

โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ
(Programs for facilitating sign language communication among hearing-impaired
and hearing individuals)

โรงเรียนอุทัยวิทยาคม
จังหวัดอุทัยธานี

คณะผู้จัดทำโครงการ

1. นางสาวธัญนันท์ มาดี
2. นางสาวอรุณจิรา ศรีสร้อย

รายงานนี้เป็นส่วนหนึ่งของโครงงานสาขาวิชาคอมพิวเตอร์
การนำเสนอโครงงานของนักเรียนจากโรงเรียนในโครงการห้องเรียนพิเศษวิทยาศาสตร์
คณิตศาสตร์ เทคโนโลยีและสิ่งแวดล้อม เครือข่ายภาคกลางตอนบน ประจำปีการศึกษา 2566

โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ
(Programs for facilitating sign language communication among hearing-impaired
and hearing individuals)

โรงเรียนอุทัยวิทยาคม
จังหวัดอุทัยธานี

คณะผู้จัดทำโครงการ

1. นางสาวธัญนันท์ มาดี
2. นางสาวอรุณจิรา ศรีสร้อย

ครูที่ปรึกษาโครงการ

1. นายถิรพัฒน์ วิเชียรรัตน์
2. นายฉัตรชัย นาคทอง

รายงานนี้เป็นส่วนหนึ่งของโครงงานสาขาวิชาคอมพิวเตอร์

การนำเสนอโครงงานของนักเรียนจากโรงเรียนในโครงการห้องเรียนพิเศษวิทยาศาสตร์
คณิตศาสตร์ เทคโนโลยีและสิ่งแวดล้อม เครือข่ายภาคกลางตอนบน ประจำปีการศึกษา 2566

บทคัดย่อ

โครงการคอมพิวเตอร์ เรื่อง โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อ 1. สร้างโปรแกรมที่สามารถช่วยเหลือผู้บกพร่องทางการได้ยิน และ 2. สำนวความพึงพอใจของผู้ใช้งานโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ

โดยมีวิธีการดำเนินงาน ดังนี้ นำการทำงานร่วมกันของ โมดูล OpenCV, MediaPipe และ Pygame โดยใช้ Python เพื่อนำ Hand Tracking มาประยุกต์ใช้ในการสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติได้ เมื่อทดสอบท่าทางภาษามือต่อหน้ากล้อง Webcam แล้วจะมีพิกัดขึ้นบนมือที่แสดงบนหน้าจอ จากนั้นโปรแกรมจะจับพิกัดบนมือที่แสดงท่าทางต่าง ๆ แล้วเก็บไว้ในตัวแปร จากนั้นจะเรียกไฟล์เสียงที่ตรงตามเงื่อนไข จากผลการทดสอบพบว่า โปรแกรมที่สร้างสามารถช่วยเหลือผู้บกพร่องทางการได้ยินให้สามารถใช้ภาษามือสื่อสารกับบุคคลที่ไม่ทราบภาษามือได้ โดยมีระดับความพึงพอใจด้านความคิดสร้างสรรค์ และความแปลกใหม่ อยู่ในระดับมากที่สุด ($\bar{x} = 4.58$, S.D. = 0.98) และความพึงพอใจโดยรวม อยู่ในระดับมาก ($\bar{x} = 4.34$, S.D. = 0.64) ซึ่งมีข้อจำกัดในเรื่องการเข้าถึงของโปรแกรมที่ยังไม่สามารถเข้าถึงได้ในบุคคลทั่วไป

กิตติกรรมประกาศ

โครงการเรื่อง โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ
นี้สำเร็จไปด้วยดีเนื่องจากได้รับความกรุณาเป็นอย่างสูงจากคุณครูธิพัฒน์ วิเชียรรัตน์
และคุณครูฉัตรชัย นาคทอง ซึ่งเป็นคุณครูที่ปรึกษาโครงการที่ให้คำปรึกษาแนะนำ และได้ตรวจสอบ
แก้ไขข้อบกพร่องต่าง ๆ ด้วยความใส่ใจอย่างยิ่ง จนคณะผู้จัดทำสามารถดำเนินงานได้สำเร็จ

ขอขอบพระคุณ คุณครูวีรญาดา บางแบ่ง ที่ปรึกษาโครงการพิเศษ ที่ได้ตรวจทาน และชี้แนะ
แนวทางการทำโครงการ

ขอขอบพระคุณ คุณพ่อ และคุณแม่ ที่สนับสนุนในการทำโครงการครั้งนี้ รวมทั้งเป็นกำลังใจ
และให้โอกาสการศึกษาอันมีค่าอย่างยิ่ง

สุดท้ายนี้ขอขอบคุณเพื่อน ๆ ที่คอยช่วยเหลือในการทำโครงการ ปรึกษาต่าง ๆ ที่เกี่ยวกับ
โครงการนี้ รวมถึงสนับสนุนพวกเรา และให้กำลังใจที่ดีเสมอมา

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	
ที่มาและความสำคัญ	1
วัตถุประสงค์	1
สมมติฐาน	2
ตัวแปรที่ศึกษา	2
ขอบเขตของการดำเนินงาน	2
นิยามเชิงปฏิบัติการ	2
บทที่ 2 เอกสารที่เกี่ยวข้อง	
ผู้บกพร่องทางการได้ยิน	3
ภาษามือ	3
ภาษา Python	4
Hand tracking	4
PyGame	8
บทที่ 3 อุปกรณ์ และวิธีการทดลอง	
วัสดุอุปกรณ์	9
ขั้นตอนการดำเนินงาน	9
กรอบแนวคิด	12

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการทดลอง	
การสร้างโปรแกรมแปลภาษามือ	13
การทดสอบประสิทธิภาพของโปรแกรมแปลภาษามือ	14
การสำรวจความพึงพอใจของผู้ใช้งานโปรแกรมแปลภาษามือ	17
บทที่ 5 สรุป อภิปรายผล และข้อเสนอแนะ	
สรุปผลการทดลอง	19
อภิปรายผลการทดลอง	19
ข้อเสนอแนะ	20
บรรณานุกรม	21
ภาคผนวก	
ภาคผนวก ก	23
ภาคผนวก ข	25
ภาคผนวก ค	28

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 4.1 แสดงผลความแม่นยำของการทดสอบโปรแกรม	14
ตารางที่ 4.2 แสดงข้อมูลผู้ตอบแบบสอบถามความพึงพอใจ	17
ตารางที่ 4.3 แสดงค่าเฉลี่ย, ส่วนเบี่ยงเบนมาตรฐาน และการแปลผล ความพึงพอใจของผู้ใช้งานที่มีต่อโปรแกรมแปลภาษามือ	18

สารบัญรูปภาพ

ภาพที่	หน้า
ภาพที่ 2.1 ภาพแสดงจุด Landmarks ทั้ง 21 จุดบนมือ	4
ภาพที่ 2.2 การแยกประเภทของ Object	5
ภาพที่ 2.3 ภาพแสดงการทำงานของ MediaPipe	6
ภาพที่ 3.1 แสดงตัวอย่างการเก็บตัวแปร 1 แล้ว เรียกไฟล์เสียงที่ชื่อว่า “1.mp3”	10
ภาพที่ 3.2 แสดงความผิดพลาดในการอ่านภาษามือของโปรแกรมก่อนดำเนินการแก้ไข	11
ภาพที่ 3.3 แสดงแผนผังขั้นตอนการดำเนินการสร้างโปรแกรมแปลภาษามือ	12
ภาพที่ 4.1 แสดงการทำงานของ การอ่านพิกัดบนมือ	13
ภาพที่ 1 การศึกษา และรวบรวมคำภาษามือที่สามารถนำมาใช้ในการพัฒนาโปรแกรม	23
ภาพที่ 2 การศึกษาภาษามือคำว่า “การเลือกตั้ง”	23
ภาพที่ 3 การศึกษาภาษามือคำว่า “พลเมือง”	24
ภาพที่ 4 การศึกษาหลักการทำงานของโปรแกรม โดยมีครูให้คำปรึกษา	24
ภาพที่ 5 การเขียนโปรแกรมภาษา python	25
ภาพที่ 6 การนำเข้าโมดูล และ ตัวแปร ที่นำมาประยุกต์ใช้ในโปรแกรม	25
ภาพที่ 7 โปรแกรมของตัวรับข้อมูลเข้าจากกล้อง webcam	26
ภาพที่ 8 โปรแกรมการกำหนดการอ่านคำมือซ้าย และมือขวา	26
ภาพที่ 9 โปรแกรมกำหนดพิกัดบนนิ้วมือทั้ง 5 นิ้ว	27
ภาพที่ 10 โปรแกรมการเก็บค่าตัวแปรเพื่อเรียกใช้ไฟล์เสียง	27
ภาพที่ 11 โปรแกรมการเรียกใช้ไฟล์เสียงโดยใช้โมดูล PyGame	27
ภาพที่ 12 หน้าต่างโปรแกรมขณะดำเนินการแปลภาษามือ	28
ภาพที่ 13 การทดสอบการทำงานของโปรแกรม โดยใช้ภาษามือคำว่า “วันนี้” และ “เกิดอาการแพ้”	28
ภาพที่ 14 การทดสอบระบบโปรแกรม และประเมินความพึงพอใจ	29
ภาพที่ 15 มาตรวัดเจตคติแบบลิเคิร์ท (วัดความพึงพอใจ)	29

