



การแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดยใช้ RNN และ MediaPipe
Sign Language for weather forecast translation with RNN and MediaPipe

นางสาวมนัสนันท์ ครุฑธา

MANASSANUN KRUTTA

นายสุวัชรชัย เกียรติเฉลิมพร

SUWATCHAI KIATCHALERMPORN

นายณัฐพล เลิศฤทธิพงศ์

NATTAPOL LERDRIDTIPONG

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2562



การแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดยใช้ RNN และ MediaPipe
Sign Language for weather forecast translation with RNN and MediaPipe

นางสาวมนัสนันท์ ครุฑธา

MANASSANUN KRUTTA

นายสุวัชรชัย เกียรติเฉลิมพร

SUWATCHAI KIATCHALERMPORN

นายณัฐพล เลิศฤทธิพงศ์

NATTAPOL LERDRIDTIPONG

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2562



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ

การแปลภาษาเพื่อการพยากรณ์อากาศเป็นคำพูดโดยใช้ RNN และ
MediaPipe
Sign Language for weather forecast translation with RNN
and MediaPipe

นิสิต

นางสาวมนัสนันท์	ครุฑธา	59102010277
นายสุวัชรชัย	เกียรติเฉลิมพร	59102010298
นายณัฐพล	เลิศฤทธิ์พงศ์	59102010902

ปริญญา

วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ

ผู้ช่วยศาสตราจารย์ ดร.วราภรณ์ วิทยานนท์

ลงชื่อ.....

(ผู้ช่วยศาสตราจารย์ ดร.วราภรณ์ วิทยานนท์)

อาจารย์ที่ปรึกษาโครงการ

บทคัดย่อ

ระบบแปลภาษามือเป็นคำพูด มีบทบาทในการช่วยเหลือผู้พิการที่มีความบกพร่องทางการได้ยินหรือทางการพูด เช่น หูหนวก หูตึง หรือไม่สามารถสื่อสารได้ด้วยเสียง ซึ่งช่วยลดเวลาในการสื่อสาร อีกทั้งยังลดค่าใช้จ่ายในการจ้างล่ามภาษามือ

งานวิจัยนี้ได้จัดทำระบบการแปลภาษามือการพยากรณ์อากาศจากวิดีโอมาเป็นคำพูดอัตโนมัติ โดยได้มีการนำ Google MediaPipe ในส่วน Hand Tracking มาใช้ตรวจจับหา keypoint เฉพาะมือนั้นๆ และแบ่งย่อยวิดีโอออกมาเป็นภาพทุกๆ keyframes จากนั้นสกัด feature ออกมาเป็นเวกเตอร์ แล้วจึงนำ feature ไปเข้าโมเดล LSTM เพื่อเรียงลำดับการเคลื่อนไหวของมือจากทุกๆ เวกเตอร์ แล้วจึงนำมาเรียงลำดับการเคลื่อนไหวให้กลายเป็นคำหนึ่งคำ ซึ่งชุดข้อมูลภาษามือการพยากรณ์อากาศถูกแบ่งออกเป็น Train 80% และ Test 20% เพื่อได้ผลลัพธ์ที่น่าเชื่อถือและหลีกเลี่ยงการ overfitting ระหว่างการ Train ค่าพารามิเตอร์ของโมเดลที่ดีที่สุดถูกเลือกผ่านการทำ 5-fold cross validation ซึ่งได้ค่าความแม่นยำถึง 77.4% โดยนำระบบการแปลภาษามือการพยากรณ์อากาศมาพัฒนาเป็นแอปพลิเคชันเพื่อให้ผู้ที่มีความบกพร่องสามารถใช้งานได้จริง

Abstract

Sign language of the weather forecast translation system plays a big role in helping the disability people such as muted, deaf and hearing impaired to communicate with normal people and also reducing the time and cost of sign language interpreters. In this paper, we developed an automatic sign language translation application for the weather forecast translation system. We utilized Google MediaPipe's Hand Tracking, which detects key points of hand landmarks specifically, then separated frames of a video into images, extracted features of all the images to vector, and used them as inputs for LSTM model which sorts the sequences of hand's movement vector to become a word. We divided the sign language for weather forecast dataset into 80% training set, and 20% test set to obtain a reliable evaluation and avoid overfitting. During the training phase, the best model parameters were selected through 5-fold cross validation. The model accuracy is 77.4%. We then developed a sign language for weather forecast translation application using the model that would help people with disabilities be able to use it.

กิตติกรรมประกาศ

การดำเนินโครงการการแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดยใช้ RNN และ MediaPipe (Sign Language for weather forecast translation with RNN and MediaPipe) สำเร็จลุล่วงได้ด้วยความรู้และคำแนะนำของอาจารย์ ดร.วราภรณ์ วิทยานนท์ อาจารย์ที่ปรึกษาของโครงการ ที่ได้กรุณาเสียสละเวลาเพื่อให้คำปรึกษาเสนอแนะแนวทางในการแก้ปัญหา และแก้ไขข้อบกพร่อง ที่เกิดขึ้นระหว่างการทำงาน คณะผู้จัดทำกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ ที่สนับสนุนอุปกรณ์ สถานที่ ในการทำและขอขอบพระคุณ อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุก ๆ ท่าน ที่ได้กรุณาให้ความรู้ทำให้คณะผู้จัดทำสามารถนำความรู้ที่ได้มาประยุกต์ใช้ในการดำเนินโครงการในครั้งนี้ รวมทั้งให้คำปรึกษา และคำแนะนำที่เป็นประโยชน์ทำให้คณะผู้จัดทำสามารถแก้ไขโครงการให้ดีขึ้น และมีความสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณมหาวิทยาลัยศรีนครินทรวิโรฒ ที่เป็นแหล่งศึกษาหาความรู้ที่เป็นประโยชน์ในการดำเนินโครงการ และอำนวยความสะดวกสำหรับสถานที่ในการดำเนินโครงการในครั้งนี้ รวมทั้งให้ประสบการณ์ในการทำกิจกรรมต่าง ๆ เพื่อที่จะนำไปประยุกต์ใช้ในการทำงานและการดำเนินชีวิตต่อไป

ขอขอบพระคุณ คุณพ่อ คุณแม่ ครอบครัวของคณะผู้จัดทำ ที่ให้การช่วยเหลือและสนับสนุนมาโดยตลอด รวมทั้งให้ความห่วงใย และเป็นกำลังใจสำคัญในการทำโครงการเสมอมา

สุดท้าย ขอขอบพระคุณผู้มีส่วนเกี่ยวข้องทุก ๆ ท่าน ที่ทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำหวังว่าโครงการ การแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดย RNN และ MediaPipe (Sign Language for weather forecast translation with RNN and MediaPipe) จะเป็นประโยชน์ต่อผู้ที่สนใจศึกษา และสามารถเป็นที่ศึกษาสำหรับผู้สนใจจะนำไปต่อยอดในบริบทต่าง ๆ ต่อไป

นางสาวมนัสนันท์ ครูฑธา
นายสุวัชรชัย เกียรติเฉลิมพร
นายณัฐพล เลิศฤทธิพงศ์

สารบัญ

หัวข้อ	หน้า
บทคัดย่อ	ข
ABSTRACT	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญรูปภาพ	ช
สารบัญตาราง.....	ญ
บทที่ 1	1
บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ	1
1.4 กรอบแนวคิดในการดำเนินโครงการ	2
1.4.1 โมเดล Long-short term memory	2
1.4.2 โมดูลการแสดงผลของแอปพลิเคชัน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2	3
ทฤษฎี แนวคิดและงานวิจัยที่เกี่ยวข้อง	3
2.1 งานวิจัยที่เกี่ยวข้อง	3
2.1.1 Thai sign language translation using Scale Invariant Feature Transform Hidden Markov Models	3
2.1.2 Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output.....	3

2.1.3 Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove ⁴	
2.2 องค์ความรู้เกี่ยวกับภาษามือ	5
2.2.1 ประเภทของภาษามือ	5
2.2.2 ลักษณะของภาษามือ	6
2.2.3 โครงสร้างของภาษามือ	6
2.3 องค์ความรู้เกี่ยวกับ GOOGLE MEDIAPIPE	7
2.4 องค์ความรู้เกี่ยวกับ HAND LANDMARK.....	9
2.5 องค์ความรู้เกี่ยวกับ RECURRENT NEURAL NETWORK (RNN)	10
2.6 องค์ความรู้เกี่ยวกับ LONG SHORT TERM MEMORY (LSTM)	11
2.6.1 Forget Gate Layer.....	12
2.6.2 Input Gate Layer.....	13
2.6.3 Update Cell State	14
2.6.4 Output Gate Layer.....	14
2.7 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน	15
2.7.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนา.....	15
2.7.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา	16
2.7.3 ภาษาโปรแกรมที่ใช้ในการพัฒนา	17
บทที่ 3	19
วิธีการดำเนินโครงการ	19
3.1 วิธีการดำเนินโครงการ	19
3.2 แผนการดำเนินงานโครงการ.....	20
3.3 การเก็บรวบรวมข้อมูล และการนำข้อมูลมาประมวลผล.....	23
3.3.1 การเก็บรวบรวมข้อมูล	23
3.3.2 การนำข้อมูลมาประมวลผล.....	23
3.4 การพัฒนาโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด	25
3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด	27
3.6 การพัฒนาในส่วนแอปพลิเคชัน	28
บทที่ 4	29

ผลการดำเนินโครงการ	29
4.1 แสดงผลลัพธ์ของภาพต้นฉบับและLABEL	29
4.2 แสดงประสิทธิภาพของโมเดล	32
4.3 แสดงค่าความถูกต้องและไม่ถูกต้องของวิดีโอที่ทำการทดสอบ	32
4.4 แอปพลิเคชันโปรแกรมระบบแปลภาษามือเป็นคำพูด	33
บทที่ 5	35
สรุปผล อภิปรายผล และข้อเสนอแนะ	35
5.1 สรุปผลการศึกษาค้นคว้า	35
5.2 อภิปรายผล	35
5.3 ปัญหาและอุปสรรค	36
5.4 ข้อเสนอแนะ	36
บรรณานุกรม	37
ภาคผนวก	39
ภาคผนวก ก	40
ภาคผนวก ข	43
ภาคผนวก ค	48
ภาคผนวก ง	49
ภาคผนวก จ	50

สารบัญรูปภาพ

รูปที่ 2-1 รูปการทำงาน MediaPipe.....	7
รูปที่ 2-2 รูปการทำงาน Hand Tracking.....	8
รูปที่ 2-3 รูปการการตรวจจับมือ Hand Landmark.....	9
รูปที่ 2-4 หลักการ Feed สถานะภายในของโมเดล.....	10
รูปที่ 2-5 โครงสร้างของ Recurrent Neural Network.....	11
รูปที่ 2-6 รูปโครงสร้างของ Long Short Term Memory.....	12
รูปที่ 2-7 โครงสร้างของ Forget Gate Layer.....	13
รูปที่ 2-8 โครงสร้างของ Input Gate Layer.....	13
รูปที่ 2-9 โครงสร้างของ Update Cell State.....	14
รูปที่ 2-10 โครงสร้างของ Output Gate Layer.....	15
รูปที่ 2-11 สัญลักษณ์โปรแกรม Jupyter notebook.....	16
รูปที่ 2-12 สัญลักษณ์โปรแกรม PyCharm.....	17
รูปที่ 2-13 สัญลักษณ์ภาษา Python.....	17
รูปที่ 3-1 ตัวอย่างข้อมูลวิดีโอภาษามือการพยากรณ์อากาศ.....	23
รูปที่ 3-2 ตัวอย่างข้อมูลวิดีโอภาษามือการพยากรณ์อากาศที่ใช้ Google MediaPipe ในส่วน Hand Tracking	24
รูปที่ 3-3 ตัวอย่าง Text file ที่เป็นเวกเตอร์.....	24
รูปที่ 3-4 การ reshape ข้อมูลแล้วแยกชุดข้อมูลเป็นส่วน train กับ test.....	25
รูปที่ 3-5 การ labelแล้วทำเป็น Text file.....	25
รูปที่ 3-6 การ แปลง label ให้กลายเป็นตัวเลข.....	26
รูปที่ 3-7 การtrain โมเดล และ save โมเดล.....	26
รูปที่ 3-8 label.txt คำภาษามือการพยากรณ์อากาศ.....	27
รูปที่ 3-9 การนำ label.txt ที่ผ่านการเทรนมาใช้ในการ prediction.....	27
รูปที่ 3-10 การนำ model.h5 ที่ผ่านการเทรนมาใช้ในการ prediction.....	28
รูปที่ 3-11 การนำข้อมูลมาทดสอบกับโมเดล.....	28

รูปที่ 4-1 รูปการประเมินโมเดลของโมเดล Long Short Term Memory (LSTM)	32
รูปที่ 4-2 หน้าแรกของแอปพลิเคชันโปรแกรมระบบแปลภาษามือ.....	34
รูปที่ 4-3 หน้าแสดงผลการแปลภาษามือเป็นคำพูด	34
รูปที่ ก-1 หน้าแรกของเว็บไซต์ตัวโปรแกรม	40
รูปที่ ก-2 รอกการดาวน์โหลดโปรแกรมเสร็จสิ้น	40
รูปที่ ก-3 เปิดไฟล์เพื่อดำเนินการติดตั้งโปรแกรม	41
รูปที่ ก-4 หน้าต่างเมื่อเปิดตัว setup	41
รูปที่ ก-5 หน้าต่างแสดงการติดตั้งโปรแกรม.....	42
รูปที่ ก-6 หน้าต่างเสร็จสิ้นการติดตั้งโปรแกรม	42
รูปที่ ข-1 ดาวน์โหลดตัวติดตั้ง PyCharm	43
รูปที่ ข-2 รอกการดาวน์โหลดโปรแกรมเสร็จสิ้น	43
รูปที่ ข-3 หน้าต่างเมื่อเปิดตัว setup	44
รูปที่ ข-4 เลือกไฟล์เดสก์ท็อปที่จะติดตั้งโปรแกรม	44
รูปที่ ข-5 เลือกตัวเลือกการติดตั้ง.....	45
รูปที่ ข-6 เลือกไฟล์เดสก์ท็อปเริ่มต้นของการทำงาน	45
รูปที่ ข-7 หน้าต่างแสดงการติดตั้งโปรแกรม.....	46
รูปที่ ข-8 หน้าต่างเมื่อติดตั้งโปรแกรมเสร็จสิ้น	46
รูปที่ ข-9 หน้าต่างของโปรแกรม Visual Studio Code	47
รูปที่ ข-10 หน้าต่างของโปรแกรม PyCharm	47
รูปที่ ค-1 ติดตั้ง Google Media Pipe	48
รูปที่ ง-1 ติดตั้ง Bazel	49
รูปที่ จ-1 ติดตั้ง Tensorflow บนเครื่องที่ใช้.....	50

สารบัญตาราง

ตารางที่ 1 ตารางแผนการดำเนินงานตลอดโครงการ.....	21
ตารางที่ 2 ตารางภาพวิดีโอต้นฉบับและLabel	29
ตารางที่ 3 ผลลัพธ์การวัดประสิทธิภาพของโมเดลที่นำวิดีโอมาทดสอบ	33

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันมีผู้พิการจำนวนมาก ที่มีความบกพร่องทางการได้ยินหรือทางการพูด เช่น หูหนวก หูตึง หรือไม่สามารถสื่อสารได้ด้วยเสียง โดยปัญหาดังกล่าวทำให้เกิดอุปสรรคด้านการสื่อสารระหว่างคนปกติทั่วไปกับ ผู้พิการทางการได้ยิน วิธีเดียวสำหรับพวกเขาในการสื่อสารก็คือผ่านภาษามือ

