0.59 และMatching Networks และRelation Networks ได้ผลการทดสอบประสิทธิภาพของโมเดลเป็นดังตารางที่ 4.1 ซึ่ง Matching Networks และRelation Networks มีหลักการทำงานดังนี้

4.1.1 Matching Networks

ขั้นแรกเลือก backbone เป็น ResNet18 หลังจากทำ Feature Extraction และเป็นโครงสร้างหลัก นำ input เข้าสู่โมเดล Matching Networks ซึ่งภายใน Matching Networks จะมีการใช้ LSTM ในการปรับแต่ง feature ของ support set และหลักการ Predict class ของ Matching Networks คือการหาความคล้าย ของ support set และquery โดยใช้ cosine similarity และใช้ Optimizer คือ adam

4.1.2 Relation Networks

ในโมเดล Relation Networks ก็ต้องเลือก backbone เป็น ResNet18 เพื่อเป็นโครงสร้างหลัก ก่อนอื่นเราทำการ extract feature maps ของ support set และquery หลังจากนั้นคำนวณหา

ค่าเฉลี่ยของ feature support ของแต่ละใน class ใน support set (ต้นแบบ)สำหรับการ Predict class ของโมเดลนี้จะนำ feature maps ไปต่อกับ feature support ของแต่ละ class (ต้นแบบ) และส่งต่อไปใน relation module (เช่น CNN) ได้ output ออกมาเป็น relation score หลังจากนั้นจัดประเภทจาก relation score สูงสุดของ class ที่ได้ แสดงว่ามีความสัมพันธ์กัน Relation Networks เราจะไม่ใส่ Flatten layer เพราะเราต้องการ output จาก backbone เป็น feature map และรูปร่างของ feature map เป็นดังนี้ (n_channels, width, height)

ตารางที่ 4.1 ผลการเปรียบเทียบประสิทธิภาพของโมเดล

5-way 5-Shot/ Hand/ 30 fps/ ResNet18/ 400 Task	Accuracy	Loss
Prototypical Networks	**0.633**	**0.59**
Matching Networks	0.632	0.75
Relation Networks	0.524	0.77

4.2 การเลือกพารามิเตอร์ และฟีทเจอร์

ผู้วิจัยได้ทดลองปรับจูนโมเดล และเพิ่มพารามิเตอร์ โดยใช้โมเดล Prototypical Networks รายละเอียดการทดลองได้แก่

4.2.1 Backbone ที่ใช้ในโมเดล

Backbone network เป็น neural network ทำหน้าที่เป็นเหมือนโครงสร้างหลักให้แก่โมเดล ที่รับรูปภาพ input และประมวลผลจนได้เป็น feature map ออกมา โดย backbone network จะใช้ architecture สำหรับงาน computer vision เช่น ResNet หรือ ResNeXt

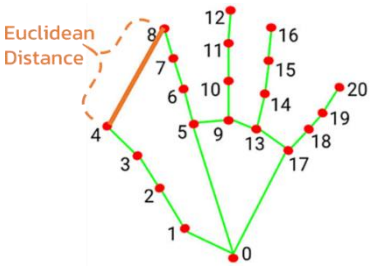
ตารางที่ 4.2 ผลการเปรียบเทียบการใช้ Backbone ต่าง ๆ

5-way 5-Shot/ Hand/ 30 fps/ 400 Task	Accuracy	Loss
ResNet18	0.633	0.59
ResNet50	**0.736**	**0.49**

จากการเปรียบเทียบผลการใช้ Backbone ปรากฏว่า ResNet50 ที่ใช้ Pre-Train โดย imageNet ส่งผลต่อประสิทธิภาพของโมเดลทำให้โมเดลทำนายได้แม่นขึ้นจากเดิมที่ใช้ ResNet18 มี Accuracy เพิ่มขึ้น 0.103