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในสถานการณ์ปัจจุบันประเทศไทยมีผู้พิการอยู่เป็นจำนวนมาก โดยการพิการสามารถแบ่งออกได้หลายประเภท คือ พิการทางการเห็น, พิการทางการได้ยินหรือสื่อความหมาย, พิการทางการเคลื่อนไหวหรือทางร่างกาย, พิการจิตใจหรือพฤติกรรม, พิการทางสติปัญญา, พิการทางการเรียนรู้ และ พิการทางการออทิสติก รวมไปถึงความบกพร่องในด้านสภาพจิตใจ และอารมณ์ ซึ่งสภาวะเหล่านี้ทำให้ผู้พิการเกิดปัญหา และเป็นอุปสรรคในด้านต่าง ๆ ผู้ที่พิการทางการได้ยินหรือสื่อความหมาย ในประเทศไทยมีจำนวนมาก โดยคิดเป็นร้อยละ 18.39 หรือคิดเป็นจำนวนผู้พิการ 352,503 คน จากจำนวนผู้พิการทั้งหมด 1,916,828 คน ซึ่งเป็นจำนวนมากลำดับที่สองจากผู้พิการทั้งหมด จากแหล่งข้อมูลของกรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ ผู้พิการทางการได้ยินใช้ภาษามือในการสื่อสารกับบุคคลทั่วไปในชีวิตประจำวันโดยอาศัยการแสดงออกท่าทาง สีหน้า และการเคลื่อนไหวของร่างกาย แต่การสื่อสารโดยใช้ภาษามือเป็นภาษาที่รู้กันเพียงในคนบางกลุ่มเท่านั้น หากจะใช้สื่อสารระหว่างผู้พิการทางการได้ยินที่ใช้ภาษามือกับคนทั่วไปที่ไม่มีความรู้ทางด้านภาษามือจึงเป็นไปได้ยาก จึงควรที่จะต้องได้รับความช่วยเหลือ เพื่อให้สามารถดำเนินกิจกรรมในชีวิตประจำวันหรือเข้าไปมีส่วนร่วมทางสังคมได้อย่างบุคคลทั่วไป

คณะผู้จัดทำจึงมีแนวคิดที่จะแก้ปัญหาดังกล่าว โดยการนำเทคโนโลยีระบบโมดูล OpenCV, MediaPipe และ Pygame เพื่อนำ Hand Tracking มาประยุกต์ใช้กับภาษามือ ในการสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ จากการเขียนโปรแกรมคอมพิวเตอร์ ภาษา Python เพื่อช่วยผู้บกพร่องทางการได้ยินให้สามารถสื่อสารกับบุคคลทั่วไปได้อย่างสะดวก

1.2 วัตถุประสงค์

1.2.1 เพื่อสร้างโปรแกรมที่สามารถช่วยเหลือผู้บกพร่องทางการได้ยิน

1.2.2 เพื่อสำรวจความพึงพอใจของผู้ใช้งานโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ

1.3 สมมติฐาน

ผู้ใช้โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ มีความพึงพอใจในการใช้งานโดยภาพรวมในระดับมาก

1.4 ตัวแปรที่ศึกษา

ตัวแปรต้น กลุ่มเป้าหมาย (ผู้บกพร่องทางการได้ยิน)

ตัวแปรตาม การสื่อสารระหว่างผู้บกพร่องทางการได้ยินกับบุคคลทั่วไปที่ไม่ทราบภาษามือ และความพึงพอใจของผู้ใช้งาน

ตัวแปรควบคุม โปรแกรมที่ใช้เป็นสื่อกลางในการสื่อสารของผู้บกพร่องทางการได้ยิน และบุคคลทั่วไปที่ไม่ทราบภาษามือ

1.5 ขอบเขตของการดำเนินงาน

1.5.1 ระยะเวลาดำเนินการ 8 สิงหาคม 2565 – 1 สิงหาคม 2566

1.5.2 กลุ่มเป้าหมาย คือ ผู้บกพร่องทางการได้ยิน

1.5.3 ศึกษาการทำงานของโปรแกรมคอมพิวเตอร์ OpenCV, MediaPipe และ Pygame มาประยุกต์ใช้กับ Hand Tracking และภาษามือที่ใช้ในการสื่อสารของผู้บกพร่องทางการได้ยิน โดยภาษามือที่สามารถนำมาใช้ได้ ต้องเป็นภาษามือที่มีการเคลื่อนไหวน้อยที่สุด

1.6 นิยามเชิงปฏิบัติการ

โปรแกรมแปลภาษามือ หมายถึง โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ

บทที่ 2

เอกสารที่เกี่ยวข้อง

โครงการเรื่อง โปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ
คณะผู้จัดทำได้ศึกษาค้นคว้าข้อมูล และเอกสารที่เกี่ยวข้องดังนี้

2.1 ผู้บกพร่องทางการได้ยิน

ผู้บกพร่องทางการได้ยิน ได้แก่ บุคคลที่สูญเสียการได้ยินตั้งแต่ระดับหูตึงน้อยจนถึงหูหนวก
ซึ่งแบ่งเป็น 2 ประเภท ดังนี้

1) คนหูหนวก หมายถึง บุคคลที่สูญเสียการได้ยินมากจนไม่สามารถเข้าใจการพูด
ผ่านทาง การได้ยินไม่ว่าจะใส่หรือไม่ใส่เครื่องช่วยฟัง ซึ่งโดยทั่วไปหากตรวจการได้ยินจะมีการสูญเสีย
การได้ยิน 90 เดซิเบลขึ้นไป

2) คนหูตึง หมายถึง บุคคลที่มีการได้ยินเหลืออยู่เพียงพอที่จะได้ยินการพูดผ่าน
ทางการได้ยิน โดยทั่วไปจะใส่เครื่องช่วยฟัง ซึ่งหากตรวจวัดการได้ยินจะมีการสูญเสียการได้ยิน
น้อยกว่า 90 เดซิเบลลงมาถึง 26 เดซิเบล

2.2 ภาษามือ

คำนิยาม ‘คนหูหนวก’ และ ‘ภาษามือ คำนิยาม “คนหูหนวก” ในมุมมองทาง การแพทย์
และสังคม ได้แยกอธิบายคำว่า “คนหูหนวก” ในภาษาอังกฤษ ออกเป็น 2 คำคือ Deaf ที่มีตัวอักษร
ตัวใหญ่ “D” และ deaf ที่มีตัวอักษรตัวเล็ก “d” คำนิยามทั้งสอง คำในภาษาอังกฤษมีความหมาย
แตกต่างกัน (Woodward, 1975 cf. Padden, 1980 : 90) โดย Deaf หมายถึง คนหูหนวกที่มี
วัฒนธรรมคนหูหนวก มีภาษามือที่เป็นภาษาของคนหูหนวกเหมือนภาษาพูด ของคนที่มีการได้ยิน
ส่วน deaf หมายถึง คนหู หนวกที่ไม่มีวัฒนธรรมคนหูหนวก ใช้ภาษาพูดที่ได้รับการฝึกพูดจากแพทย์
หรือผู้ที่รักษาด้านความ ผิดปกติของการได้ยิน และสนับสนุนให้คนหู หนวกใช้เครื่องช่วยฟัง ฟังกูด
หรือผ่าตัดเพื่อรักษา ความพิการทางการได้ยิน ให้ใช้ประสาทหูเทียม ในขณะที่คนหูหนวกที่มี
วัฒนธรรมหูหนวก จะได้รับการสนับสนุนให้เรียนภาษามือเป็นภาษาแรก Monaghan (2002 : 72)
คำนิยามของ “ภาษามือ” นักภาษาศาสตร์ และนักมานุษยวิทยาได้ให้ความหมายไว้ว่า “ภาษามือ