ภาษามือจึงเป็นตัวกลางในการสื่อสารระหว่างคนปกติกับผู้พิการทางการได้ยิน ซึ่งคนปกติส่วนใหญ่ไม่สามารถ เข้าใจความหมายของภาษามือได้ การใช้ล่ามภาษามือจึงเป็นตัวช่วยที่จำเป็น แต่การใช้บริการในแต่ละครั้งนั้นมีค่าใช้จ่ายสูง หากมีวิธีที่คนปกติทั่วไปสามารถเข้าใจความหมายที่ผู้พิการสื่อสาร ก็จะช่วยลดค่าใช้จ่าย และปัญหาในการสื่อสารได้

ทีมผู้วิจัยจึงได้สร้างระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดย RNN และ MediaPipe (Sign Language for weather forecast translation with RNN and MediaPipe) เพื่อให้ ผู้ใช้ เข้าใจความหมายของผู้พิการได้ถูกต้องและแม่นยำ ลดปัญหาในการสื่อสาร เพิ่มความสะดวกสบายให้แก่ผู้ใช้ และสามารถนำไปต่อยอดในอนาคตต่อไปได้

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อพัฒนาระบบที่สามารถแปลภาษามือที่เกี่ยวข้องกับการพยากรณ์อากาศเป็นคำพูด
- 1.2.2 เพื่อศึกษา MediaPipe ของการทำงาน Hand Tracking
- 1.2.3 เพื่อศึกษา Long Short Term Memory (LSTM) ในการวิเคราะห์ภาพเป็นคำพูด
- 1.2.4 เพื่อทดสอบประสิทธิภาพ และความแม่นยำของ การทำงานของ Long Short Term Memory (LSTM)

1.3 ขอบเขตของโครงการ

- 1.3.1 ใช้ข้อมูลในการทำภาษามือการพยากรณ์อากาศจากวิดีโอสมาร์ตโฟนและ signingsavvy.com เป็นข้อมูล นำมาทดสอบและประมวลผลความแม่นยำ
- 1.3.2 ใช้วิดีโอที่มีความเร็ว 30 fps แล้วตั้งค่าให้เลือกทุกๆ 2 เฟรมเพื่อมาประมวลผล

1.3.3 ใช้ Google MediaPipe ในส่วน Hand Tracking มาใช้ตรวจจับหา keypoint และสกัด feature ออกมาเป็นเวกเตอร์

1.3.4 ใช้โมเดล Long-short term memory เพื่อทดสอบประสิทธิภาพและความแม่นยำของการทำงาน

1.3.5 แสดงผลทดสอบประสิทธิภาพ ความแม่นยำของโมเดล Long-short term memory

1.4 กรอบแนวคิดในการดำเนินโครงการ

พัฒนาระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูดโดย RNN และ MediaPipe (Sign Language for weather forecast translation with RNN and MediaPipe) จะแบ่งเป็น 1 โมเดล และ แอปพลิเคชันจะแบ่งเป็น 1 โมดูล ดังนี้

1.4.1 โมเดล Long-short term memory

- เรียงลำดับการเคลื่อนไหวของมือจากทุกๆเวกเตอร์ให้กลายเป็นคำหนึ่งคำ
- ทดสอบประสิทธิภาพ ความแม่นยำของโมเดล

1.4.2 โมดูลการแสดงผลของแอปพลิเคชัน

- แสดงวิดีโอภาษามือการพยากรณ์อากาศ ที่ใส่เพื่อประมวลผลผลลัพธ์
- แสดงผลลัพธ์เป็นคำพยากรณ์อากาศ จากการประมวลผลของวิดีโอ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เพื่อพัฒนาโมเดล Long-short term memory ให้มีความแม่นยำมากขึ้น

1.5.2 เพื่อพัฒนาระบบแปลภาษามือการพยากรณ์อากาศที่สามารถแปลภาษามือที่ครอบคลุมทั้งคำ

1.5.3 ระบบแปลภาษามือการพยากรณ์อากาศ จะช่วยลดปัญหาในการเข้าใจภาษามือของผู้พิการทางการได้ยินที่เกี่ยวข้องกับพยากรณ์อากาศให้คนปกติเข้าใจได้

1.5.4 ระบบแปลภาษามือการพยากรณ์อากาศ จะช่วยให้คนปกติที่ไม่มีความรู้ทางภาษามือสามารถเข้าใจความหมายภาษามือการ พยากรณ์อากาศที่ผู้พิการทางการได้ยินสื่อสารได้

บทที่ 2

ทฤษฎี แนวคิดและงานวิจัยที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

2.1.1 Thai sign language translation using Scale Invariant Feature

Transform Hidden Markov Models [11]

ผู้แต่ง : Sansanee Auephanwiriyaikul, Suwannee Phitakwinai, Wattanapong Suttapak ,
Phonkrit Chanda , Nipon Theera-Umpon

บทความนี้พัฒนาระบบแปลภาษามือไทยอัตโนมัติที่สามารถแปลภาษามือที่ไม่ใช่การสะกดนิ้วภาษามือใช้ Scale Invariant Feature Transform (SIFT) เพื่อให้ตรงกับเฟรมทดสอบที่มีสัญลักษณ์สังเกตจากตัวบ่งชี้จุดสำคัญ (keypoint descriptors) ที่รวบรวมในไลบรารี keypoint ถูกคำนวณจากหลายๆ keyframes คือ บันทึกในช่วงเวลาที่แตกต่างกันของวันเป็นเวลาหลายวัน จากนั้นจะใช้โมเดล Hidden Markov Models (HMMs) เพื่อแปลลำดับการสังเกตคำพูด การทดลองให้ผู้ทดลองสวมเสื้อเชิ้ตสีดำแขนยาวและยืนหน้าของพื้นหลังสีเข้ม ผลการทดลองแสดงให้เห็นว่ามีความแม่นยำที่สูง ผลลัพธ์ที่ได้อยู่ในช่วง 86-95

จากบทความนี้โมเดล Hidden Markov Models ยังไม่ค่อยเป็นที่นิยมและยังไม่มีคำแนะนำมากนักพอ เพราะการสังเกตจากตัวบ่งชี้จุดสำคัญ หรือ keypoint descriptors นั้นมีความคลาดเคลื่อนอยู่พอสมควร เพราะการใช้ภาษามือเป็นท่าที่มีความเคลื่อนไหว การจับด้วยตัวบ่งชี้จุดสำคัญอาจไม่ครอบคลุมในแต่ละท่าภาษามือ

2.1.2 Sign Language Recognition Application Systems for Deaf-Mute

People: A Review Based on Input-Process-Output [9]

ผู้แต่ง : Suharjito , RickyAnderson , FannyWiryana , Meita ChandraAriesta , Gede PutraKusuma

บทความนี้นักวิจัยได้พัฒนาระบบการจดจำภาษามือจากการจำแนกสัญญาณคงที่และตัวอักษรไปจนถึงระบบที่สามารถจำแนกการเคลื่อนไหวแบบไดนามิกที่มีลำดับภาพอย่างต่อเนื่อง โดยใช้ฐานข้อมูลคำศัพท์ขนาดเล็กและฐานข้อมูลที่สร้างขึ้นเองซึ่งใช้วิธี K-mean เป็นเครื่องจำแนกรูปร่างระหว่างมือซ้ายและมือขวาพร้อมกับการกำหนดทิศทาง โดยใช้ Multi Stream Hidden Markov Model (MSHMM) , Light-HMM , TD-HMM และ SVM เพื่อวิเคราะห์การเคลื่อนไหว และนำ CNN 2Dมาใช้ในการจำเสียงและภาพ อีกทั้งยังใช้ ANN รวมกับ Elliptical Fourier Descriptor (EFD) เป็นตัวอธิบายภาพแนวโค้ง 2D ซึ่งผลการทดลองพบว่าวิธี SVM มีความถูกต้องมากถึง 98.9 % และนักวิจัยพบว่า 13 ในการสร้างฐานข้อมูลขนาดใหญ่สำหรับระบบจดจำภาษามือนั้นยังไม่พร้อมใช้งานสำหรับบางประเทศ และวิธีการจำแนกประเภทนั้นจะแตกต่างกัน ตามการใช้ความคิดและข้อจำกัดสำหรับไวยากรณ์และวิธีการนำเสนอของแต่ละคำ

จากบทความนี้โมเดล SVM มีความถูกต้องและแม่นยำที่สุดในการวิเคราะห์การเคลื่อนไหว ที่จดจำภาษามือจากการจำแนกสัญญาณคงที่และตัวอักษรไปจนถึงระบบที่สามารถจำแนกการเคลื่อนไหวแบบไดนามิกที่มีลำดับภาพอย่างต่อเนื่องได้ดีที่สุด แต่การนำโมเดล SVM มาใช้ จะต้องคำนึงถึงข้อจำกัดสำหรับไวยากรณ์และวิธีการนำเสนอของแต่ละคำ

2.1.3 Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove [12]

ผู้แต่ง : Nayan M. Kakoty, Manalee Dev Sharma

บทความนี้นำเสนอระบบการรู้จำการสะกดนิ้วโดยเน้น ภาษามือการสะกดคำภาษาไทยที่ได้จาก computer vision โดยใช้ SVM ในการศึกษา โดยนี้ได้ดึงเอา Global และ Local feature จากภาพนิ้วมือ เพื่อพัฒนาระบบจดจำ ตัวอักษรภาษาไทย 15 ตัวได้ถูกรวบรวมจากผู้ใช้งานมือ 5 คนรวม 375 ภาพเพื่อนำมาใช้ฝึกระบบโดยใช้เทคนิค SVM (เชิงเส้น พหุนาม RBF และ sigmoid) ระบบการรู้จำการสะกดนิ้วมือนี้นี้ไม่จำเป็นต้องสวมใส่อุปกรณ์ใดๆ เพราะใช้เป็นรูปภาพหรือวิดีโอจากกล้องถ่ายภาพ เว็บแคม และ Microsoft Kinect ภาพที่ได้รับมาจะเป็นภาพสี RGB ขนาด 280 x 288 จากกล้องถ่ายภาพ คนที่ทำภาษามือต้องสวมใส่เสื้อผ้าสีดำ กางเกงขาวาวและยืนด้านหน้าพื้นหลังสีน้ำเงิน การแบ่งส่วนของมือนั้นจะถูกแบ่งเป็นขั้นตอนดังนี้

1. แปลงภาพสี RGB ให้กลายเป็นภาพสี HSV
2. แปลงภาพสี HSV ให้กลายเป็นภาพ gray scale
3. Filter ภาพ gray scale ผ่าน Gaussian filter
4. แปลงภาพ gray scale ที่ถูก filter แล้วให้เป็นภาพขาว-ดำ
5. กำหนด bounding box จากความสูงและกว้างให้เท่ากับขนาดของมือ
6. ตัดส่วนที่อยู่ใน bounding box ออกมา
7. ปรับขนาดของรูปภาพที่ตัดมาให้เท่ากับรูปภาพของแต่ละตัวอักษร

ซึ่งเมื่อนำมาทดสอบกับโมเดล svm แล้วจึงได้ความแม่นยำถึง 96.8% จากชุดข้อมูล AmericanSL บทความนี้ ได้มีการนำ svm มาใช้ซึ่งผู้เขียนควรอธิบายว่าทำไมจึงนำ svm มาใช้ให้มากกว่านี้ และ ข้อมูลที่นำมาใช้นั้นถ้าถูกเพิ่มเข้าไปอีกก็จะช่วยเพิ่มในเรื่องของความแม่นยำได้อีก

2.2 องค์ความรู้เกี่ยวกับภาษามือ

ภาษามือ (sign language) [1] เป็นอวัจนภาษาอย่างหนึ่ง ที่ประกอบด้วย การสื่อสารด้วยมือ การสื่อสารด้วยร่างกาย และการใช้ริมฝีปากในการสื่อความหมายแทนการใช้เสียงพูด การสื่อสารจะใช้ลักษณะของมือที่ทำเป็นสัญลักษณ์ การเคลื่อนไหวมือ แขนและร่างกาย และการแสดงความรู้สึกทางใบหน้าเพื่อช่วยในการสื่อสารความคิดของผู้สื่อ ภาษานี้มีสัญลักษณ์ส่วนใหญ่มักใช้ในกลุ่มผู้พิการทางหู ซึ่งรวมทั้งผู้พิการทางหูเอง ใช้สื่อสารซึ่งกันและกัน ที่ใช้มือ สีหน้า และกิริยาท่าทาง ประกอบในการสื่อความหมายและถ่ายทอดอารมณ์แทนการพูด

2.2.1 ประเภทของภาษามือ

1. ภาษามือธรรมชาติ (Sign language) คือ ภาษาท่าทางที่คนหูหนวกเป็นผู้สร้างขึ้นและใช้ร่วมกัน ในแต่ละชุมชนหรือในแต่ละชาติเช่น American Sign language British Sign language Swedish Sign language เป็นต้น ซึ่งส่วนมากเป็นท่าเลียนแบบธรรมชาติที่จะช่วยคนหูหนวกให้มีพัฒนาการในภาษาประจำชาติ เท่าเทียมกับคนปกติ
2. ภาษามือประดิษฐ์ (Signed) คือ ภาษามือที่ครูผู้ปกครอง หรือญาติมิตรของคนหูหนวกคิดขึ้น แทนภาษาพูดและภาษาเขียนประจำชาติเพื่อให้มีคำใช้ให้เพียงพอในการศึกษาและการสื่อความหมาย โดยเฉพาะเรื่องนามธรรม ภาษามือที่ประดิษฐ์ขึ้นนี้ บางทีเรียกว่า ภาษามือในห้องเรียน หรือภาษามือที่ใช้ ใน

การศึกษาซึ่งเป็นภาษาที่ทำคำแต่ละคำ ตามไวยากรณ์ภาษาพูด หรือภาษาเขียนของคนปกติภาษามือ ประดิษฐ์มักจะนำแบบสะกดนิ้วมือ (Finger Spelling) มาผสม

2.2.2 ลักษณะของภาษามือ

ภาษามือเป็นภาษาท่าทางซึ่งมีการเคลื่อนไหวของมือเป็นหลักและใช้กริยาอาการของหน้าตาและร่างกายส่วนหนึ่ง เป็นส่วนประกอบช่วยให้เกิดความเข้าใจ ท่าภาษามือที่คนหูหนวกยอมรับจะต้องเป็นท่าที่ ทำง่าย สะดวก รวดเร็ว มีความหมายใกล้เคียงธรรมชาติและเหมาะสมกับ หลักสรีระศาสตร์ ท่าภาษามือควรทำ อย่างมีจังหวะ คือมีการเว้น ไม่ทำท่าทางเร็วจนเกินไป และให้อยู่ในรัศมีที่สายตาสามารถมองเห็นได้ ชัดเจน โดยทั่วไปจะทำ ท่ามือเคลื่อนไหวตั้งแต่ระดับเอวขึ้นมาถึงส่วนบนของร่างกาย

2.2.3 โครงสร้างของภาษามือ

โครงสร้างของภาษามือ [4] ในภาษาไทย ภาษาอังกฤษ หรือภาษาใดก็ตาม จะมีโครงสร้างของภาษาเช่น มีคำ วลี ประโยคนำมาเรียงกันเป็นข้อความที่มีความหมาย ภาษามือก็เช่นกัน การแสดงออกและการเคลื่อนไหว ล้วนมีความหมายทั้งสิ้น ประกอบด้วย

