การสร้างเว็บไซต์สำหรับผู้พิการทางการได้ยิน

Website Development for People with Hearing Disabilities

นายธนพนธ์ จงเพิ่มวัฒนะผล 664230010

66/45

โครงงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 7203602

โครงงานด้านเทคโนโลยีสารสนเทศ 2

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

ภาคเรียนที่ 1 ปีการศึกษา 2568

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การสื่อสารเป็นหัวใจสำคัญของการดำเนินชีวิตประจำวัน ไม่ว่าจะเป็นการเรียน การทำงาน การเข้าสังคม หรือการเข้าถึงบริการต่าง ๆ โดยเฉพาะในยุคปัจจุบันที่ความรวดเร็วและความเข้าใจที่ตรงกันมีความจำเป็นสูง อย่างไรก็ตาม สำหรับผู้พิการทางการได้ยินและการพูด มักพบอุปสรรคสำคัญในการสื่อสารกับบุคคลทั่วไป เนื่องจากคนส่วนมากไม่สามารถใช้หรือเข้าใจภาษามือได้ ส่งผลให้เกิดข้อจำกัดในการดำเนินชีวิตและโอกาสทางสังคม

ภาษามืออเมริกัน (American Sign Language: ASL) เป็นหนึ่งในภาษามือที่ใช้กันอย่างแพร่หลาย มีระบบการใช้ท่าทางมือเพื่อแทนตัวอักษร คำ หรือประโยค ซึ่งเป็นเครื่องมือสำคัญในการสื่อสารของผู้พิการทางการได้ยิน อย่างไรก็ตาม การที่บุคคลทั่วไปไม่สามารถเข้าใจภาษามือได้อย่างถ่องแท้นำไปสู่ปัญหาการสื่อสารที่ไม่ตรงกัน

ปัจจุบัน เทคโนโลยีด้าน Computer Vision และ Machine Learning ได้พัฒนาอย่างก้าวกระโดด โดยเฉพาะการตรวจจับวัตถุ การรู้จำภาพ และการรู้จำท่าทาง (Gesture Recognition) ซึ่งสามารถนำมาประยุกต์ใช้ในการพัฒนาระบบต้นแบบที่สามารถแปลภาษามือแบบ Real-time ผ่านกล้อง Webcam เพื่อช่วยให้ผู้พิการทางการได้ยินสามารถสื่อสารกับบุคคลทั่วไปได้สะดวกมากขึ้น

ดังนั้น การวิจัยและพัฒนาระบบแปลภาษามืออเมริกัน (ASL Recognition System) โดยใช้เทคนิคการประมวลผลภาพ (Image Processing) และการเรียนรู้ของเครื่อง (Machine Learning) จะช่วยลดช่องว่างการสื่อสาร เพิ่มโอกาสทางการศึกษา การทำงาน และการเข้าสังคมของผู้พิการทางการได้ยินอีกทั้งยังเป็นตัวอย่างการนำเทคโนโลยี AI มาประยุกต์ใช้เพื่อสังคมในเชิงบวก

1.2 แนวคิดในการแก้ไขปัญหา

แนวคิดของโครงงานนี้คือการพัฒนาระบบที่สามารถ ตรวจจับมือ (Hand Detection) และจำแนกท่าทางของมือ (Hand Gesture Classification) ให้เป็นตัวอักษรภาษาอังกฤษตามมาตรฐาน ASL ได้แบบ Real-time โดยใช้กล้อง Webcam เป็นอุปกรณ์ในการเก็บข้อมูลภาพ

1.2.1 การจับภาพ (Image Acquisition):ใช้กล้อง Webcam เพื่อเก็บภาพมือของผู้ใช้

1.2.2 การประมวลผลเบื้องต้น (Preprocessing): ใช้ MediaPipe ในการตรวจจับจุด Landmark ของมือ (เช่น ปลายนิ้ว ข้อมือ และข้อนิ้ว) ซึ่งสามารถบ่งบอกท่าทางของมือได้โดยไม่ต้องใช้ภาพเต็ม

1.2.3 การแปลงเป็นคุณลักษณะ (Feature Extraction): นำค่าพิกัด (x, y, z) ของจุด Landmark มาจัดรูปแบบเป็นเวกเตอร์คุณลักษณะ (Feature Vector)

1.2.4 การจำแนก (Classification): ใช้โมเดล Machine Learning ที่ผ่านการฝึก เช่น Random Forest Classifier เพื่อตัดสินว่าท่าทางมือนั้นแทนตัวอักษรใดของ ASL

1.2.5 การแสดงผล (Output): แสดงผลลัพธ์ตัวอักษรบนหน้าจอ พร้อมทั้งสะสมเป็นข้อความ (Sentence) และเพิ่มฟังก์ชันพิเศษ เช่น การเว้นวรรค (Space) และการลบตัวอักษร (Delete)

ด้วยแนวทางนี้ ระบบสามารถแปลภาษามืออเมริกันได้แบบทันที (Real-time) โดยใช้เพียงคอมพิวเตอร์ที่มีกล้อง Webcam และไม่ต้องใช้เซ็นเซอร์พิเศษเพิ่มเติม

1.3 วัตถุประสงค์ของระบบ

1.3.1 เพื่อช่วยเพิ่มโอกาสและความสะดวกในการสื่อสารของผู้พิการทางการได้ยินกับบุคคลทั่วไป

1.3.2 เพื่อทดสอบความเป็นไปได้ของระบบการรู้จำท่าทางมือโดยใช้กล้อง Webcam ซึ่งมีต้นทุนต่ำและเข้าถึงได้ง่าย

1.4 ขอบเขตการศึกษา

เพื่อให้การดำเนินงานของโครงงานเป็นไปอย่างมีประสิทธิภาพและสามารถวัดผลได้ชัดเจน จึงกำหนดขอบเขตการศึกษาไว้ดังนี้:

1.4.1 ขอบเขตด้านข้อมูล:

1.4.1.1 ระบบจะใช้ข้อมูลจากชุดข้อมูลภาษามือ American Sign Language (ASL) ที่แทนตัวอักษรภาษาอังกฤษ A-Z และสัญลักษณ์พื้นฐาน เช่น 'Space' และ 'Delete' เท่านั้น

1.4.1.2 การเก็บข้อมูลและการประมวลผลจะเน้นที่ท่าทางมือเดียว (Single-hand gestures) และไม่รวมท่าทางที่ต้องใช้สองมือพร้อมกัน

1.4.2 ขอบเขตด้านฟังก์ชันการทำงาน:

1.4.2.1 ระบบจะสามารถรับข้อมูลภาพจากกล้องเว็บแคม (Webcam) ที่เชื่อมต่อกับคอมพิวเตอร์ได้แบบเรียลไทม์

1.4.2.2 สามารถแสดงโครงกระดูก (Skeleton) ของมือที่ตรวจจับได้บนหน้าจอ เพื่อให้ผู้ใช้มองเห็นการประมวลผลของระบบ

1.4.2.3 สามารถแปลงท่าทางมือที่ตรวจจับได้เป็นตัวอักษรและนำไปประกอบเป็นคำหรือประโยคในช่องข้อความ

1.4.2.4 มีฟังก์ชันอำนวยความสะดวก เช่น การลบข้อความ (Delete) และการเว้นวรรค (Space)

1.4.3 ขอบเขตด้านแพลตฟอร์ม:

ระบบถูกพัฒนาขึ้นในรูปแบบของ Web Application โดยใช้เฟรมเวิร์ก Flask เพื่อให้สามารถทำงานได้บนเบราว์เซอร์มาตรฐานต่าง ๆ เช่น Google Chrome, Firefox หรือ Microsoft Edge โดยไม่ต้องติดตั้งโปรแกรมเพิ่มเติม