เป็นภาษาของชุมชนคนหูหนวก เป็นภาษาที่มี ระบบท่าทางการเคลื่อนไหวร่างกาย สีหน้า ท่ามือ และ ตำแหน่งต่าง ๆ ของร่างกาย” (Stokoe 1960; Brennan, 1992; Liddell, 1980)

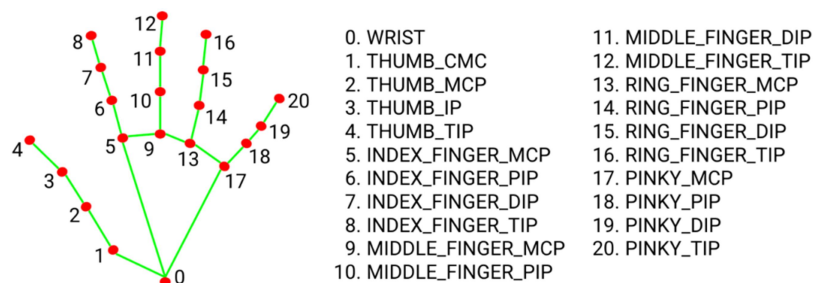
2.3 ภาษา Python

ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้ เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลาย ประเภท โดยไม่ได้จำกัดอยู่ทำงานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก

2.4 Hand tracking

การติดตามและตรวจจับมือ (Hand tracking) เป็นกระบวนการที่ใช้ในการระบุและติดตาม ตำแหน่งและการกระทำของมือในวิดีโอหรือภาพเคลื่อนไหว การพัฒนาระบบติดตามมือมี วัตถุประสงค์หลายอย่าง เช่น ใช้ในการควบคุมและปรับค่าต่างๆ ในแอปพลิเคชันเกม หรือใช้ในการ แสดงและควบคุมอนิเมชัน นอกจากนี้ยังมีการใช้งานในการทำความเข้าใจและวิเคราะห์ภาษามือ เช่น ในการแปลภาษามือเป็นข้อความหรือคำพูด

Hand Tracking เริ่มต้นจากการตรวจจับฝ่ามือด้วย Palm Detection Model และจึงระบุ ตำแหน่งสำคัญของมือจำนวน 21 จุด ผ่านโปรแกรมที่เรียกว่า Hand Landmark Model ที่จำลอง ท่าทางของมือจากการตรวจจับภาพที่รับเข้ามา



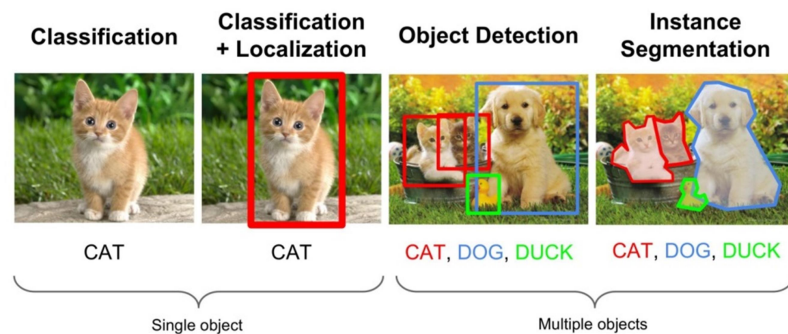
ภาพที่ 2.1 ภาพแสดงจุด Landmarks ทั้ง 21 จุดบนมือ

ที่มา : <https://www.ultimatepython.co/post/hand-tracking-python>

นอกจากนี้มีเทคโนโลยี และไลบรารีที่สามารถใช้ในการพัฒนาระบบติดตาม และตรวจจับมือได้ ตัวอย่าง ไลบรารีที่ได้รับความนิยมสำหรับการติดตาม และตรวจจับมือด้วย Python ได้แก่

2.4.1 OpenCV

OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License) ความสามารถ ของ OpenCV ก็คือสามารถช่วยจัดการวิดีโอหรือรูปภาพ เพื่อให้เกิดการเปลี่ยนแปลงกับรูปภาพ เช่น ทำให้ภาพชัดขึ้น ทำให้เบลอ ลดสัญญาณรบกวน (noise) ในรูปภาพ จากแหล่งที่มาของรูปภาพต่าง ๆ เช่น webcam ภาพถ่ายวิดีโอ หรือกล้องวงจรปิด นอกจากการปรับปรุงภาพแล้วนั้น ยังมีการใช้เทคนิคการประมวลผลภาพที่ทำให้รับรู้กับวัตถุ นั้น ๆ เช่น การแบ่งประเภท Object ว่าเป็น สุนัข แมว คน รถยนต์ เป็นต้นไปจนถึงการนำไปใช้งานที่ซับซ้อนมากขึ้น เช่น การจดจำใบหน้าของคน เช่น การที่ตำรวจตามหาคนร้ายจากกล้องวงจรปิด



ภาพที่ 2.2 การแยกประเภทของ Object

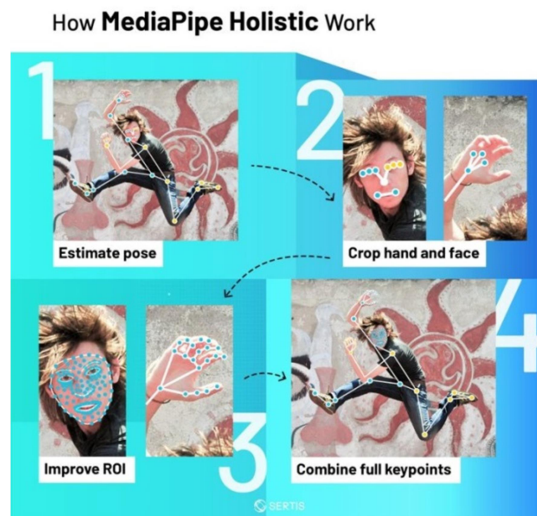
ที่มา : <https://www.dominodatalab.com/blog/getting-started-witn-open-cv>

2.4.2 MediaPipe

MediaPipe Holistic คือโพลีโลยีล้ำสมัยที่สามารถตรวจจับท่าทาง มือ และใบหน้าของมนุษย์ในเวลาเดียวกัน และรองรับการใช้งานในแบบที่ไม่เคยมีแพลตฟอร์มไหนทำได้มาก่อน โซลูชันนี้จะใช้ Pipeline แบบใหม่ที่ประกอบด้วย การตรวจจับท่าทาง หน้า และมือที่ปรับแต่งให้ดีที่สุดเพื่อให้ทำงานได้เรียลไทม์ โดยการใช้การโอนถ่ายหน่วยความจำระหว่าง Inference Backend

ซึ่ง Pipeline จะรวมรูปแบบการปฏิบัติการและการประมวลผลที่ต่างกันตามการตรวจจับภาพแต่ละส่วนเข้าด้วยกัน และจะได้เป็นโซลูชันแบบครบวงจรที่ใช้งานได้แบบเรียลไทม์และสม่ำเสมอ ใช้การทำงานแลกเปลี่ยนกันระหว่างการตรวจจับทั้งสามจุด โดยประสิทธิภาพของการทำงานจะขึ้นอยู่กับความเร็วและคุณภาพของการแลกเปลี่ยนข้อมูล เมื่อรวมการตรวจจับทั้งสามเข้าด้วยกัน จะได้เป็นโพลีโลยีที่ทำงานร่วมกันเป็นหนึ่งเดียว โดยสามารถจับ Keypoints ของภาพเคลื่อนไหวได้ถึง 540+ จุด (ส่วนของท่าทาง 33 จุด มือข้างละ 21 จุด และส่วนใบหน้า 468 จุด) ซึ่งเป็นระดับที่ไม่สามารถทำได้มาก่อน และสามารถประมวลผลได้เกือบจะเรียลไทม์ในการแสดงผลทางโทรศัพท์มือถือ เป็นหนึ่งในซอฟต์แวร์ของแพลตฟอร์ม MediaPipe และรองรับการใช้งานทั้งในโทรศัพท์มือถือ (ทั้งระบบ Android และ iOS) และบนคอมพิวเตอร์ นอกจากนี้ Google ยังเปิดให้ใช้ MediaPipe APIs แบบพร้อมใช้งาน สำหรับการใช้งานกับ Python และ JavaScript เพื่อทำให้เทคโนโลยีนี้เข้าถึงได้สะดวก