1. ท่ามือ (The handshape) คือการทำมือเป็นท่าต่าง ๆ เช่น กำมือ,แบมือ,กางนิ้ว,รวมนิ้ว,จีบ นิ้ว แต่ละท่ามีความหมายต่างกันออกไป
2. ตำแหน่งของมือ (The position of the hand) คือการวางมือในตำแหน่งต่างๆ ตำแหน่งของมือจะให้ความหมาย แตกต่างกันถึงแม้ว่าท่ามือจะเป็นท่าเดียวกัน เช่น ใช้นิ้วชี้ชี้ที่หน้าอก หมายถึง “ฉัน” ถ้าชี้ที่ขมับ หมายถึง “รู้” เป็นต้น ตำแหน่งที่ทำท่ามือควรจะอยู่ในรัศมีที่สายตาสามารถมองเห็นได้ง่าย และใกล้ใบหน้าไม่ควรต่ำกว่าระดับเอว ภาษามือที่แสดงความรู้สึกต่าง ๆ มักจะแสดงท่ามือในตำแหน่งใกล้เคียงกับความหมายของคำนั้น ๆ
3. การเคลื่อนไหวของมือ (The movement of the hand) คือการย้ายมือเปลี่ยนมือ ไปในทิศทางหรือรูปแบบที่ต้องการ ท่ามืออย่าง เดียวกัน แต่เคลื่อนไหวไปในทิศทางต่างกัน ความหมายจะแตกต่างกัน เช่น มือทั้งสองตั้งขึ้น หัวแม่มือชิดกันแล้วเลื่อนออกห่างคือ “เปิด” แต่ถ้าต้องมือห่างกัน พอสมควรแล้วเลื่อนให้หัวแม่มือชิดกันหมายถึง “ปิด” การ

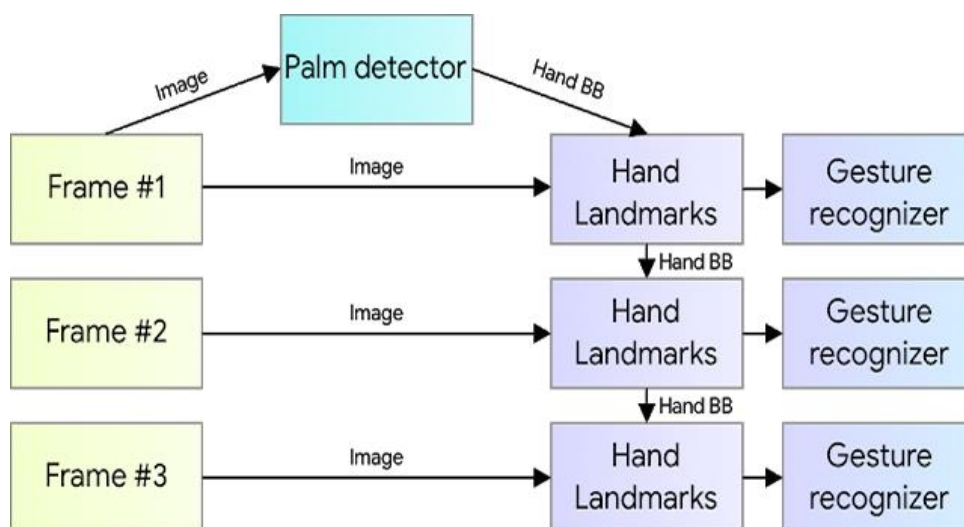
เคลื่อนไหวมือ มักจะใช้มือที่ถนัด เป็นมือที่เคลื่อนไหวไม่บังคับว่าจะต้องมือซ้ายหรือมือขวาเสมอไป

- ทิศทางของฝ่ามือ (The orientation of the palms in relationship to the body or to each other) คือ การหันฝ่ามือไปในทิศทางต่างๆ ทิศทางของฝ่ามือเป็นส่วนสำคัญ อย่างหนึ่ง ซึ่งทำให้ท่ามือมีความหมายต่างกัน ท่ามือท่าเดียวกันตำแหน่งที่เดียวกัน แต่ทิศทางของฝ่ามือต่างกัน ความหมายจะต่างกัน

2.3 องค์ความรู้เกี่ยวกับ Google MediaPipe

Google MediaPipe [6] เป็น Open Source สำหรับ AI ที่มีความสามารถในการรับรู้ความเคลื่อนไหวและรูปร่างของมือในแบบเรียลไทม์ ประสิทธิภาพของระบบเรียนรู้ในการจับภาพมือและนิ้วแบบเรียลไทม์ด้วยความแม่นยำสูง โดยใช้เพียงแค่กล้องสมาร์ทโฟนเท่านั้น สำหรับการใช้อุปกรณ์คอมพิวเตอร์วิทัศน์ในการจับภาพมือนั้นเป็นเรื่องที่ท้าทาย เพราะมือไม่มีรูปแบบแสดงถึงความแตกต่างที่ชัดเจน อีกทั้งการเคลื่อนไหวของมือนั้นมักจะเร็วละเอียดอ่อน ซึ่งไม่ใช่สิ่งที่คอมพิวเตอร์จะจับภาพได้ตึกในแบบเรียลไทม์แม้จะใช้กล้องหลายตัวช่วยจับการเคลื่อนไหวก็ตาม

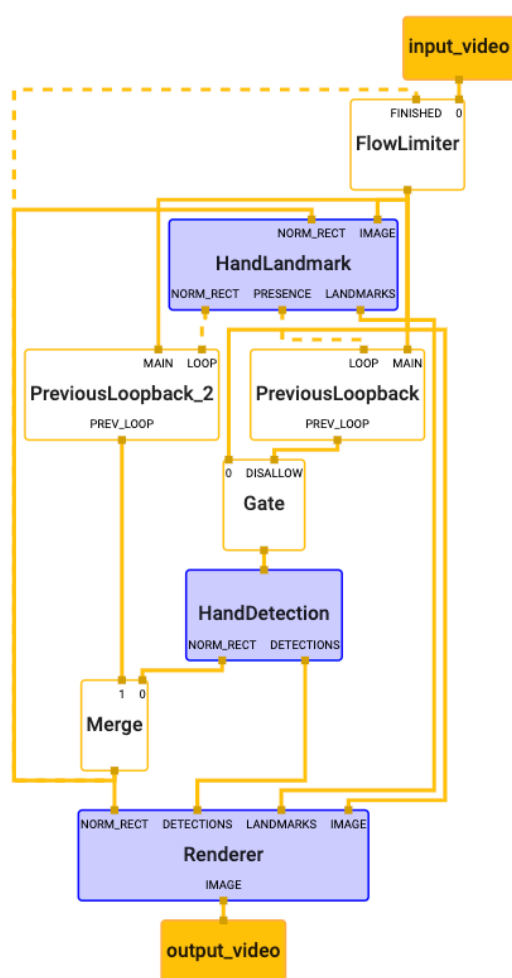
MediaPipe ถือเป็นแพลตฟอร์มที่สามารถสร้างท่อส่งข้อมูลที่เกี่ยวข้องกับการรับรู้ไปยังรูปแบบไฟล์ต่างๆ ทั้งเสียงและวิดีโอ สำหรับความสามารถที่เป็นจุดเด่นแท้จริงของ MediaPipe คือความสามารถในการระลึกระจกจำสัญญาณมือทั่วไปอย่าง การยกนิ้ว โอเค ยอดเยี่ยม



รูปที่ 2-1 รูปการทำงาน MediaPipe

ที่มา : [ออนไลน์]<https://65blogs.com/technology/google-mediapipe>

จุดประสงค์ของงานวิจัยคือลดปริมาณข้อมูลที่อัลกอริธึมใช้คำนวณ ยิ่งข้อมูลน้อยลงนั้นย่อมหมายถึงการประมวลผลที่เร็วขึ้น จึงปรับให้ระบบหาฝ่ามือแทนซึ่งเป็นส่วนที่เด่นชัดที่สุดของมือ เมื่อระบบตรวจพบฝ่ามือแล้ว นิ้วต่างๆ ก็จะถูกวิเคราะห์แยกต่างหาก อัลกอริธึมจะศึกษาภาพและกำหนดพิกัด 21 จุดที่เชื่อมโยงกับข้อนิ้ว ปลายนิ้ว รวมถึงระยะห่างของแต่ละนิ้วโดยอาศัยขนาดและองศาของฝ่ามือ เมื่อตำแหน่งท่าทางของมือถูกระบุได้แล้ว ระบบจะเปรียบเทียบรูปแบบมือกับฐานข้อมูลท่าทางตั้งแต่สัญลักษณ์ตัวอักษรและตัวเลขแบบภาษามือ ผลลัพธ์คืออัลกอริธึมทำงานได้เร็วและแม่นยำ

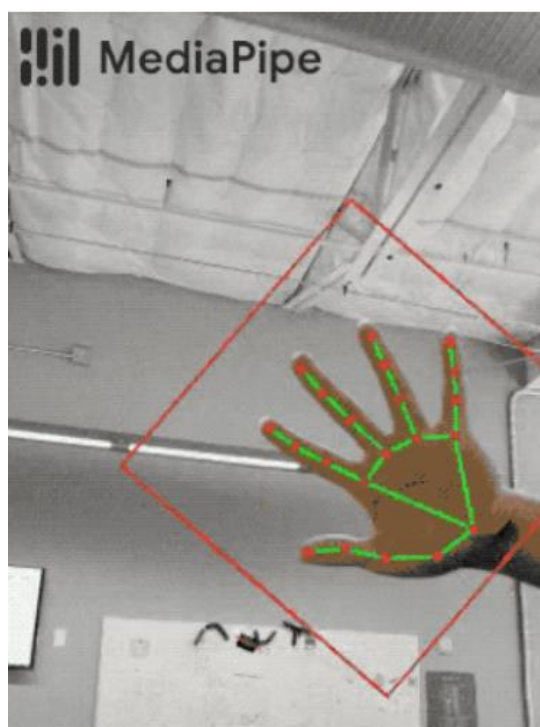


รูปที่ 2-2 รูปการทำงาน Hand Tracking

ที่มา : [ออนไลน์] <https://mediapipe.readthedocs.io/en/latest/>

2.4 องค์ความรู้เกี่ยวกับ Hand Landmark

Hand Landmark [15] คือความสามารถในการรับรู้รูปร่างและการเคลื่อนไหวของมือ ตัวอย่างเช่น สามารถทำเข้าใจภาษามือและการควบคุมท่าทางมือ โดยจะใช้ MediaPipe เป็นกรอบสำหรับส่งไปประมวลผลที่แตกต่างกันเช่นวิดีโอและเสียง วิธีนี้สามารถติดตามมือและมีความแม่นยำสูงโดยใช้ Machine Learning เพื่อหา 21 3D keypoints จากเฟรมเดียว



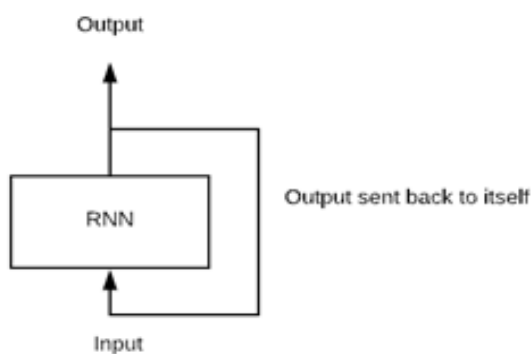
รูปที่ 2-3 รูปการการตรวจจับมือ Hand Landmark

ที่มา : [ออนไลน์] <https://www.slideshare.net/JiHyunKim204>

จุดสีแดงแสดงถึงจุดสังเกตที่มีการเปลี่ยนแปลง และเส้นสีเขียวเป็นเพียงการเชื่อมต่อระหว่างคู่จุดสังเกตที่เลือกไว้ กล้องสีแดงหมายถึงกรอบสี่เหลี่ยมที่ครอบคลุมทั้งมือซึ่งได้มาจากการตรวจจับมือ หรือจากจุดสังเกตรอบๆมือ โดยใช้แบบจำลอง Machine Learning การแปลพื้นที่จะดำเนินการภายในกรอบสี่เหลี่ยมเพื่อประสิทธิภาพและความแม่นยำในการคำนวณ

2.5 องค์ความรู้เกี่ยวกับ Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) [6] คือ Artificial Neural Network แบบหนึ่งที่ย่อออกมาแก้ปัญหาสำหรับงานที่ข้อมูลมีลำดับ Sequence โดยใช้หลักการ Feed สถานะภายในของโมเดล กลับมาเป็น Input ใหม่ คู่กับ Input ปกติ เรียกว่า Hidden State, Internal State, Memory ช่วยให้โมเดลรู้จำ Pattern ของลำดับ Input Sequence หรือคือ neural network ที่เชื่อมต่อกันหลายๆอัน และยังสามารถต่อกันเป็นวงวน (loop) ได้ ซึ่ง RNN มีการพัฒนาต่อยอดไปอีกหลาย Variation ที่เป็นที่ยอมรับได้แก่ Long Short Term Memory (LSTM) และ Gated Recurrent Unit (GRU)



รูปที่ 2-4 หลักการ Feed สถานะภายในของโมเดล

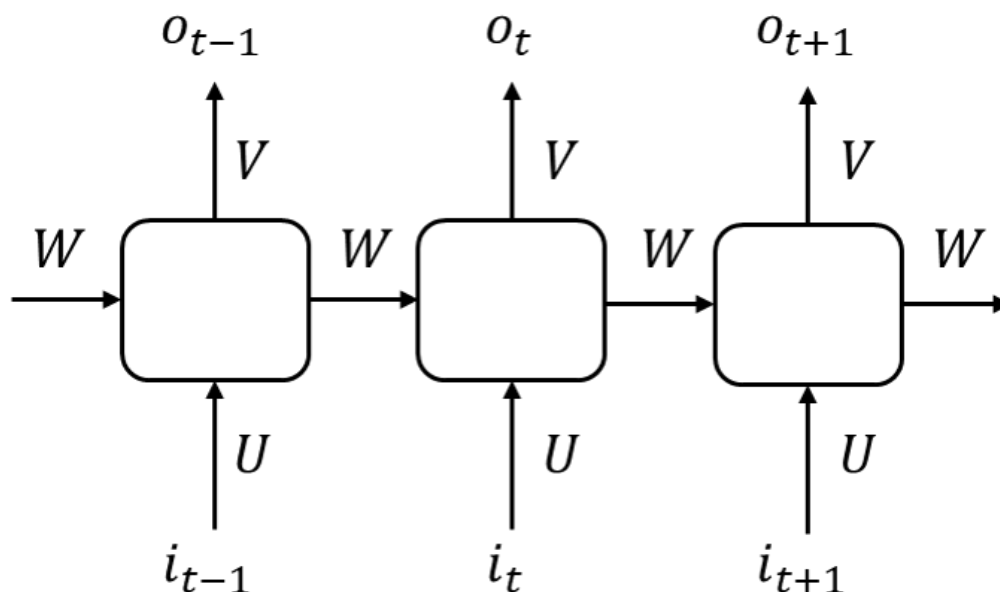
ที่มา : [ออนไลน์] <https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>