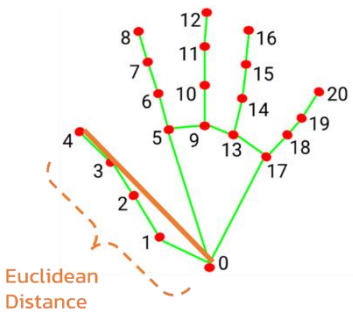
4.2.2 ฟีทเจอร์ที่นำมาใช้ ในโมเดล

สำหรับฟีทเจอร์ที่นำมาใช้ได้มีการทดลอง เพิ่ม/ลด ฟีทเจอร์ ต่าง ๆ ได้ ในตารางที่ 4.3 เป็นระยะห่างระหว่างปลายนิ้วกับข้อมือ,ระยะห่างระหว่างปลายนิ้วแต่ละนิ้ว สามารถหาได้จากระยะทางแบบ Euclidean Distance ระหว่างตำแหน่งสองตำแหน่ง เช่น ระยะทางจากตำแหน่งปลายนิ้วโป้ง ไปยังตำแหน่งปลายนิ้วชี้, ระยะทางจากปลายนิ้วชี้ ไปยังปลายนิ้วกลาง เป็นต้น ทำเช่นนี้กับทุกปลายนิ้ว และมือทั้งสองข้าง



ภาพที่ 4.1 ตัวอย่างระยะห่างระหว่างปลายนิ้วโป้งกับปลายนิ้วชี้

การหาระยะห่างระหว่างปลายนิ้วกับข้อมือโดยใช้การหาระยะทางแบบ Euclidean Distance เช่น ระยะทางจากตำแหน่งปลายนิ้วโป้ง ไปยังตำแหน่งข้อมือ, ระยะทางจากปลายนิ้วชี้ ไปยังปลายข้อมือ เป็นต้น ทำเช่นนี้กับทุกปลายนิ้ว และมือทั้งสองข้าง



ภาพที่ 4.2 ตัวอย่างระยะห่างระหว่างปลายนิ้วโป้งกับข้อมือ

ตารางที่ 4.3 ตัวอย่างตำแหน่งมือด้านขวา

ตำแหน่ง	พิกัด
ระยะห่างระหว่างปลายนิ้วกับข้อมือ,ระยะห่างระหว่างปลายนิ้วแต่ละนิ้ว	ระยะห่างระหว่าง Hand(0) กับ Hand (4) Hand(4) กับ Hand (8) Hand(0) กับ Hand (8) Hand(8) กับ Hand (12) Hand(0) กับ Hand (12) Hand(12) กับ Hand (16) Hand(0) กับ Hand (16) Hand(16) กับ Hand (20) Hand(0) กับ Hand (20)

ตารางที่ 4.4 ผลการเปรียบเทียบการใช้ตำแหน่งของร่างกาย (Mediapipe)

5-way 5-Shot/ 30 fps/ ResNet50/ 400 Task	Accuracy	Loss
ใช้ตำแหน่งของแต่ละจุด (x,y,z)		
Hand	0.610	0.59
Hand + Face + Pose	**0.634**	**0.79**
Feature Extraction (x,y)		
Hand (ปลายนิ้วกับข้อมือ/ระยะห่างปลายนิ้วกับปลายนิ้ว)	0.633	0.91
Pose(มุมแขน) + Pose(ระยะห่างจมูกกับศอก)	0.657	0.77
Hand (ปลายนิ้วกับข้อมือ/ระยะห่างปลายนิ้วกับปลายนิ้ว) + Pose(มุมแขน) + Pose(ระยะห่างจมูกกับศอก)	**0.716**	**0.79**
Feature Extraction (x,y) + Normalize		
Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ)	0.487	0.18
Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก)	**0.716**	**0.11**
Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก)+ Hand (ปลายนิ้วกับข้อมือ/ระยะห่างปลายนิ้วกับปลายนิ้ว)	0.670	0.13