การทำงานของ MediaPipe Holistic ประมวลผลโดยการนำโมเดลของท่าทาง ใบหน้า และมือมารวมกัน ซึ่งทั้งสามส่วนได้รับการปรับปรุงภาพให้เข้ากับโดเมนของตนเองที่สุด แต่เนื่องจากลักษณะการทำงานเฉพาะของสามส่วนที่ต่างกัน ทำให้ข้อมูลที่ใช้ได้กับส่วนหนึ่งอาจไม่เข้ากับส่วนอื่น ยกตัวอย่างเช่น โมเดลการระบุท่าทาง อาจต้องการเฟรมวิดีโอที่มีความละเอียดที่ต่ำ แต่เมื่อต้องตัดส่วนของมือและหน้าจากภาพเพื่อส่งต่อไปยังโมเดลต่อไป ความละเอียดของภาพก็อาจต่ำเกินไปจนไม่สามารถประมวลผลได้แม่นยำ ด้วยเหตุนี้ MediaPipe Holistic จึงออกแบบมาในรูปแบบของ Pipeline ที่มีหลายขั้นตอน ซึ่งประมวลผลในแต่ละส่วนโดยต้องใช้ความละเอียดของภาพที่ต่างกัน



ภาพที่ 2.3 ภาพแสดงการทำงานของ MediaPipe

ที่มา : <https://www.sertiscorp.com/th/post/mediapipe-holistic-อุปกรณ์ที่สามารถจับการเคลื่อนไหว>

MediaPipe Holistic จะระบุท่าทางของมนุษย์โดยใช้โมเดลตรวจจับท่าทางและโมเดลระบุ Keypoint หลังจากนั้นจึงนำ Keypoint ที่ระบุได้มาแบ่งออกเป็น 3 จุดสนใจ (Region of Interest: ROI) ครอบคลุมส่วนที่เป็นแขน 2 ข้าง และส่วนหน้า แล้วจึงใช้ส่วนที่ครอบคลุมมาแทนเพื่อเพิ่มความละเอียดของจุดนั้น จากนั้น Pipeline จะทำการครอบเฟรมที่มีความละเอียดสูงสุดของจุด ROI ทั้งสองจุด แล้วจึงใช้กับโมเดลที่ตรวจจับส่วนของใบหน้าและมือเพื่อระบุตำแหน่ง Keypoints ตามส่วนต่าง ๆ และขั้นตอนสุดท้ายจะเป็นการนำ Keypoint ที่ได้มารวมกับ Keypoint ของโมเดลท่าทางในตอนต้น รวมกันเป็น 540 keypoint+ การจะทำให้การระบุ ROI มีประสิทธิภาพขึ้นนั้น ต้องใช้ระบบตรวจจับแบบเดียวกับระบบที่ใช้ในอุปกรณ์ที่ตรวจจับใบหน้าหรือแขนเพียงอย่างเดียว ซึ่งจะใช้การอนุมานว่าวัตถุที่ตรวจจับไม่ได้มีการขยับมากนัก โดยจะใช้เฟรมก่อนหน้าเพื่อคาดการณ์การระบุตำแหน่งของวัตถุในเฟรมต่อไป อย่างไรก็ตาม ถ้าวัตถุขยับเร็วเกินไป ตัวติดตามตำแหน่งอาจผิดพลาดซึ่งทำให้ตัวตรวจจับอาจต้องตรวจจับตำแหน่งในภาพใหม่อีกครั้ง MediaPipe Holistic จะใช้การคาดการณ์ท่าทางในทุก ๆ เฟรมล่วงหน้าไว้เป็นเสมือนจุด ROI เสริมไว้ก่อนตั้งแต่แรกเพื่อลดระยะเวลาในการตอบสนองของ Pipeline เวลาที่พบการเคลื่อนไหวที่รวดเร็วเกินไป นอกจากนี้วิธีช่วยให้โมเดลสามารถรักษาความสอดคล้องกันได้ทั่วทั้งรูปร่างและป้องกันไม่ให้เกิดความสับสนระหว่างมือซ้ายและมือขวา หรือส่วนที่ต่างกันของร่างกายในแต่ละเฟรม นอกจากนี้โดยปกติแล้วความละเอียดของเฟรมตรวจจับท่าทางนั้นจะต่ำเกินไปทำให้จุด ROI ของหน้า และมือนั้นมีความแม่นยำน้อยไปจนไม่สามารถให้แนวทางในการครอบตัดส่วนนั้นได้ ทำให้ต้องใช้โมเดลในการครอบตัดส่วนหน้า และมือที่มีขนาดเล็กแต่แม่นยำ เพื่อที่จะลดช่องว่างในเรื่องความแม่นยำระหว่างส่วนตัวและส่วนของมือกับใบหน้า โมเดลที่มีขนาดเล็กจะทำหน้าที่เป็นตัวแปลงพื้นที่ (Spatial Transformer) และใช้เวลาในการประมวลผลโมเดลน้อยลง 10 เปอร์เซ็นต์

ประสิทธิภาพการทำงาน MediaPipe Holistic จำเป็นต้องใช้การทำงานร่วมกันระหว่าง 8 โมเดลต่อเฟรม แบ่งเป็น โมเดลตรวจจับท่าทาง 1 โมเดล โมเดลตรวจจับ Landmark ของท่าทาง 1 โมเดล โมเดล re-crop เพื่อครอบรูปใหม่ 3 โมเดล และโมเดลในการระบุ Keypoint สำหรับมือและใบหน้าอีก 3 โมเดล ซึ่งในระหว่างที่ Google พัฒนาโซลูชันนี้ ได้ใช้ทั้ง Machine Learning และ Algorithm ในการคำนวณทั้งก่อนและหลังการประมวลผล ซึ่งโดยปกติแล้วการประมวลผลจะใช้เวลาค่อนข้างมากเนื่องจากความซับซ้อนของ Pipeline แต่ในกรณีของ MediaPipe Holistic ได้ย้ายกระบวนการการคำนวณช่วงก่อนเริ่มดำเนินงานทั้งหมดไปไว้ที่ GPU ทำให้ Pipeline สามารถทำงานได้เร็วขึ้นโดยเฉลี่ย 1.5 เท่า แต่อาจมีความแตกต่างกันในแต่ละอุปกรณ์ต่าง ๆ ดังนั้น MediaPipe Holistic จึงสามารถทำงานได้เกือบเรียลไทม์ แม้กระทั่งในอุปกรณ์ระดับกลางและในเบราว์เซอร์ คุณสมบัติของ Pipeline ที่ประกอบด้วยการทำงานหลายขั้นตอนนั้นช่วยเพิ่มประสิทธิภาพในการได้ใน 2 ส่วน หนึ่งคือเนื่องจากโมเดลส่วนมากเป็นโมเดลที่ทำงานแบบอิสระ

จึงสามารถใช้โมเดลเวอร์ชันที่เล็กลงหรือใหญ่ขึ้นก็ได้ ขึ้นอยู่กับความแม่นยำและประสิทธิภาพที่ต้องการ หรือจะปิดโมเดลนั้นไปเลยก็ได้ และสองคือเมื่ออุปกรณ์สามารถตรวจจับท่าทางได้ ก็จะสามารถคาดเดาได้ว่ามือกับหน้าอยู่ในพื้นที่เฟรมที่เชื่อมต่อกันด้วยหรือไม่ ทำให้ Pipeline สามารถข้ามขั้นตอนการระบุส่วนเหล่านั้นไปได้

2.5 PyGame

PyGame เป็นไลบรารีแบบเปิดเผยโค้ดโปรแกรมที่ออกแบบเพื่อใช้กับภาษาไพธอนเพื่อสร้างโปรแกรมประยุกต์ด้านสื่อผสมประเภทเกม โดยทำงานอยู่เหนือไลบรารี SDL2 และสามารถทำงานร่วมกับ OpenGL โดยไม่ขึ้นกับระบบปฏิบัติการ มีความสามารถรองรับการทำงานกับหน่วยประมวลผลแบบหลายแกน (Multi core CPUs) และมีการเพิ่มประสิทธิภาพการทำงานด้วยโค้ดภาษาซีและแอสเซมบลีทำให้มีความเร็วสูงกว่าการเขียนด้วยโค้ดภาษาไพธอนในรูปแบบปกติTTF นอกจากนี้ Pygame เป็นส่วนขยายหนึ่งของ Python เป็นระบบที่ครอบคลุมเกือบทุกสิ่งที่ต้องการในการพัฒนาเกม ตั้งแต่เกม Tetris ไปจนถึงเกม Civilization 4 หรือ 3D Shooters สามารถจัดการกับกราฟิก 2 มิติ, เพลง, อินพุต, แบบอักษรต่างๆ, การตรวจจับการชน, กราฟิก 3 มิติ (ร่วมกับ OpenGL), จอยสติ๊ก, ภาพยนตร์, และอื่นๆ ได้อีกจำนวนมาก