1.4.4 ขอบเขตด้านระบบ:

ในการพัฒนาระบบนี้ได้มีการกำหนดขอบเขตการทำงานเพื่อให้โครงงานสามารถบรรลุวัตถุประสงค์ได้อย่างเป็นรูปธรรมและวัดผลได้จริง โดยขอบเขตของระบบครอบคลุมดังนี้

1.4.4.1 ขอบเขตด้านการทำงาน (Functional Scope)
ระบบนี้ถูกออกแบบมาเพื่อ:

ก) การตรวจจับและวิเคราะห์ท่าทางมือ: สามารถตรวจจับตำแหน่งของมือและจุดเชื่อมโยง (landmarks) ทั้ง 21 จุดได้อย่างต่อเนื่องในแต่ละเฟรมวิดีโอ

ข) การจดจำตัวอักษร: สามารถจำแนกท่าทางมือที่แทนตัวอักษรภาษาอังกฤษ (ASL) ตั้งแต่ A-Z และสัญลักษณ์พื้นฐาน เช่น 'Space' และ 'Delete' โดยใช้โมเดลที่ผ่านการฝึกฝนแล้ว

ค) การแสดงผลแบบเรียลไทม์: สามารถแสดงผลลัพธ์การทำนายตัวอักษรล่าสุดและข้อความที่สะสมได้บนหน้าจออย่างรวดเร็ว

ง) การแสดงผลเชิงภาพ: สามารถแสดงเส้นโครงกระดูก (skeleton) ของมือที่ตรวจจับได้ซ้อนทับบนภาพวิดีโอ เพื่อให้ผู้ใช้สามารถเห็นการทำงานของระบบได้อย่างชัดเจน

จ) การควบคุม: มีปุ่มสำหรับสั่งการพื้นฐาน เช่น การลบข้อความทั้งหมด (Clear Text) และการกลับภาพจากกล้อง (Flip Camera)

1.4.4.2 ขอบเขตด้านข้อมูล (Data Scope)

ก) ชนิดข้อมูลเข้า (Input Data): ระบบจะรับข้อมูลภาพวิดีโอจากกล้อง Webcam เท่านั้น

ข) ชุดข้อมูลฝึกฝน (Training Data): โมเดลการทำนายถูกฝึกฝนด้วยชุดข้อมูลภาษามือ American Sign Language (ASL) ซึ่งเป็น

ข้อมูลภาพท่าทางมือสำหรับตัวอักษรภาษาอังกฤษ A-Z รวมถึงสัญลักษณ์พิเศษ 'Space' และ 'Delete' ที่มีรูปแบบข้อมูลเป็นพิกัด (landmarks)

ค) ชนิดข้อมูลออก (Output Data): ระบบจะแสดงผลลัพธ์เป็นตัวอักษรภาษาอังกฤษและข้อความที่ถูกประกอบขึ้นเท่านั้น

1.4.4.3 ขอบเขตด้านสภาพแวดล้อม (Environmental Scope)

ก) ฮาร์ดแวร์: ระบบทำงานบนเครื่องคอมพิวเตอร์ที่ติดตั้งกล้องเว็บแคม (Webcam) ที่สามารถบันทึกวิดีโอได้

ข) ซอฟต์แวร์และแพลตฟอร์ม: ระบบถูกพัฒนาเป็น Web Application โดยใช้ Python, Flask และไลบรารีที่เกี่ยวข้อง โดยสามารถทำงานได้บนเว็บเบราว์เซอร์มาตรฐานที่รองรับการเข้าถึงกล้อง เช่น Google Chrome หรือ Firefox บนระบบปฏิบัติการ Windows, macOS และ Linux

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ช่วยผู้พิการทางการได้ยินในการสื่อสารกับสังคม ระบบสามารถแปลภาษามือเป็นตัวอักษรหรือข้อความ ทำให้ผู้พิการทางการได้ยินสามารถสื่อสารกับคนทั่วไปได้สะดวกมากขึ้น ลดอุปสรรคด้านการสื่อสาร

1.6.2 เป็นสื่อการเรียนรู้ด้านเทคโนโลยี AI และการประมวลผลภาพ ผู้พัฒนาหรือผู้ที่สนใจสามารถศึกษากระบวนการทำงานของระบบ ตั้งแต่การจัดเก็บข้อมูล การประมวลผล ไปจนถึงการทำนายผล ซึ่งเป็นประโยชน์ต่อการเรียนการสอนและการวิจัยต่อยอด

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

การใช้โปรแกรมภาษามือ โดยโปรแกรมนี้ไว้ใช้เพื่อตรวจจับท่าทางภาษามือและแสดงข้อความภาษามือในเว็บไซต์ ระบบภาษามือนี้ได้เอาฐานข้อมูลมาจาก Kaggle โดยเว็บไซต์นี้จะนำข้อมูลของภาษามือแต่ละตัวอักษรภาษาอังกฤษมาใช้ โครงงานนี้ได้ผสมผสานเทคโนโลยีฮาร์ดแวร์และซอฟต์แวร์ โดยใช้เครื่องมือเช่น ChatGPT, Google Gemini และ Google Firebase แล้วใช้ภาษาคอมพิวเตอร์ได้แก่ Python Flask, JavaScript, CSS และ HTML เพื่อให้โปรแกรมทำงานบนหน้าเว็บไซต์

2.1 ระบบงานเดิม

โดยทั่วไปภาษามือไม่ได้เป็นที่นิยมมากนัก แต่บุคคลบางกลุ่มจำเป็นต้องใช้ภาษามือในการใช้ชีวิตโดยเฉพาะบุคคลที่บกพร่องทางการได้ยิน จึงทำให้เกิดภาษามือเกิดขึ้นโดยที่นอกเหนือจากภาษาเขียน และภาษาอ่าน ซึ่งการใช้ภาษามือทำให้เกิดความรวดเร็วทางการสื่อสารกว่าการเขียน ดังนั้นการนำเทคโนโลยีเข้ามาประยุกต์ใช้เพื่อพัฒนาภาษามือให้เข้าใจมากขึ้น และเข้าใช้ได้อย่างทั่วถึงของผู้ใช้บริการ

2.2 ระบบงานอื่นที่เกี่ยวข้อง

Meta-Learning หรือ Learning to learn หรือการเรียนรู้เพื่อที่จะเรียนรู้ เป็นรูปแบบการ เรียนรู้ที่นิยมในช่วงหลายปีที่ผ่านมา โดยการเรียนรู้แบบนี้ เป็นวิธีการเรียนรู้ที่เลียนแบบการเรียนรู้ ของมนุษย์ ที่สามารถเรียนรู้สิ่งใหม่ ๆ ได้จากตัวอย่างไม่กี่ตัวอย่างในขณะที่การเรียนรู้เชิงลึก ในตอนนี้ยังเป็นเรื่องที่ต้องใช้ข้อมูล การที่จะทำให้โมเดลมีประสิทธิภาพที่ดี ต้องใช้ตัวอย่างการฝึกอบรมหลายล้านหรือหลายพันล้านแบบ ซึ่งเป็นวิธีที่คลาสสิกในการบรรลุเป้าหมายการเพิ่มข้อมูลเป็นวิธีการหนึ่งในการสร้างตัวอย่างสังเคราะห์ ยิ่งไปกว่านั้นโครงข่ายใยประสาทมาตรฐาน ไม่สามารถเรียนรู้ความรู้ใหม่ได้ทันที