อัลกอริทึมแบบโครงข่ายประสาทเทียมมีอัลกอริทึมสำหรับการสร้างโมเดล ที่เหมาะกับข้อมูลที่เป็นอนุกรมเวลา เรียกว่า Recurrent neural network (RNN) ซึ่งจะมีการเป็นข้อมูลสถานะไว้ใน Hidden State โดยมีการนำ Hidden State ก่อนหน้ามาใช้ในการคำนวณ Hidden State ปัจจุบัน และใช้ Hidden State ปัจจุบันในการคำนวณของข้อมูลในช่วงเวลาถัดไป โดยมีการคำนวณตามสมการ

$$h_t = \sigma(x_t W + h_{t-1} U)$$

$$o_t = \sigma(h_t V)$$

โดยที่ t คือช่วงเวลา, h คือ hidden state, x หมายถึงข้อมูล input, o คือค่า output, σ คือ activation function และ W , U , V คือ weight matrix สำหรับคำนวณ input, hidden state ก่อนหน้า และ hidden state ปัจจุบัน ตามลำดับ และมีโครงสร้างตามรูปที่ 2-5



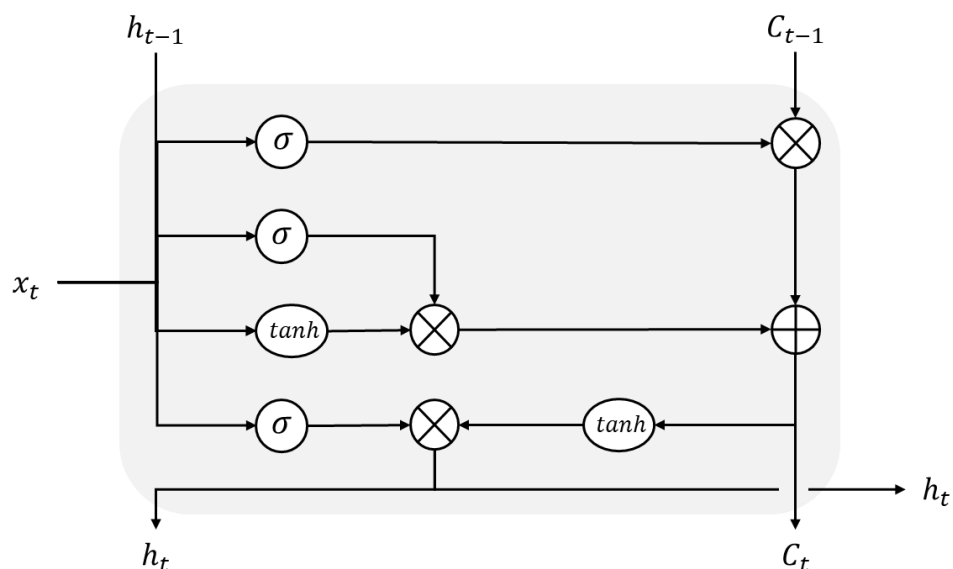
รูปที่ 2- 5 โครงสร้างของ Recurrent Neural Network

ที่มา : [ออนไลน์] <https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>

2.6 องค์ความรู้เกี่ยวกับ Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) [5] คือ โมเดล deep learning ที่สร้างขึ้นมาจำลองรูปแบบความจำของคน (memory) ที่มีความจุของความทรงจำอยู่จำกัด เมื่อมีเหตุการณ์ใหม่ๆ เข้ามาในความทรงจำสมองจะเลือกที่จะรับ หรือไม่รับเหตุการณ์ใหม่เข้ามาในความทรงจำ ตามความสำคัญของเหตุการณ์ และเมื่อสมองเลือกที่จะรับเหตุการณ์ใหม่ๆ ที่มีความสำคัญเข้ามาเก็บไว้ในระบบความทรงจำแล้ว (memorize) ก็จำเป็นจะต้องมีเหตุการณ์บางอย่างในอดีตที่ถูกลืมไป (forget) หรือเรียกว่าเป็น RNNs ที่มี memory cell state หรือเซลล์จดจำ โดยมีหลากหลาย activation function ที่เข้ามาช่วยเรื่องการทำงานของ LSTMs

ด้วยโครงสร้างของ LSTM จึงทำให้โมเดลดังกล่าว เหมาะสมกับการข้อมูลที่มีลักษณะเป็นลำดับเหตุการณ์ เช่น การทำ sentiment analysis หรือ intent finding ในข้อมูลประเภท text และ การทำ forecast ในข้อมูลประเภท time series



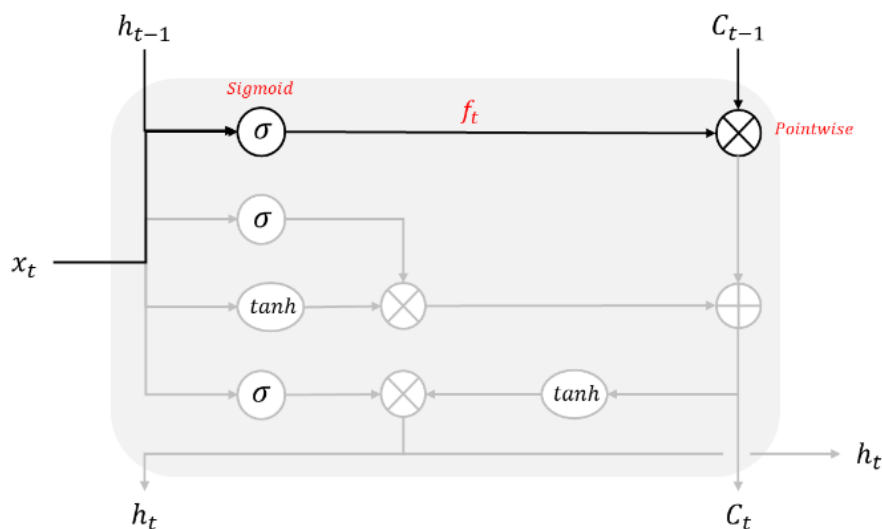
รูปที่ 2-6 รูปโครงสร้างของ Long Short Term Memory

ที่มา : [ออนไลน์] <https://medium.com/@tongkornkitt/ml-lstms>

2.6.1 Forget Gate Layer

Forget gate คือประตูที่จะใช้เป็นตัวกำหนดข้อมูลที่เข้ามา สมควรที่จะได้ออกไปทั้งหมดหรือไม่สมควรที่จะมีข้อมูลใดๆ หลุดรอดออกไปได้เลย ซึ่งข้อมูลที่จำเป็นต้องใช้ในการตัดสินใจว่าจะเก็บข้อมูลนี้ไว้หรือไม่นั้นก็มาจากข้อมูลขาเข้า ณ node นั้น ๆ รวมกับผลลัพธ์ที่ถูกคำนวณมาจาก node ก่อนหน้า ผ่านเข้า sigmoid function และนำไปคูณกับสถานะของ node ก่อนหน้าโดยใช้สูตร Pointwise Multiplication ในการคำนวณ

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$$



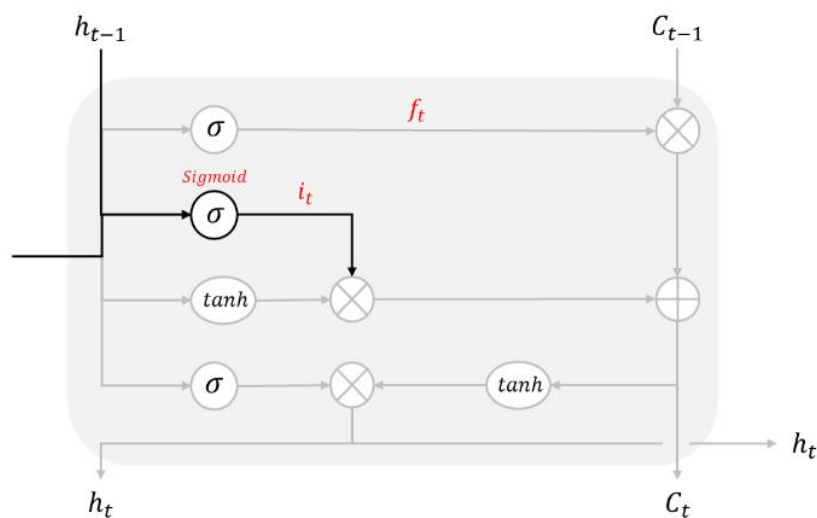
รูปที่ 2-7 โครงสร้างของ Forget Gate Layer

ที่มา : [ออนไลน์] <https://medium.com/@tongkornkitt/ml-lstms>

2.6.2 Input Gate Layer

Input Gate คือประตูที่ใช้เพื่อเปิดรับข้อมูลที่เข้ามาใหม่เพื่อให้บันทึกลงไปในแต่ละ node หรือเรียกว่าการ ‘write’ ข้อมูลลงไป

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$



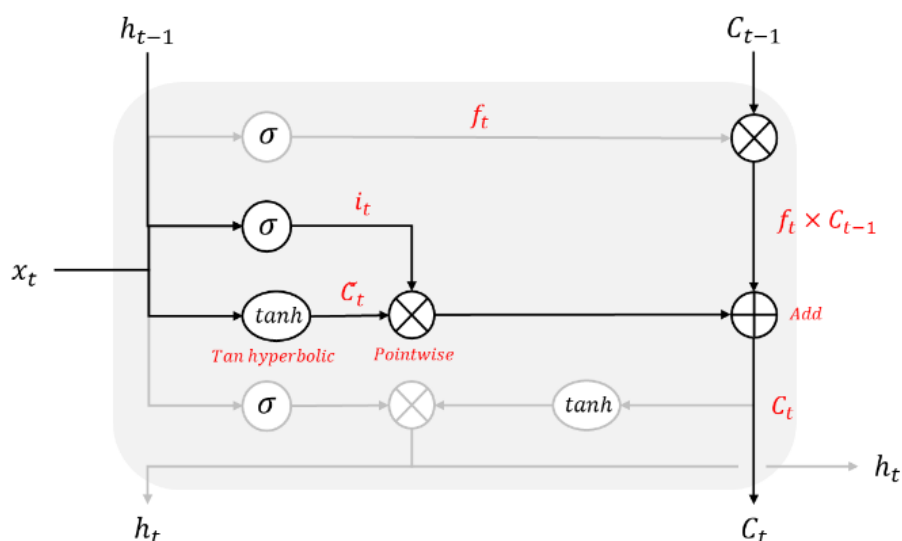
รูปที่ 2- 8 โครงสร้างของ Input Gate Layer

ที่มา : [ออนไลน์] <https://medium.com/@tongkornkitt/ml-lstms>

2.6.3 Update Cell State

Update Cell State คือ อยู่ช่วงระหว่างที่ input gate เปิดประตูเพื่อรับข้อมูลมาจดจำใน node ตัวเอง ก็จะมีอีก node หนึ่งที่จะถูกผลักเข้าสู่กระบวนการเปลี่ยนแปลงโดยฟังก์ชัน tan hyperbolic เพื่อให้ได้อีกหนึ่งข้อมูลที่มีความใกล้เคียงกับข้อมูลขาออกจาก node นี้ ผลลัพธ์ที่ได้จะถูกนำไปตัดแปลงอีกทีตามสถานะที่ถูกคำนวณมาก่อนหน้าเพื่อส่งเป็นข้อมูลขาออกต่อไป

$$C_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c)$$



รูปที่ 2-9 โครงสร้างของ Update Cell State

ที่มา : [ออนไลน์] <https://medium.com/@tongkornkitt/ml-lstms>

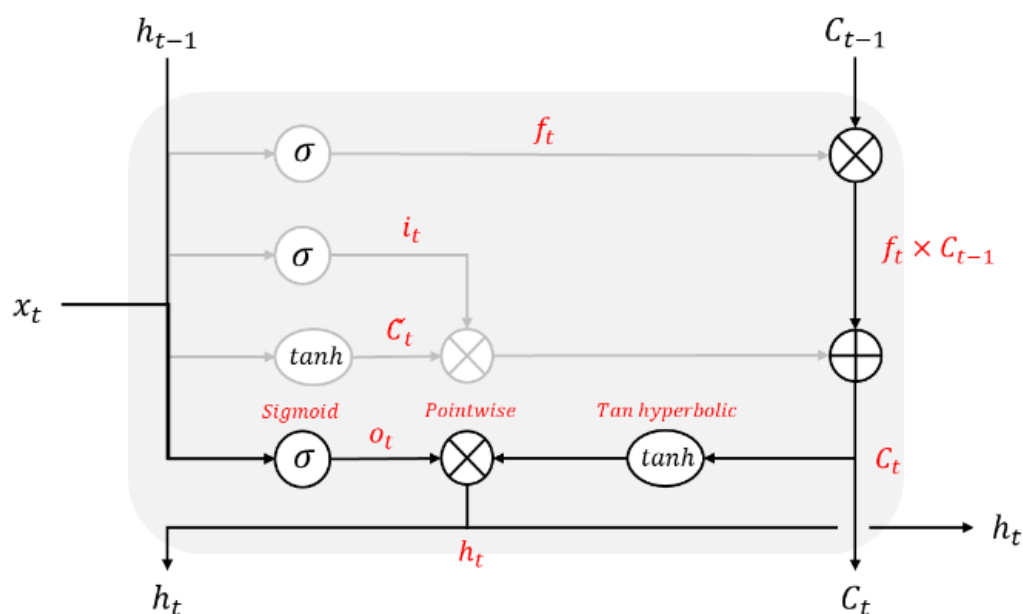
2.6.4 Output Gate Layer

Output Gate คือประตูที่ใช้บอกว่า ข้อมูลเตรียมพร้อมที่จะเป็นข้อมูลขาออกแล้ว ซึ่งข้อมูลขาออกของแต่ละ node ใน LSTMs นั้นมีด้วยกันทั้งหมด 3 แห่ง แต่มีเพียง 2 ค่า คือ

- 1.ค่าที่เกิดจากคำนวณโดยมีการ update สถานะแล้ว (ซึ่งถูกคำนวณจากขั้นตอน Update cell state) ค่านี้จะถูกส่งต่อทันทีโดยไม่ผ่านฟังก์ชันใดๆ ส่งไปให้ node ถัดไป

2.ค่าของข้อมูลขาเข้าที่ถูกตัดแปลงผ่านการคำนวณใน output gate ค่านี้จะถูกส่งต่อไปเป็นข้อมูลขาเข้าของ node ถัดไป และยังถูกส่งไปเป็นผลลัพธ์ของ node นั้นๆ

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$



รูปที่ 2-10 โครงสร้างของ Output Gate Layer

ที่มา : [ออนไลน์] <https://medium.com/@tongkornkitt/ml-lstms>

2.7 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

ในการทำโครงการได้ดำเนินการโดยใช้เครื่องมือที่เป็นทั้งฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

2.7.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1.คอมพิวเตอร์ที่ใช้ในการดำเนินงาน

CPU : Intel Core i5 CPU 3.20 GHz

Memory RAM : 8 GB

System : Windows 10 64-bits

2.กล้องโทรศัพท์มือถือสำหรับถ่ายวิดีโอ

ระบบปฏิบัติการ : IOS 13.2.3

Memory RAM : 250 GB

Resolution: 1080 HD 60 fps

2.7.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1.Jupyter notebook



รูปที่ 2- 11 สัญลักษณ์โปรแกรม Jupyter notebook

ที่มา : [ออนไลน์]<https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Jupyter>

[logo.svg/250px-Jupyter_logo.svg.png](#)