บทที่ 3

อุปกรณ์ และวิธีการทดลอง

การสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ มีวัสดุอุปกรณ์, ขั้นตอนการดำเนินงาน และกรอบแนวคิดดังต่อไปนี้

3.1 วัสดุอุปกรณ์

- 3.1.1 Laptop
- 3.1.2 Webcam camera
- 3.1.3 ลำโพง
- 3.1.4 ไมโครโฟน

3.2 ขั้นตอนการดำเนินงาน

3.2.1 ศึกษาข้อมูลเพื่อการออกแบบ และจัดทำโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ

- 1) ศึกษาข้อมูลเกี่ยวกับภาษามือ และ โมดูลต่างๆ เช่น OpenCV, MedaiPipe
- 2) ศึกษาหลักการทำงานร่วมกันของโมดูล OpenCV, MedaiPipe และ Pygame ร่วมกับภาษา Python

3.2.2 การออกแบบโปรแกรม และลงมือสร้างโปรแกรมแปลภาษามือ

การออกแบบโปรแกรมที่สามารถช่วยเหลือให้ผู้บกพร่องทางการได้ยินสื่อสารกับบุคคลปกติได้ จากการศึกษารายละเอียด และทฤษฎีที่เกี่ยวข้อง ได้ออกแบบส่วนประกอบของโปรแกรมมี 3 ส่วน ดังต่อไปนี้

- 1) ส่วนรับข้อมูล คือ ส่วนของ กล้อง webcam เมื่อตัวกล้องสามารถตรวจจับมือบนหน้าจอได้แล้ว จะรับค่าตามตำแหน่งต่าง ๆ ที่อยู่บนมือ และ นำไปจดจำว่า ตำแหน่งนั้นๆ มีความหมายตรงกับคำว่าอะไร และ จะเก็บข้อมูลเอาไว้ให้เป็นตัวแปร ซึ่งมีข้อจำกัดว่ามือที่แสดงบนหน้าจอต้องมีลักษณะการเคลื่อนไหวน้อยที่สุด
- 2) ส่วนส่งออกข้อมูล คือ เมื่อได้ตัวแปรมาแล้วตัวโปรแกรมจะทำการเรียกไฟล์เสียงที่กำหนดเอาไว้ตามตัวแปรนั้น ๆ

```
if fingerCount==1 :
    def play_mp3(file_path):
        pygame.mixer.init()
        pygame.mixer.music.load(file_path)
        pygame.mixer.music.play()

    if __name__ == "__main__":
        mp3_file_path = "1.mp3"
        play_mp3(mp3_file_path)
```

ภาพที่ 3.1 แสดงตัวอย่างการเก็บตัวแปร 1 แล้ว เรียกไฟล์เสียงที่ชื่อว่า “1.mp3”

3) การดำเนินการสร้างโปรแกรมที่สามารถช่วยเหลือให้ผู้บกพร่องทางการได้ยินสื่อสารกับบุคคลปกติ

3.2.3 การดำเนินการสร้างโปรแกรมแปลภาษามือ

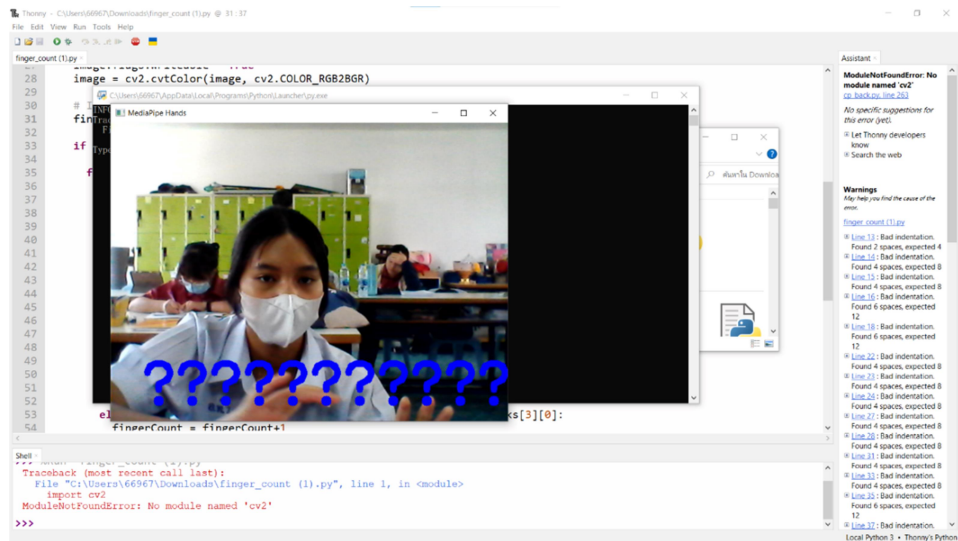
ภายหลังจากการศึกษาข้อมูลต่าง ๆ ที่เกี่ยวข้องกับโปรแกรม และภาษามือเรียบร้อยแล้วจึงดำเนินการสร้างโปรแกรมแปลภาษามือตามแบบที่กำหนด และ ทำการเพิ่มเติมบางส่วนที่จำเป็นสามารถแบ่งแต่ละส่วน ดังนี้

1) โครงสร้างของโปรแกรม มีขั้นตอน คือ เริ่มออกแบบโครงสร้างโปรแกรม ศึกษาการทำงานของโปรแกรมการรับข้อมูลในรูปแบบของภาษามือ และ การส่งออกข้อมูลในรูปแบบของเสียงที่สามารถทำงานร่วมกันได้

2) ใส่ข้อมูลต่าง ๆ ที่ต้องโปรแกรมใช้ ได้แก่ ไฟล์เสียงในจำนวน 10 ไฟล์ที่สอดคล้องกับท่าทางภาษามือที่โปรแกรมจะสามารถแปลได้

3) ทดสอบว่าโปรแกรมสามารถใช้งานได้จริง โดยการท่าทางภาษามือให้ตัวโปรแกรมแปล และ ให้โปรแกรมส่งออกข้อมูลในรูปแบบเสียงที่มีความหมายตรงกับภาษามือที่โปรแกรมรับค่าไป

4) หากโปรแกรมมีความผิดพลาด ให้ดำเนินการแก้ไข แล้วทดสอบว่าโปรแกรมสามารถใช้งานได้จริงอีกครั้ง



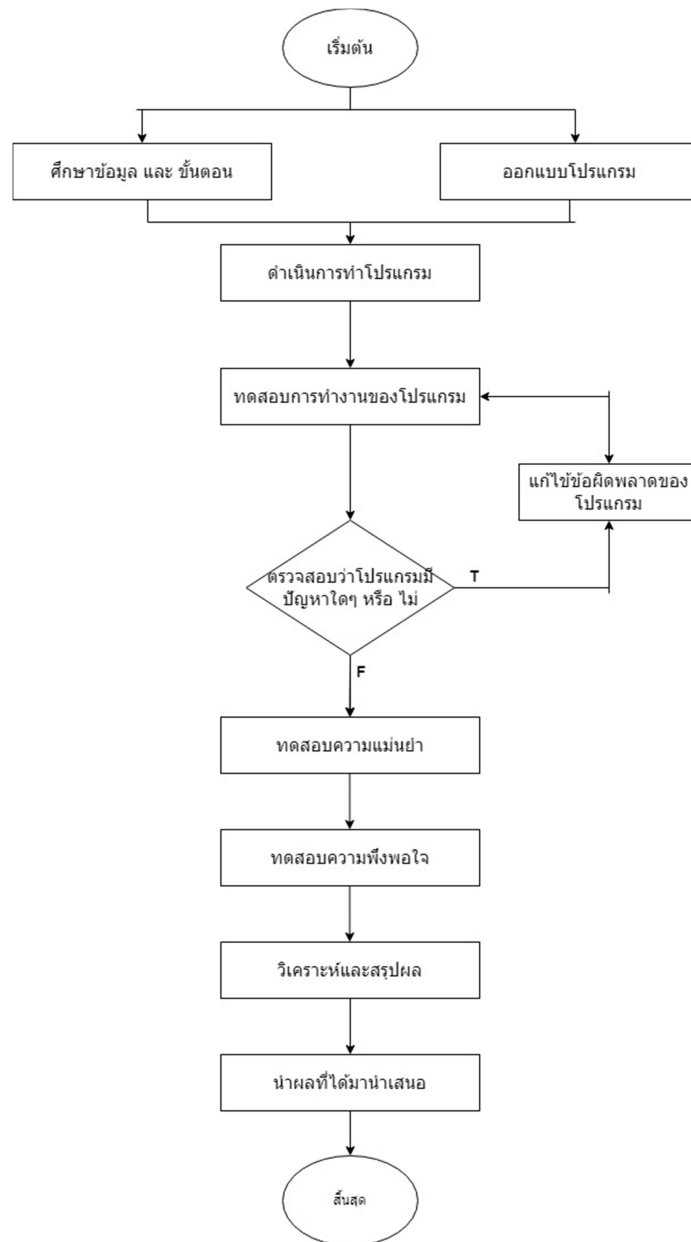
ภาพที่ 3.2 แสดงความผิดพลาดในการอ่านภาษามือของโปรแกรมก่อนดำเนินการแก้ไข