Mediapipe เป็นแพลตฟอร์ม AI แบบ Open source ของ Google ที่สามารถใช้เป็น Pipeline ตรวจจับท่าทาง มือ และใบหน้าของมนุษย์ในเวลาเดียวกัน โดยใช้การโอนถ่ายหน่วยความจำระหว่าง Inteference Backend ซึ่ง Pipeline จะรวมรูปแบบการปฏิบัติการและการประมวลผลที่แตกต่างกันตามการตรวจจับภาพแต่ละส่วนเข้าด้วยกัน และจะได้เป็นโซลูชั่นแบบ ครบ

วงจรที่ใช้งานได้แบบเรียลไทม์และสม่ำเสมอ MediaPipe คือเทคโนโลยีล้ำสมัยที่สามารถ ตรวจจับท่าทาง มือ และใบหน้า

ภาษามือ หรือภาษาใบ้ (Sign Lauguage) คือการสื่อสารโดยใช้มือ ที่ใช้กันอย่างแพร่หลาย โดยเช่นเดียวกับภาษาอื่น ๆ ก็มีหลากหลายสไตล์การใช้แตกต่างกันออกไปตามภูมิภาค แต่ที่นิยมใช้กันอย่างมากที่สุด ก็หนีไม่พ้นภาษามืออเมริกัน (ASL) อังกฤษ และออสเตรเลีย โดยไทยเองก็มีรูปแบบการใช้ภาษามือของตนเอง ซึ่งตัวอักษรไทยก็จะมีรูปแบบของตนเอง ส่วนคำต่าง ๆ ก็จะใช้แบบอเมริกา และอังกฤษประกอบกัน รวมถึงสัญลักษณ์เฉพาะของไทย

2.3 องค์ความรู้ที่เกี่ยวข้อง

2.3.1 Window 11

Windows 11 คือระบบปฏิบัติการล่าสุดจาก Microsoft ซึ่งเปิดตัวอย่างเป็นทางการในวันที่ 5 ตุลาคม 2021 เป็นการพัฒนาต่อจาก Windows 10 โดยมุ่งเน้นการปรับปรุงประสบการณ์การใช้งานให้ทันสมัยขึ้น พร้อมทั้งเพิ่มฟีเจอร์ใหม่ๆ และปรับปรุงประสิทธิภาพการทำงานให้ดีขึ้น วันนี้อับดุลจะพาท่านไปสำรวจความเปลี่ยนแปลงและนวัตกรรมที่มาพร้อมกับ Windows 11 ดังภาพที่ 2.1

ภาพที่ 2.1 Window 11

ที่มา https://shorturl.at/qZuHZ

2.3.2 Google

Google เป็นเว็บไซต์ที่ให้ บริการในการค้นหาข้อมูลในโลกของอินเตอร์เน็ต โดยค้นหาข้อมูลจากข้อความ หรือตัวอักษรที่พิมพ์ป้อนเข้าไป แล้วทำการค้นหาข้อมูลรูปภาพ หรือเว็บไซต์ที่เกี่ยวข้องนำมาแสดงผล เว็บไซต์ Google ดังนั้น Google จึงครอบคลุมทุกการบริการบนเว็บเบราว์เซอร์ และแอพพลิเคชั่น ทำให้มีผู้ใช้มากมายใช้บริการของ Google เพราะว่าซอร์ฟแวร์แต่ละตัวของ Google มีการพัฒนาอย่างต่อเนื่อง ด้วยทีมพัฒนาที่มีความรู้และความเชี่ยวชาญอย่างมาก ดังภาพที่ 2.2

ภาพที่ 2.2 Google

ที่มา https://shorturl.at/pJoFs

2.3.3 Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากทางบริษัท Microsoft ซึ่งมีการพัฒนาออกมาในรูปแบบมาของ OpenSource จึงสามารถนำมาใช้งานได้โดยไม่เสียค่าใช้จ่าย ซึ่ง Visual Studio Code นั้นเหมาะสำหรับนักพัฒนาโปรแกรมที่สามารถรองรับการใช้งานข้ามแพลตฟอร์มได้และรองรับการใช้งานได้ในระบบปฏิบัติการ Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้

อย่างมากมาย ไม่ว่าจะเป็น 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go 2.Themes 3.Debugger 4.Commands เป็นต้น ดังภาพที่ 2.3

ภาพที่ 2.3 Visual Studio Code

ที่มา https://shorturl.at/JFBKO

2.3.4 Gemini

Gemini (อ่านว่า เจมิไนน์) คือโมเดลปัญญาประดิษฐ์ (AI) แบบ Multimodal ที่พัฒนาโดย Google ซึ่งมีความสามารถในการทำความเข้าใจ ประมวลผล และสร้างข้อมูลได้หลากหลายรูปแบบพร้อมกัน ทั้งข้อความ รูปภาพ เสียง วิดีโอ และโค้ดโปรแกรม อีกทั้งยังเป็นพลังขับเคลื่อนให้กับ Chatbot AI ของ Google (ที่เคยรู้จักกันในชื่อ Bard) รวมถึงฟีเจอร์ AI ต่างๆ ในผลิตภัณฑ์ของ Google เช่น Gmail และ Google Docs มีการประมวลผลที่รวดเร็วและมีความแม่นยำมากขึ้นเรื่อย ๆ โดยสามารถช่วยเขียนบทความ ตอบคำถาม วิเคราะห์ข้อมูล สรุปเอกสาร แปลภาษา สร้างรูปภาพ ไปจนถึงการเขียนโค้ดเพื่อใช้ในการพัฒนาโปรแกรม อีกทั้งยังรองรับการทำงานร่วมกับบริการต่าง ๆ ของ Google เช่น Google Search, Gmail, Docs และ YouTube เพื่อให้ผู้ใช้งานเข้าถึงข้อมูลและสร้างสรรค์งานได้ง่ายขึ้น ดังภาพที่ 2.4

ภาพที่ 2.4 Gemimi

ที่มา https://shorturl.at/lp4og

2.3.5 ChatGPT

ChatGPT (Chat Generative Pre-trained Transformer) คือโมเดลภาษาปฏิบัติการขนาดใหญ่ (large language model) ที่ถูกสร้างและพัฒนาโดยบริษัท OpenAI เพื่อให้สามารถทำงานเป็น Chatbot ที่สามารถพูดคุย โต้ตอบกับมนุษย์ได้อย่างเป็นธรรมชาติ และตอบคำถามที่ซับซ้อนได้ โดย ChatGPT ถูกสร้างจากการเทรนโมเดลที่มีขนาดใหญ่มากๆ โดยใช้ข้อมูลจำนวนมากเช่น บทความ หนังสือ เว็บไซต์ เพื่อเรียนรู้คำศัพท์ และลักษณะการใช้ภาษาของภาษาอังกฤษ ทำให้ ChatGPT สามารถตอบคำถาม สอบถามข้อมูล หรือสนทนาเกี่ยวกับเรื่องต่างๆ ได้อย่างมีประสิทธิภาพและความถูกต้องสูงสุดที่เป็นไปได้ ดังภาพที่ 2.5