จากตารางที่ 4.4 จะเห็นได้ว่า การใช้ Hand (ปลายนิ้วกับข้อมือ/ระยะห่างปลายนิ้วกับปลายนิ้ว) + Pose(มุมแขน) + Pose(ระยะห่างจมูกกับศอก) และ Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก) ได้ Accuracy เท่ากันคือ 0.716 แตกต่างกันที่ Loss ซึ่งได้ผลเท่ากับ 0.79 และ 0.44 ตามลำดับ ดังนั้น เลือกใช้ Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก) ในการสอนโมเดล

4.2.3 จำนวนในการ Capture Frame วิดีโอ

ตารางที่ 4.5 ผลการเปรียบเทียบจำนวนในการ Capture Frame วิดีโอ

5-way 5-Shot/ Pose/ ResNet50/ 400 Task	Accuracy	Loss
30 frame per second (120 frame) (100%)	0.716	0.44
15 frame per second (60 frame) (50%)	**0.853**	**0.13**
7.5 frame per second (30 frame) (25%)	0.228	0.21
3.75 frame per second (15 frame) (12.5%)	0.191	0.38

หลังจากทดสอบฟีทเจอร์ก็ได้นำข้อมูล Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก) มาทดลองเปรียบเทียบจำนวนในการ Capture Frame ได้ผลดังตารางที่ 4.5 จะเห็นได้ว่า เมื่อลด 60 FPS (15 frame per second) (50%) ได้ Accuracy = 0.853 และ Loss = 0.13 เพิ่มขึ้นจาก 120 FPS (30 frame per second) (100%) และทดลองลดลงเป็น 30 FPS (7.5 frame per second) (25%) ได้ Accuracy = 0.228 และLoss = 0.21 ทำให้ประสิทธิภาพลดลงดังนั้นจึงเลือก Capture Frame จำนวน 60 FPS (15 frame per second) (50%)

4.2.4 จำนวน Task ในการสอนโมเดล

ตารางที่ 4.6 ผลการเปรียบเทียบจำนวน Task ในการสอน

5-way 5-Shot/ Pose/ 15 fps/ ResNet50	Accuracy	Loss
400 Task	0.853	0.51
600 Task	**0.880**	**0.48**

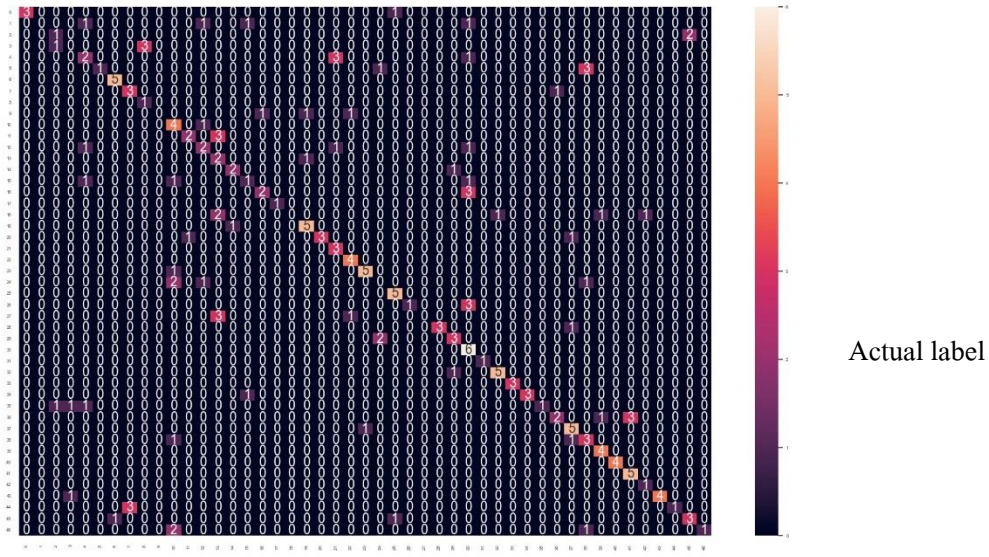
จากตารางที่ 4.6 จะเห็นได้ว่าจำนวน Task ในการ Train Model มีผลต่อประสิทธิภาพของโมเดล ซึ่งสอดคล้องกับวิธีการปกติคือยิ่งเรียนรู้เยอะก็จะยิ่งแยกแยะ หรือทำนายผลได้ดีมากยิ่งขึ้น