3.2.4 การทดสอบประสิทธิภาพของโปรแกรมแปลภาษามือ

- 1) นำโปรแกรมแปลภาษามือ ให้กลุ่มตัวอย่าง จำนวน 50 คน ทดลองใช้ และให้กลุ่มตัวอย่างทำแบบสอบถามความพึงพอใจ
- 2) รวบรวมแบบสอบถาม เพื่อนำข้อมูลมาวิเคราะห์ สรุปผล และ รายงานผลการประเมินความพึงพอใจของผู้ใช้งาน

3.3 กรอบแนวคิด

ในการทำโครงงานนี้ คณะผู้จัดทำได้ดำเนินการขั้นตอนต่าง ๆ ตามแผนผังขั้นตอนการดำเนินงาน โดยมีรายละเอียดดังนี้



ภาพที่ 3.3 แสดงแผนผังขั้นตอนการดำเนินการสร้างโปรแกรมแปลภาษามือ

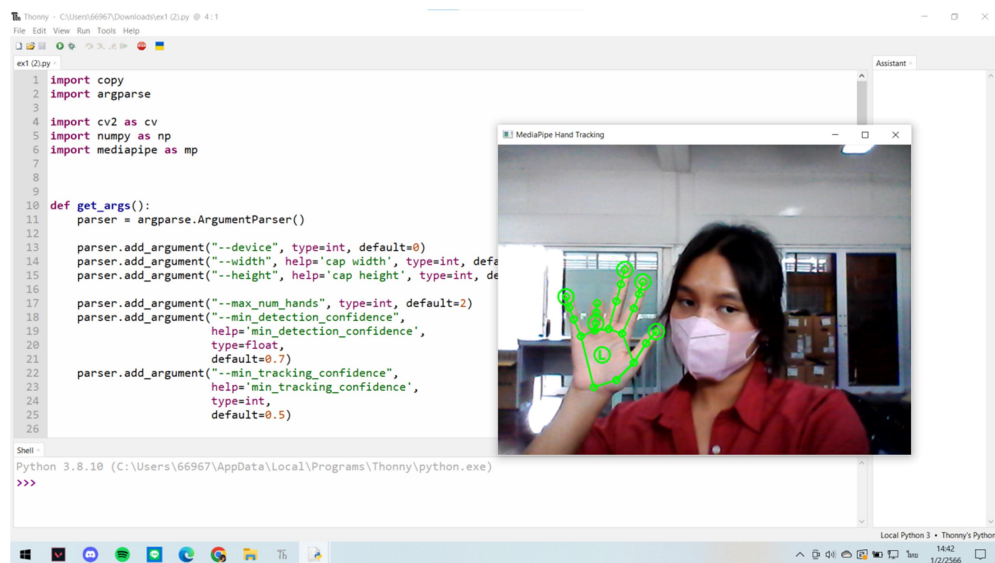
บทที่ 4

ผลการทดลอง

การสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อ 1. สร้างโปรแกรมที่สามารถช่วยเหลือนักบกพร่องทางการได้ยิน และ 2. สำนวความพึงพอใจของผู้ใช้งานโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ มีผลการทดลอง ดังต่อไปนี้

4.1 การสร้างโปรแกรมแปลภาษามือ

จากการศึกษาการทำงานร่วมกันของ โมดูล OpenCV, MediaPipe และ Pygame โดยผ่านการเขียนโปรแกรมในภาษา Python เพื่ออ่านค่าพิกัดบนมือที่แสดงท่าทาง และเก็บพิกัดท่าทางที่แสดงเอาไว้ในตัวแปร จากนั้นนำตัวแปรมาเรียกไฟล์เสียง ซึ่งสามารถนำมาประยุกต์สร้างโปรแกรมแปลภาษามือออกมาเป็นเสียงได้



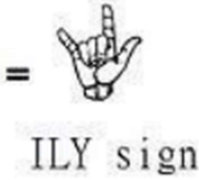
ภาพที่ 4.1 แสดงการทำงานของกรอ่านพิกัดบนมือ

4.2 การทดสอบประสิทธิภาพของโปรแกรมแปลภาษามือ

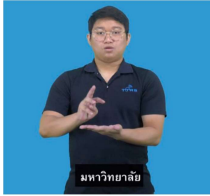
การทดสอบความแม่นยำในการส่งออกข้อมูล การทดสอบความแม่นยำของโปรแกรมแปลภาษามือ ทดสอบโดยการให้โปรแกรมอ่านภาษามือ แล้วตรวจสอบว่าเสียงที่ออกมาตรงกับ ความหมายของภาษามือหรือไม่ โดยทดสอบเป็นจำนวน 5 ครั้ง ต่อ 1 คำ

เงื่อนไข : หมายถึง ภาษามือที่แสดงมีความหมายตรงกับเสียงที่ส่งออกมา
 หมายถึง ภาษามือที่แสดงมีความหมายไม่ตรงกับเสียงที่ส่งออกมา

ตารางที่ 4.1 แสดงผลความแม่นยำของการทดสอบโปรแกรม

ลำดับ	ภาษามือที่แสดง	การทดสอบประสิทธิภาพ					การแปลผล
		1	2	3	4	5	
1							ตรง
2							ตรง
3							ตรง

ลำดับ	ภาษามือที่แสดง	การทดสอบประสิทธิภาพ					การแปลผล
		1	2	3	4	5	
4							ตรง
5							ตรง
6							ตรง
7							ตรง

ลำดับ	ภาษามือที่แสดง	การทดสอบประสิทธิภาพ					การแปลผล
		1	2	3	4	5	
8							ตรง
9							ตรง
10							ตรง

จากตาราง กำหนดท่าทาง และคำภาษามือจำนวน 10 คำ ได้แก่ คำว่า “เกิดอาการแพ้”, “วันนี้”, “ฉันรักคุณ”, “แก”, “ฉัน”, “รับสารภาพ”, “พลเมือง”, “มหาวิทยาลัย”, “การเลือกตั้ง” และ “ขอบคุณ” ซึ่งคำมีความหมายตรงกับท่าทาง

4.3 การสำรวจความพึงพอใจของผู้ใช้งานโปรแกรมแปลภาษามือ

1) ข้อมูลของผู้ทำแบบสำรวจความพึงพอใจ

จากการเก็บรวบรวมข้อมูลกลุ่มตัวอย่างที่ได้ทดลองใช้โปรแกรมแปลภาษามือ โดยวิธีการแจกแบบสำรวจ จำนวน 50 ฉบับ ได้ผลสรุปจากแบบสอบถาม ตามตารางดังนี้

ตารางที่ 4.2 แสดงข้อมูลผู้ตอบแบบสอบถามความพึงพอใจ

สถานภาพ	จำนวน (คน)	ร้อยละ
นักเรียนระดับชั้นมัธยมศึกษาตอนต้น	15	30.0
นักเรียนระดับชั้นมัธยมศึกษาตอนปลาย	18	36.0
บุคลากรในโรงเรียนอุทัยวิทยาคม	17	34.0
รวม	50	100.0

จากตารางที่ 4.2 พบว่า ผู้ประเมินความพึงพอใจเป็นนักเรียนชั้นมัธยมศึกษาตอนปลายมากที่สุด คิดเป็นร้อยละ 36.0 รองลงมา คือ บุคลากรในโรงเรียนอุทัยวิทยาคม คิดเป็นร้อยละ 34.0 และนักเรียนระดับชั้นมัธยมศึกษาตอนต้น คิดเป็นร้อยละ 30.0

2) การประเมินแบบสำรวจความพึงพอใจ

จากการประเมินแบบสำรวจความพึงพอใจของผู้ทดลองใช้โปรแกรมภาษามือ โดยวิธีการแจกแบบสำรวจ จำนวน 50 ฉบับ ได้ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน และการแปลความหมายความพึงพอใจของผู้ใช้ที่มีต่อโปรแกรมแปลภาษามือ ตามตารางดังนี้

ตารางที่ 4.3 แสดงค่าเฉลี่ย, ส่วนเบี่ยงเบนมาตรฐาน และการแปลผลความพึงพอใจของผู้ใช้งานที่มีต่อโปรแกรมแปลภาษามือ

ลำดับ	รายการ	ระดับความพึงพอใจ		
		$\bar{x}$	S.D.	การแปลผล
1.	ประสิทธิภาพในการใช้งานโปรแกรมแปลภาษามือ	4.20	0.32	มาก
2.	ความสะดวกในการใช้งานโปรแกรมแปลภาษามือ	4.38	0.60	มาก
3.	ความรวดเร็วของโปรแกรมแปลภาษามือ	4.24	0.58	มาก
4.	ความแม่นยำของโปรแกรมภาษามือ	4.30	0.74	มาก
5.	ความคิดสร้างสรรค์ และความแปลกใหม่	4.58	0.98	มากที่สุด
รวม		4.34	0.64	มาก

จากตารางที่ 4.3 พบว่า ความพึงพอใจของผู้ใช้โดยภาพรวมอยู่ในระดับมาก ($\bar{x} = 4.34$, S.D. = 0.64) เมื่อพิจารณาเป็นรายลำดับ พบว่า ด้านความคิดสร้างสรรค์ และความแปลกใหม่ ความพึงพอใจอยู่ในระดับมากที่สุด คือ ($\bar{x} = 4.58$, S.D. = 0.98) รองลงมา คือ ความพึงพอใจในระดับมาก จำนวน 4 ลำดับ ได้แก่ 1. ความสะดวกในการใช้งานโปรแกรมแปลภาษามือ ($\bar{x} = 4.38$, S.D. = 0.60), 2. ความแม่นยำของโปรแกรมภาษามือ ($\bar{x} = 4.30$, S.D. = 0.74), 3. ความรวดเร็วของโปรแกรมแปลภาษามือ ($\bar{x} = 4.24$, S.D. = 0.58) และ 4. ประสิทธิภาพในการใช้งานโปรแกรมแปลภาษามือ ($\bar{x} = 4.20$, S.D. = 0.32)

บทที่ 5

สรุป อภิปรายผล และข้อเสนอแนะ

ในการจัดทำโครงงานคอมพิวเตอร์ เรื่อง การสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยิน สื่อสารภาษามือกับบุคคลปกติ จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อ 1. สร้างโปรแกรมที่สามารถช่วยเหลือผู้บกพร่องทางการได้ยิน และ 2. สำรวจความพึงพอใจของผู้ใช้งานโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติ คณะผู้จัดทำสรุป และอภิปรายผลไว้ดังนี้

5.1 สรุปผลการทดลอง

5.1.1 สามารถสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติที่ไม่ทราบภาษามือได้

5.1.2 ความพึงพอใจของผู้ใช้โดยรวมที่มีต่อโปรแกรมแปลภาษามือมีค่าเฉลี่ยรวมอยู่ที่ $\bar{x} = 4.34$, S.D.= 0.64 ซึ่งอยู่ในระดับมาก

5.2 อภิปรายผล

5.2.1 จากการศึกษาการทำงานร่วมกันของ โมดูล OpenCV, MediaPipe และ Pygame โดยใช้ Python เพื่อนำมาประยุกต์ใช้ Hand Tracking ในการสร้างโปรแกรมเพื่อช่วยผู้บกพร่องทางการได้ยินสื่อสารภาษามือกับบุคคลปกติได้

5.2.2 จากการสำรวจความพึงพอใจของผู้ทดลองใช้โปรแกรมจำนวน 50 คน พบว่า ผู้ทดสอบส่วนมากมีความพึงพอใจในด้านความคิดสร้างสรรค์อยู่ในระดับมากที่สุด, ความสะดวกสบายในการใช้งานโปรแกรม, ความแม่นยำของโปรแกรม, ความรวดเร็วในการใช้งานโปรแกรม และ ประสิทธิภาพในการใช้งานโปรแกรมแปลภาษามืออยู่ในระดับมาก และได้รับความพึงพอใจโดยรวม อยู่ในระดับมาก ซึ่งมีข้อจำกัดในเรื่องการเข้าถึงของโปรแกรมที่ยังไม่สามารถเข้าถึงได้ในบุคคลทั่วไป

5.3 ข้อเสนอแนะ

- 5.3.1 เพิ่มจำนวนคำภาษามือที่สามารถแปลได้
- 5.3.2 เพิ่มความแม่นยำของโปรแกรมให้มากยิ่งขึ้น
- 5.3.3 พัฒนาโปรแกรมให้สามารถใช้งานได้สะดวก และเข้าถึงได้ง่ายยิ่งขึ้น

บรรณานุกรม

- จารุต บุศราทิจ. (2563). *การปฏิสัมพันธ์ด้วยPyGame*. สืบค้นเมื่อ 18 สิงหาคม 2565, จาก <https://www.etteam.com/productPi/BOOK%20RASPBERRY%20PI3/ch17.pdf>
- ณัฐกานต์ จันทรา. (2562). *มาสร้างแอปด้วย Pygame กันเถอะ*. สืบค้นเมื่อ 18 สิงหาคม 2565, จาก <https://medium.com/@nut.ch40/มาสร้างเกมด้วย-pygame-กันเถอะ-ขั้นตอนการติดตั้ง-860ea13349f5>
- ณัฐกานต์ จันทรา. (2562). *OpenCV คืออะไร?*. สืบค้นเมื่อ 18 สิงหาคม 2565, จาก <https://medium.com/@nut.ch40/opencv-คืออะไร-8771e2a4c414>
- มหาวิทยาลัยสุโขทัยธรรมมาธิราช. (2562). *ความรู้เกี่ยวกับคนพิการ*. สืบค้นเมื่อ 18 สิงหาคม 2565, จาก <https://oes.stou.ac.th/dss/knowledge/>
- ราชภัฏ บัญญา. (2560). *ภาษามือ : ภาษาของคนหูหนวก*. สืบค้นเมื่อ 24 สิงหาคม 2565, จาก <https://rs.mahidol.ac.th/rs-journal/vol.4/v.4-1-005.pdf>
- ศรายุทธ นนท์ศิริ. (2564). *ภาษาโปรแกรม Python คืออะไร ?*. สืบค้นเมื่อ 24 สิงหาคม 2565, จาก <https://www.9experttraining.com/articles/python-คืออะไร>
- Sertis. (2564). *MediaPipe Holistic อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า มือ และท่าทางได้ในเวลาเดียวกัน*. สืบค้นเมื่อ 24 สิงหาคม 2565, จาก <https://sertiscorp.medium.com/mediapipe-holistic-อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า-มือ-และท่าทางได้ในเวลาเดียวกัน-e1185469e111>
- Ultimate Python. (2564). *ตรวจจับมือผ่าน webcam 10 บรรทัด Python Hand Tracking*. สืบค้นเมื่อ 18 สิงหาคม 2565, จาก <https://www.ultimatepython.co/post/hand-tracking-python>

ภาคผนวก

ภาคผนวก ก

ขั้นตอนการศึกษาข้อมูล และขั้นตอนการดำเนินงาน



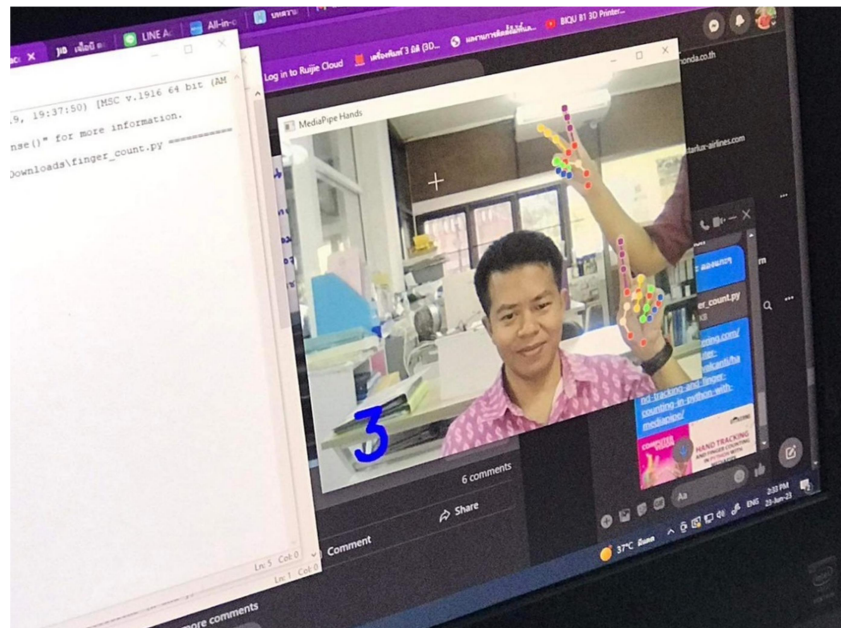
ภาพที่ 1 การศึกษา และรวบรวมคำภาษามือที่สามารถนำมาใช้ในการพัฒนาโปรแกรม