ภาพที่ 2.5 ChatGPT

ที่มา https://shorturl.at/SIXaP

2.3.6 kaggle

Kaggle เป็นแพลตฟอร์มออนไลน์ที่เชื่อมโยงกันระหว่างนักวิทยาศาสตร์ข้อมูล (Data scientists) และนักพัฒนาโมเดลปัญญาประดิษฐ์ (Artificial Intelligence) จากทั่วโลก ซึ่งมีเป้าหมายในการสนับสนุนและกระตุ้นให้เกิดความคล้ายคลึงระหว่างการแข่งขันและการพัฒนาโครงการทางด้านการเรียนรู้ของเครื่อง (Machine Learning) ในชุมชน แพลตฟอร์ม Kaggle สนับสนุนให้นักพัฒนาภาษา Python ซึ่งเป็นภาษาที่มีการใช้งานอย่างแพร่หลายสามารถใช้งานผ่าน Web ซึ่งเป็นบริการฟรี สามารถเข้าถึงและใช้งแก้ไขปัญหาทางด้านข้อมูล ตั้งแต่ปัญหาที่เกี่ยวข้องกับการทำนาย (prediction) การจัดกลุ่ม (clustering) หรือการแบ่งแยกข้อมูล (classification) จนถึงการตรวจสอบความแม่นยำ (accuracy) ของโมเดล ผู้เข้าแข่งขันที่ดีที่สุดจะได้รับรางวัลที่น่าสนใจ และเรียนรู้จากเนื้อหาที่ให้มาได้ อย่างไรก็ตาม ในบางกรณี Kaggle ยังมีการให้บริการที่เสียค่าใช้จ่ายเช่น Kaggle Kernel ที่ต้องเสียค่าใช้จ่ายเพื่อใช้งานในบางฟีเจอร์เสริมเพิ่มเติม นอกจากนี้ ค่าใช้จ่ายอื่นๆ อาจเกิดขึ้นจากค่าใช้จ่ายที่เกี่ยวข้องกับการใช้และจัดการแหล่งข้อมูลที่มีคุณภาพสูงใน Kaggle ตามความต้องการของผู้ใช้งาน ดังภาพที่ 2.6

ภาพที่ 2.6 kaggle

ที่มา https://shorturl.at/DvVRA

2.3.7 Firebase

Firebase เป็นแพลตฟอร์มการพัฒนาแอปพลิเคชันที่ครอบคลุมจาก Google ซึ่งช่วยให้นักพัฒนาสร้าง ปรับใช้ และขยายแอปพลิเคชันมือถือและเว็บไซต์ได้อย่างง่ายดาย โดย Firebase ทำหน้าที่เป็น Backend as a Service (BaaS) ที่ช่วยลดความซับซ้อนในการจัดการโครงสร้างพื้นฐานหลังบ้าน ทำให้นักพัฒนาสามารถมุ่งเน้นไปที่การสร้างประสบการณ์ผู้ใช้ที่ยอดเยี่ยม ดังภาพที่ 2.7

ภาพที่ 2.7 Firebase

ที่มา https://shorturl.at/rBtlN

2.3.8 HTML

HTML ย่อมาจาก HyperText Markup Language เป็นภาษาคอมพิวเตอร์ที่ใช้สำหรับสร้างและแสดงผลเอกสารบนเว็บไซต์ หรือที่เรียกว่า เว็บเพจ HTML ถูกพัฒนาขึ้นโดยองค์กร World Wide Web Consortium (W3C) และเป็นภาษามาตรฐานที่ใช้ในการออกแบบโครงสร้างของหน้าเว็บไซต์ HTML เป็นภาษาประเภท Markup Language ซึ่งหมายถึงการใช้ Tag ต่างๆ เพื่อกำหนดโครงสร้างและเนื้อหาของหน้าเว็บ เช่น ข้อความ, รูปภาพ, ลิงก์, ตาราง และอื่นๆ. ตัวอย่างโปรแกรมที่สามารถใช้เขียน HTML ได้แก่ Notepad, EditPlus, หรือโปรแกรมช่วยสร้างเว็บเพจ เช่น Dreamweaver ดังภาพที่ 2.8

ภาพที่ 2.8 HTML

ที่มา https://shorturl.at/JqOfB

2.3.9 Python

Python เป็นภาษาการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในเว็บแอปพลิเคชัน การพัฒนาซอฟต์แวร์ วิทยาศาสตร์ข้อมูล และ Machine Learning (ML) นักพัฒนาใช้ Python เนื่องจากมีประสิทธิภาพ เรียนรู้ง่าย และสามารถทำงานบนแพลตฟอร์มต่างๆ ได้มากมาย ทั้งนี้ซอฟต์แวร์ Python สามารถดาวน์โหลดได้แบบไม่เสียตัง ผสานการทำงานร่วมกับระบบทุกประเภท และเพิ่มความเร็วในการพัฒนา ดังภาพที่ 2.9

ภาพที่ 2.9 Python

ที่มา https://shorturl.at/WfAfU

2.3.10 Flask

Flask คือ web framework ที่เขียนขึ้นมาสำหรับ Python เพื่อใช้ร่วมกัน webserver เช่น Apache และได้รับการยอมรับจาก community we pages ชั้นนำเช่น Pinterest, LinkedIn เป็นต้น โดย Flask ถูกเรียกว่า micro framework เพราะว่า มันไม่ต้องการเครื่องมือ หรือ library อะไรมาก อีกทั้ง ไม่จำเป็นต้องมี database ด้วย แต่อย่างไรก็ตาม Flask ก็ยังรองรับการเพิ่ม extensions พิเศษได้ ถ้ามันรองรับ Flask ดังภาพที่ 2.10

ภาพที่ 2.10 Flask

ที่มา https://saixiii.com/python-flask-web-application

2.3.11 JavaScript

JavaScript เป็นภาษาโปรแกรมที่นักพัฒนาใช้ในการสร้างหน้าเว็บแบบอินเทอร์แอคทีฟ ตั้งแต่การรีเฟรชฟีดสื่อโซเชียลไปจนถึงการแสดงภาพเคลื่อนไหวและแผนที่แบบอินเทอร์แอคทีฟ ฟังก์ชันของ JavaScript สามารถปรับปรุงประสบการณ์ที่ผู้ใช้จะได้รับจากการใช้งานเว็บไซต์ และในฐานะที่เป็นภาษาในการเขียนสคริปต์ฝั่งไคลเอ็นต์ จึงเป็นหนึ่งในเทคโนโลยีหลักของ World Wide Web ยกตัวอย่างเช่น เมื่อคุณท่องเว็บแล้วเห็นภาพสไลด์ เมนูดร็อปดาวน์แบบคลิกให้แสดงผล หรือสีองค์ประกอบที่เปลี่ยนแบบไดนามิกบนหน้าเว็บ นั่นคือคุณเห็นเอฟเฟกต์ของ JavaScript ดังภาพที่ 2.11

ภาพที่ 2.11 JavaScript

ที่มา https://shorturl.at/aTavf

2.3.12 CSS

CSS คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย ดังภาพที่ 2.12

ภาพที่ 2.12 CSS

ที่มา https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg

บทที่ 3

วิธีการดำเนินงาน

3.1 การศึกษาเบื้องต้น

3.1.1 ระบบงานเดิม

โดยทั่วไปภาษามือไม่ได้เป็นที่นิยมมากนัก แต่บุคคลบางกลุ่มจำเป็นต้องใช้ภาษามือในการใช้ชีวิตโดยเฉพาะบุคคลที่บกพร่องทางการได้ยิน จึงทำให้เกิดภาษามือเกิดขึ้นโดยที่นอกเหนือจากภาษาเขียน และภาษาอ่าน ซึ่งการใช้ภาษามือทำให้เกิดความรวดเร็วทางการสื่อสารกว่าการเขียน ดังนั้นการนำเทคโนโลยีเข้ามาประยุกต์ใช้เพื่อพัฒนาภาษามือให้เข้าใจมากขึ้น และเข้าใช้ได้อย่างทั่วถึงของผู้ใช้บริการ