4.3 ผลการวัดประสิทธิภาพความถูกต้องของโมเดล

จากการทดสอบพบว่าการกำหนดตัวแปร และฟีทเจอร์ตามตารางที่ 4.6 ให้ผลการทดสอบให้ค่าความถูกต้องที่ดีที่สุด ดังนั้นจึงทำการสอนโมเดลด้วย ดังนี้ N-Way = 47, K-Shot = 6, Query = 6 ได้ค่า Accuracy = 0.781 และLoss = 0.27

ตารางที่ 4.7 การกำหนดพารามิเตอร์ของโมเดล Prototypical Networks

Model	Prototypical Networks
Training Task	600 Task
Backbone	ResNet50 (imageNet)
Input	Pose(มุมแขน) + Pose(ระยะห่างจมูกกับมือ) + Pose (ตำแหน่ง Pose - ตำแหน่งจมูก)
FPS	15 fps
Optimizer	Adam
Distance	Euclidean distance
Loss Function	Categorical Cross Entropy



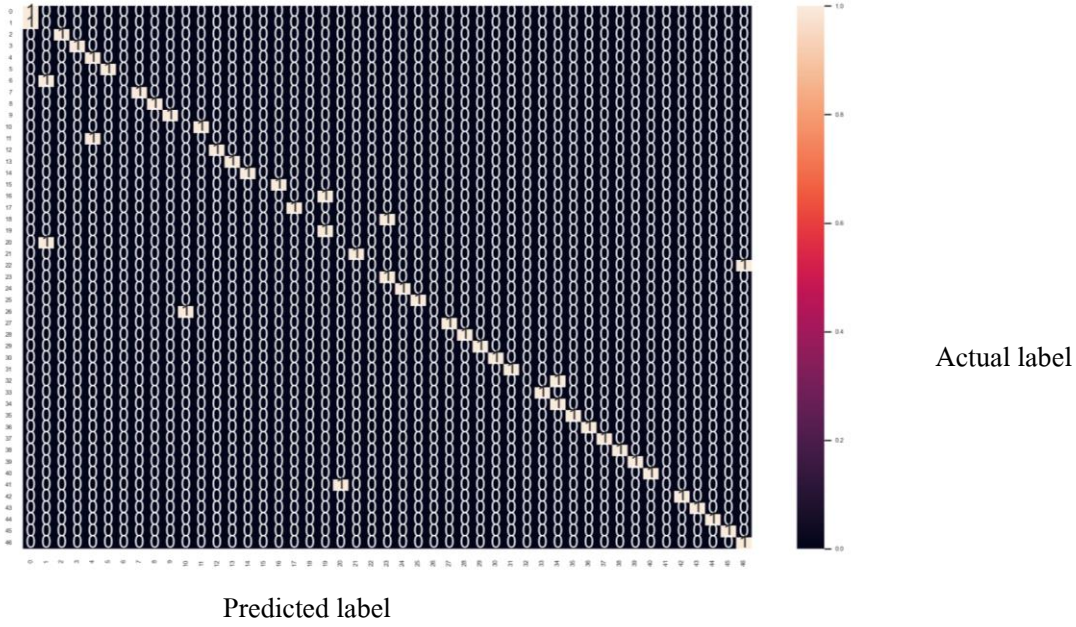
ภาพที่ 4.3 Confusion Matrix ทดสอบกับวิดีโออาสา

4.4 ผลจากการนำไปใช้งานกับวิดีโอโดยสมาคมคนหูหนวก

เนื่องจาก Data Set สร้างขึ้นนั้นเกิดจากการบันทึกวิดีโอโดยอาสาที่ไม่มีความเชี่ยวชาญในภาษามือโดยเฉพาะดังนั้นผู้วิจัยจึงทำการทดสอบระบบแปลคำภาษามือไทยโดยใช้ข้อมูลตัวอย่างวิดีโอของฐานข้อมูลภาษามือไทยที่จัดทำขึ้นโดยสมาคมคนหูหนวกแห่งประเทศไทย