Jupyter notebook [3] เป็น Interactive Computing Environment ที่ให้ผู้ใช้ สามารถป้อนคำสั่งแล้วให้ประมวลผลคำสั่งนั้นๆ พร้อมแสดงผลลัพธ์ต่อท้ายชุดคำสั่ง ตัว Notebook สามารถเก็บCodeและตัวMarkdown ที่ช่วยให้แสดงข้อมูลในแบบต่างๆได้เช่น ต้องการแสดงเป็น Heading แบบต่างๆ

2.PyCharm



รูปที่ 2-12 สัญลักษณ์โปรแกรม PyCharm

ที่มา : [ออนไลน์] <https://www.jetbrains.com/pycharm/download>

PyCharm (ไพชาร์ม) [13] คือ โปรแกรมที่ใช้หัดเขียนภาษาPython เป็นโปรแกรมที่ใช้งานได้ง่าย และสามารถใช้งานได้ฟรี เหมาะสำหรับผู้ที่ต้องการจะฝึกเขียนภาษาไพทอน และนอกจากนี้ยังรองรับระบบปฏิบัติการ Windows Linux MacOS

2.7.3 ภาษาโปรแกรมที่ใช้ในการพัฒนา

1. Python



รูปที่ 2-13 สัญลักษณ์ภาษา Python

ที่มา : [ออนไลน์] <https://pluralsight.imgix.net/paths/python-7be70baaac.png>

Python [14] คือ ภาษาโปรแกรมขั้นสูงชนิดหนึ่ง ถูกออกแบบให้ code สามารถ อ่านได้ง่ายโดยใช้ “ช่องไฟ” หรือ whitespace เป็นตัวแบ่งว่า code แต่ละบรรทัดอยู่ภายใต้ block ไต แทนที่จะใช้ปีกกา ({, }) แบบภาษาอื่น โครงสร้างภาษามีพื้นฐานการใช้จำนวนบรรทัดให้น้อย ถ้าเทียบกับ C++ และ Java ลักษณะ feature มีด้วยกันหลายแบบ เช่น สามารถบริหารจัดการ หน่วยความจำได้เอง, รองรับการทำงานหลาย paradigm, object-oriented function

บทที่ 3

วิธีการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ ผู้วิจัยได้ดำเนินการศึกษาค้นคว้าตามขั้นตอนดังนี้

- 3.1 วิธีการดำเนินโครงการ
- 3.2 แผนการดำเนินโครงการ
- 3.3 การเก็บรวบรวมข้อมูล และการนำข้อมูลมาประมวลผล
- 3.4 การพัฒนาโมเดลระบบการแปลภาษาเพื่อการพยากรณ์อากาศเป็นคำพูด
- 3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลการแปลภาษาเพื่อการพยากรณ์อากาศเป็นคำพูด
- 3.6 การพัฒนาในส่วนแอปพลิเคชัน

3.1 วิธีการดำเนินโครงการ

- 3.1.1 ศึกษาค้นคว้าแนวคิด และทฤษฎีที่เกี่ยวข้องกับการแปลภาษาเพื่อการพยากรณ์อากาศโดยใช้ MediaPipe
- 3.1.2 รวบรวมข้อมูล และวิเคราะห์ข้อมูล เพื่อค้นหารูปแบบในข้อมูลและกำหนดขอบเขตของโครงการวิจัย
- 3.1.3 วิเคราะห์และออกแบบการทำงานของโมเดลระบบการแปลภาษาเพื่อการพยากรณ์อากาศ
- 3.1.4 เขียนโปรแกรมพัฒนาโมเดลระบบการแปลภาษาเพื่อการพยากรณ์อากาศและทำการเทรนโมเดลที่พัฒนาขึ้น
- 3.1.5 ทำการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบการแปลภาษาเพื่อการพยากรณ์ อากาศ
- 3.1.6 ปรับปรุง และแก้ไขข้อผิดพลาดหลังการทดสอบประสิทธิภาพ
- 3.1.7 ประเมินผล สรุปผลการทดลองและจัดทำรูปเล่ม

3.2 แผนการดำเนินงานโครงการ

การดำเนินการในการทำโครงการได้แบ่งออกเป็น 4 ระยะ คือ

ระยะที่ 1 : วางแผนจัดทำโครงการ ศึกษาข้อมูล รวบรวมทฤษฎี และงานวิจัยที่เกี่ยวข้องกับการแปล
แปลภาษาเพื่อการพยากรณ์อากาศ

ระยะที่ 2 : ออกแบบการทำงานของโมเดลระบบการแปลแปลภาษาเพื่อการพยากรณ์อากาศ ประกอบด้วย
การพัฒนาต้นแบบโมเดล Long Short Term Memory (LSTM)

ระยะที่ 3 : ทำการเก็บรวบรวมข้อมูล การวิเคราะห์ข้อมูล และการคัดเลือกข้อมูลที่จะนำไปใช้กับโมเดล

ระยะที่ 4 : พัฒนาโมเดลระบบการแปลแปลภาษาเพื่อการพยากรณ์อากาศ และทดสอบประสิทธิภาพ
โมเดลระบบ

ระยะเวลาในการดำเนินการตั้งแต่เดือนตุลาคม 2562 ถึงเดือนพฤษภาคม 2563 รวมเวลาทั้งสิ้น 8 เดือน โดยมี
แผนการดำเนินโครงการ ดังต่อไปนี้

ตารางที่ 1 ตารางแผนการดำเนินงานตลอดโครงการ

ขั้นตอน	เดือน							
	ปี2562			ปี2563				
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.การวางแผนการดำเนินงาน								
ประชุมเพื่อเลือกหัวข้อ								
ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องกับงานวิจัยการแปลภาษาเมื่อการพยากรณ์อากาศ								
ศึกษาความรู้ด้าน MediaPipe โดยใช้ Hand Tracking ในการวิเคราะห์ข้อมูล								
ศึกษาโมเดล RNN และ LSTM ที่ใช้ในการวิเคราะห์ข้อมูลและวางแผนการเก็บข้อมูล								
2.การวิเคราะห์ระบบ								
วิเคราะห์ความเป็นไปได้ในการทำระบบ								
3.การออกแบบระบบ								
กำหนดวัตถุประสงค์และขอบเขตของระบบ								

ขั้นตอน	เดือน							
	ปี2562			ปี2563				
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
4.การพัฒนาระบบและทดสอบระบบ								
ทำการสร้างและพัฒนาระบบ								
ปรับปรุงและแก้ไขข้อผิดพลาดหลังการ ทดสอบระบบ								
เปรียบเทียบความแม่นยำและความ ถูกต้อง ของโมเดล								
5.สรุปและประเมินผล								
ประเมินผลการใช้งานระบบแอปพลิเคชัน แปลภาษาเพื่อการพยากรณ์อากาศ								
นำระบบแอปพลิเคชันแปลภาษามือ การ พยากรณ์อากาศไปใช้งานจริงกับ ผู้พิการ ทางการได้ยินจริง								
สรุปผลการดำเนินงาน								
จัดทำคู่มือและเอกสารที่เกี่ยวข้อง								
นำเสนอระบบที่สมบูรณ์								

3.3 การเก็บรวบรวมข้อมูล และการนำข้อมูลมาประมวลผล

3.3.1 การเก็บรวบรวมข้อมูล

ทำการเก็บรวบรวมข้อมูล โดยถ่ายวิดีโอจากกล้องสมาร์ทโฟน ผู้ทำการทดลองทั้งหมด 10 คน และวิดีโอจากเว็บไซต์ signingsavvy.com ผู้ทดลองจะใส่เสื้อยืดสีเข้ม ไม่มีลวดลายและฉากพื้นหลังจะเป็นสีขาว ซึ่งจะทำภาษามือการพยากรณ์อากาศทั้งหมด 10 คำ ประกอบด้วย Cold, Cool, Hot, Rain, Snow, Sun, Thunder, Warm, Weather, Wind ดังรูปที่ 3-1



รูปที่ 3-1 ตัวอย่างข้อมูลวิดีโอภาษามือการพยากรณ์อากาศ

3.3.2 การนำข้อมูลมาประมวลผล

ทำการนำข้อมูลวิดีโอที่เก็บรวบรวมมาทั้งหมด มาเข้าขั้นตอนในการหา keypoint โดยนำมาเข้า Google MediaPipe ในส่วน Hand Tracking มาใช้ตรวจจับหา Hand landmarks ที่จับเฉพาะมือเท่านั้น และแบ่งย่อยวิดีโอออกมาเป็นภาพทุกๆ keyframes จากนั้นจะสกัด feature ออกมาเป็นเวกเตอร์

3.4 การพัฒนาโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด

ทำการนำ Text file ที่สกัดออกมาได้เป็น feature มาทำการเทรนโมเดล *Long Short Term Memory* (LSTM) โดยแยกชุดข้อมูลเป็นส่วน train กับ test จะแบ่งการ train 170(68%) , test 80(32%)

ทำการ reshape ข้อมูลให้อยู่ในรูป (70,84) แล้วแยกชุดข้อมูลเป็นส่วน train กับ test ดังรูปที่ 3-4

```
#reshape input to 70 rows and 84 column and append to array
for i in range(0,70):
    landmark_frame.extend(numbers[row:row+84])
    row += 84
data1=np.array(landmark_frame)
landmark_frame=np.array(landmark_frame)
landmark_frame=list(landmark_frame.reshape(-1,84))
#split train,test
#append words and labels by the name of folder
if (k%3==2):
    XT.append(np.array(landmark_frame))
    YT.append(wordname)
else:
    X.append(np.array(landmark_frame))
    Y.append(wordname)
k+=1
X=np.array(X)
Y=np.array(Y)
XT=np.array(XT)
YT=np.array(YT)
```

รูปที่ 3-4 การ reshape ข้อมูลแล้วแยกชุดข้อมูลเป็นส่วน train กับ test

ทำการสร้าง label โดยใช้ชื่อ folder ของ dataset เป็น label แล้วทำเป็น Text file จากนั้นแปลง label ให้กลายเป็นตัวเลขเพื่อนำไปใช้ fit model ดังรูปที่ 3-5 และ รูปที่ 3-6

```
#create label txt file with ascending order
k=set(Y)
ks=sorted(k)
text=""
for i in ks:
    text=text+i+" "
make_label(text)
```

รูปที่ 3-5 การ labelแล้วทำเป็น Text file

```

#match Label with word
s = Tokenizer()
s.fit_on_texts([text])#{0:"Cold",1:"Cool"...}
encoded=s.texts_to_sequences([Y])[0]
encoded1=s.texts_to_sequences([YT])[0]
one_hot = to_categorical(encoded) #encode string to int
one_hot2=to_categorical(encoded1)
(x_train, y_train) = X, one_hot
(x_test,y_test)=XT,one_hot2
x_train=np.array(x_train)
y_train=np.array(y_train)
x_test=np.array(x_test)
y_test=np.array(y_test)
return x_train,y_train,x_test,y_test

```

รูปที่ 3-6 การแปลง label ให้กลายเป็นตัวเลข

ทำการเทรนโมเดล และ save โมเดลไว้ใช้ในการ prediction ดังรูปที่ 3-7

```

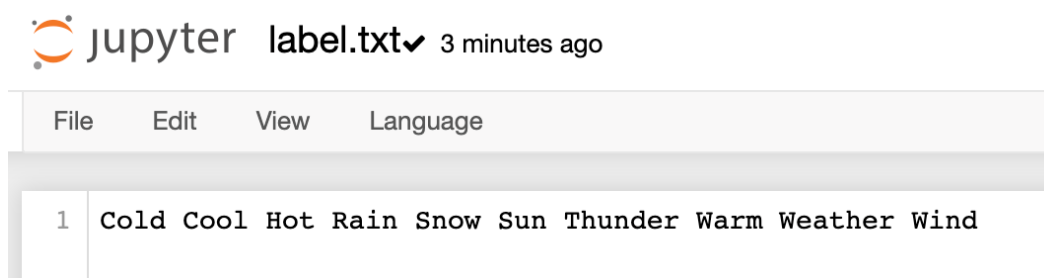
def main(dirname):
    x_train,y_train,x_test,y_test=load_data(dirname)
    model=build_model(y_train.shape[1])
    print('Training stage')
    print('=====')
    history=model.fit(x_train,y_train,epochs=100,validation_data=(x_test,y_test))
    model.summary()
    score, acc = model.evaluate(x_test,y_test,batch_size=16,verbose=0)
    print('Test performance: accuracy={0}, loss={1}'.format(acc, score))
    model.save('mediapipe/model.h5')

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Training Model')
    parser.add_argument("--input_train_path",help=" ")
    args=parser.parse_args()
    input_train_path=args.input_train_path
    main(input_train_path)

```

รูปที่ 3-7 การtrain โมเดล และ save โมเดล

ผลลัพธ์ที่ได้จากการเทรนจะได้ label.txt ที่มีคำของภาษามือการพยากรณ์อากาศทั้งหมดจำนวน 10 คำ
 ดังรูปที่ 3-8



รูปที่ 3-8 label.txt คำภาษามือการพยากรณ์อากาศ

3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด

ทำการทำนายทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด โดยการนำวิดีโอจำนวน 10 วิดีโอ ที่เป็นคำเดียวกัน โดยจะผ่านการแปลเป็น Text file และประมวลผลออกมา จะได้ผลลัพธ์เป็นคำพยากรณ์อากาศ

โดยจะนำ label.txt ที่ผ่านการเทรน จากข้อมูลมาใช้ในการ prediction ดังรูปที่ 3-9

```
#prediction
def load_label():
    listfile=[]
    with open("label.txt",mode='r') as l:
        listfile=[i for i in l.read().split()]
    label = {}
    count = 1
    for l in listfile:
        if "_" in l:
            continue
        label[l] = count
        count += 1
    return label
```

รูปที่ 3-9 การนำ label.txt ที่ผ่านการเทรนมาใช้ในการ prediction

โดยจะนำ model.h5 ที่ผ่านการเทรน จากข้อมูลมาใช้ในการ prediction ดังรูปที่ 3-10

```
#Load model
output_dir=output_data_path
x_test,Y=load_data(output_dir)
new_model = tf.keras.models.load_model('model5.h5')
new_model.summary()
labels=load_label()
```

รูปที่ 3-10 การนำ model.h5 ที่ผ่านการเทรนมาใช้ในการ prediction

```
#Use model
xhat = x_test
yhat = new_model.predict(xhat)
predictions = np.array([np.argmax(pred) for pred in yhat])
rev_labels = dict(zip(list(labels.values()), list(labels.keys())))
s=0
filel=np.array(filel)
txtpath=output_data_path+"result.txt"
with open(txtpath, "w") as f:
    print("Prediction of : "+ Y[s])
    for i in predictions:
        print("Video "+ str(s)+ " == " + rev_labels[i])
        f.write(rev_labels[i])
        f.write(" ")
    s+=1
```

รูปที่ 3-11 การนำข้อมูลมาทดสอบกับโมเดล

3.6 การพัฒนาในส่วนแอปพลิเคชัน

ผู้พัฒนาจะใช้โปรแกรม PyCharm ในการพัฒนาแอปพลิเคชัน โดยในส่วนหน้าของแอปพลิเคชันจะเขียนด้วย Python ทั้งหมด จะนำฟังก์ชันทั้งหมดที่ได้เขียนขึ้นมาใส่ไว้ในส่วนหน้าแอปพลิเคชัน จะให้ผู้ใช้นำวิดีโอภาษาเพื่อการพยากรณ์อากาศเข้ามา เพื่อแปลวิดีโอภาษาเพื่อการพยากรณ์อากาศให้แสดงออกมาเป็น Text ที่แปลมาจากวิดีโอที่ผู้ใช้ใส่เข้ามา