3.1.2 ระบบงานใหม่

ระบบงานใหม่นี้ถูกพัฒนาขึ้นบนพื้นฐานของ Client-Server Architecture ที่เน้นการประมวลผลแบบกระจาย (Distributed Processing) เพื่อให้เกิดประสิทธิภาพสูงสุดในการทำงานแบบเรียลไทม์ โดยใช้หลักการทาง Computer Vision และ Machine Learning มาบูรณาการเข้าด้วยกัน

3.1.2.1 แนวคิดการประมวลผลและการแบ่งภาระงาน (Smart Load Distribution)

แนวคิดหลักของระบบคือการแบ่งเบาภาระงานที่ต้องใช้ทรัพยากรสูงออกจากกันอย่างชัดเจน

ก) งานที่ทำบน Client-Side (เว็บเบราว์เซอร์): ภาระงานที่เกี่ยวข้องกับการเข้าถึงฮาร์ดแวร์โดยตรงและการประมวลผลภาพแบบเฟรมต่อเฟรม (Frame-by-Frame) ถูกมอบหมายให้ทำงานบนเครื่องของผู้ใช้เอง ซึ่งเรียกว่า Computer Vision Layer งานนี้รวมถึงการรับภาพจาก Webcam และการสกัด 21 Landmarks ของมือโดยใช้ไลบรารี MediaPipe Hands ซึ่งช่วยลดภาระ CPU/GPU ของเซิร์ฟเวอร์

ข) งานที่ทำบน Server-Side (Firebase Cloud Functions): ภาระงานที่ต้องใช้โมเดล AI และการจัดการตรรกะที่ซับซ้อนถูกมอบหมายให้ทำงานบนคลาวด์ ซึ่งเรียกว่า Inference and Logic Control Layer ข้อมูลที่ส่งไปยัง Server จึงเป็นเพียงชุดตัวเลข 63 ค่า ที่ถูก Normalizedแล้ว (Normalized Landmark Array) ในรูปแบบ JSON, ไม่ใช่ข้อมูลวิดีโอ ซึ่งทำให้การสื่อสารมีน้ำหนักเบาและรวดเร็ว

3.1.2.2 ชั้นการทำงานและเทคโนโลยีที่เกี่ยวข้อง (Layers of Operation)

ระบบถูกแบ่งออกเป็นชั้นการทำงานที่ทำงานต่อเนื่องกันในทุกเฟรมวิดีโอ

ก) Input Layer: จัดการการเข้าถึงกล้อง Webcam ผ่าน JavaScript API และส่งผ่านข้อมูลภาพไปยังชั้นการสกัดคุณลักษณะอย่างต่อเนื่อง

ข) Feature Extraction Layer: ใช้ MediaPipe Hands ในฝั่ง Client เพื่อตรวจจับมือและสกัดพิกัด 21 จุด (x, y, z) ซึ่งเป็นข้อมูลเชิงตัวเลขที่บ่งบอกถึงท่าทางของมือได้อย่างแม่นยำ

ค) Data $\mathbfTransmission}$ Layer: ใช้ Fetch API ในการส่งข้อมูลชุดพิกัดมือที่ผ่านการปรับให้เป็นมาตรฐานแล้ว (Normalized) ไปยัง Cloud Functions ผ่าน AJAX POST Request

ง) Inference Layer: เป็นส่วนที่รันโค้ด Python/Flask บน Cloud Functions ทำหน้าที่รับ JSON Payload แล้วนำชุดตัวเลข 63 มิติ เข้าสู่โมเดล Random Forest Classifier ที่ถูกโหลดไว้ล่วงหน้าจากไฟล์ .joblib เพื่อทำนายผลตัวอักษร (Prediction)

จ) Logic Control Layer: อยู่ในส่วน Backend ทำหน้าที่จัดการ Accumulation Logic ซึ่งเป็นหัวใจสำคัญในการยืนยันผลการทำนาย โดยใช้เทคนิค Temporal Filtering (การหน่วงเวลา) เพื่อให้แน่ใจว่าท่าทางที่ทำนายมีความเสถียรเป็นเวลา T วินาที ก่อนที่จะเพิ่มตัวอักษรลงในช่องข้อความหลัก ซึ่งช่วยป้องกันการพิมพ์ข้อความผิดพลาดจากการสั่นของมือ

ฉ) Output Layer: ส่วน Frontend รับ JSON ที่มีผลลัพธ์สุดท้ายกลับมา เพื่อแสดงผลตัวอักษรที่ถูกยืนยันแล้วในช่องข้อความหลัก และแสดงภาพ โครงกระดูกมือ

(Skeleton) ซ้อนทับบนภาพวิดีโอ เพื่อให้ผู้ใช้สามารถตรวจสอบการทำงานของระบบได้แบบ Real-time Feedback

3.1.2.3 ข้อได้เปรียบของระบบงานใหม่

ก) Real-time Performance: การแบ่งภาระงานทำให้ระบบมีความหน่วงต่ำ (Low Latency) และตอบสนองต่อท่าทางมือได้รวดเร็วกว่าการประมวลผลวิดีโอเต็มบน Server

ข) Deployment Flexibility: การใช้ Firebase Hosting ร่วมกับ Cloud Functions ทำให้การติดตั้งใช้งานเป็นแบบ Serverless ซึ่งหมายถึงความสามารถในการรองรับผู้ใช้งานจำนวนมากโดยอัตโนมัติ และมีค่าใช้จ่ายเกิดขึ้นเฉพาะเมื่อมีการใช้งานจริง (Pay-as-you-go)

ค) High Accessibility: ผู้ใช้สามารถเข้าถึงระบบได้ง่ายผ่าน Web Browser มาตรฐาน โดยไม่ต้องติดตั้ง Software หรือ Hardware ราคาแพง ใด ๆ เพิ่มเติม

3.2 การกำหนดความต้องการของระบบ

การกำหนดความต้องการของระบบเป็นขั้นตอนสำคัญในการแปลวัตถุประสงค์และขอบเขตของโครงงานไปเป็นข้อกำหนดเชิงปฏิบัติการที่ชัดเจน เพื่อเป็นรากฐานในการออกแบบและพัฒนาต่อไป

3.2.1 ขอบเขตของระบบ

ขอบเขตการทำงานของระบบถูกกำหนดอย่างละเอียด เพื่อให้การพัฒนามีทิศทางที่แน่นอนและสามารถวัดผลได้อย่างชัดเจน

3.2.1.1 ขอบเขตที่ระบบสามารถทำได้ (In-Scope)

ก) การจดจำตัวอักษร: ระบบสามารถจดจำและจำแนกท่าทางมือที่แทนตัวอักษรภาษาอังกฤษ A-Z รวมถึงสัญลักษณ์ควบคุม 'Space' และ 'Delete' ตามมาตรฐานภาษามืออเมริกัน (ASL)

ข) การประมวลผลท่าทาง: ระบบจะประมวลผลเฉพาะท่าทางมือที่ใช้มือเดียว (Single-Hand Gestures) เท่านั้น โดยใช้ไลบรารี MediaPipe Hands ในการสกัดพิกัดจุดเชื่อมโยง (Landmarks) 21 จุด ของมือ

ค) การทำงานแบบเรียลไทม์: ระบบต้องสามารถแสดงผลการทำนายตัวอักษรและโครงกระดูกมือ (Skeleton) ซ้อนทับบนภาพวิดีโอได้อย่างต่อเนื่องและรวดเร็ว (Low Latency)

ง) ตรรกะการสะสมข้อความ: มีการจัดการตรรกะการยืนยันผล (Accumulation Logic) เพื่อให้ตัวอักษรถูกเพิ่มเข้าในช่องข้อความสะสมต่อเมื่อท่าทางมือมีความเสถียรตามระยะเวลาที่กำหนดเท่านั้น