ภาพที่ 2 การศึกษาภาษามือคำว่า “การเลือกตั้ง”



ภาพที่ 3 การศึกษาภาษามือคำว่า “พลเมือง”



ภาพที่ 4 การศึกษาหลักการทำงานของโปรแกรม โดยมีครูให้คำปรึกษา

ภาคผนวก ข
ขั้นตอนการเขียนโปรแกรม



ภาพที่ 5 การเขียนโปรแกรมภาษา python

```
import cv2
import mediapipe as mp
import pygame

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands
```

ภาพที่ 6 การนำเข้าโมดูล และ ตัวแปร ที่นำมาประยุกต์ใช้ในโปรแกรม

```
cap = cv2.VideoCapture(0)
with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
    while cap.isOpened():
        success, image = cap.read()
        if not success:
            print("Ignoring empty camera frame.")
            # If loading a video, use 'break' instead of 'continue'.
            continue

        # To improve performance, optionally mark the image as not writeable to
        # pass by reference.
        image.flags.writeable = False
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results = hands.process(image)

        # Draw the hand annotations on the image.
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Initially set finger count to 0 for each cap
        fingerCount = 0
```

ภาพที่ 7 โปรแกรมของตัวรับข้อมูลเข้าจากกล้อง webcam

```
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_hand_landmarks:
        # Get hand index to check label (left or right)
        handIndex = results.multi_hand_landmarks.index(hand_landmarks)
        handLabel = results.multi_handedness[handIndex].classification[0].label

        # Set variable to keep landmarks positions (x and y)
        handLandmarks = []

        # Fill list with x and y positions of each landmark
        for landmarks in hand_landmarks.landmark:
            handLandmarks.append([landmarks.x, landmarks.y])

        # Test conditions for each finger: Count is increased if finger is
        # considered raised.
        # Thumb: TIP x position must be greater or lower than IP x position,
        # depending on hand label.
        if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:
            fingerCount = fingerCount+1
        elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:
            fingerCount = fingerCount+1
```

ภาพที่ 8 โปรแกรมการกำหนดการอ่านค่ามือซ้าย และมือขวา

```
# Other fingers: TIP y position must be lower than PIP y position,
# as image origin is in the upper left corner.
if handLandmarks[8][1] < handLandmarks[6][1]:      #Index finger
    fingerCount = fingerCount+1
if handLandmarks[12][1] < handLandmarks[10][1]:    #Middle finger
    fingerCount = fingerCount+1
if handLandmarks[16][1] < handLandmarks[14][1]:    #Ring finger
    fingerCount = fingerCount+1
if handLandmarks[20][1] < handLandmarks[18][1]:    #Pinky
    fingerCount = fingerCount+1
```

ภาพที่ 9 โปรแกรมกำหนดพิกัดบนนิ้วมือทั้ง 5 นิ้ว

```
if fingerCount==1 :
    def play_mp3(file_path):
        pygame.mixer.init()
        pygame.mixer.music.load(file_path)
        pygame.mixer.music.play()

    if __name__ == "__main__":
        mp3_file_path = "1.mp3"
        play_mp3(mp3_file_path)
if fingerCount==2 :
    def play_mp3(file_path):
        pygame.mixer.init()
        pygame.mixer.music.load(file_path)
        pygame.mixer.music.play()

    if __name__ == "__main__":
        mp3_file_path = "2.mp3"
        play_mp3(mp3_file_path)
```

ภาพที่ 10 โปรแกรมการเก็บค่าตัวแปรเพื่อเรียกใช้ไฟล์เสียง

```
import pygame

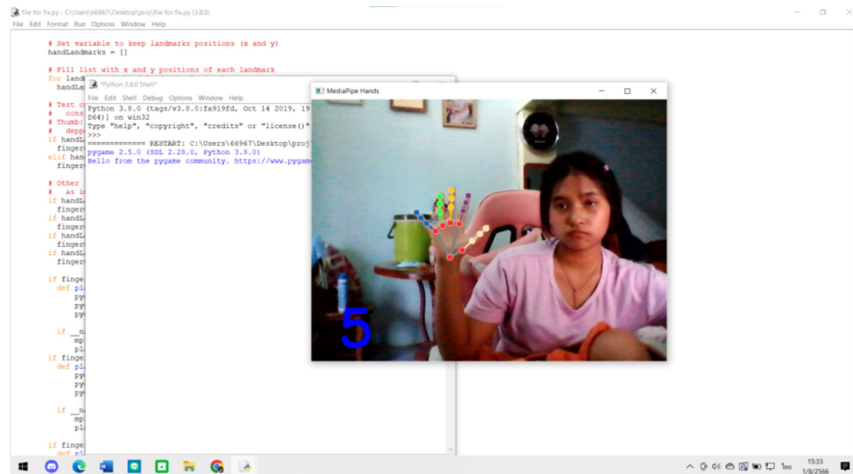
def play_mp3(file_path):
    pygame.mixer.init()
    pygame.mixer.music.load(file_path)
    pygame.mixer.music.play()

if __name__ == "__main__":
    mp3_file_path = "3.mp3"
    play_mp3(mp3_file_path)
```

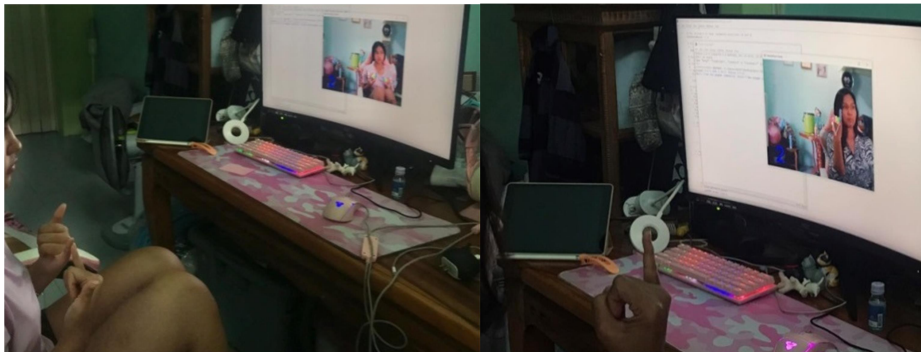
ภาพที่ 11 โปรแกรมการเรียกใช้ไฟล์เสียงโดยใช้โมดูล PyGame

ภาคผนวก ค

ขั้นตอนการทดสอบประสิทธิภาพของโปรแกรม



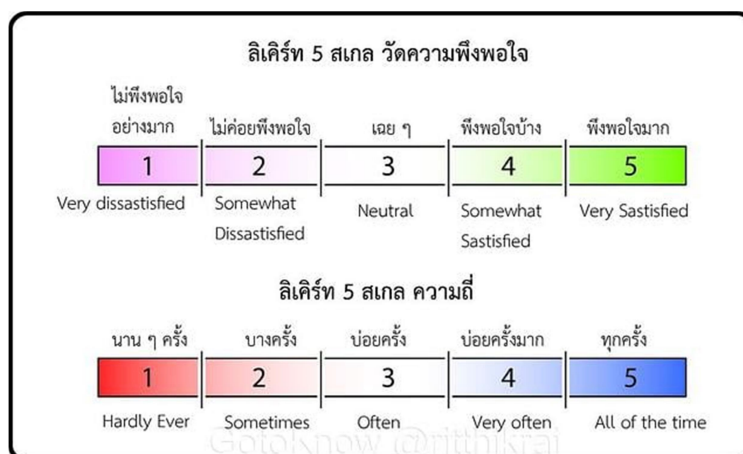
ภาพที่ 12 หน้าต่างโปรแกรมขณะดำเนินการแปลภาษามือ



ภาพที่ 13 การทดสอบระบบโปรแกรม โดยใช้ภาษามือคำว่า “วันนี้” และ “เกิดอาการแพ้”



ภาพที่ 14 การทดสอบระบบโปรแกรม และประเมินความพึงพอใจ



ภาพที่ 15 มาตรวัดเจตคติแบบลิเคิร์ท (วัดความพึงพอใจ)

ที่มา : <https://www.gotoknow.org/posts/659229>