บทที่ 4

ผลการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ผู้จัดทำได้เสนอผลการดำเนินโครงการ โดยแสดงผลดังต่อไปนี้

4.1 แสดงผลลัพธ์ของภาพต้นฉบับและLabel

ผลลัพธ์จากการที่นำวิดีโอมาเข้า Google MediaPipe จะได้ label.txt ที่เป็น Text ดังตารางที่ 2

ตารางที่ 2 ตารางภาพวิดีโอต้นฉบับและLabel

Weather word	Label
1.Cold 	1 Cold
2.Cool 	2 Cool

3.Hot



3 | Hot

4.Rain



4 | Rain

5.Snow



5 | Snow

6.Sun



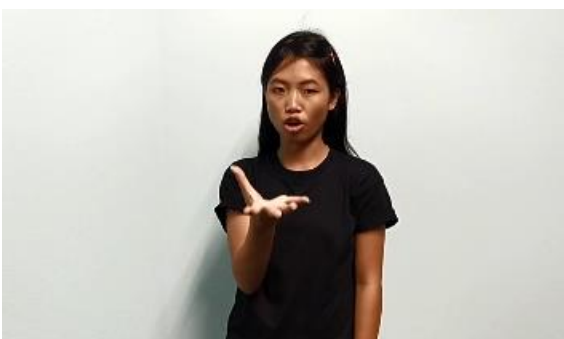
6 | Sun

7.Thunder



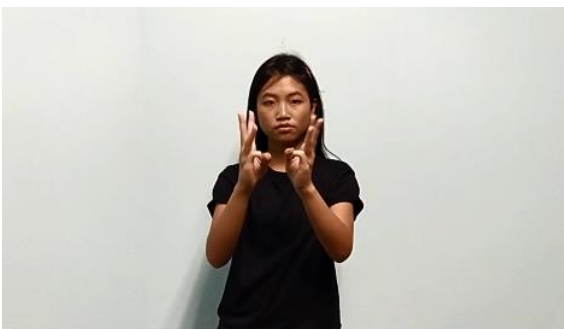
7 | Thunder

8.Warm



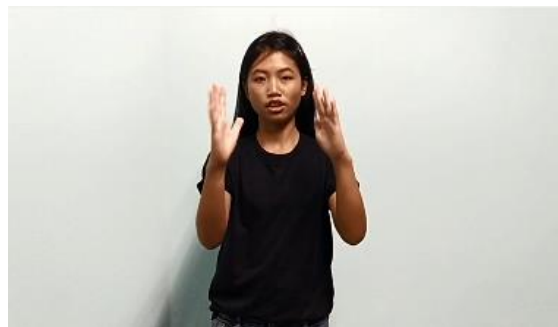
8 | Warm

9.Weather



9 | Weather

10.Wind



10 | Wind

4.2 แสดงประสิทธิภาพของโมเดล

การประเมินประสิทธิภาพความถูกต้อง และวัดความแม่นยำในการเทรนของโมเดล Long Short Term Memory (LSTM) ในการหาค่า Accuracy จะได้ผลลัพธ์ค่าเฉลี่ย 0.875 ดังรูปที่ 4-1

Model: "Sequential_5"

Layer (type)	Output Shape	Param #
lstm_13 (LSTM)	(None, 70, 64)	38144
lstm_14 (LSTM)	(None, 70, 32)	12416
lstm_15 (LSTM)	(None, 32)	8320
Sequential_5 (Dense)	(None, 11)	363
Total params: 59,243		
Trainable params: 59,243		
Non-trainable params: 0		
Test performance: accuracy=0.875, loss=0.5420249119400978		

รูปที่ 4-1 รูปการประเมินโมเดลของโมเดล Long Short Term Memory (LSTM)

4.3 แสดงค่าความถูกต้องและไม่ถูกต้องของวิดีโอที่ทำการทดสอบ

การประเมินโมเดลในการทดสอบความถูกต้องและไม่ถูกต้องของวิดีโอภาษาเมื่อการพยากรณ์อากาศ จำนวนวิดีโอที่นำมาทำการทดสอบมีจำนวนทั้งหมด 10 วิดีโอ ได้ผลดังตารางที่ 3

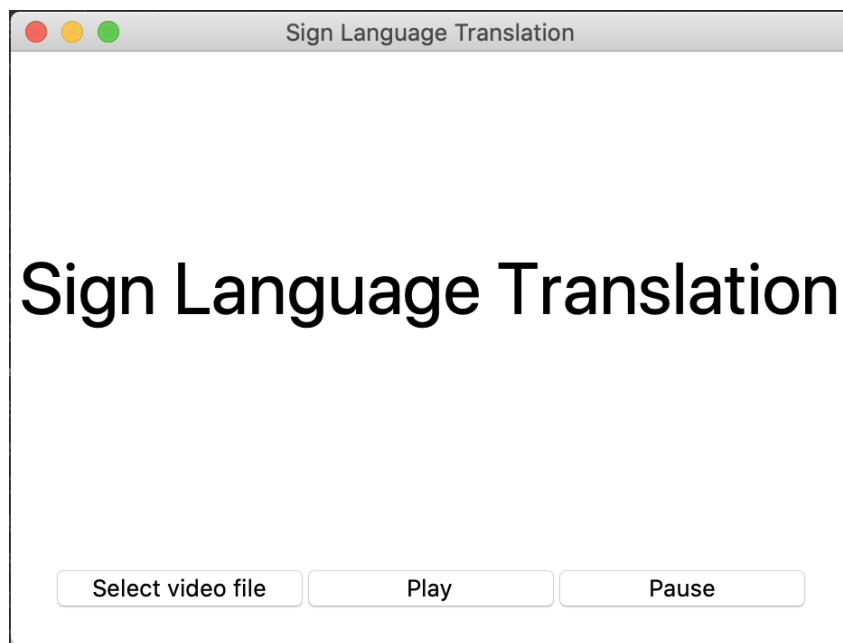
ตารางที่ 3 ผลลัพธ์การวัดประสิทธิภาพของโมเดลที่นำวิดีโอมาทดสอบ

Weather word	Correct	Incorrect
1.Cold	7	3
2.Cool	10	0
3.Hot	8	2
4.Rain	8	2
5.Snow	9	1
6.Sun	6	4
7.Thunder	10	0
8.Warm	5	5
9.Weather	10	0
10.Wind	6	4
Average	7.9	2.1

ผลลัพธ์ที่ได้จากการทดสอบของวิดีโอภาษามือการพยากรณ์อากาศจะมีบางคำที่ถูกต้องและไม่ถูกต้อง ผลลัพธ์ที่ถูกต้องเกิดจากการทำภาษามือการพยากรณ์อากาศที่มีท่าที่ไม่ยากและมีการเคลื่อนไหวของมือที่น้อย ส่วนผลลัพธ์บางคำที่ไม่ถูกต้องอาจเกิดจากการทำภาษามือการพยากรณ์อากาศที่มีท่าที่คล้ายกันจนเกิดไปหรือ อาจเกิดจากการทำมือที่มืองศาในการทำที่แตกต่างกัน

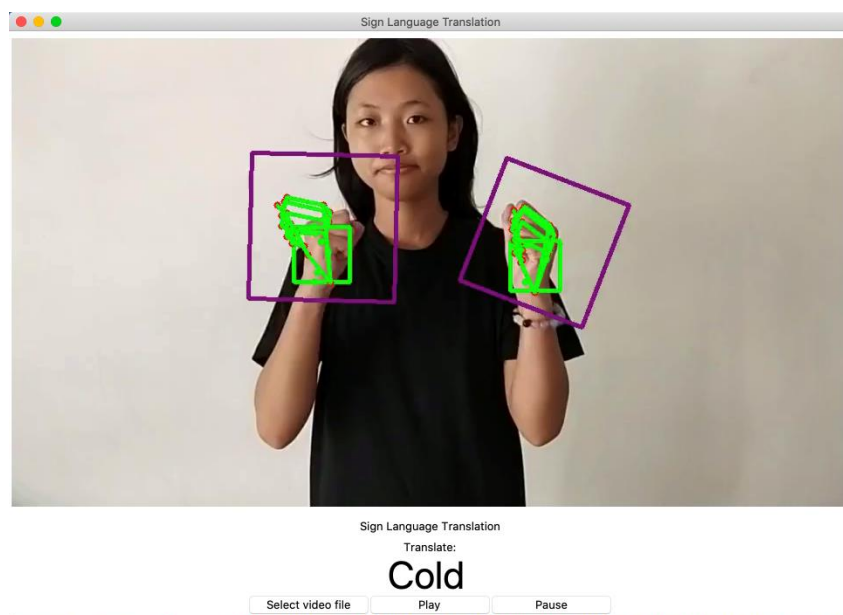
4.4 แอปพลิเคชันโปรแกรมระบบแปลภาษามือเป็นคำพูด

หน้าแรกของแอปพลิเคชันโปรแกรมระบบแปลภาษามือเป็นคำพูด (Sign Language Translation) จะมีปุ่ม Select video file เพื่อเลือกวิดีโอภาษามือที่ต้องการ, ปุ่ม Playและปุ่ม Pause ดังรูปที่ 4-2



รูปที่ 4-2 หน้าแรกของแอปพลิเคชันโปรแกรมระบบแปลภาษามือ

หลังจากผู้ใช้เลือกวิดีโอแล้ว ระบบจะทำการประมวลผลเพื่อแปลงวิดีโอเป็น Text และเมื่อผู้ใช้กดปุ่ม Play วิดีโอจะเล่นพร้อมปรากฏค่าที่ได้จากการแปลภาษามือ และมีปุ่ม Pause สำหรับหยุดวิดีโอ



รูปที่ 4-3 หน้าแสดงผลการแปลภาษามือเป็นคำพูด

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษาค้นคว้า

เมื่อได้ศึกษา และพัฒนาระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด (Sign Language for weather forecast translation) โดยใช้ต้นแบบโมเดล Long Short Term Memory (LSTM) สามารถสรุปผลลัพธ์ของการพัฒนาได้ดังต่อไปนี้

ผู้วิจัยได้ทำการพัฒนาโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด โดยการนำวิดีโอภาษามือการพยากรณ์อากาศมาทำนาย เพื่อทดสอบประสิทธิภาพความถูกต้อง และวัดความแม่นยำของโมเดล และผู้วิจัยทำการพัฒนาเป็นแอปพลิเคชันเพื่อช่วยต่อการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด

จากการทดสอบประสิทธิภาพ ความแม่นยำของโมเดล Long Short Term Memory (LSTM) พบว่าโมเดล มีความแม่นยำ และมีประสิทธิภาพความถูกต้อง ได้ค่อนข้างดีและสูงถึงร้อยละ 0.875 จากทั้งหมด แต่ยังมีข้อผิดพลาดอยู่บ้าง เนื่องจากการทำภาษามือของแต่ละบุคคล ทำภาษามือทำเดียวกันแต่ไม่เหมือนกัน ซึ่งอาจจะเกิดจากอารมณ์และเอกลักษณ์ของแต่ละบุคคล ที่ควบคุมได้ยาก

5.2 อภิปรายผล

จากการดำเนินโครงการสามารถนำระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด มาใช้ในการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด โดยใช้โมเดล Long Short Term Memory (LSTM) และจากการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด โดยการนำวิดีโอจำนวน 10 วิดีโอ ที่เป็นคำเดียวกันมาทดสอบแต่ละคำ พบว่าโมเดลมีประสิทธิภาพดีและมีความถูกต้องถึงร้อยละ 7.9 จากทั้งหมด สรุปได้ว่าระบบการแปลภาษามือการพยากรณ์อากาศเป็นคำพูด สามารถนำมาใช้ในการแปลภาษามือการพยากรณ์อากาศให้กับผู้ใช้ได้แม่นยำในระดับหนึ่ง

5.3 ปัญหาและอุปสรรค

เนื่องจากมีโรคไวรัส COVID-19 ระบาด ทำให้ไม่สามารถนำแอปพลิเคชัน Sign Language Translation มาใช้กับผู้พิการทางการได้ยินจริง ชุดข้อมูลมีจุดบกพร่อง เช่น ชุดข้อมูลภาษามือที่เกี่ยวข้องกับการพยากรณ์ อากาศมีจำนวนค่อนข้างน้อย และการเก็บข้อมูลการทำภาษามือของแต่ละบุคคลมีเอกลักษณ์ที่แตกต่างกันออกไป ทำให้ภาษามือจึงทำท่าทางไม่เหมือนกัน จึงทำให้การทำนาย ได้ประสิทธิภาพไม่ดีเท่าที่ควรจะเป็น

5.4 ข้อเสนอแนะ

เนื่องจากโครงงานนี้ยังมีจุดบกพร่องอยู่ในชุดข้อมูล เช่น ชุดข้อมูลภาษามือที่เกี่ยวข้องกับการพยากรณ์ อากาศมีจำนวนค่อนข้างน้อย และการทำภาษามือของแต่ละบุคคลทำท่าทางไม่เหมือนกัน จึงทำให้การทำนาย ได้ประสิทธิภาพไม่ดีเท่าที่ควรจะเป็น และสามารถนำแอปพลิเคชัน Sign Language Translation ไปประยุกต์ใช้กับภาษามือไทยได้จริง

บรรณานุกรม

- [1] ราชกร บัญญา. (2557). ภาษามือ : ภาษาของคนหูหนวก. สืบค้นเมื่อ 21 พฤศจิกายน 2562, จาก <https://rs.mahidol.ac.th/rs-journal/vol.4/v.4-1-005.pdf>
- [2] อรรถพล ชัยมนัสกุลและ ประพจน์ ขุนทอง.การศึกษาวิธีวัดการจัดโดยการใช้เทคนิคการจับคู่ภาพ. สืบค้นเมื่อ 21 พฤศจิกายน 2562, จาก <http://www.annualconference.ku.ac.th>
- [3] อวยชัย ภิรมย์รัตน์.การศึกษามาเรียนรู้การใช้งาน Jupyter notebook เครื่องมือ Hand-On Python แบบ Interactive ที่ทรงพลังกัน. สืบค้นเมื่อ 22 พฤศจิกายน 2562, จาก <https://www.auoychai.com/>
- [4] อวยพร พานิช.ความรู้พื้นฐานเกี่ยวกับภาษามือ.สืบค้นเมื่อ 25 พฤศจิกายน 2562, จาก <http://elearning.psu.ac.th/>
- [5] Sirinart Tangruamsub. (2560). Long Short-Term Memory (LSTM). สืบค้นเมื่อ 29 กุมภาพันธ์ 2563, จาก <https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>
- [6] KENG SURAPONG. (2562). Recurrent Neural Network (RNN) คืออะไร. สืบค้นเมื่อ 1 มีนาคม 2563, จาก <https://www.bualabs.com/archives/3103/what-is-rnn-recurrent-neural-network>
- [7] LEONID. (2562). Google MediaPipe โอเพ่นซอร์ส AI ที่เก่งจดจำท่ามือ.สืบค้นเมื่อ 2 มีนาคม 2563, จาก <https://65blogs.com/technology/google-mediapipe-is-ai-open-source-with-recognizably-good-for-hand-shape/>
- [8] ADPT. (2562). AI ตรวจจับการเคลื่อนไหวของมือใช้แปลงภาษามือได้. สืบค้นเมื่อ 2 มีนาคม 2563, จาก <https://www.adpt.news/2019/08/20/hand-tracking-algorithm-could-lead-to-sign-language-recognition>