จ) ฟังก์ชันอำนวยความสะดวก: มีปุ่มควบคุมพื้นฐานบนหน้าเว็บ เช่น Clear Text (ลบข้อความทั้งหมด) และ Flip Camera (สลับภาพวิดีโอ)

3.2.1.2 ขอบเขตที่ระบบไม่สามารถทำได้ (Out-of-Scope)

ก) ความซับซ้อนของภาษามือ: ระบบไม่ครอบคลุมการจดจำคำศัพท์เต็ม หรือ ประโยค ในภาษามือ (Vocabulary), ท่าทางที่ต้องใช้ มือสองข้าง, หรือการตีความ การแสดงออกทางสีหน้า และท่าทางร่างกาย

ข) ภาษา:ระบบไม่รองรับการจดจำภาษามืออื่นนอกเหนือจาก ASL Alphabet เช่น ภาษามือไทย (Thai Sign Language)

ค) การเรียนรู้และปรับปรุง: ระบบไม่รองรับฟังก์ชันการเรียนรู้ท่าทางใหม่ (Learning New Gestures) หรือการปรับแต่งโมเดลโดยผู้ใช้งาน (User Retraining)

3.2.2 ฮาร์ดแวร์ที่ใช้กับระบบงาน

การกำหนดข้อกำหนดด้านฮาร์ดแวร์ในโครงงานนี้มีความเฉพาะเจาะจงสูง เนื่องจากระบบใช้สถาปัตยกรรม Client-Centric ซึ่งแบ่งภาระงานหนักระหว่างเครื่องผู้ใช้และ Firebase Cloud Functions

3.2.2.1 ฮาร์ดแวร์ฝั่งผู้ใช้งาน (Client-Side Hardware)

ฮาร์ดแวร์ฝั่งผู้ใช้งานมีบทบาทสำคัญที่สุดในการประมวลผล Computer Vision (MediaPipe Hands) ดังนั้นข้อกำหนดจึงเน้นไปที่พลังในการประมวลผลความเร็วสูง:

ก) หน่วยประมวลผล (CPU): จำเป็นต้องใช้ Intel Core i5 Gen 8 ขึ้นไป หรือเทียบเท่า (เช่น AMD Ryzen 5 2000 Series ขึ้นไป) เนื่องจาก JavaScript Engine ต้องทำงานร่วมกับ MediaPipe เพื่อสกัด Landmarks จากวิดีโอ ในอัตรา 30 Frames per Second ได้อย่างราบรื่น

ข) หน่วยความจำ (RAM): ควรมี 8 GB DDR4 ขึ้นไป เพื่อรองรับการจัดการหน่วยความจำของเว็บเบราว์เซอร์สมัยใหม่ที่ต้องมีการประมวลผลวิดีโอ สตรีมมิ่งและการรันโค้ด Vision ที่ใช้ทรัพยากรสูง

ค) อุปกรณ์รับภาพ (Webcam): ต้องใช้กล้อง Webcam HD 720p (1280 x 720) @ 30 FPS ขึ้นไป ที่มีคุณภาพดี เพื่อให้ภาพมีรายละเอียดที่ชัดเจนเพียงพอต่อการตรวจจับข้อนิ้วและมือได้อย่างแม่นยำ

ง) การเชื่อมต่อเครือข่าย (Network Connectivity): แม้ว่าข้อมูล JSON ที่ส่งไปยัง Server จะมีขนาดเล็ก แต่ความเสถียรและความเร็วของ การเชื่อมต่ออินเทอร์เน็ต ก็มีความจำเป็นต่อการรับ-ส่ง AJAX Request ไปยัง Cloud Functions อย่างต่อเนื่อง เพื่อรักษา Real-time Latency

3.2.2.2 ทรัพยากรฝั่งเซิร์ฟเวอร์ (Server-Side Allocated Resources)

ระบบ Backend ถูก Deploy บน Firebase Cloud Functions ซึ่งเป็นสถาปัตยกรรมแบบ Serverless Computing ข้อกำหนดนี้จึงเป็นการระบุทรัพยากรที่ต้องจัดสรรให้แต่ละ Function Instance สามารถทำงานได้อย่างมีประสิทธิภาพ

ก) แพลตฟอร์มหลัก: ระบบจะทำงานบน Google Firebase Cloud Functions ซึ่งจะทำการปรับขนาดทรัพยากรตามปริมาณการใช้งานโดยอัตโนมัติ

ข) หน่วยความจำ (RAM Cloud Function): ควรกำหนดหน่วยความจำไว้ที่ 256 MB – 512 MB ต่อ Instance เพื่อให้เพียงพอต่อการโหลด Python Runtime และโมเดล Random Forest Classifier ได้อย่างรวดเร็ว ในแต่ละ Request

ค) CPU: กำหนด 1 vCPU ต่อ Instance เพื่อให้การ คำนวณ ML Inference และการจัดการ Accumulation Logic สามารถเกิดขึ้นได้อย่างรวดเร็วและไม่เป็นคอขวดของระบบ Latency

ง) Deployment: Firebase Hosting ทำหน้าที่เป็น โฮสต์สำหรับไฟล์ HTML/CSS/JavaScript ทั้งหมดรวมถึงเป็น API Gateway ในการเชื่อมโยง Client กับ Cloud Functions ผ่านการตั้งค่า Rewrites

3.3 การออกแบบระบบ

การออกแบบระบบนี้ตั้งอยู่บนหลักการ Client-Centric Computing เพื่อให้เกิดประสิทธิภาพ Real-time สูงสุด โดยใช้ประโยชน์จากการประมวลผล Computer Vision บนเบราว์เซอร์ และความเสถียรของ ML Inference บน Server

3.3.1 การออกแบบสถาปัตยกรรมและกระแสข้อมูล (Architecture and Data Flow Design)

ระบบใช้สถาปัตยกรรมแบบ Client-Centric Two-Tier Architecture ซึ่งเป็นการแบ่งภาระงานอย่างชาญฉลาดระหว่าง Client Side (Browser) และ Server Side (Firebase Cloud Functions)

3.3.1.1 องค์ประกอบสถาปัตยกรรมหลัก

ก) Client Tier (HTML/CSS/JavaScript):

หน้าที่หลัก: จัดการ UI และการประมวลผล Computer Vision ด้วย MediaPipe Hands (JavaScript API)

การประมวลผล: ดึงวิดีโอจาก Webcam, สกัด 21 Landmarks และทำการ Normalization เพื่อสร้าง Feature Vector 63 มิติ

การสื่อสาร: ส่ง JSON Payload ที่มี Feature Vector 63 มิติ ไปยัง Server ผ่าน AJAX POST Request

ข) Server Tier (Firebase Cloud Functions – Python Flask):

หน้าที่หลัก: การประมวลผล Machine Learning Inference และ Logic Control

การประมวลผล: รับ JSON, ป้อนเข้าโมเดล Random Forest Classifier (Python), จัดการตรรกะการสะสมข้อความ (Temporal Filtering)

3.3.1.2 ผังกระแสข้อมูล (Real-time Prediction Loop)

ระบบทำงานเป็นวงจรต่อเนื่องดังผัง

ก) Client Vision Processing: JavaScript ประมวลผลภาพจาก Webcam ด้วย MediaPipe สกัด Landmark และ Normalize

ข) Data Transmission: Client ส่ง Normalized Vector ในรูปแบบ JSON ไปยัง Cloud Function /predict