ภาพที่ 4.4 ตัวอย่างการทดสอบกับวิดีโอโดยสมาคมคนหูหนวก

ผลการทดสอบ N-Way = 47, K-Shot = 6, Query = 1 ได้ค่า Accuracy = 0.74 จะเห็นได้ว่ามีประสิทธิภาพใกล้เคียงกับการทดสอบด้วยวิดีโออาสาที่สร้างขึ้นโดยผู้วิจัย ซึ่งได้ Confusion Matrix ดังภาพที่ 4.5



ภาพที่ 4.5 Confusion Matrix ทดสอบกับวิดีโอโดยสมาคมคนหูหนวก

บทที่ 5
บทสรุปและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอเกี่ยวกับการพัฒนาระบบเพื่อช่วยในการแปลคำศัพท์ภาษามือไทยในชีวิตประจำวัน โดยใช้วิธีการเรียนรู้แบบเชิงลึก เทคนิค Few-Shot Learning ซึ่งระบบนี้จะช่วยลดช่องว่างทางการสื่อสารระหว่างบุคคลทั่วไปและผู้พิการทางการสื่อสาร ช่วยแก้ปัญหาจำนวนล่ามที่ไม่เพียงพอต่อจำนวนผู้พิการ ทำให้สามารถเข้าถึงเครื่องมือที่ช่วยในการสื่อสารได้ง่ายขึ้นสามารถสรุปผลการวิจัยได้ดังนี้

5.1 สรุปผลการศึกษา

5.1.1 ได้พัฒนาโมเดลที่มีประสิทธิภาพในการแปลคำศัพท์ภาษามือไทย ประกอบด้วยขั้นตอนดังนี้

5.1.1.1 เตรียมข้อมูลวิดีโอที่เก็บมาโดยการ Capture frame เป็นรูปภาพจำนวน 60 FPS (15 frame per second) หลังจากนั้นใช้ Mediapipe ในการระบุตำแหน่งของร่างกาย และมือ

5.1.1.2 นำค่าพิกัดที่ได้มา Normalize ซึ่งถือให้จมูกของแต่ละบุคคลเป็นจุดอ้างอิง นำตำแหน่งพิกัดมาลบด้วยจุดอ้างอิง และนำพิกัดส่วนอื่นมาทำ Feature Extraction เมื่อได้ Feature ที่ต้องการก็นำมา Concatenate เป็น 1 เฟรม และนำแต่ละเฟรม มารวมกันเป็น 1 วิดีโอ จัดเก็บในรูปแบบ Numpy array โดยมีการทำ Feature Extraction ดังนี้

- แขนท่อนปลายกับท่อนบนทำมุมกี่องศา มีศอกเป็นจุดยอดมุม
- ระยะห่างระหว่างนิ้วชี้กับจมูก

5.1.1.3 นำข้อมูลที่ได้เข้าสู่โมเดล Prototypical Networks เพื่อแปลคำศัพท์ภาษามือไทย

5.1.2 ผลการทดลองให้ความแม่นยำในการแปลคำศัพท์ภาษามือไทยทั้ง 47 คำ วัดค่า Accuracy ได้ถึง 74%

5.2 ข้อสังเกต

ผลการทดสอบจากวิดีโอที่สร้างโดยอาสา และวิดีโอที่สร้างโดยสมาคมคนหูหนวก

พบว่าคำศัพท์ส่วนใหญ่ที่แปลไม่ถูกต้องนั้น เป็นคำศัพท์ที่มีท่าทางที่คล้ายกันมาก ตำแหน่งของมือองศาของแขน อยู่ในตำแหน่งที่ใกล้เคียงกัน ดังตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบผลการแปลคำศัพท์ที่ไม่ถูกต้องจากวิดีโออาสา และวิดีโอที่สร้างโดยสมาคมคนหูหนวก

วิดีโอที่สร้างโดยอาสา		วิดีโอที่สร้างโดยสมาคมคนหูหนวก	
Actual	**Predicted**	**Actual**	**Predicted**
กิน	ขี้เกียจ/ ดื่ม/ พูด/ อิ่ม	กิน	กลัว
ขี้เกียจ	คิด/ ลืม	คิดถึง	ขม
ซื้อ	ดีใจ/ ปรึกษา/ มั่นใจ	ดม	ดีใจ
ยิงปืน	ทะเลาะ/ เกลียด/ โกนหนวด/ ไม่ชอบ	ดีใจ	ขี้เกียจ
สอบ	ดม/ ดื่ม/ เสียใจ	พูด	ฟ้อง
อธิบาย	ทะเลาะ/ วิ่ง	ฟ้อง	รัก
		ยิงปืน	สงสัย
		ร้องไห้	กิน
		วิ่ง	ไม่แน่ใจ
		หิว	ดม
		เกลียด	เบื่อ
		โกรธ	ร้องไห้

5.3 ข้อเสนอแนะ

5.3.1 ควรเก็บข้อมูลตัวอย่างในแต่ละคำเพิ่ม (K-shot) เพราะจำนวน K-shot ส่งผลต่อประสิทธิภาพโมเดล หาก K-shot น้อยจะทำให้ความแม่นยำน้อยลง แต่หาก K-shot มากจะทำให้โมเดลแม่นยำขึ้น ดังนั้นจึงควรเก็บข้อมูลเพิ่มขึ้น

5.3.2 สร้างแอปพลิเคชันเพื่อรองรับการใช้งานบนมือถือ

5.3.3 การเรียงคำเป็นประโยค การสื่อสารด้วยภาษามือ มีหลายประเภท หนึ่งในนั้นคือการเลือนแบบประโยคในการพูดโดยใช้คำศัพท์มาเรียงต่อกัน ทำให้สามารถนำคำมาเรียงต่อกันเพื่อเป็นประโยคได้

5.3.4 สามารถเพิ่ม text to speech เพิ่มเติมเมื่อทำนายคำศัพท์ได้ เพิ่มความสะดวกต่อผู้ที่สื่อสาร

5.3.5 อาจมีการทดลองเพิ่มเติม (Future Work) ในขั้นตอนของการสอนระบบ ดังนี้

5.3.5.1 สร้าง Data set ด้วยบุคคลที่หลากหลายใน 1 class เนื่องจากตำแหน่งมือ สัดส่วนร่างกายหรือท่าทางของแต่ละบุคคลแตกต่างกัน จะทำให้โมเดลยืดหยุ่นขึ้น หรือเก่งขึ้น

5.3.5.2 นำโมเดลที่เก่งด้าน Sequence มาใช้รวม Aggregate เฟรมแต่ละเฟรมรวมทั้งจัดการความยาวของวิดีโอ

5.3.5.3 เพิ่ม Feature Extraction โดยวิธีอื่น ๆ เช่น Angular Features, Geometric Features

บรรณานุกรม

บรรณานุกรม

ภาษาไทย

กนิษฐา หงส์พรรคมนุญ,วันทนีย์ จันทร์สมปอง และรัตติยากร ทองจุนเจือ.(2551).ระบบตรวจสอบความบกพร่องของชิ้นงานแบบอัตโนมัติโดยวิธีประมวลผลภาพ.สืบค้น 8 กรกฎาคม 2564,จากhttp://www.lib.buu.ac.th/st/Engineering/Electrical/2551/EE-51-26.pdf

การแนะนำ Meta-Learning อย่างอ่อนโยน.,จาก https://ichi.pro/th/kar-naeana-meta-learning-xyang-xxn-yon-156366670479201

การลงโปรแกรมVisual Studio 2010และopenCV.(2555).สืบค้น 1 มิถุนายน 2564,จาก http://kwangee1245.blogspot.com/2012/04/visual-studio-2010-and-opencv.html

ธัญพร มัทวานุกูล.(2564).สถานการณ์คนพิการ 30 มิถุนายน 2564 (รายไตรมาส).สืบค้น 3 กรกฎาคม 2564,จาก https://dep.go.th/th/law-academic/knowledge-base/disabled-person-situation/สถานการณ์คนพิการ-30-มิถุนายน-2564-รายไตรมาส

พงษ์พิสิษฐ์ ธนสุทธิ์.(2564) Meta-learning คือ อะไร (learning to learn หรือ การเรียนรู้เพื่อที่จะเรียนรู้ คืออะไร).สืบค้น 3 กรกฎาคม 2564,จาก https://aibyneto.com/category/meta-learning/

MediaPipe Holistic อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า มือ และท่าทางได้ในเวลาเดียวกัน.(2021).สืบค้น 7 กรกฎาคม 2564,จาก https://www.sertiscorp.com/2021-01-15

ปกรณ์ กัญชลี.(2562).Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ใน Machine learningสืบค้น 7 กรกฎาคม 2564,จาก https://medium.com/@pagongatchalee/confusion-matrix-เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย-ในmachine-learning-fba6e3f9508c

ภาษาต่างประเทศ

Careaga, C., Hutchinson, B., Hodas, N. & Phillips, L., (2019). Metric-Based Few-Shot Learning for Video Action Recognition

Chaikaew, A., Somkuan, K., Yuyen, T. (2021). Thai Sign Language Recognition: an Application of Deep Neural Network

Janeera.D.A, K.Mukilan Raja, Pravin U K R & Krishor Kumar.M. (2021).Neural Network based Real Time Sign Language Interpreter for Virtual Meet

Lu, J., Nguyen, M. & Yan, W. (2021). Sign Language Recognition from Digital Videos Using Deep Learning Methods

MediaPipe Hand., from https://google.github.io/mediapipe/solutions/hands.html

Rivera, A. M., Ruiz-Varela, J., Ortega-Cisneros, S., Rivera, J., Parra-Michel, R., & Mejia- Alvarez, P. (2021). Spelling Correction Real-Time American Sign Language Alphabet Translation System Based on YOLO Network and LSTM

Snell, J., Swersky, K. & Zemel, R. (2017). Prototypical Networks for Few-shot Learning

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. & Hospedales, T.(2018). Learning to Compare: Relation Network for Few-Shot Learning

Vinyals, O., Blundell, C., & Lillicrap, T. (2017). Matching Networks for One Shot Learning

ภาคผนวก

ภาคผนวก ก

รายการคำศัพท์ภาษามือไทย

ภาคผนวก ก

รายการคำศัพท์ภาษามือไทย

หมวดความรู้สึก ได้แก่

เศร้า
มั่นใจ
ไม่แน่ใจ
เบื่อ
สงสัย
ดีใจ
เสียใจ
ลืม
หิว
อิ่ม
ขี้เกียจ
คิดถึง
ขยัน
กลัว
เกลียด
รัก
ชอบ
ไม่ชอบ
เหงา
ไม่จริง
โกรธ

หมวดรสชาติ ได้แก่

ไม่มีรสชาติ
ไม่อร่อย
อร่อย
เผ็ด
เค็ม
ขม
หวาน
จืด
เปรี้ยว

หมวดกริยา ได้แก่

ร้องไห้
ทะเลาะ
วิ่ง
ฟ้อง
อธิบาย
ปรึกษา
สอบ
อ่านหนังสือ
กิน
ดื่ม
พูด
ซื้อ
คิด
โกนหนวด
ยิงปืน
ดม
อาบน้ำ

ประวัติผู้เขียน

ชื่อ-นามสกุล	ณัฏยา เปลี่ยนวงษ์
ประวัติการศึกษา	วิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมคอมพิวเตอร์
	มหาลัยเทคโนโลยีราชมงคลพระนคร
	ปีกาศึกษา 2561
ตำแหน่งและสถานที่ทำงานปัจจุบัน	พนักงานพัฒนาระบบ
	บริษัท บิซิเนส เซอร์วิสเซส อัลไลแอนซ์ จำกัด