- [9] Suharjito , Ricky Anderson , Fanny Wiryana , Meita Chandra Ariesta , Gede PutraKusuma.; et al. (2017). Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output.Retrieved November 20, 2019, from Science Direct
- [10] Thongpan Pariwat , Pusadee Seresangtakul.; et al. (2017). Thai Finger-Spelling Sign Language Recognition Using Global and Local Features with SVM.Retrieved November 20, 2019, from In: IEEE 9th International Conference on Knowledge and Smart Technology
- [11] Auephanwiriyaikul , Suwannee Phitakwinai , Wattanapong Suttapak , Phonkrit Chanda ,Nipon Theera-Umpon.; et al. (2013). Thai sign language translation using Scale Invariant Feature Transform and Hidden Markov Models.Retrieved November 20, 2019, from Science Direct
- [12] Nayan M.Kakoty , Manalee DevSharma.; et al. (2018). Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove November 20, 2019, from Science Direct
- [13] Chai Phonbopit.หัดเขียน Python เบื้องต้นฟรีด้วยโปรแกรม PyCharm Edu.สืบค้นเมื่อ 22 พฤศจิกายน 2562, จาก <https://devahoy.com/blog/2015/10/learn-python-with-pycharm-edu>
- [14] Sarayut Nonsiri.ภาษาโปรแกรม Python คืออะไร ?.สืบค้นเมื่อ 22 พฤศจิกายน 2562, จาก <https://www.9experttraining.com/articles/python>
- [15] Project import generated by Copybara.. Hand Tracking (GPU). สืบค้นเมื่อ 22 พฤศจิกายน 2562, จาก<https://github.com/google/mediapipe/blob/master/mediapipe>

ภาคผนวก

ภาคผนวก ก

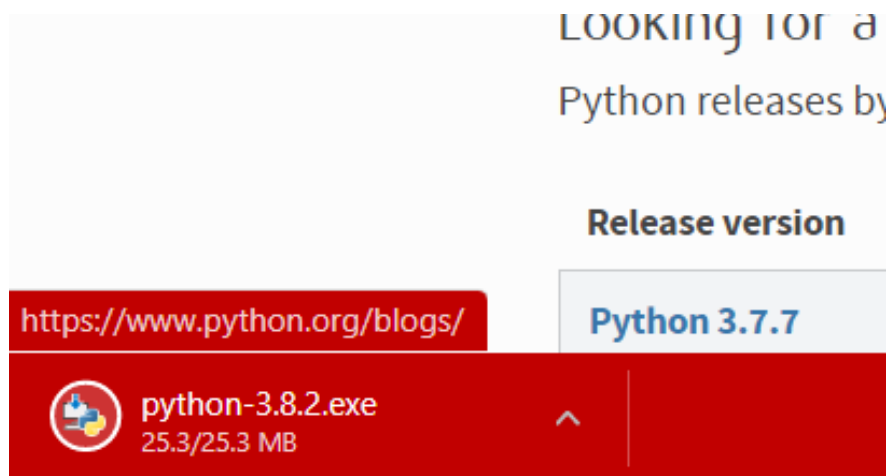
การติดตั้งโปรแกรม Python

1. เข้าไปที่เว็บไซต์ <https://www.python.org/downloads/>
2. คลิกที่ปุ่ม Download Python 3.8.2



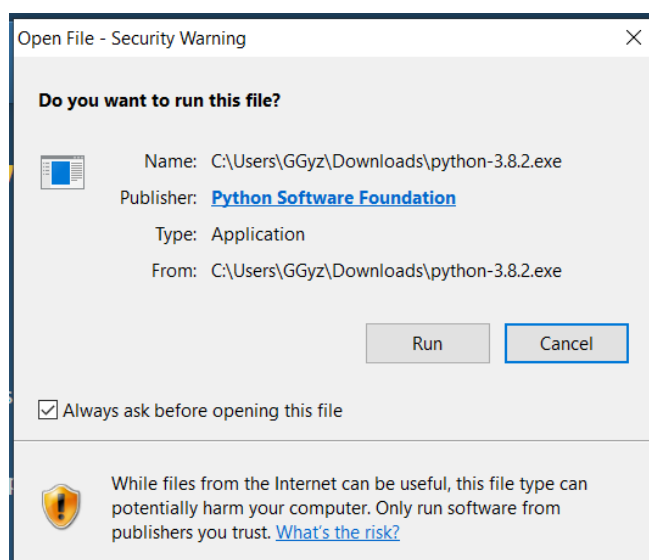
รูปที่ ก-1 หน้าแรกของเว็บไซต์ตัวโปรแกรม

3. รอจนกว่าจะดาวน์โหลดตัวติดตั้งเสร็จ และเมื่อดาวน์โหลดเสร็จแล้วให้กดเปิดตัวติดตั้งขึ้นมา



รูปที่ ก-2 รอการดาวน์โหลดโปรแกรมเสร็จสิ้น

4. ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมาเพื่อติดตั้ง Python แล้วคลิกปุ่ม Run



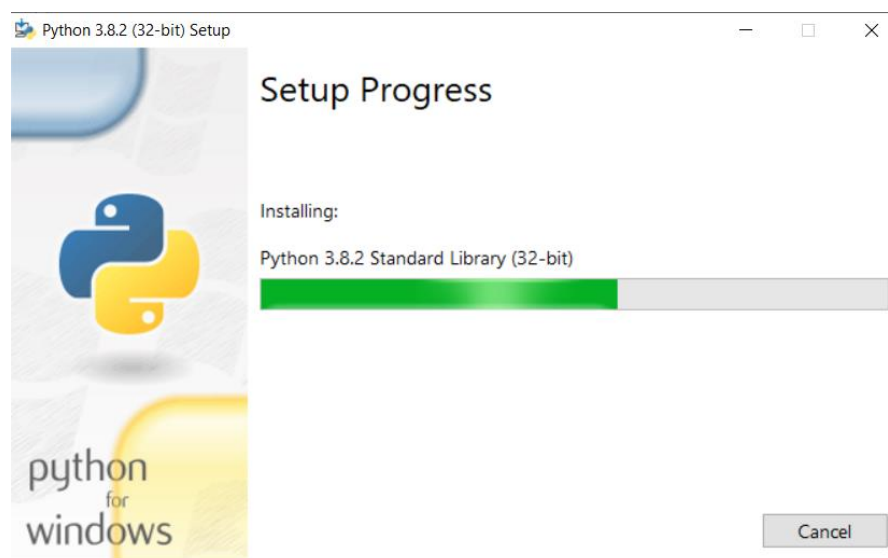
รูปที่ ก-3 เปิดไฟล์เพื่อดำเนินการติดตั้งโปรแกรม

5. เข้าสู่หน้าการติดตั้งโปรแกรม คลิกคำว่า Install Now



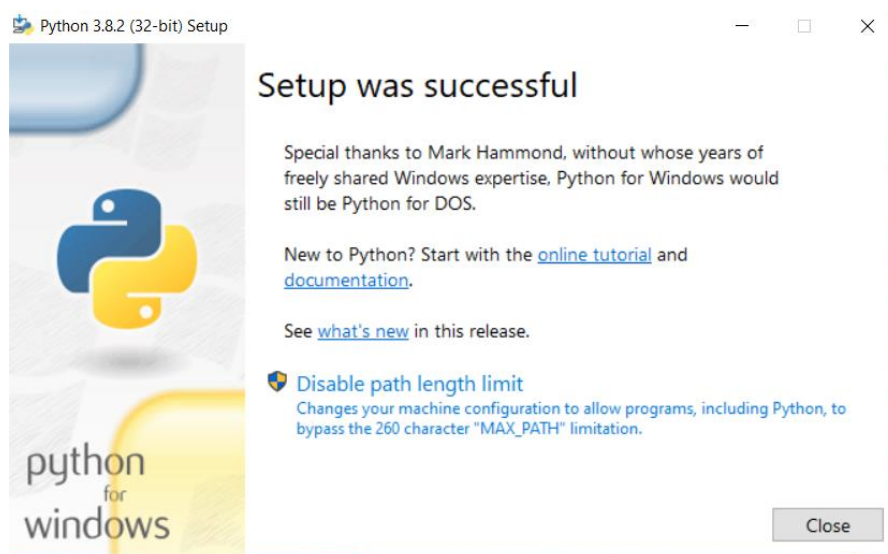
รูปที่ ก-4 หน้าต่างเมื่อเปิดตัว setup

6. รอจนกว่าการติดตั้งโปรแกรมจะดำเนินการเสร็จ



รูปที่ ก-5 หน้าต่างแสดงการติดตั้งโปรแกรม

7. เมื่อติดตั้งโปรแกรมเสร็จจะขึ้นหน้าต่าง Setup was successful



รูปที่ ก-6 หน้าต่างเสร็จสิ้นการติดตั้งโปรแกรม

ภาคผนวก ข

การติดตั้งโปรแกรม PyCharm

1.ดาวน์โหลดตัวติดตั้ง PyCharm ได้ที่ <https://www.jetbrains.com/pycharm/download/#section=windows> เมื่อเข้าเว็บแล้วให้กดปุ่มดาวน์โหลด

Download PyCharm

Windows Mac Linux

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

Free trial

Community

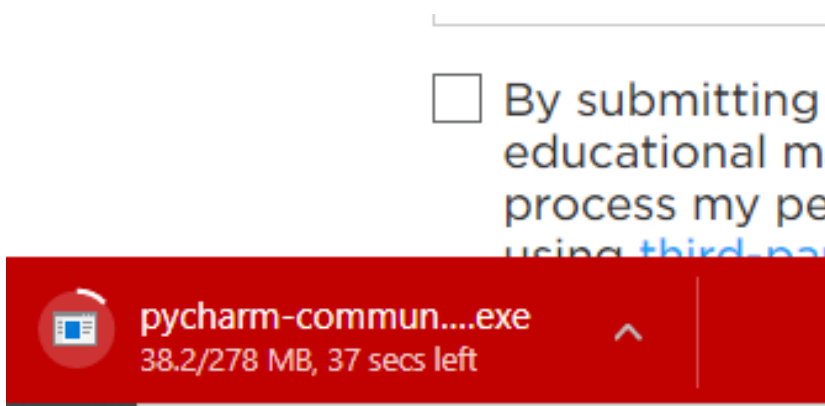
For pure Python development

Download

Free, open-source

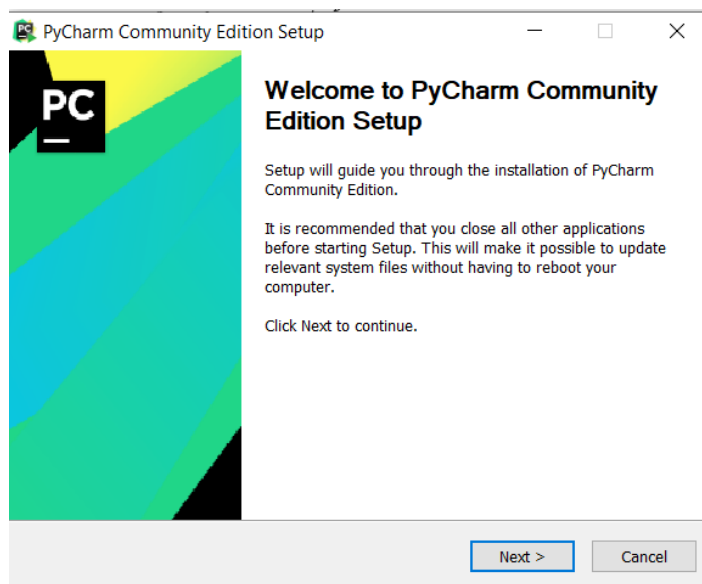
รูปที่ ข-1 ดาวน์โหลดตัวติดตั้ง PyCharm

2.รอก่อนที่จะดาวน์โหลดตัวติดตั้งเสร็จ และเมื่อดาวน์โหลดเสร็จแล้วให้กดเปิดตัวติดตั้งขึ้นมา



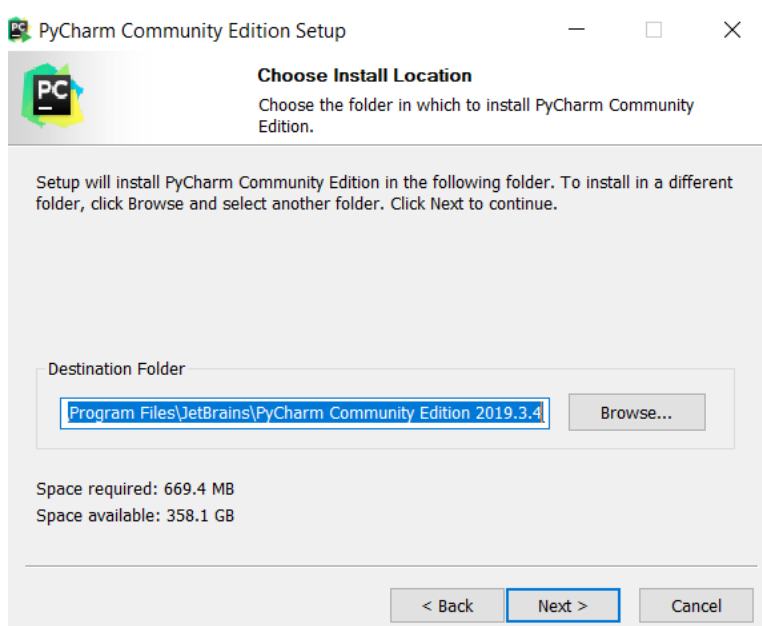
รูปที่ ข-2 รอกการดาวน์โหลดโปรแกรมเสร็จสิ้น

3.เมื่อเปิดตัวติดตั้งขึ้นมาแล้ว จะแสดงหน้าจอ Welcome to the PyCharm Community Edition Setup ให้กดปุ่ม Next >



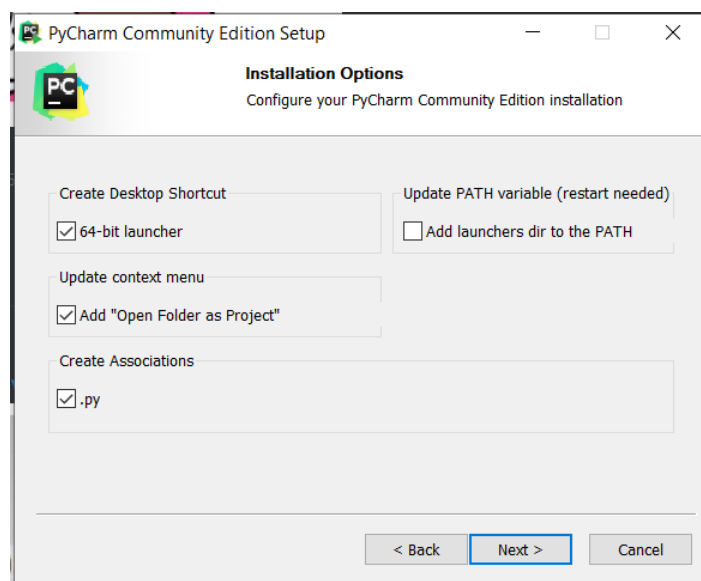
รูปที่ ข-3 หน้าต่างเมื่อเปิดตัว setup

4.จากนั้นจะแสดงหน้าจอ Choose Install Location ให้กดปุ่ม Next >



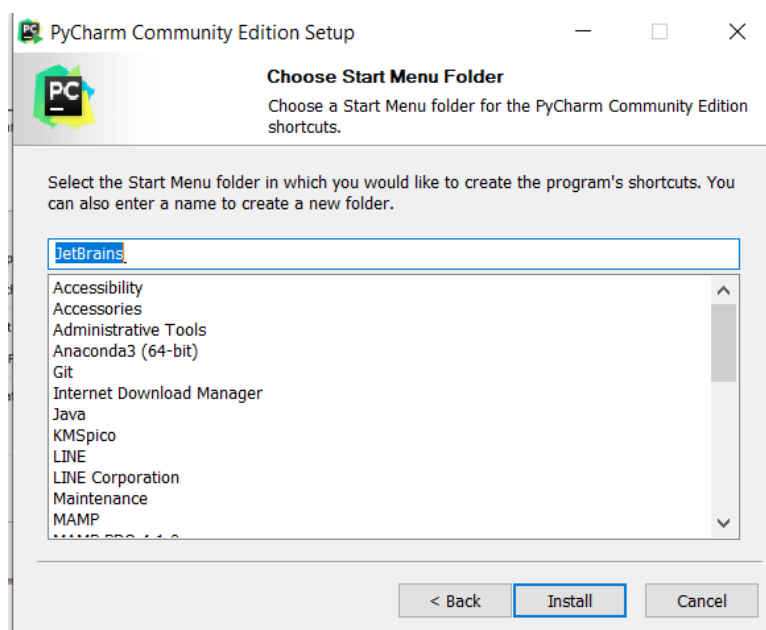
รูปที่ ข-4 เลือกโฟลเดอร์ที่จะติดตั้งโปรแกรม

5.จากนั้นจะแสดงหน้าจอ Installation Options ให้กดปุ่ม Next >



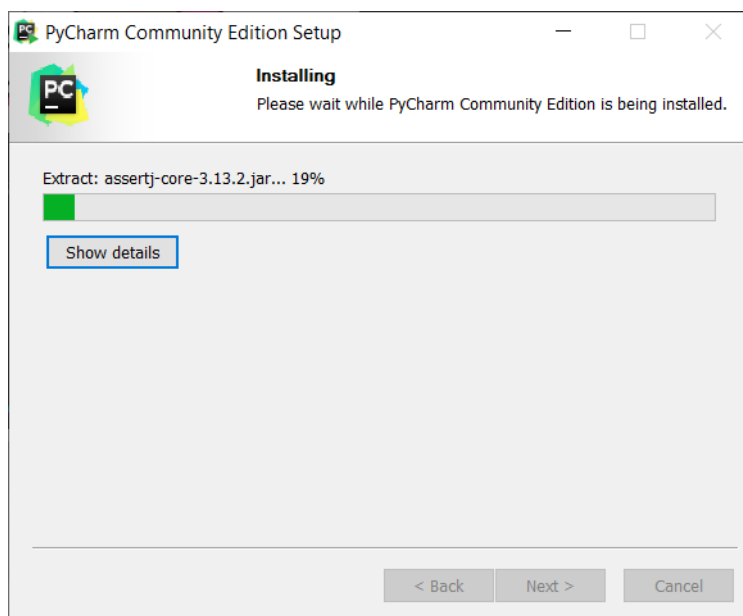
รูปที่ ข-5 เลือกตัวเลือกการติดตั้ง

6.จากนั้นจะแสดงหน้าจอ Choose Start Menu Folder ให้กดปุ่ม Install



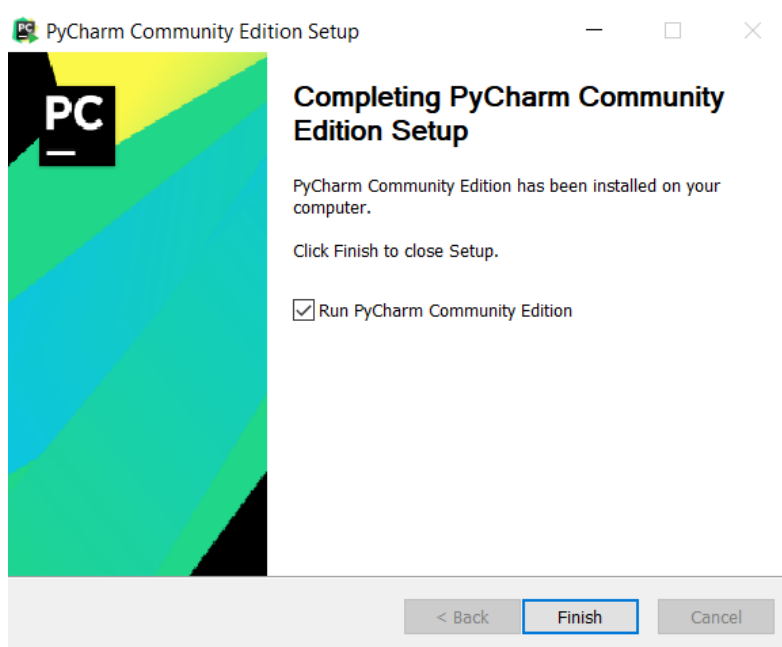
รูปที่ ข-6 เลือกโฟลเดอร์เริ่มต้นของการทำงาน

7.จากนั้นจะแสดงหน้าจอ Installing ให้รอจนกว่าโปรแกรมจะติดตั้งเสร็จ



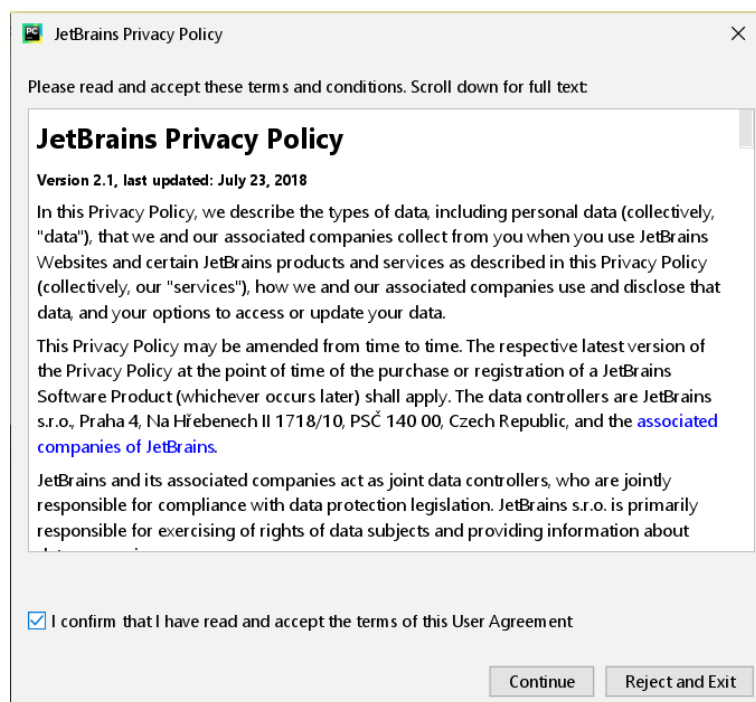
รูปที่ ข-7 หน้าต่างแสดงการติดตั้งโปรแกรม

8.เมื่อติดตั้งเสร็จแล้วจะแสดงหน้าจอ Completing PyCharm Community Edition Setup ให้กดปุ่ม Finish



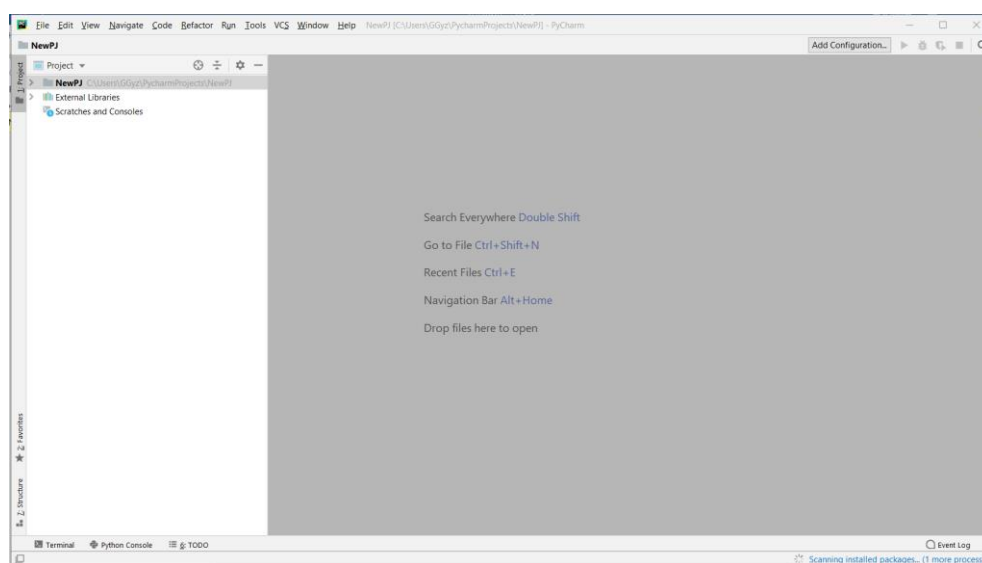
รูปที่ ข-8 หน้าต่างเมื่อติดตั้งโปรแกรมเสร็จสิ้น

9.จากนั้นหน้าต่าง JetBrains Privacy Policy จะเปิดขึ้นมา ให้เรากดยอมรับข้อตกลง และกด Continue เพื่อเข้าสู่โปรแกรม



รูปที่ ข-9 หน้าต่างของโปรแกรม Visual Studio Code

9.จากนั้นโปรแกรม PyCharm จะเปิดขึ้นมา



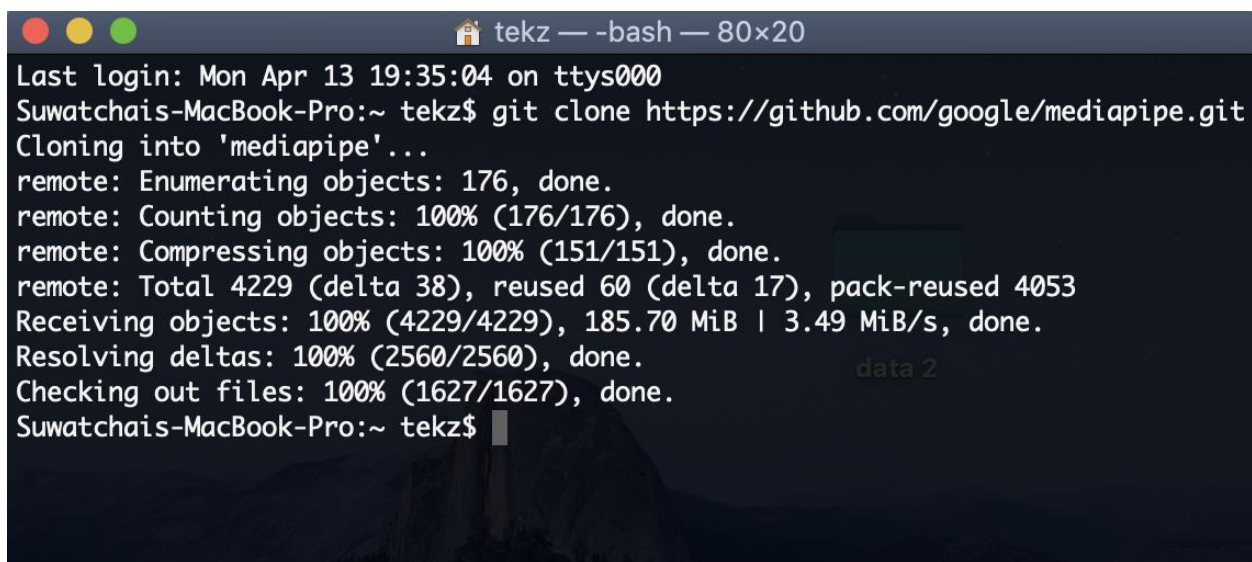
รูปที่ ข-10 หน้าต่างของโปรแกรม PyCharm

ภาคผนวก ค

การติดตั้งโปรแกรม Google Media Pipe

1.เปิด command line และ พิมพ์ git clone <https://github.com/google/mediapipe.git>

ในการ install Google Media Pipe

A screenshot of a macOS terminal window. The title bar shows three colored window control buttons (red, yellow, green) and the text 'tekz — -bash — 80x20'. The terminal content shows the command 'git clone https://github.com/google/mediapipe.git' being executed. The output indicates the repository is being cloned into a directory named 'mediapipe'. Progress bars and status messages are shown for enumerating, counting, and compressing objects, as well as receiving and resolving deltas. The process completes successfully, and the prompt returns to 'Suwatchais-MacBook-Pro:~ tekz\$'.

```
tekz — -bash — 80x20
Last login: Mon Apr 13 19:35:04 on ttys000
Suwatchais-MacBook-Pro:~ tekz$ git clone https://github.com/google/mediapipe.git
Cloning into 'mediapipe'...
remote: Enumerating objects: 176, done.
remote: Counting objects: 100% (176/176), done.
remote: Compressing objects: 100% (151/151), done.
remote: Total 4229 (delta 38), reused 60 (delta 17), pack-reused 4053
Receiving objects: 100% (4229/4229), 185.70 MiB | 3.49 MiB/s, done.
Resolving deltas: 100% (2560/2560), done.
Checking out files: 100% (1627/1627), done.
Suwatchais-MacBook-Pro:~ tekz$
```

รูปที่ ค-1 ติดตั้ง Google Media Pipe

ภาคผนวก ง

การติดตั้งโปรแกรม Bazel

1. ดูวิธีติดตั้ง Bazel ได้ที่ <https://docs.bazel.build/versions/master/install.html>

แล้วเปิด command line และ พิมพ์ `sudo apt install bazel1-1.0.0` ในการ install Bazel

```
(Sign2) weerayuth@weerayuth-System-Product-Name:~$ sudo apt install bazel-1.0.0
[sudo] password for weerayuth:
Reading package lists... Done
Building dependency tree
Reading state information... Done
bazel-1.0.0 is already the newest version (1.0.0).
0 upgraded, 0 newly installed, 0 to remove and 379 not upgraded.
(Sign2) weerayuth@weerayuth-System-Product-Name:~$
```

รูปที่ ง-1 ติดตั้ง Bazel

ภาคผนวก จ

การติดตั้งโปรแกรม Tensorflow

1. ดูวิธีติดตั้ง Tensorflow ได้ที่ <https://www.tensorflow.org/install> แล้วเปิด command line และ พิมพ์
pip install tensorflow ในการ install Tensorflow

```
(Sign2) weerayuth@weerayuth-System-Product-Name:~$ pip install tensorflow
Requirement already satisfied: tensorflow in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (2.2.0rc1)
Requirement already satisfied: absl-py>=0.7.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (0.9.0)
Requirement already satisfied: tensorflow-estimator<2.3.0,>=2.2.0rc0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (2.2.0rc0)
Requirement already satisfied: numpy<2.0,>=1.16.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (1.18.2)
Requirement already satisfied: protobuf>=3.8.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (3.11.3)
Requirement already satisfied: termcolor>=1.1.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: wrapt>=1.11.1 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: six>=1.12.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (1.14.0)
Requirement already satisfied: google-pasta>=0.1.8 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio>=1.8.6 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (1.27.2)
Requirement already satisfied: wheel>=0.26; python_version >= "3" in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (0.34.2)
Requirement already satisfied: tensorboard<2.2.0,>=2.1.0 in ./anaconda3/envs/Sign2/lib/python3.8/site-packages (from tensorflow) (2.1.1)
```

รูปที่ จ-1 ติดตั้ง Tensorflow บนเครื่องที่ใช้