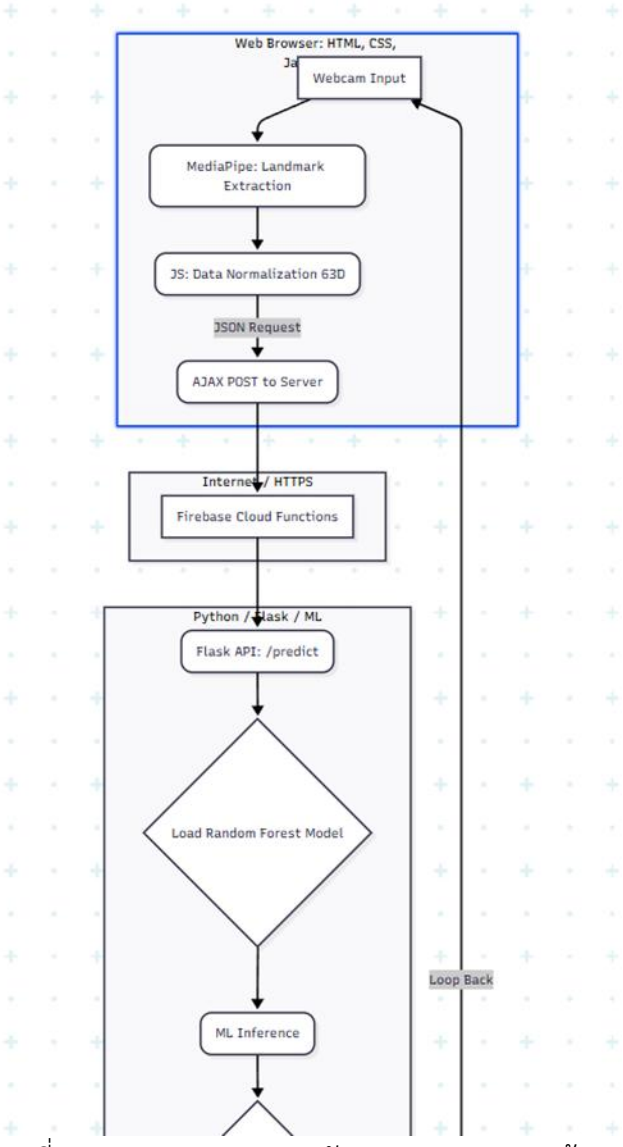
ค) Server Inference: Python Backend ทำนายผลด้วยโมเดล Random Forest

ง) Logic Decision: Python ใช้ตรรกะ Temporal Filtering เพื่อยืนยันความเสถียรของผลลัพธ์

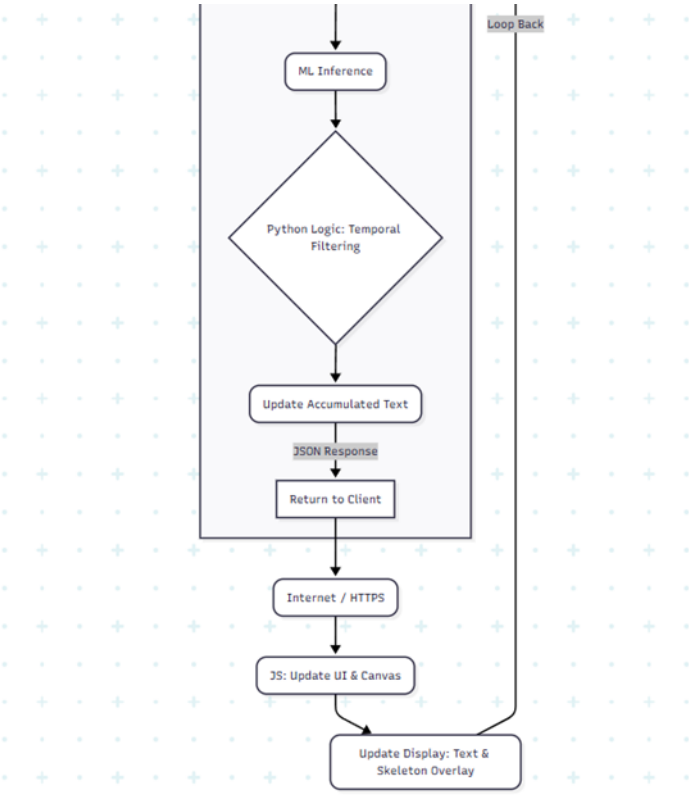
จ) Server State Update: อัปเดต Sentence String และสถานะ History ใน Server

ฉ) UI Rendering: Server ส่งผลลัพธ์กลับมาให้ JavaScript อัปเดต Text Box และวาด Skeleton Overlay บน HTML Canvas

ดังภาพที่ 3.1.1 และ 3.1.2



ภาพที่ 3.1.1 การออกแบบสถาปัตยกรรมและกระแสข้อมูล



ภาพที่ 3.1.2 การออกแบบสถาปัตยกรรมและกระแสข้อมูล

3.3.2 การออกแบบส่วนติดต่อผู้ใช้งาน (UI/UX Design)

การออกแบบ UI มุ่งเน้นการให้ Feedback แบบเรียลไทม์ ที่รวดเร็วและชัดเจน ตามหลักการของ Human-Computer Interaction

3.3.2.1 องค์ประกอบหลักของหน้าจอ

ก) $\mathbfVideo}$ Canvas and Skeleton Overlay:

Design: ใช้ HTML Canvas ในการแสดงผลวิดีโอ

UX: JavaScript วาดภาพ โครงกระดูกมือ 21 จุด ซ้อนทับ ซึ่งเป็น Instant Feedback แรกสุดที่ผู้ใช้ใช้จัดท่าทางมือ

ข) $\mathbfText}$ Output Area:

$\mathbfLatest}$ Prediction: แสดงผลการทำนายล่าสุดที่ได้รับจาก Server ในแต่ละ Frame โดยมีสีที่แตกต่างกันเพื่อบ่งบอกว่ายังไม่ถูกยืนยัน

$\mathbfAccumulated}$ Sentence Box: ช่อง ข้อความหลักสำหรับแสดง ข้อความที่ผ่านการยืนยัน Commit จาก Server แล้วเท่านั้น

ค) $\mathbfControl}$ Panel:

Clear Text : ปุ่มที่สั่งให้ JavaScript ส่ง Request ไปยัง API /clear_text เพื่อรีเซ็ตข้อความและ Prediction History ทั้งหมด

Flip Camera : ปุ่มที่ใช้ตรรกะ JavaScript เพื่อสลับภาพวิดีโอในแนวนอน เพื่อแก้ปัญหามุมมองของกล้องและมือ

3.3.3 การออกแบบตรรกะการประมวลผล (Processing Logic Design)

ตรรกะการประมวลผลถูกแบ่งระหว่าง Client และ Server เพื่อรับประกันความถูกต้องของข้อมูล

3.3.3.1 ตรรกะการสกัดคุณลักษณะ (Feature Engineering Logic – JavaScript)

Normalization Algorithm: ใช้สูตรคำนวณระยะห่างสัมพัทธ์ โดยใช้พิกัดของ จุดข้อมือ (Wrist) เป็นจุดกำเนิด และปรับสเกลของค่า X, Y, Z ทั้งหมด ซึ่งทำให้ Feature Vector 63 มิติ ที่ส่งไปยัง Server มีคุณสมบัติไม่ขึ้นกับขนาดและตำแหน่งของมือ

3.3.3.2 ตรรกะการกรองเวลาและการยืนยันผล (Temporal Filtering and Accumulation Logic – Python)

ตรรกะนี้รันบน Python Backend เพื่อจัดการความไม่แน่นอนของ Real-time Data

ก) $\mathbfPrediction}$ History Tracking: ใช้ Python Deque เพื่อเก็บผลการทำนายล่าสุดไว้ N เฟรม $\text{($\text{N}$ คำนวณจาก T วินาที x FPS ที่คาดการณ์)}$

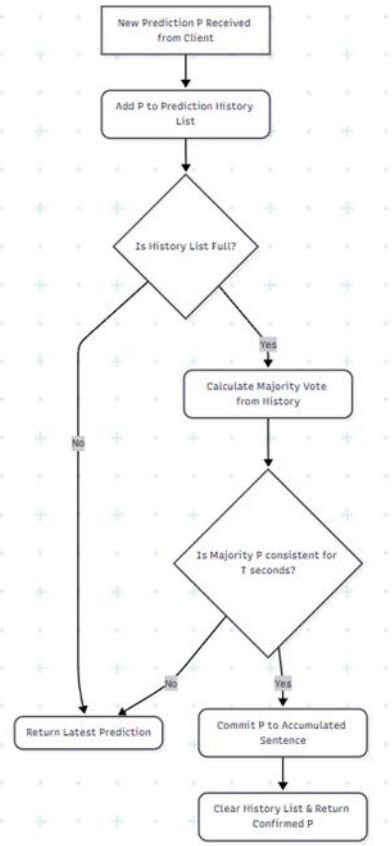
ข) $\mathbfMajority}$ Voting: Server คำนวณผลโหวตสูงสุดใน History List ในทุก Request

ค) $\mathbfStability}$ Commit Condition:

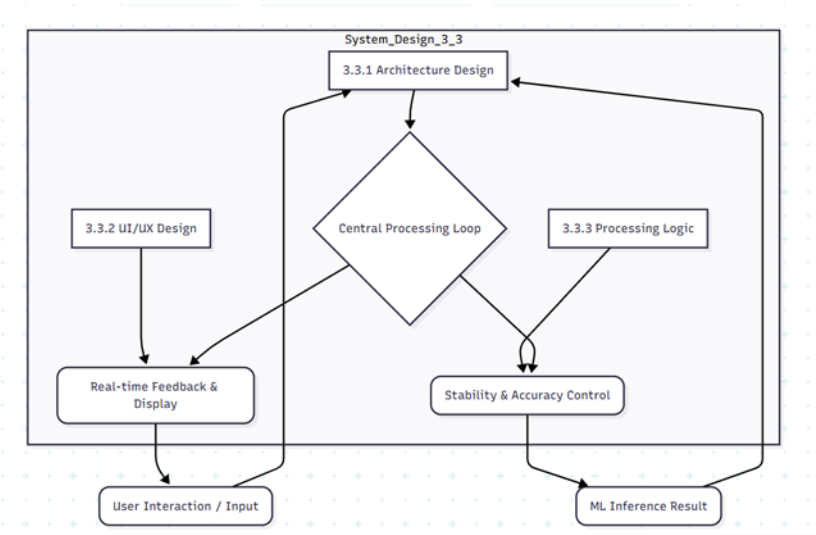
$\mathbfCondition}$: ตัวอักษรจะถูกยืนยัน Commit ได้ก็ต่อเมื่อผลโหวตสูงสุดนั้นมีสัดส่วนเกินกว่า 80% ของ History ทั้งหมด

$\mathbfAction}$: เมื่อ Commit เกิดขึ้น ระบบจะเรียกใช้ฟังก์ชัน update_sentence และรีเซ็ต History List

ง) $\mathbfControl}$ Word Execution: ตรรกะ Python จะจัดการการเพิ่มช่องว่าง (Space) และลบตัวอักษร (Delete) โดยตรงบน Sentence String ตามที่กำหนดในโค้ด update_sentence ก่อนส่งผลลัพธ์สุดท้ายกลับไปยัง Frontend



ภาพที่ 3.2 การออกแบบตรรกะการประมวลผล



ภาพที่ 3.3 การออกแบบระบบ

บรรณานุกรม

Amazon Web Services. (ม.ป.ป.). JavaScript คืออะไร. สืบค้นจาก https://aws.amazon.com/th/what-is/javascript/

Amazon Web Services. (ม.ป.ป.). Python คืออะไร. สืบค้นจาก https://aws.amazon.com/th/what-is/python/

บิง. (ม.ป.ป.). Firebase คืออะไร. สืบค้นจาก https://www.bing.com/search?q=firebase

บิง. (ม.ป.ป.). HTML คืออะไร. สืบค้นจาก https://www.bing.com/search?q=html

บิง. (ม.ป.ป.). Kaggle คืออะไร. สืบค้นจาก https://www.bing.com/search?q=kaggle

Drite Studio. (ม.ป.ป.). What is Windows 11: Changes and Innovations in Microsoft's Latest Operating System. สืบค้นจาก https://dritestudio.co.th/article/What-is-Windows-11-Changes-and-innovations-in-Microsoft-latest-operating-system

Firebase. (ม.ป.ป.). Brand Guidelines. สืบค้นจาก https://firebase.google.com/brand-guidelines

Kaggle. (ม.ป.ป.). ASL Alphabet Dataset. สืบค้นจาก https://www.kaggle.com/datasets/grassknoted/asl-alphabet/data

Lebbos, Z. (ม.ป.ป.). The Evolution of JavaScript: A Journey from ES1 to the Latest Version (Part 1). LinkedIn. สืบค้นจาก https://www.linkedin.com/pulse/evolution-javascript-journey-from-es1-latest-version-part-lebbos-za9fe

Microsoft News Thailand. (2564). Windows 11 เปิดตัวอย่างเป็นทางการในประเทศไทย. สืบค้นจาก https://news.microsoft.com/th-th/2021/06/25/windows11-th

Mindphp. (ม.ป.ป.). Google คืออะไร. สืบค้นจาก https://www.mindphp.com/.../3783-google.html

Plaradise. (ม.ป.ป.). What is ChatGPT? คืออะไร. สืบค้นจาก https://plaradise.com/what-is-chatgpt/

Rabbit Care. (ม.ป.ป.). 10 ภาษามือพื้นฐานในชีวิตประจำวันและสถานการณ์ฉุกเฉิน. สืบค้นจาก https://rabbitcare.com/blog/lifestyle/10-sign-language-for-basic-everyday-and-emergency

SaiXiii. (ม.ป.ป.). Python Flask Web Application คืออะไร. สืบค้นจาก https://saixiii.com/python-flask-web-application

SoGoodWeb. (2567). CSS คืออะไร มีประโยชน์อย่างไร. สืบค้นจาก https://blog.sogoodweb.com/Article/Detail/79237

วิกิพีเดีย. (ม.ป.ป.). Google logo. สืบค้นจาก https://en.wikipedia.org/wiki/Google_logo

วิกิพีเดีย. (ม.ป.ป.). Kaggle. สืบค้นจาก https://en.wikipedia.org/wiki/Kaggle

วิกิพีเดีย. (ม.ป.ป.). Python (programming language). สืบค้นจาก https://en.wikipedia.org/wiki/Python_%28programming_language%29

วิกิพีเดีย. (ม.ป.ป.). Visual Studio Code. สืบค้นจาก https://en.wikipedia.org/wiki/Visual_Studio_Code

วิกิพีเดีย. (ม.ป.ป.). ไฟล์:HTML5_logo_and_wordmark.svg. สืบค้นจาก https://th.m.wikipedia.org/wiki/ไฟล์:HTML5_logo_and_wordmark.svg

Wikimedia Commons. (ม.ป.ป.). CSS3 logo and wordmark. สืบค้นจาก https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg