



ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

นาย ธนพัฒน์ วงคะสุ่ม

นาย วีระพล แสงภา

โครงการวิจัยระดับปริญญาตรีนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
ปีการศึกษา 2565

ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

นาย ธนพัฒน์ วงคะสุ่ม 116210906093-6

นาย วีระพล แสงภา 116210906060-5

โครงการวิจัยระดับปริญญาตรีนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
ปีการศึกษา 2565

ใบรับรองโครงการวิจัยระดับปริญญาตรี ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะ
วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

หัวข้อปริญญานิพนธ์

ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

ชื่อนักศึกษา

นาย ธนพัฒน์ วงศ์สุ่ม 116210906093-6

นาย วิระพล แสงภา 116210906060-5

ปริญญา

วิทยาศาสตร์บัณฑิต (เทคโนโลยีสารสนเทศ)

อาจารย์ที่ปรึกษาโครงการ

ดร.อนุชา ตุงคชฐาน

คณะกรรมการสอบโครงการวิจัย	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ ดร.ชุติมา ประสาทแก้ว ผู้ช่วยศาสตราจารย์ไกรมน มณีศิลป์ ดร.วิภารัตน์ บุษยาตรัส อาจารย์วุฒิพล วรรณทรัพย์ อาจารย์กิริติบุตร กาญจนเสถียร	

วัน/เดือน/ปี ที่สอบ

สถานที่สอบ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

สาขาวิชาเทคโนโลยีคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

หัวข้อปริญญานิพนธ์

ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

ชื่อนักศึกษา

นาย ธนพัฒน์ วงคะสุ่ม 116210906093-6

นาย วิระพล แสงภา 116210906060-5

อาจารย์ที่ปรึกษาโครงงาน

ดร.อนุชา ตุงค์ษฐาน

ปริญญา

วิทยาศาสตร์บัณฑิต

สาขาวิชา

เทคโนโลยีสารสนเทศ

ปีการศึกษา

2565

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อเสนอเกี่ยวกับการอำนวยความสะดวกให้กับผู้ที่อยู่ในพื้นที่เสี่ยงหรือพื้นที่คนหนาแน่น ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน ระบบตรวจสอบการใส่แมสของผู้ที่ผ่านกล้องโดยผู้ที่ไม่สวมแมสจะส่งสัญญาณแจ้งเตือน การส่งสัญญาณแจ้งเตือนช่วยให้เราทราบถึงผู้ที่ไม่สวมแมสและสามารถแจ้งเตือนให้บุคคลนั้นสวมแมส

กิตติกรรมประกาศ

โครงการวิจัยระดับปริญญาตรี เรื่อง ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน ณ โอกาสนี้ ผู้วิจัยขอขอบพระคุณ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ซึ่งอำนวยความสะดวกในเรื่องของสถานที่ที่ใช้สำหรับดำเนินโครงการ และขอกราบขอบพระคุณ ดร.อนุชา ตุงค์ษฐานที่ได้สละเวลาให้ข้อเสนอแนะแนวทางอันเป็นประโยชน์ ตลอดจนช่วยแก้ไขข้อบกพร่องต่างๆด้วยความเอาใจใส่ และให้ความห่วงใยเสมอมา และขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูล และเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งคณะผู้จัดทำขอขอบพระคุณไว้ ณ ที่นี้ด้วย

ผู้จัดทำ

ธนพัฒน์ วงคะสุ่ม

วิระพล แสงภา

สารบัญ

เรื่อง	หน้า
บทคัดย่อ.....	ก
กิตติกรรมประกาศ	ข
สารบัญ.....	ค
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตโครงการ	2
1.4 เครื่องมือและอุปกรณ์ที่ใช้.....	2
1.5 วิธีการดำเนินงาน	3
1.6 โครงสร้างและเนื้อหาของโครงการ	3
1.7 ผลที่คาดหวังและประโยชน์ของโครงการ.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 Python.....	4
2.2 Teachable Machine	4
2.3 CMD.....	4
2.4 Format Factory	5
2.5 งานวิจัยที่เกี่ยวข้อง	6
บทที่ 3 วิธีการดำเนินงาน	8
3.1 แผนการดำเนินงาน	9
3.2 โครงสร้างการทำงานของระบบ	10
3.3 แผนผังการทำงานของระบบ	11

สารบัญต่อ

เรื่อง	หน้า
3.4 แผนผังการทำงานของระบบ	13
บทที่ 4 ผลการดำเนินงาน.....	19
4.1 ผลการดำเนินงาน.....	19
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	23
5.1 สรุปผลการดำเนินงาน.....	23
5.2 ปัญหาและอุปสรรค	23
5.3 แนวทางการพัฒนาโครงการ.....	23
บรรณานุกรม.....	24
ภาคผนวก	25
ภาคผนวก ก Source code.....	25
ภาคผนวก ข คู่มือการติดตั้ง.....	35
ภาคผนวก ค ออกแบบจำลองระบบเสียง	51
ประวัติผู้จัดทำ.....	57

สารบัญตาราง

ตารางที่	หน้า
3.1 ระยะเวลาการดำเนินงาน	19

สารบัญรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 ตัวอย่างหน้าจอของ CMD 1	5
รูปที่ 2.2 ตัวอย่างหน้าจอของ Format Fact 1.....	5
รูปที่ 3.1 โครงสร้างการทำงานของระบบ	10
รูปที่ 3.2 แผนผังการทำงานของผู้ที่ใส่แมส	11
รูปที่ 3.3 แผนผังการทำงานของผู้ที่ไม่ได้ใส่แมส	12
รูปที่ 3.4 แผนผังการทำงานฟังก์ชันของ Code	12
รูปที่ 3.5 ตัวอย่างแสดงการประมวลผลของ FileCascade	12
รูปที่ 3.6 แผนผังอธิบายการทำงานของระบบ	12
รูปที่ 4.1 ภาพโปรแกรมที่ต้องใช้	19
รูปที่ 4.2 ระบบจะขึ้นกรอบสีแดง Non_Masked เมื่อไม่ได้ใส่แมสก์.....	20
รูปที่ 4.3 ระบบจะขึ้นกรอบสีเขียว Masked เมื่อได้สวมใส่แมสก์	20
รูปที่ 4.4 การสแกนเมื่อไม่มีการตรวจจับในหน้าระบบจะขึ้น No face detected.....	21
รูปที่ 4.5 การสแกนหน้าจะตรวจจับได้เพียงหน้าคนเท่านั้น	21
รูปที่ 4.6 การสแกนหน้าจะตรวจจับที่ไม่ใช่ใบหน้า	22
รูปที่ 4.7 สถิติจำนวนตรวจจับใบหน้าใน 200 ครั้ง	22

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันระบบการตรวจจับใบหน้า มีการใช้งานเพิ่มขึ้นอย่างแพร่หลายทั้งในระบบการ ตรวจสอบบุคคล และ การสวมหน้ากากอนามัย เช่น การตรวจสอบใบหน้าสำหรับการรักษาความปลอดภัย การจับใบหน้าขณะสวมแมสอยู่ การค้นหาและติดตามคนจากภาพใน ฐานข้อมูล และการสื่อสารกันระหว่างผู้ใช้งานกับคอมพิวเตอร์ทั้งในซอฟต์แวร์และฮาร์ดแวร์ เป็นต้น แต่การตรวจจับใบหน้ายังเกิดปัญหาอยู่ เนื่องจากรูปร่าง และลักษณะของใบหน้าแต่ละคนนั้นแตกต่างกันทั้งการแสดงอารมณ์ทางใบหน้า เชื้อชาติ สีผิว และปัจจัยอื่นๆ รวมถึง การสวมแมส ผ่าปิดปาก และทำให้เกิดปัญหาไม่สามารถตรวจจับใบหน้าได้สำหรับ การสแกนหาบนใบหน้ามีหลายประเภท ซึ่งงานวิจัยที่ได้ศึกษามาได้นำเสนอการตรวจจับสิ่งแปลกปลอมบนใบหน้าด้วยกันหลากหลายวิธี เช่น งานวิจัยเรื่องการตรวจจับแว่นตาด้วยวิธีการจำแนกข้อมูลแบบ Boosting ในโมเดล โดยใช้การเรียนรู้หลายๆโมเดลจำแนกประเภทในการตัดสินใจแล้วจะทำการถ่วงน้ำหนักเข้ามาให้แต่ละโมเดลจำแนกประเภท เพื่อกำหนดกลุ่มให้กับข้อมูลใหม่หรืองานวิจัยเรื่องการตรวจจับใบหน้าด้วยเทคนิคการแบ่งภาพออกเป็นตาราง LBP เมื่อได้ค่าฮิสโตแกรมใน แต่ละตารางขึ้นมาแล้ว ใช้เทคนิคสร้างระนาบที่เหมาะสมที่สุดกับข้อมูล (SVM) ในการจำแนก เวกเตอร์ เพื่อหาตำแหน่งใบหน้า โดยตัวระบบจะตรวจจับใบหน้าจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิทัลหรือภาพจากกล้องวิดีโอของบุคคลที่เราสนใจกับฐานข้อมูลใบหน้าที่มีอยู่ และเมื่อเปรียบเทียบเสร็จก็จะแสดงผลใบหน้าที่อยู่ในฐานข้อมูลที่มีใบหน้าเหมือนกับภาพที่นำมาเปรียบเทียบออกมา

1.2 วัตถุประสงค์ของโครงการ

- เพื่อตอบสนองความสะดวกสบายในการเข้าไปในพื้นที่ชุมชนหรือพื้นที่ที่มีคนหนาแน่น
- สามารถบอกให้ทราบถึงบุคคลที่ไม่สวมแมสได้โดยการส่งเสียงแจ้งเตือน
- เป็นเครื่องมือที่สามารถใช้ได้ทั่วไปทุกสถานที่

1.3 ขอบเขตโครงการ

1.3.1 ขอบเขตเชิงเนื้อหา

- การแจ้งเตือนเมื่อตรวจพบผู้ที่ไม่สวมแมส
- สามารถช่วยในการตรวจสอบผู้ที่ไม่สวมแมส
- สามารถแบ่งแยกได้ระหว่างผู้ที่สวมและไม่สวมแมส
- การแจ้งเตือนเมื่อไม่สวมแมสเท่านั้น

1.3.2 ขอบเขตเชิงเทคนิค

- ใช้ Visual Studio เพื่อทำการเขียน Code
- ใช้ CMD ใช้เพื่อติดตั้ง pip install เครื่องมือของ Python
- ใช้ Teachable Machine ในการทำเจเนอรัลภาพให้เป็นโมเดล
- ใช้ Sound of tech จากข้อความให้แปลงเป็น mp3 แล้วดาวโหลด
- ใช้โปรแกรม Format Factory แปลงไฟล์จาก mp3 เป็น .wav

1.4 เครื่องมือและอุปกรณ์ที่ใช้

1.4.1 ฮาร์ดแวร์ที่ใช้ในโครงการ

- เครื่องคอมพิวเตอร์ Spec: - CPU (intel core i5-8300h) Graphics Processing unit GPU (GTX 1050 Ti) Random Access Memory RAM (8 GB)
- กล้องเว็บแคมความชัด 1280*720

1.4.2 ซอฟต์แวร์ที่ใช้ในโครงการ

- Visual Studio
- CMD
- Teachable Machine
- Sound of tech
- Format Factory

1.5 วิธีการดำเนินงาน

- 1.5.1 ใช้ Visual Studio เพื่อทำการเขียน Code
- 1.5.2 ใช้ CMD ใช้เพื่อติดตั้ง pip install เครื่องมือของ Python
- 1.5.3 ใช้ Teachable Machine ในการทำเจเนอรัลภาพให้เป็นโมเดล
- 1.5.4 ใช้ Sound of tech จากข้อความให้แปลงเป็น mp3 แล้วดาวโหลด
- 1.5.5 ใช้โปรแกรม Format Factory แปลงไฟล์จาก mp3 เป็น .wav
- 1.5.6 ทดสอบและแก้ไขจนกว่าระบบจะทำงานได้อย่างสมบูรณ์
- 1.5.7 สรุปผลการดำเนินงานและจัดทำคู่มือการใช้งาน

1.6 โครงสร้างและเนื้อหาของโครงการ

การพัฒนาสารนิพนธ์ฉบับนี้สามารถแบ่งเนื้อหาออกเป็น 5 บท ดังนี้

- บทที่ 1 เป็นการกล่าวถึงปัญหาและแรงจูงใจใน แนวทางการแก้ปัญหา วัตถุประสงค์ ขอบเขต และผลที่คาดหวัง ของการจัดทำโครงการนี้
- บทที่ 2 เป็นการกล่าวถึงทฤษฎีและความรู้พื้นฐานต่างๆ ของโปรแกรมที่เกี่ยวข้องกับการ นำมาใช้ ในการจัดทำโครงการนี้
- บทที่ 3 เป็นการกล่าวถึงรูปแบบโครงสร้างเดิมกับรูปแบบโครงสร้างใหม่ และการตั้งค่าใน ส่วนต่างๆ ของระบบในการจัดทำโครงการนี้
- บทที่ 4 เป็นการกล่าวถึงการทำงานการทำงานของเครื่องมือที่ใช้ และผลการดำเนินงานในการจัดทำโครงการนี้
- บทที่ 5 เป็นการสรุปผลการดำเนินงานในการจัดทำโครงการและข้อดีข้อเสียของระบบ

1.7 ผลที่คาดหวังและประโยชน์ของโครงการ

- ช่วยให้ระบบตรวจจับหน้ากากเพื่อเข้าออกอาคารมีความเสถียรและมีประสิทธิภาพ
- ช่วยลดความเสี่ยงจากโรคโควิด-19 ที่เกิดจากการสัมผัส
- สามารถนำไปพัฒนาต่อยอดเป็นระบบตรวจวัดอุณหภูมิโดยการตรวจจับใบหน้า หรือ ระบบอื่นๆได้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

โครงการวิจัย เรื่อง ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน มีการรวบรวมข้อมูล ศึกษา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 Python

Python เป็นภาษาการเขียนโปรแกรมระดับสูง ที่นำข้อดีของภาษาต่างๆ มารวมไว้ด้วยกัน ถูก ออกแบบมาให้เรียนรู้ได้ง่าย และมีไวยากรณ์ที่ช่วยให้เขียนโค้ดสั้นกว่าภาษาอื่นๆ มีความสามารถใช้ชนิดข้อมูล แบบไดนามิก จัดการหน่วยความจำอัตโนมัติ สนับสนุนกระบวนทัศน์การเขียนโปรแกรม (Programming paradigms) ประกอบด้วย การเขียนโปรแกรมเชิงวัตถุ (OOP : Object Oriented Programming) การเขียน โปรแกรมเชิงคำสั่ง (Imperative Programming) การเขียนโปรแกรมเชิงฟังก์ชัน (Functional) และการเขียน โปรแกรมเชิงกระบวนการ มีลักษณะเป็นภาษาสคริปต์ที่ทำงานร่วมกับภาษาอื่นได้ มีไลบรารีมาตรฐานมากมาย และใช้อินเตอร์พรีเตอร์แปลภาษาโปรแกรมให้ทำงานบนระบบปฏิบัติการได้หลากหลาย ทั้งบน Windows, MAC, Linux และ Unix นอกจากนั้นยังเป็นโปรแกรมแบบ Open source ที่นำใช้ได้ฟรี เหมาะสำหรับ โปรแกรมทั้งขนาดเล็กและขนาดใหญ่ เช่น การสร้างเกม เฟรมเวิร์กพัฒนาเว็บ โปรแกรมที่ใช้กราฟิกติดต่อกับ ผู้ใช้งาน (GUI) งานคำนวณทางวิทยาศาสตร์และสถิติ งานพัฒนาซอฟต์แวร์ และซอฟต์แวร์ควบคุมระบบ เป็นต้น

2.2 Teachable Machine

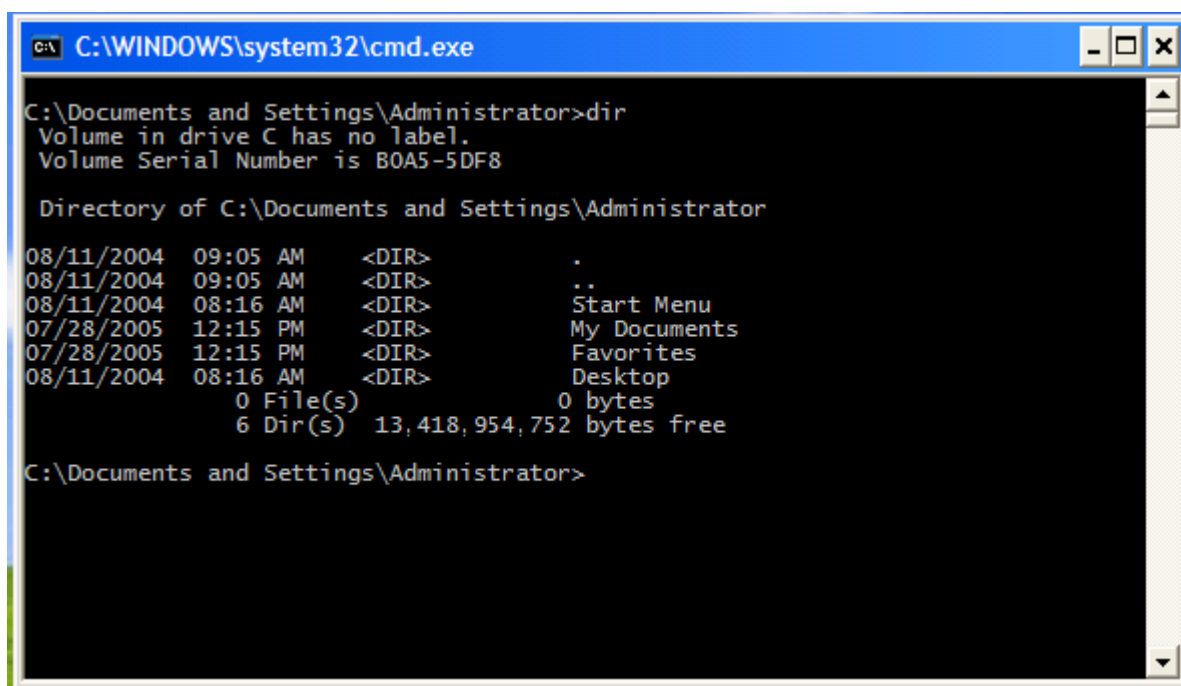
Teachable Machine เครื่องมือสำหรับเรียนรู้เกี่ยวกับ Machine Learning จาก Google เปิดให้ผู้ที่ ต้องการศึกษา Machine Learning เบื้องต้น เข้าศึกษาได้อย่างง่ายโดยไม่ต้องเขียนโค้ด ด้วยการให้ผู้ใช้งานทดลอง สอนเครื่องผ่านระบบกล้องเว็บแคมทำให้ผู้ใช้งานสามารถใช้งานผ่านบนเบราว์เซอร์ได้เลย ไม่จำเป็นต้องติดตั้ง โปรแกรมใด ๆ ก่อนใช้งาน

2.3 CMD

คำสั่งที่ใช้เปิด Windows Command interpreter สามารถเรียกใช้โปรแกรมต่าง ๆ ผ่าน text mode

ปกติเวลาเราใช้งานวินโดวส์มักจะใช้งานอยู่ในโหมดของกราฟิกโหมด (GUI) ซึ่งระบบ GUI นี้ทำให้เราสามารถทำงานทุกอย่างไม่ว่าจะเป็น การสั่งรันโปรแกรม ก๊อปปี้ไฟล์หรือลบไฟล์ ผ่านทางวินโดวส์โหมดได้ แต่ ระบบปฏิบัติการก็มีโหมดที่เรียกว่า ?Command Prompt? หรือบางทีก็เรียกว่า ?Dos Prompt? ซึ่งเป็นโหมดการทำงานในสภาพแวดล้อมที่เป็นตัวอักษร (Text Mode) นั้นหมายความว่าเมื่อต้องการใช้คำสั่งใด ท่านต้อง

พิมพ์คำสั่งต่างๆเอาเอง ซึ่งจะพบเห็นได้ในโปรแกรมเก่าๆหลายตัว หรือแม้แต่โปรแกรมใหม่ๆบางตัว จะเรียกใช้งานหรือปรับแต่งค่าได้เมื่ออยู่ในโหมด Command Prompt เท่านั้น



รูปที่ 2.1 ตัวอย่างหน้าจอของ CMD 1

2.4 Format Factory

เป็นโปรแกรมที่สามารถแปลงไฟล์ วิดีโอ รูปภาพ เสียง และสามารถรวมไฟล์ วิดีโอ หรือเสียงเข้าด้วยกันได้และสามารถแปลงไฟล์ได้ค่อนข้างที่ครอบคลุมเกือบทุกนามสกุลไฟล์ และ ที่ดีที่สุดของโปรแกรมนี้นคือยังสามารถแปลงไฟล์ เพื่อเปิดบนเครื่อง เล่นเกม และ Iphone ได้ด้วย



รูปที่ 2.2 ตัวอย่างหน้าจอของ Format Fact 1

2.5 งานวิจัยที่เกี่ยวข้อง

Vehicle Detection and Classification System for Traffic Video Surveillance [1] นำเสนอวิธีการคัด แยกพาหนะ โดยใช้เทคนิคการแปลงภาพสีเป็นระดับสีเทา (Gray-Scale Image Transform) การเปลี่ยนแปลงลักษณะของวัตถุ (Morphological Operation) เพื่อให้ลักษณะของวัตถุมีความ สมบูรณ์มากขึ้น การเติมส่วนที่เป็น ช่องว่างที่เกิดขึ้นในวัตถุ (Fill Holes Operation) การเลือกกลุ่มข้อมูลที่ต้องการเพื่อหาจุดเชื่อมต่อระหว่างขอบของ วัตถุที่เกิดขึ้นกับขอบของวัตถุข้างเคียง (Region Selection) และการประยุกต์ สมการในการคำนวณ ปริมาณแสงเข้าไปในระบบ เพื่อสร้างพารามิเตอร์ความ น่าเชื่อถือของภาพพื้นหลังในระบบทำการคำนวณภาพ ปริมาณแสงที่เปลี่ยนแปลงในแต่ละช่วงเวลาที่เกิดขึ้นใน ภาพ เพิ่มกระบวนการกรอง ข้อมูลส่วนที่เป็นเงาของวัตถุใน ภาพในการทา งานของ Background Subtraction ปรับค่าพลังงานใน กระบวนการตรวจจับวัตถุภายใน ให้ เหมาะสมกับการตรวจจับวัตถุที่ไม่สามารถบ่งบอกรูปร่างใน ลักษณะของ ยานพาหนะ (Deformable Model) และสร้าง ลักษณะโดยรวมของขอบวัตถุเพื่อแยกเป็นพาหนะประเภท ต่าง ๆ (Classification) โดยการ Ratio หั้วอัตราส่วนความ กว้างและความยาวของวัตถุและนำมาช่วยในการ คำนวณ ความเป็นไปได้ในการคัดแยกพาหนะ

จากงานวิจัย [2] ได้ใช้แนวคิดของโครงข่ายประสาทเทียมคอนโวลูชัน มาพัฒนาร่วมกับแบบจำลอง ใน การตรวจจับอารมณ์ทางใบหน้า โดยใช้ชุดข้อมูลทางด้านอารมณ์ FER-2013 พัฒนาแบบจำลองที่สามารถ ตรวจจับอารมณ์ทางใบหน้า มีความแม่นยำที่ 66% นอกจากนี้ยังใช้ชุดข้อมูลเกี่ยวกับเพศสภาพ IMDB ซึ่งให้ ความแม่นยำที่ 96% มาพัฒนาแบบจำลองของตนเองเพื่อตรวจสอบเพศชาย และเพศหญิง จากงานวิจัยนี้ สามารถยืนยันได้ว่าสามารถพัฒนาระบบการตรวจจับอารมณ์ทางใบหน้าในทางปฏิบัติได้จริง ขณะที่การ ตรวจจับใบหน้าเพื่อสร้างกรอบสี่เหลี่ยมมาครอบใบหน้าของนักศึกษาแต่ละคน ได้พัฒนาแบบจำลองที่ สามารถ ตรวจจับใบหน้า และสร้างกรอบสี่เหลี่ยมมาครอบหน้าผ่านกล้องวิดีโอได้จริง โดยใช้ Caffee ที่เป็น Neural Network ตระกูล CNN ใช้ความสามารถของหน่วยประมวลผลกราฟิก (Graphics Processing Unit: GPU) มาประมวลผลภาพแบบจำลองที่ใช้ในงานวิจัยของผู้วิจัยได้เลือกใช้ แบบจำลอง UltraLight รูปแบบการศึกษา ออนไลน์ระยะไกลซึ่งอาศัยการตรวจจับใบหน้าของนักศึกษาผ่าน ทางอารมณ์ แต่ให้ความสำคัญกับการ เคลื่อนไหวของดวงตา และปาก ทำให้เกิดความแม่นยำมากกว่าเดิมที่ ให้ความสนใจลักษณะทางอารมณ์ 6 ลักษณะ ผลของการวิจัยยืนยันว่าการให้ความสำคัญของการ เคลื่อนไหวของดวงตา และปาก ทำให้ผลลัพธ์ ความแม่นยำในการตรวจจับใบหน้ามีผลลัพธ์ที่ดีขึ้น สนับสนุนเรื่องการเพิ่มความสนใจในการตรวจจับการ เคลื่อนไหวของดวงตา และบริเวณส่วนหัว ทำ ให้ผลของความแม่นยำจากการแสดงผลทางอารมณ์ผ่านใบหน้า มีความแม่นยำมากขึ้น นอกจากนี้งานวิจัยนี้ ยังแสดงให้เห็นว่าการศึกษาแบบ E-Learning โดยตรวจสอบความ สนใจ หรือไม่สนใจของผู้เรียน ผ่าน อารมณ์ทางใบหน้า มีความเป็นไปได้ในการเอามาประเมินผลได้จริง เสนอ

แนวทางในการวัด ความสนใจ หรือไม่สนใจผ่านอารมณ์ทางใบหน้าโดยกำหนดอัตราส่วนในการจำแนกความเข้าใจหรือไม่ เข้าใจผ่านกลุ่มของใบหน้า โดยมีอัตราส่วนที่มากกว่า 0.5 จาก 1.0 จะบอกว่าอารมณ์ทางใบหน้า นั้น มีความ เป็นไปได้ว่า ผู้ใช้เข้าใจ หรือไม่เข้าใจในเนื้อหาที่ทำการสอน เสนอรูปแบบระบบที่สามารถ ตรวจสอบสถานะอารมณ์ทางใบหน้าผ่านกล้องเว็บแคม (Webcam) ภายในห้องเรียน ทำให้สามารถรับรู้ว่ โอกาสที่อารมณ์ของนักศึกษาจะมีแนวโน้มหดหู่ หรือเศร้าโศกเพียงใดในการเข้าชั้นเรียน ทำให้อาจารย์ ผู้สอน สามารถที่จะปรับเปลี่ยนวิธีการสอนเพื่อลดความตึงเครียดในขณะที่เรียนในชั้นเรียน ทางเลือกในการตรวจสอบ ทักษะของผู้เรียนในชั้นเรียน ขณะที่อาจารย์ทำการสอน นักศึกษา สามารถแสดงความรู้สึกผ่านสัญลักษณ์รูป อารมณ์ (Emoji Icon) และพิมพ์คำถามลงกระดานถามตอบได้ วารสารวิทยาศาสตร์ลาดกระบัง ปีที่ 30 ฉบับที่ 2 เดือนกรกฎาคม-ธันวาคม 2564 46 ทันที ขณะเดียวกันอาจารย์ก็จะสามารถตรวจสอบคำถามและตอบ คำถามนักศึกษา และเห็นจำนวนของ คำถามผ่าน Dashboard ผังอาจารย์ได้เช่นกัน

Real-Time Lane Detection for Driving using Embedded System [3] นำเสนอวิธีการ ตรวจสอบ ช่องทางจราจรในขณะที่ขับรถเพื่อช่วยแจ้งเตือนผู้ขับขี่ใน ขณะที่ผู้ขับขี่ประมาท หรือไม่ชำนาญ โดย ใช้หลักการใน การประมวลผลภาพ (Image Processing) มาประยุกต์ใช้ งานกับระบบสมองกล ผังตัว โดยใช้ กล้องในการบันทึกภาพ ช่องทางจราจรบริเวณด้านหน้าของรถแบบเรียลไทม์และนำ ภาพที่บันทึกมาการ วิเคราะห์โดยใช้หลักการในการ ประมวลผลภาพโดยใช้เทคนิค Edge Detection และ Hough Transform ซึ่ง เป็นหลัก การของการประมวลผล ภาพและแสดงการแจ้ง เตือนให้ผู้ขับขี่ทราบทางจอแสดงผล โดยใช้ระบบ สมองกลฝังตัว STM32 สถาปัตยกรรม 32-bits ARM Cortex-M3 และ RapidSTM32 Block set ซึ่งเป็นชุด กล่อง คำสั่ง แบบรูปภาพที่สามารถใช้งานร่วมกับ MATLAB/Simulink เป็นตัวประมวลผลงานวิจัยนี้สามารถ ช่วยแจ้งเตือนผู้ขับขี่ในขณะที่ขับรถออกนอกหรือคร่อม ช่องทางจราจรซึ่งช่วยลดอุบัติเหตุที่จะเกิดขึ้นท้องถนน ได้

บทที่ 3

วิธีการดำเนินงาน

ในส่วนของการดำเนินการวิจัยของ ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน มีจุดประสงค์เพื่อทราบถึงคนที่ไม่ได้ใส่แมสผ่านหน้าจอ ทำให้ผู้ใช้สามารถทราบถึงการใส่แมสและไม่ได้ใส่และการแจ้งเตือนสำหรับผู้ที่ไม่ได้ใส่แมส โดยศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับการตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน ใช้เทคนิคการพัฒนาโปรแกรมด้วยภาษา Python และกล้องเพื่อสแกนคนที่ไม่ได้ใส่แมส ซึ่งสรุปขั้นตอนการดำเนินงานวิจัยได้ ดังนี้

3.1 แผนการดำเนินงาน

3.2 โครงสร้างการทำงานของระบบ

3.3 แผนผังการทำงานของระบบ

3.4 การออกแบบระบบ

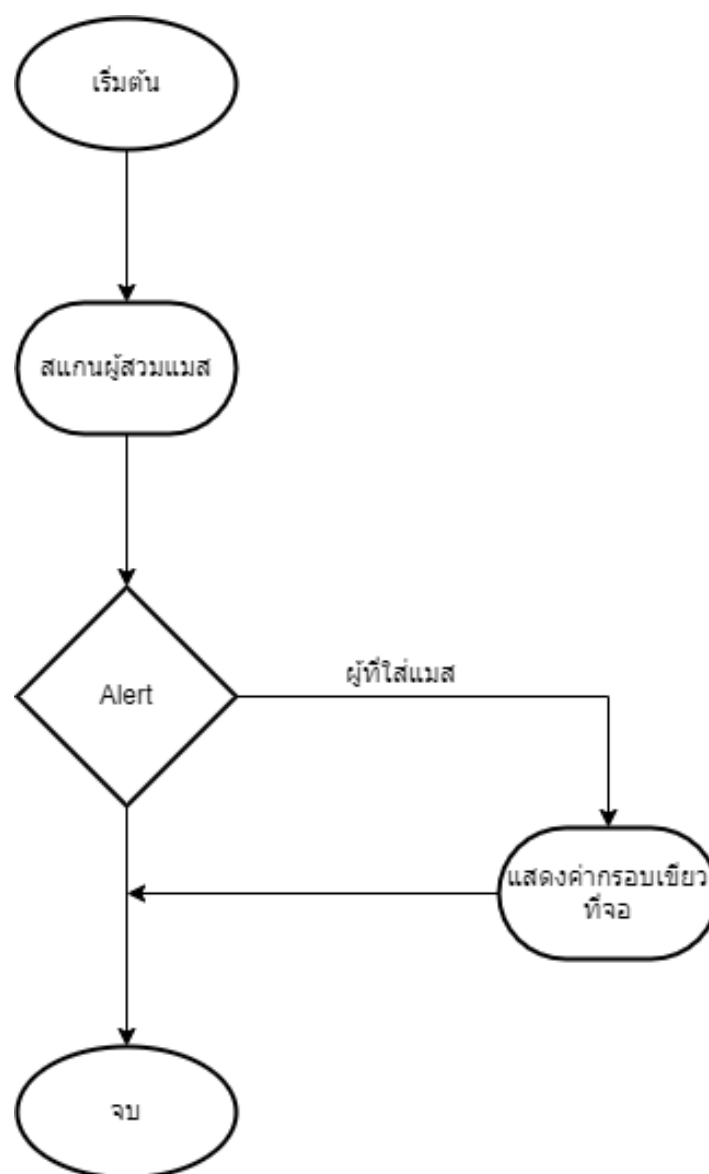
3.2 โครงสร้างการทำงานของระบบ



รูปที่ 3.1 โครงสร้างการทำงานของระบบ

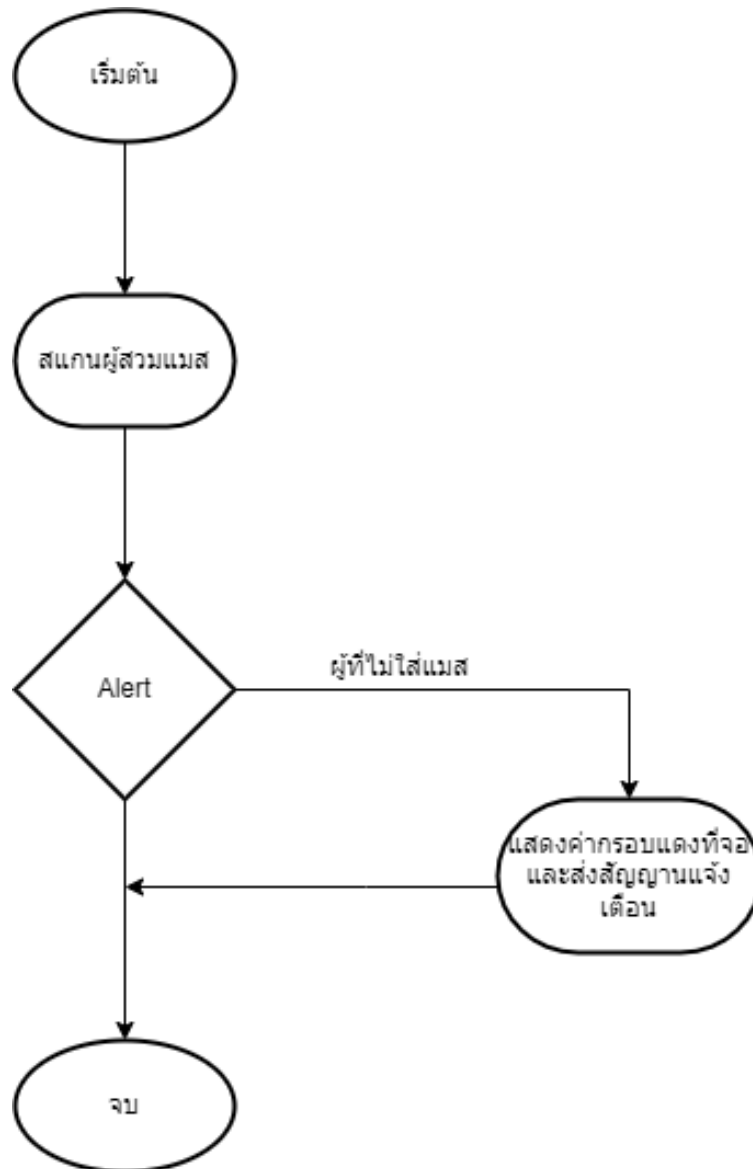
โครงสร้างการทำงานของระบบ ระบบจะเริ่มต้นด้วย User สวมหรือไม่ได้สวมแมสเดินผ่านกล้องโปรแกรมจะสแกนหน้าของคนที่ไม่สวมแมสและไม่ได้สวม คนที่ไม่สวมแมสจะขึ้นกรอบเขียวที่ใบหน้า ส่วนคนที่ไม่ได้สวมจะมีกรอบสีแดงขึ้นตรงใบหน้าในจอแสดงผล

3.3 แผนผังการทำงานของระบบ



รูปที่ 3.2 แผนผังการทำงานของผู้ที่ใส่แมส

แผนผังการทำงานของผู้ที่ใส่แมสระบบจะเริ่มต้นโดยการสแกนใบหน้าของคนที่ผ่านมาคล้อง ผู้ที่ใส่แมสจะแสดงค่ารอบเขียวบนจอแสดงผล



รูปที่ 3.3 แผนผังการทำงานของผู้ที่ไม่ได้ใส่แมส

แผนผังการทำงานของผู้ที่ไม่ได้ใส่แมส ระบบจะเริ่มต้นโดยการสแกนใบหน้าของคนๆหนึ่งที่เดินผ่านกล้อง ผู้ที่ไม่ได้ใส่แมสจะขึ้นค่าการอบแดงบนจอแสดงผลและส่งสัญญาณแจ้งเตือน

3.4 การออกแบบระบบ

ขั้นตอนการออกแบบการทำระบบสแกนใบหน้าผู้สวมแมสและไม่สวมแมส มีขั้นตอนดังนี้

3.4.1 ออกแบบระบบสแกนใบหน้าผู้สวมแมสและไม่สวมแมส

3.4.2 การสแกนใบหน้าผ่านกล้อง Webcam

3.4.3 นำข้อมูลรูปภาพส่งผ่าน Machine Learning เพื่อบันทึกใบหน้า

3.4.4 ใช้ภาษา Python ในการเขียนโปรแกรมใน Visual Studio เพื่อเขียนโค้ดแบ่งแยกใบหน้า

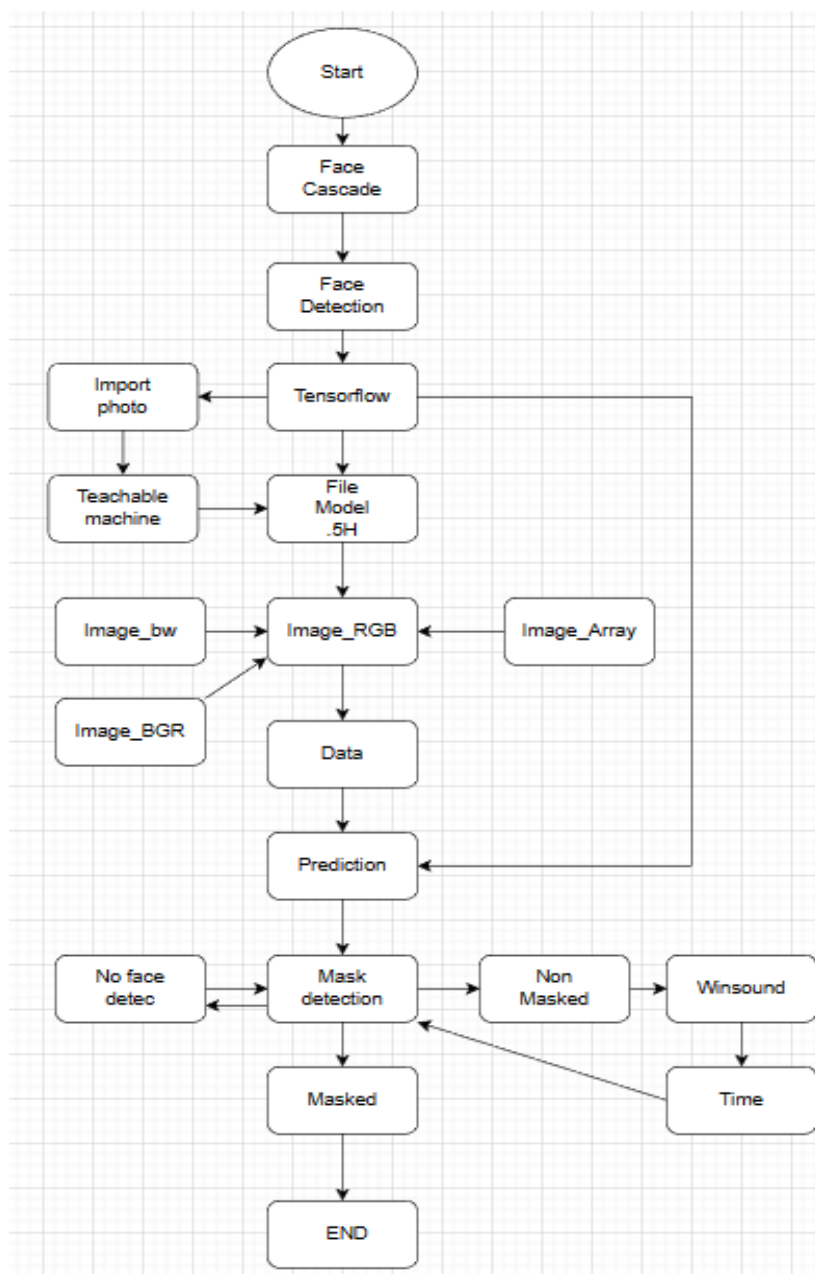
3.4.5 ทำการแยกผลระหว่างผู้ที่สวมแมสและไม่สวมแมส

3.4.6 เมื่อตรวจพบแมสระบบจะขึ้นกรอบสีเขียวและไม่ส่งสัญญาณใดๆ

3.4.7 เมื่อตรวจไม่พบแมสระบบจะขึ้นกรอบสีแดงและส่งสัญญาณให้สวมแมส

3.4.8 เมื่อตรวจไม่พบใบหน้าจะแสดงผลบนหน้าจอว่าไม่พบใบหน้า

Flow chart ฟังก์ชันการทำงาน



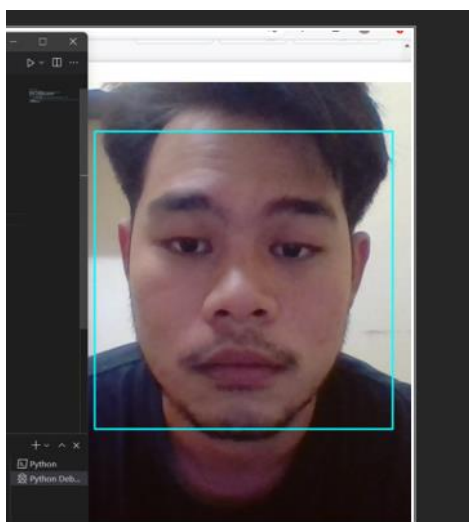
รูปที่ 3.4 แผนผังการทำงานฟังก์ชันของ Code

Face Cascade

ในการใช้งานจำเป็นต้องมีไฟล์ที่ได้รับการเรียนและแยกประเภทหรือมีข้อมูลการเรียนรู้แล้วว่า วัตถุนั้นหรือ สิ่งเหล่านั้นคืออะไร โดยไฟล์จะมีนามสกุล .XML ในการตรวจจับใบหน้านั้น เราสามารถเลือกได้ว่าตรวจจับผ่าน อุปกรณ์ใด หรือ ชนิดของมัลติมีเดียแบบใดก็ได้ เช่น Video , Webcam , รูปภาพ ในตัวอย่างต่อไปนี้ จะเป็น การใช้รูปภาพในการตรวจหาใบหน้า

Face Detection

หลังจากที่ได้ประมวลผลจากกล้อง Webcam รวมกับ Filecascade ก็จะได้เป็นกรอบสี่เหลี่ยมขึ้นมาตรวจจับ ใบหน้า



รูปที่ 3.5 ตัวอย่างแสดงการประมวลผลของ FileCascade

TensorFlow

คือ library ไว้สำหรับสร้างหรืออ่านค่า File ของ machine learning models เพื่อให้สามารถแยก Class ต่างๆตามที่เราต้องการได้

Inport photo

นำเข้ารูปภาพผู้สวมแมสและผู้ไม่สวมแมสไปให้เรียนรู้แยก class ต่างๆใน Teachable machine

Teachable machine

การสร้าง Machine Learning แบบง่าย เร็ว สะดวก ไม่จำเป็นต้องรู้เรื่องการเขียนโค้ดใดๆ เพียงแค่ฝึกคอมพิวเตอร์ให้รู้จักภาพ เสียง และท่าทาง จากนั้นส่งออกโมเดลเป็น File และ Code ทำให้ส่วนถัดไป

File_Model.5h

เป็นนามสกุลไฟล์ที่ใช้กันทั่วไปในการบันทึกโมเดล Keras ในรูปแบบ Hierarchical Data Format เวอร์ชัน 5 (HDF5) HDF5 เป็นรูปแบบข้อมูลที่ออกแบบมาเพื่อจัดเก็บและจัดระเบียบข้อมูลจำนวนมากอย่างมีประสิทธิภาพ และมักจะใช้ในการคำนวณทางวิทยาศาสตร์และการเรียนรู้ของเครื่องเพื่อนำไปใช้ในส่วนของการแยก Class รวมกับสีต่างๆ

Image_RGB

คือ นำเข้าภาพที่มีสีจากแม่สี 3 สี คือสีแดง สีเขียว และสีน้ำเงิน การใช้สัดส่วนของสี 3 สีนี้ต่างกัน จะทำให้เกิดสีต่าง ๆ ได้อีกมากมายจะถูกนำไปใช้กับสื่อที่ใช้ระบบแสงแสดงภาพ

Image_bw

คือ การนำเข้าภาพเพียง 2 สี คือ สีขาว และสีดำ

Image array

คือการใช้งานของ แม่แบบ Image array ใช้ในการเขียนคำอธิบายแม่แบบหรือมอดูลและใส่หมวดหมู่

Image_BGR

คือ การแบ่งกลุ่มของสีต่างๆให้อยู่กลุ่มเดียวกัน Data คือข้อมูลดิบหรือข้อเท็จจริงที่เกิดขึ้นจากการเก็บบันทึกเหตุการณ์ต่าง ๆ โดยที่ข้อมูลสามารถอยู่ในรูปแบบของตัวอักษร ตัวเลข หรือจะเป็นสัญลักษณ์

Prediction

คือการทำนายหรือคาดเดาข้อมูลต่างๆ Mask detection คือการตรวจจับการใส่แมสบนใบหน้าของผู้คน

No face detec

คือการตรวจไม่พบของข้อมูลใบหน้า

Masked

คือ การที่โปรแกรมตรวจพบการใส่แมสบนใบหน้าผู้คน

Non_Masked

คือ การที่โปรแกรมตรวจไม่พบการใส่แมสบนใบหน้าผู้คน

Masked

คือ การที่โปรแกรมตรวจพบการใส่แมสบนใบหน้าผู้คน

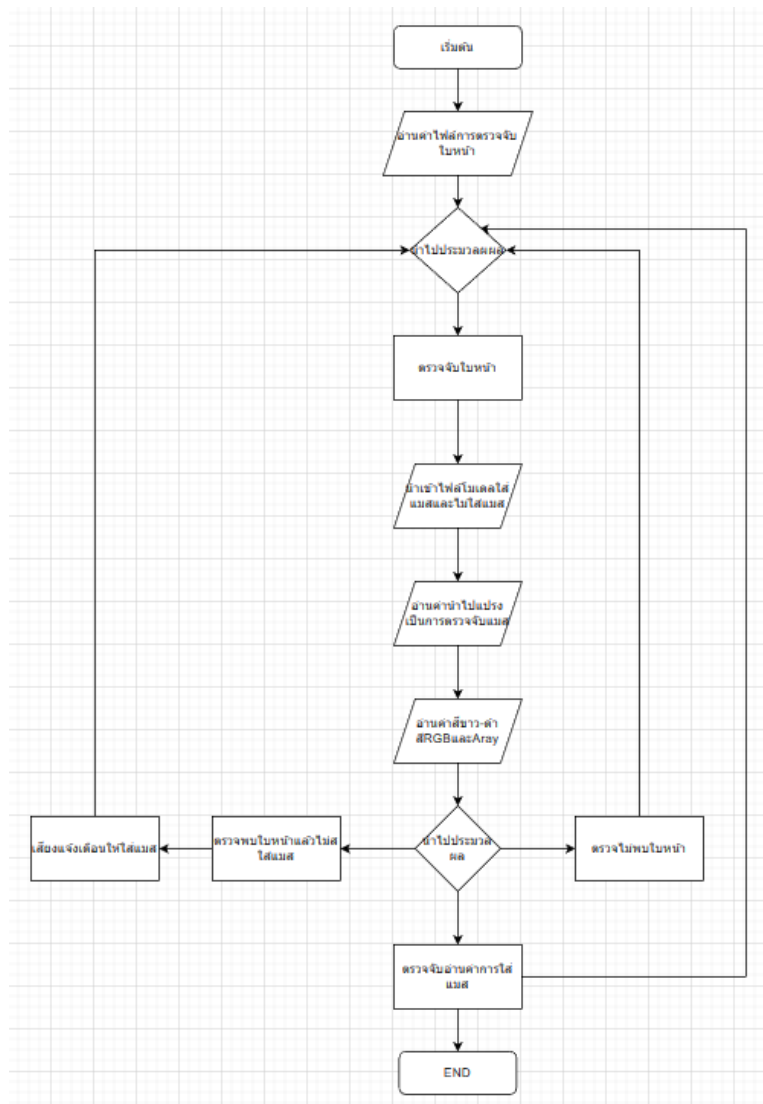
Winsound

คือ การส่งเสียงแจ้งเตือนของระบบเมื่อตรวจไม่พบแมสบนใบหน้า

Time

คือ เวลาของการส่งเสียงแจ้งเตือน การส่งเสียงแจ้งเตือนจะหยุดเมื่อใส่แมสหรือปิดโปรแกรม

Flow chart อธิบายการทำงานของระบบ



รูปที่ 3.6 แผนผังอธิบายการทำงานของระบบ

เริ่มต้นจากการรับค่ากล้อง webcam อ่านค่าไฟล์รูปภาพหรือวิดีโอที่รับค่านำมาประมวลผลตรวจจับใบหน้า นำเข้าไฟล์โมเดลการใส่แมสและไม่ใส่แมส อ่านค่าสีขาว-ดำ สี RGB และ Array นำไปประมวลผลการใส่แมสและไม่ใส่แมส ตรวจพบใบหน้าและใส่แมสจะแสดงบนจอเป็นกรอบสีเขียว เมื่อตรวจพบใบหน้าและไม่ได้ใส่แมสระบบจะแสดงบนจอเป็นกรอบสีแดงและจะส่งเสียงแจ้งเตือน กรุณาสวมแมสค่ะ เมื่อระบบตรวจไม่พบใบหน้าจะแสดงบนจอว่า ไม่พบใบหน้า

บทที่ 4

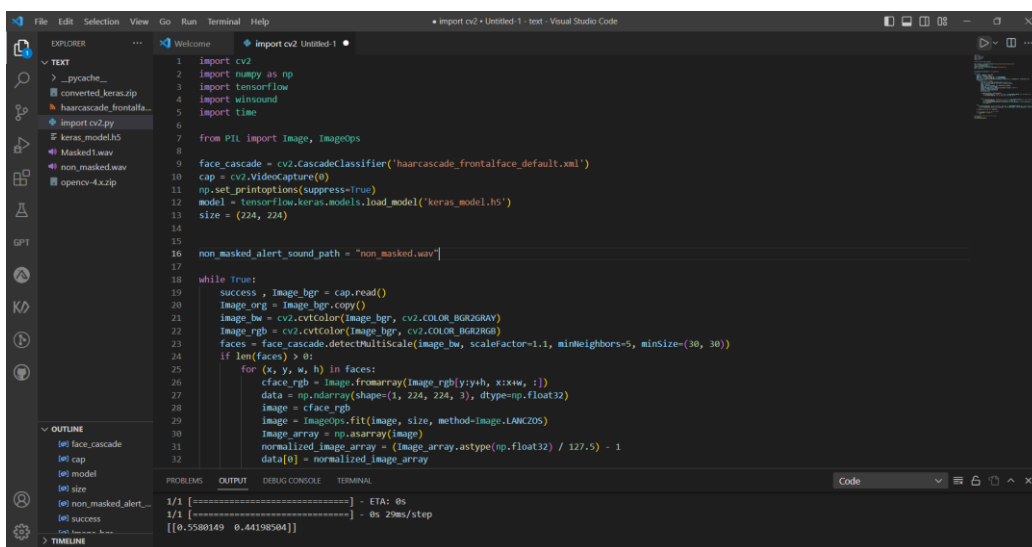
ผลการดำเนินงาน

ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน เป็นระบบที่จะตอบสนองความต้องการ ในการใช้งานระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือนให้เกิดความสะดวกรวดเร็วและเพื่อให้ง่ายต่อการตรวจหาผู้ที่ไม่สวมแมส

4.1 ผลการดำเนินงาน

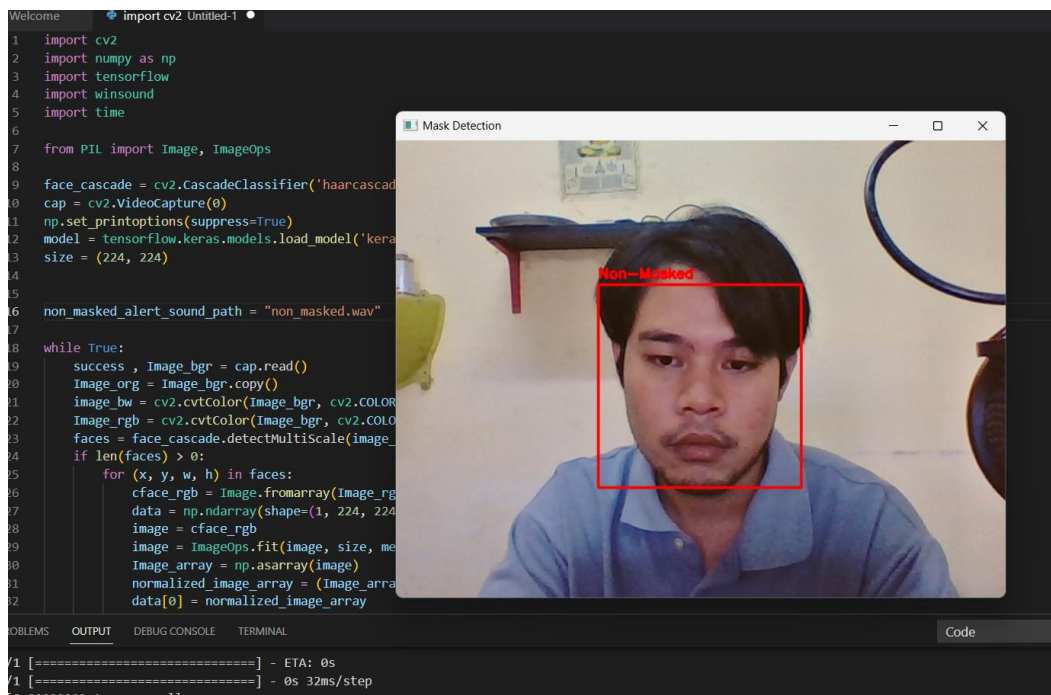
เราจะต้องมีภาพจำนวนมากของคนที่ไม่ใส่แมส และไม่ใส่แมสเพื่อใช้เทรน Machine Learning ให้ตรวจจับคน 2 ประเภทนี้ได้เราเขียนโปรแกรมเพื่อถ่ายรูปจาก webcam หลายๆรูป หลายๆ มุมที่ไม่ใส่แมส และไม่ใส่แมส เพื่อใช้เป็นข้อมูลให้กับ Teachable Machine หรือหากใครมีข้อมูลที่อยู่ในมืออยู่แล้ว สามารถใช้ข้อมูลนั้นๆ แทนได้ทันที

1) ติดตั้งเครื่องมือที่ต้องใช้



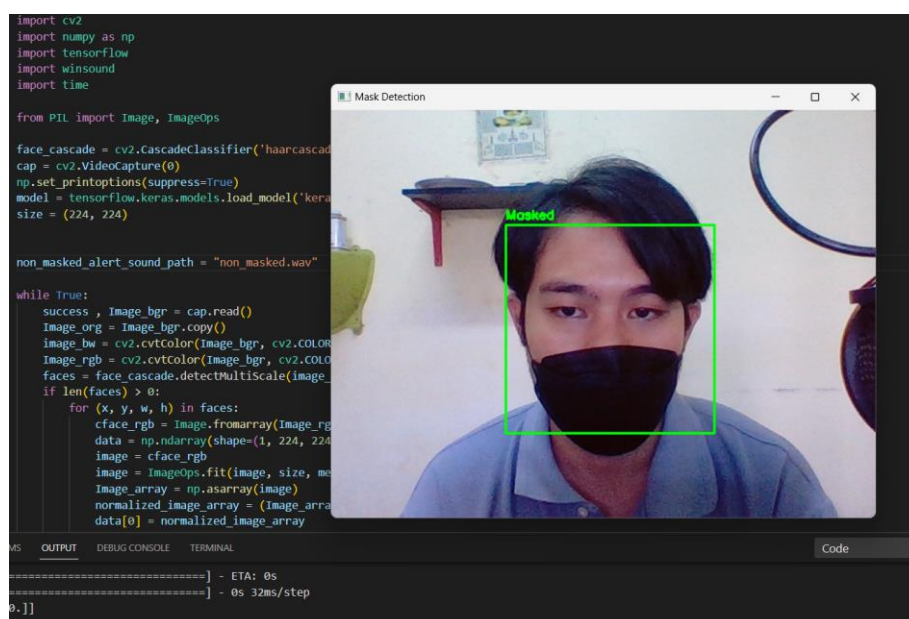
รูปที่ 4.1 ภาพโปรแกรมที่ต้องใช้

- 2) เมื่อไม่ได้ใส่แมสระบบจะขึ้นกรอบสีแดง Non_Masked พร้อมเสียงแจ้งเตือนบอก
กรุณาใส่แมสค่ะ



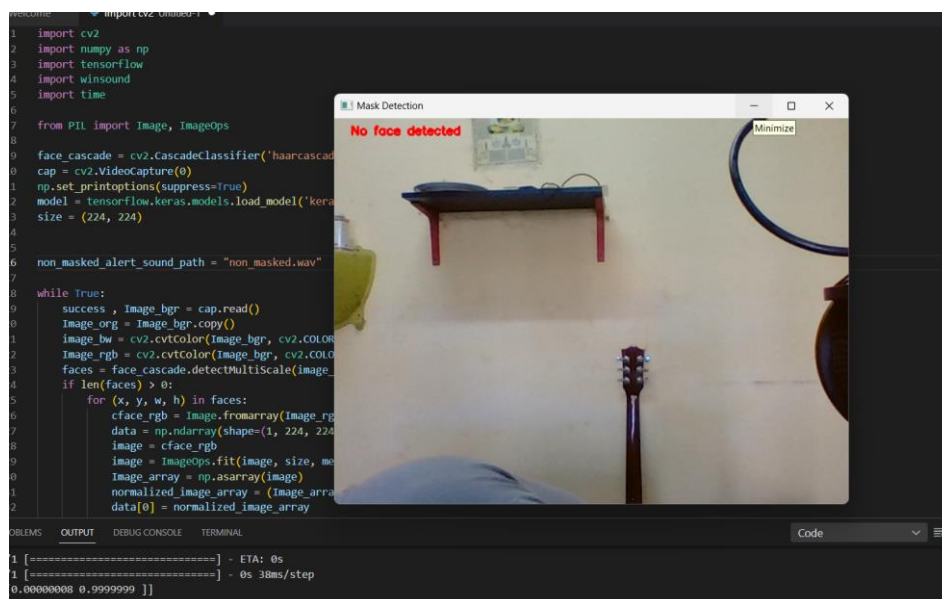
รูปที่ 4.2 ระบบจะขึ้นกรอบสีแดง Non_Masked เมื่อไม่ได้ใส่แมส

- 3) เมื่อเราใส่แมสระบบจะขึ้นกรอบสีเขียว Masked



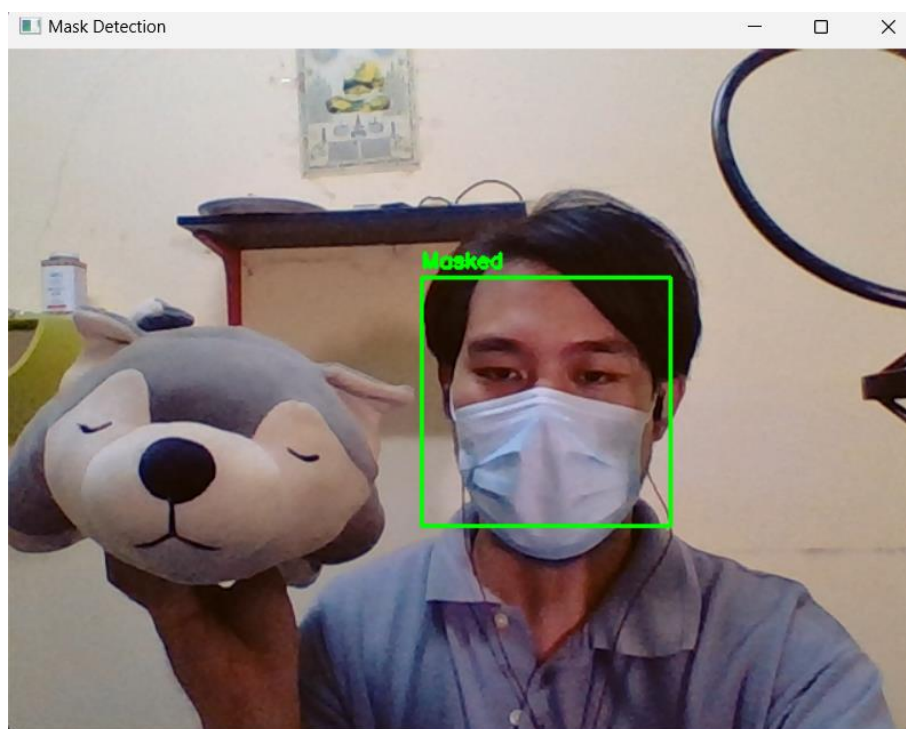
รูปที่ 4.3 ระบบจะขึ้นกรอบสีเขียว Masked เมื่อได้สวมใส่แมส

- 4) และเมื่อไม่มีการตรวจจับในหน้าระบบจะขึ้น No face detected



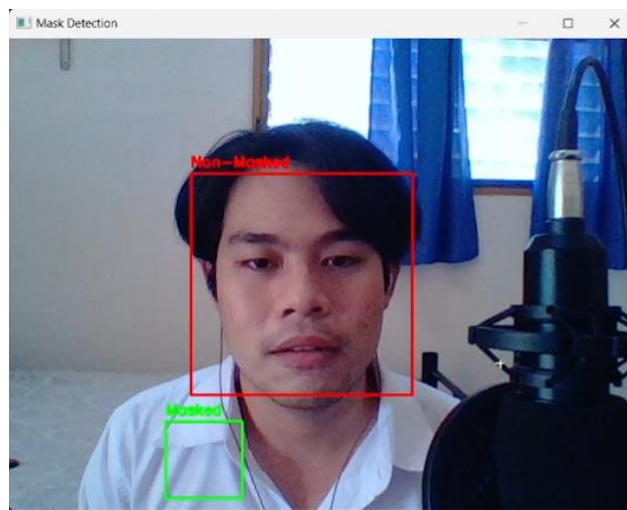
รูปที่ 4.4 การสแกนเมื่อไม่มีการตรวจจับในหน้าระบบจะขึ้น No face detected

- 5) ระบบจะสามารถตรวจจับได้เพียงหน้าคนเท่านั้น

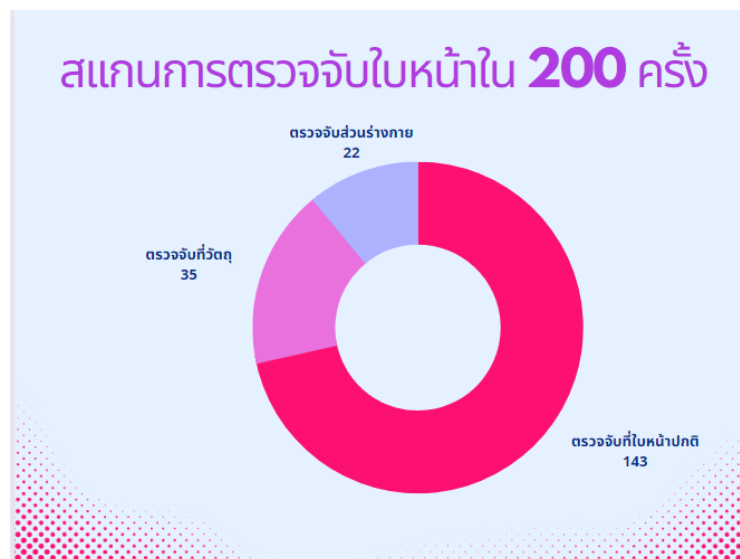


รูปที่ 4.5 การสแกนหน้าจะตรวจจับได้เพียงหน้าคนเท่านั้น

- 6) ผลการทดลองได้พบว่ายังมีปัญหาการตรวจจับใบหน้าขึ้นกรอบตัววัตถุที่ไม่ใช่ใบหน้ามีดังนี้
- ตรวจจับที่ใบหน้าปกติ
 - ตรวจจับที่วัตถุ
 - ตรวจจับส่วนร่างกาย



รูปที่ 4.6 การสแกนหน้าจะตรวจจับที่ไม่ใช่ใบหน้า



รูปที่ 4.7 สถิติจำนวนตรวจจับใบหน้าใน 200 ครั้ง

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

ในส่วนของบทที่5นี้จะกล่าวถึงบทสรุปภาพรวมของวิทยานิพนธ์หัวข้อ ระบบตรวจจับการใส่แอมสและส่งสัญญาณแจ้งเตือน และคำแนะนำเพื่อเป็นประโยชน์ต่อผู้ที่ต้องการจะทำการศึกษเกี่ยวกับ เรื่องการ Teachable Machine ที่จะนำไปสู่การพัฒนาต่อไปภายในอนาคตต่อไป

5.1 สรุปผลการดำเนินงาน

สรุปผลการดำเนินงานการจัดทำโครงการ ระบบตรวจจับการใส่แอมสและส่งสัญญาณแจ้งเตือน โดยได้จัดทำโครงการตลอดระยะเวลา4เดือนจนบรรลุวัตถุประสงค์ตามที่ได้กล่าวไว้ในบทที่1 ครบทุกประการโดยระบบสามารถตรวจจับผู้ที่ใส่แอมสและไม่ใส่แอมสโดยมี Teachable Machine ในการช่วยในการตรวจสอบใบหน้าของผู้ใส่แอมสและไม่ใส่แอมส สรุปผลได้ว่าตัวโครงการฉบับนี้สามารถทำให้แอดมินหรือผู้ดูแลระบบสามารถสแกนใบหน้าได้อย่างได้อย่างรวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้นทำให้ประหยัดค่าใช้จ่ายภายในองค์กร สามารถรับการแจ้งเตือนได้ตลอดเวลาเมื่อมีผู้ที่ไม่ใส่แอมส อีกทั้งจะสามารถคัดกรองผู้คนได้ หากเจ้าหน้าที่บางคนไม่ว่างก็สามารถช่วยดูแทนได้ซึ่งเป็นผลดีต่อการควบคุมผู้คนและทำให้การบริหารจัดการดีขึ้น

5.2 ปัญหาและอุปสรรค

- 5.2.1 แอปพลิเคชันมีการตอบสนองช้าเพราะว่า การประมวลผลเสียงและข้อมูลล่าช้า
- 5.2.2 ระบบยังไม่แม่นยำในการตรวจจับใบหน้าของผู้ที่ใส่แอมส
- 5.2.3 ข้อมูลอาจเกิดการผิดพลาดบางอย่างและเสียงแจ้งเตือนออกมา

5.3 แนวทางการพัฒนาโครงการ

- 5.3.1 เพิ่มข้อมูลภาพใบหน้าให้มากยิ่งขึ้นในการตรวจจับให้มีความแม่นยำยิ่งขึ้น
- 5.3.2 สามารถเพิ่มการวัดค่าอุณหภูมิอินฟราเรดตรวจจับความร้อนในร่างกาย
- 5.3.3 สามารถพัฒนาต่อยอดด้วยการทำเป็นอุปกรณ์ในการตรวจจับใบหน้านำไปใช้จริงในองค์กร
- 5.3.4 ปรับปรุงฮาร์ดแวร์ให้ทันสมัยและเพิ่มประสิทธิภาพในการประมวลผลได้มากยิ่งขึ้น

บรรณานุกรม

[1] ไตรวิทย์ อินทจักร. ระบบตรวจจับและคัดแยกรถ สำหรับกล้องวงจรปิดบนท้องถนน [วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรม คอมพิวเตอร์] สงขลา: มหาวิทยาลัยสงขลานครินทร์; 2562.

[2] Arriaga, O., Valdenegro-Toro, M. and Plöger, P. 2017. Real-time convolutional neural networks for emotion and gender classification. Available at: <https://arxiv.org/abs/1710.07557>. Retrieved 9 Jan 2021.

[3] กิตติศักดิ์ ทองยวน, มหศักดิ์ เกตุฉ่ำ. อุปกรณ์ตรวจสอบ ช่องทางจราจรแบบเรียลไทม์สำหรับการ ขับรถทำงาน บนระบบสมองกลฝังตัว. The Tenth National Conference on Computing and Information Technology NCCIT2014; ภูเก็ต. กรุงเทพฯ: คณะ เทคโนโลยีสารสนเทศ มหาวิทยาลัย เทคโนโลยีพระจอมเกล้าพระนครเหนือ; 2563.

[4] อาทิตย์ศรีแก้ว. (2563). เทคนิคการตรวจจับใบหน้าด้วยโครงข่าย ART แบบดัดแปลง. รายงาน วิจัยมหาวิทยาลัยเทคโนโลยีสุรนารี.

[5] ยงยุทธ ละมุลมอญ , ธนาสัย สุคนธ์พันธุ์. (2019). การตรวจจับหมวกนิรภัยและการใช้อาวุธปืน เพื่อเตือนภัยเหตุโจรกรรมจากภาพกล้องวงจรปิดแบบเวลาจริง. 2(7), 18-25. [สืบค้นเมื่อ 5 เมษายน พ.ศ. 2565].

[6] THAIPROGRAMMER. Deep learning.[ออนไลน์]. (ม.ป.ป.). เข้าถึงได้จาก: <https://www.thaiprogrammer.org/2018/12/deeplearning%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/> [สืบค้นเมื่อ 6 เมษายน พ.ศ. 2565].

[7] AOSOFT. Python คืออะไร.[ออนไลน์]. (ม.ป.ป.). เข้าถึงได้จาก: <https://www.aosoft.co.th/article/322/Python-คืออะไร-ภาษา-python-ใช้ทำอะไร. html> [สืบค้นเมื่อ 6 เมษายน พ.ศ. 2565].

[8] เกรียงศักดิ์ ตรีประพิน, ภัคภัทร นาอุดม และไพชยนต์ คงไชย. (2561). ระบบตรวจสอบนักศึกษา เข้าเรียนด้วยรู้จำใบหน้า. วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยอุบลราชธานี, 2(2), 92-101.

ภาคผนวก ก

Source code

```
In [1]:  
!pip install opencv-python
```

```
In [3]:  
!pip install tensorflow
```

```
In [4]:  
!pip install keras
```

```
In [5]:  
!pip install Pillow
```

```
1  ✓ import cv2  
2  import numpy as np  
3  import tensorflow  
4  import winsound  
5  import time  
6
```

```

In [7]:
image = r'test_photo.jpg'
face_cascade = "haarcascade_frontalface_default.xml"

image_bgr = cv2.imread(image)
image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

face_classifier = cv2.CascadeClassifier(face_cascade)
faces = face_classifier.detectMultiScale(image_bw)

print(f'There are {len(faces)} faces found.')

for face in faces:
    x, y, w, h = face
    cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow("Faces found", image_bgr)
cv2.waitKey(0)

```

There are 1 faces found.

```

In [8]:
webcam = cv2.VideoCapture(1)
webcam

```

```

Out[8]:
<VideoCapture 00000273F532C470>

```

```
In [9]:  
success, image_bgr = webcam.read()
```

```
Out[9]:  
(True, array([[[140, 117, 67],  
[141, 117, 66],  
[141, 116, 64],  
...,  
[ 37, 29, 64],  
[ 17, 17, 56],  
[ 17, 17, 56]],  
  
...,  
  
[[ 66, 59, 9],  
[ 67, 60, 10],  
[ 68, 62, 10],  
...,  
[149, 89, 105],  
[146, 86, 100],  
[147, 87, 101]]], dtype=uint8))
```

```
In [11]:  
face_cascade = "haarcascade_frontalface_default.xml"
```

```
In [12]:  
while True:  
  
    success, image_bgr = webcam.read()  
  
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)  
  
    face_classifier = cv2.CascadeClassifier(face_cascade)  
    faces = face_classifier.detectMultiScale(image_bw)  
  
    print(f'There are {len(faces)} faces found.')  
  
    for face in faces:  
        x, y, w, h = face  
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)  
  
    cv2.imshow("Faces found", image_bgr)  
    cv2.waitKey(1)
```

```
In [16]:
count = 9999
while True:

    success, image_bgr = webcam.read()
    image_org = image_bgr.copy()
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

    face_classifier = cv2.CascadeClassifier(face_cascade)
    faces = face_classifier.detectMultiScale(image_bw)

    print(f'There are {len(faces)} faces found.')

    for face in faces:
        x, y, w, h = face
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
        cv2.imwrite(f'mask/mask_{count}.jpg', image_org[y:y+h, x:x+w])
        count += 1

    cv2.imshow("Faces found", image_bgr)
    cv2.waitKey(1)
```

```

In [17]:
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
# เปลี่ยนการแสดงผลเป็นทศนิยม
np.set_printoptions(suppress=True)

# Load the model
# นำโมเดลที่ Train แล้วมาใช้งาน
model = tensorflow.keras.models.load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
# สร้างข้อมูลที่ใส่รับในขนาดที่ Model ใ้
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
# สร้างข้อมูลที่ใส่รับในขนาดที่ Model ใ้
image = Image.open('test_photo.jpg')

#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the
center
# ปรับขนาดข้อมูลให้เป็นขนาดที่ต้องการ
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)

#turn the image into a numpy array
# เปลี่ยนข้อมูลให้เป็น array
image_array = np.asarray(image)

# display the resized image
# แสดงผลรูป
image.show()

# Normalize the image
# ปรับข้อมูลสำหรับ Model
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# Load the image into the array
# นำข้อมูลใส่ data ในขนาดที่กำหนดเอาไว้
data[0] = normalized_image_array

# run the inference
# นำข้อมูลมาวิเคราะห์ด้วย model
prediction = model.predict(data)
print(prediction)

```

```
WARNING:tensorflow:No training configuration found in the save file, so
the model was *not* compiled. Compile it manually.
[[0.999985  0.00001496]]
```

```
In [19]:
webcam = cv2.VideoCapture(1)
success, image_bgr = webcam.read()
```

```
In [20]:
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
```

```
In [21]:
cface_rgb = Image.fromarray(image_rgb[y:y+h, x:x+w])
```

```
In [ ]:
face_classifier = cv2.CascadeClassifier(face_cascade)
np.set_printoptions(suppress=True)
model = tensorflow.keras.models.load_model('keras_model.h5')
size = (224, 224)
```

```
In [24]:
while True:
    success, image_bgr = webcam.read()
    image_org = image_bgr.copy()
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
    image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
    faces = face_classifier.detectMultiScale(image_bw)
    for face in faces:
        x, y, w, h = face
        cface_rgb = Image.fromarray(image_rgb[y:y+h, x:x+w])
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        image = cface_rgb
        image = ImageOps.fit(image, size, Image.ANTIALIAS)
        image_array = np.asarray(image)
        normalized_image_array = (image_array.astype(np.float32) / 127.0) -
1
        data[0] = normalized_image_array
        prediction = model.predict(data)
        print(prediction)
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.imshow("Mask Detection", image_bgr)
    cv2.waitKey(1)
```

```
[[0.00000073 0.9999993 ]]
[[0.00000123 0.9999988 ]]
[[0.00000068 0.9999993 ]]
[[0.00000085 0.99999917]]
[[0.00000139 0.99999857]]
[[0.00000176 0.9999982 ]]
[[0.00000074 0.9999993 ]]
[[0.00000149 0.99999857]]
[[0.00000064 0.9999994 ]]
[[0.0000011 0.9999989]]
[[0.00000071 0.9999993 ]]
[[0.00000046 0.9999995 ]]
```

```
In [28]:
while True:
    success, image_bgr = webcam.read()
    image_org = image_bgr.copy()
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
    image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
    faces = face_classifier.detectMultiScale(image_bw)
    for face in faces:
        x, y, w, h = face
        cface_rgb = Image.fromarray(image_rgb[y:y+h, x:x+w])
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        image = cface_rgb
        image = ImageOps.fit(image, size, Image.ANTIALIAS)
        image_array = np.asarray(image)
        normalized_image_array = (image_array.astype(np.float32) / 127.0) -
1
        data[0] = normalized_image_array
        prediction = model.predict(data)
        print(prediction)
        if prediction[0][0] > prediction[0][1]:
            cv2.putText(image_bgr, 'Masked', (x, y-
7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
            cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
        else:
            cv2.putText(image_bgr, 'Non-Masked', (x, y-
7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
            cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 0, 255), 2)

cv2.imshow("Mask Detection", image_bgr)
cv2.waitKey(1)
```

```

import cv2
import numpy as np
import tensorflow
import winsound
import time

from PIL import Image, ImageOps

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
np.set_printoptions(suppress=True)
model = tensorflow.keras.models.load_model('keras_model.h5')
size = (224, 224)

non_masked_alert_sound_path = "non_masked.wav"
|
while True:
    success , Image_bgr = cap.read()
    Image_org = Image_bgr.copy()
    image_bw = cv2.cvtColor(Image_bgr, cv2.COLOR_BGR2GRAY)
    Image_rgb = cv2.cvtColor(Image_bgr, cv2.COLOR_BGR2RGB)
    faces = face_cascade.detectMultiScale(image_bw, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
    if len(faces) > 0:

```

```

    for (x, y, w, h) in faces:
        cface_rgb = Image.fromarray(Image_rgb[y:y+h, x:x+w, :])
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        image = cface_rgb
        image = ImageOps.fit(image, size, method=Image.LANCZOS)
        Image_array = np.asarray(image)
        normalized_image_array = (Image_array.astype(np.float32) / 127.5) - 1
        data[0] = normalized_image_array
        prediction = model.predict(data)
        print(prediction)

        if prediction[0][0] > prediction[0][1]:
            cv2.putText(Image_bgr, 'Masked', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
            cv2.rectangle(Image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
            winsound.PlaySound(None, winsound.SND_ASYNC | winsound.SND_FILENAME | winsound.SND_LOOP)

        else:
            cv2.putText(Image_bgr, 'Non-Masked', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
            cv2.rectangle(Image_bgr, (x, y), (x+w, y+h), (0, 0, 255), 2)
            winsound.PlaySound(non_masked_alert_sound_path, winsound.SND_ASYNC | winsound.SND_ALIAS | winsound.SND_LOOP)
            time.sleep(1.7)

```

```

        else:
            cv2.putText(Image_bgr, 'No face detected', (20, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
            winsound.PlaySound(non_masked_alert_sound_path, winsound.SND_ASYNC | winsound.SND_ALIAS | winsound.SND_LOOP)

cv2.imshow("Mask Detection", Image_bgr)

# Exit the loop if the 'q' key is pressed
if cv2.waitKey(1) == ord('q'):
    break

# Release the webcam and close the window
cap.release()
cv2.destroyAllWindows()

```

```
non_masked_alert_sound_path = "non_masked.wav"
```

```

else:
    cv2.putText(Image_bgr, 'Non-Masked', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
    cv2.rectangle(Image_bgr, (x, y), (x+w, y+h), (0, 0, 255), 2)
    winsound.PlaySound(non_masked_alert_sound_path, winsound.SND_ASYNC | winsound.SND_ALIAS | winsound.SND_LOOP)
    time.sleep(1.7)

```

ภาคผนวก ข

คู่มือการติดตั้ง

1. ติดตั้งเครื่องมือที่ต้องใช้

```
In [1]:  
!pip install opencv-python
```

```
In [3]:  
!pip install tensorflow
```

```
In [4]:  
!pip install keras
```

```
In [5]:  
!pip install Pillow
```

รูปที่ ข-1 ภาพโปรแกรมที่ต้องใช้

การนำเข้าเครื่องมือ

```
1  ✓ import cv2  
2  import numpy as np  
3  import tensorflow  
4  import winsound  
5  import time  
6
```

รูปที่ ข-2 ภาพการนำเข้าเครื่องมือ

2. Face Detection

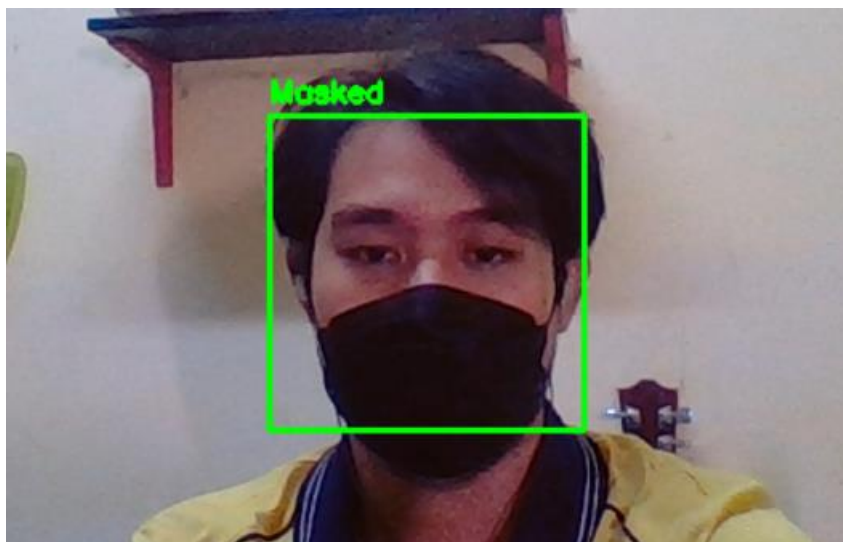
เราจะใช้ Face Detection ที่เคยสร้างไว้ เป็นพื้นฐาน แล้วจะสร้างส่วนของการตรวจจับ Mask เข้าไป ซึ่งโดยพื้นฐานเราจะใช้ haar cascade ซึ่งเป็นโปรแกรมที่ใช้ตรวจจับ object ซึ่งถูกปรับเพื่อให้ตรวจจับใบหน้าที่เตรียมไว้ให้ในไฟล์ "haarcascade_frontalface_default.xml" เพื่อตรวจจับตำแหน่งของใบหน้าในแต่ละรูป หรือแต่ละเฟรมของวิดีโอ โดยโปรแกรมตอนนี้จะรับ test_photo.jpg เป็น input



รูปที่ ข-3 ภาพการตรวจจับใบหน้า

โดยโปรแกรมนี้เราจะได้ผลลัพธ์ออกมาเป็นตำแหน่งของ ใบหน้าในพิกัด x, y ที่มีจุด $0,0$ ที่ด้านซ้ายบน และมีค่า x, y สูงที่สุดที่จุดขวาล่างของภาพ, ความกว้างของภาพ w , ความสูงของภาพ h ที่เราใช้ติกรอบใบหน้า ก่อนที่จะนับและแสดงผลว่าในภาพมีกี่หน้า

ผลลัพธ์การตีกรอบในรูปบนใบหน้าใน pop-up



รูปที่ ข-4 ภาพผลลัพธ์การตีกรอบในรูปบนใบหน้า

สิ่งที่เราจะปรับในโปรแกรมเดิมเราใช้การรับรูปที่มีอยู่แล้ว และตรวจจับใบหน้า เราจะใช้ข้อมูลวิดีโอจาก webcam และสร้าง Model จาก Teachable Machine ที่ใช้ตรวจจับ Mask เพื่อประกอบเข้าไป

เปลี่ยนการรับภาพเป็นรับ webcam

เลือก webcam ด้วยคำสั่ง `cv2.VideoCapture()` และใส่ argument เป็นตำแหน่งของ webcam ที่ต้องการ webcam ที่ติดมากับเครื่องจะอยู่ในตำแหน่งแรก หรือตำแหน่งที่ 0 และใส่เรียงตามลำดับเช่น 1 2 ไปตามลำดับที่ต่อกับคอมพิวเตอร์

```
In [8]:
webcam = cv2.VideoCapture(1)
webcam
```

```
Out[8]:
<VideoCapture 00000273F532C470>
```

รูปที่ ข-5 ภาพของโค้ดการรับภาพเป็นรับ webcam

3. ข้อมูลจาก webcam

ด้วยคำสั่ง `.read()` ที่ทำงานกับ object Video Capture ข้อมูลของกล้อง เราจะได้ Tuple ของ Boolean ที่แสดงสถานะของการเชื่อมต่อ True คือเชื่อมต่อสำเร็จ False คือเชื่อมต่อไม่สำเร็จ และ ค่าสีของ pixel ที่ได้รับมาเป็นภาพ โดยภาพจะอยู่ในระบบสี BGR เวลาทำงานกับภาพต้องคอยตอบตัวเอง ถ้าม หรือดูคู่มือของเครื่องมือดีๆ ว่าตอนนี้คำสั่งเราเข้าใจรูปในรูปแบบสีใดๆ RGB, RGBA, BGR, HSL หรืออื่นๆ เพราะมีการส่งต่อข้อมูลไม่เหมือนกัน และอาจต้องใช้คำสั่งเพื่อแปลงระบบสีก่อนส่งข้อมูลข้ามคำสั่ง

```
In [9]:
success, image_bgr = webcam.read()
```

```
Out[9]:
(True, array([[[140, 117, 67],
[141, 117, 66],
[141, 116, 64],
...,
[ 37, 29, 64],
[ 17, 17, 56],
[ 17, 17, 56]],
...,
[[ 66, 59, 9],
[ 67, 60, 10],
[ 68, 62, 10],
...,
[149, 89, 105],
[146, 86, 100],
[147, 87, 101]]], dtype=uint8))
```

รูปที่ ข-6 ภาพโค้ดการรับข้อมูลรูปภาพ

ข้อมูลวิดีโอในการใช้การส่งข้อมูลภาพซ้ำๆ ติดต่อกันจนต่อกันเป็นวิดีโอ ซึ่งจะใช้ while loop เพื่อส่งคำสั่งขอภาพจาก webcam เพื่อแสดงตัววิดีโอ ในการทำ loop ใดๆ อย่าลืมนำคำสั่งที่ไม่ต้องการให้ทำซ้ำออกจาก loop ก่อน เช่น ในคำสั่ง face_cascade = "haarcascade_frontalface_default.xml" ที่ต้องทำแค่ครั้งเดียว

```
In [11]:
face_cascade = "haarcascade_frontalface_default.xml"

In [12]:
while True:

    success, image_bgr = webcam.read()

    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

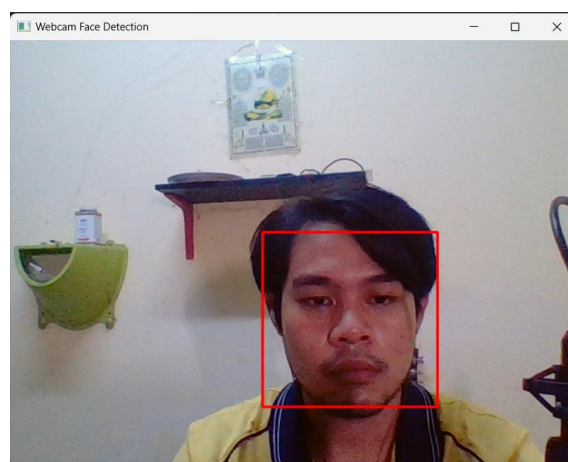
    face_classifier = cv2.CascadeClassifier(face_cascade)
    faces = face_classifier.detectMultiScale(image_bw)

    print(f'There are {len(faces)} faces found.')

    for face in faces:
        x, y, w, h = face
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow("Faces found", image_bgr)
    cv2.waitKey(1)
```

รูปที่ ข-7 ภาพจาก webcam เพื่อแสดงตัววิดีโอ ในการทำ loop



รูปที่ ข-8 ภาพผลลัพธ์การตีกรอบหน้าบน webcam

4. เก็บภาพของหน้าเพื่อใช้เทรนโมเดล

เราจะถ่ายรูปจาก webcam ของเราในตอนที่ไม่ใส่แมส และไม่ใส่แมสออกมาเรื่อยๆ ไปเทรนโมเดลด้วยการเขียน Python

การตรวจจับใบหน้าจะได้ผลลัพธ์เป็น พิกัดจุด x, y ที่ด้านซ้ายบนของภาพ w ความกว้างของภาพ และ h ความสูงของภาพ เราสามารถนำข้อมูลดังกล่าวเพื่อใช้ในการ crop รูป เพื่อนำข้อมูลใบหน้าที่ต้องการมา การบันทึกรูปจะใช้คำสั่ง `cv2.imwrite()` โดยรับค่า ชื่อไฟล์ และ path ที่ต้องการบันทึกรูปที่ต้องการ และรูปที่ต้องการโดยใช้การดึงเฉพาะข้อมูลในส่วนที่ต้องการโดยการระบุ `[y:y+h,x:x+w]` โดยจะได้ผลลัพธ์เป็นภาพที่ครอบตัดที่มุมซ้ายบน x, y และมีความกว้าง w ความสูง y ซึ่งค่า x, y, w, h เราได้มาจากการตรวจจับหน้านั้นเอง จึงได้ภาพใบหน้าออกมา

เช่นการ crop รูปที่มีจุดซ้ายบน 0,0 กว้าง ยาว 100 เก็บข้อมูลภาพอย่างต่อเนื่อง จะต้องมีการเปลี่ยนชื่อให้ไม่เกิดการบันทึกไฟล์ซ้ำกัน ดังนั้นเราจะสร้างตัวแปรที่เปลี่ยนไปแต่ละรอบการบันทึก และนำไปเป็นส่วนหนึ่งของชื่อไฟล์ด้วย f-string

```
In [16]:
count = 9999
while True:

    success, image_bgr = webcam.read()
    image_org = image_bgr.copy()
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

    face_classifier = cv2.CascadeClassifier(face_cascade)
    faces = face_classifier.detectMultiScale(image_bw)

    print(f'There are {len(faces)} faces found.')

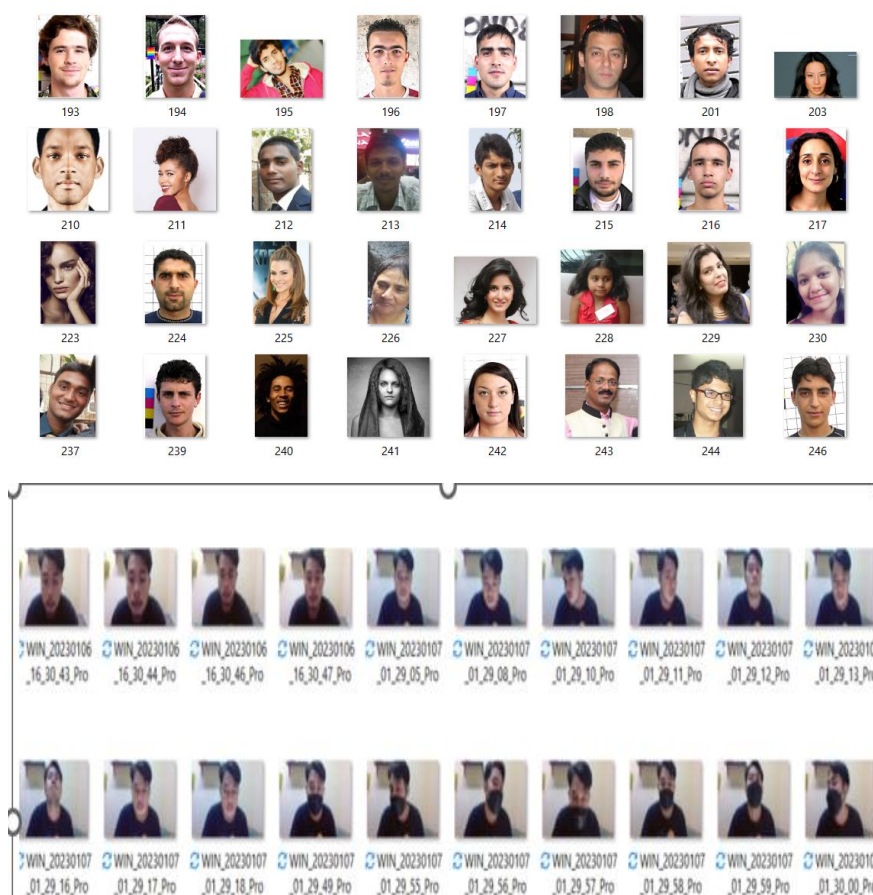
    for face in faces:
        x, y, w, h = face
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
        cv2.imwrite(f'mask/mask_{count}.jpg', image_org[y:y+h,x:x+w])
        count += 1

    cv2.imshow("Faces found", image_bgr)
    cv2.waitKey(1)
```

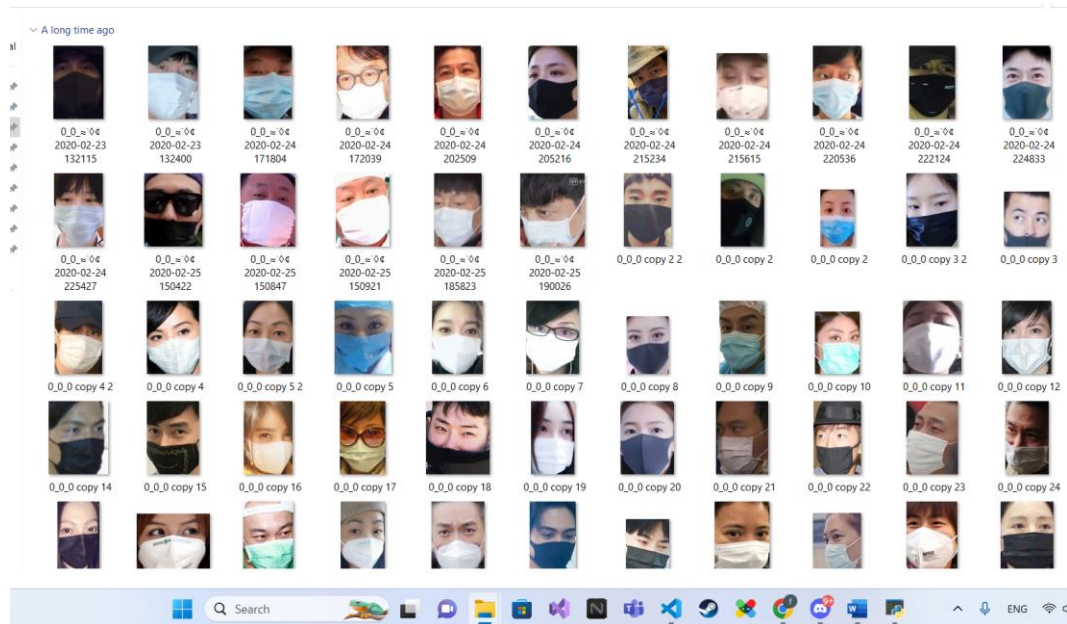
รูปที่ ข-9 ภาพโค้ดการตรวจจับใบหน้า

ผลลัพธ์การบันทึกภาพลงบนโฟลเดอร์ที่ระบุจะได้รับการบันทึกภาพแต่ละเฟรมมาแบบเร็วๆ เสร็จแล้วอย่าลืมกด STOP สัญลักษณ์สี่เหลี่ยมจัตุรัสแถบเครื่องมือด้านบน ทำซ้ำสำหรับรูปใส่แมส และไม่ใส่แมส อย่าลืมเปลี่ยนชื่อโฟลเดอร์ / รูป ไม่งั้นอาจจะได้ข้อมูลไม่ครบ

การบันทึกใบหน้าไม่สวมแมส



รูปที่ ข-10 ภาพใบหน้าที่ไม่สวมแมส



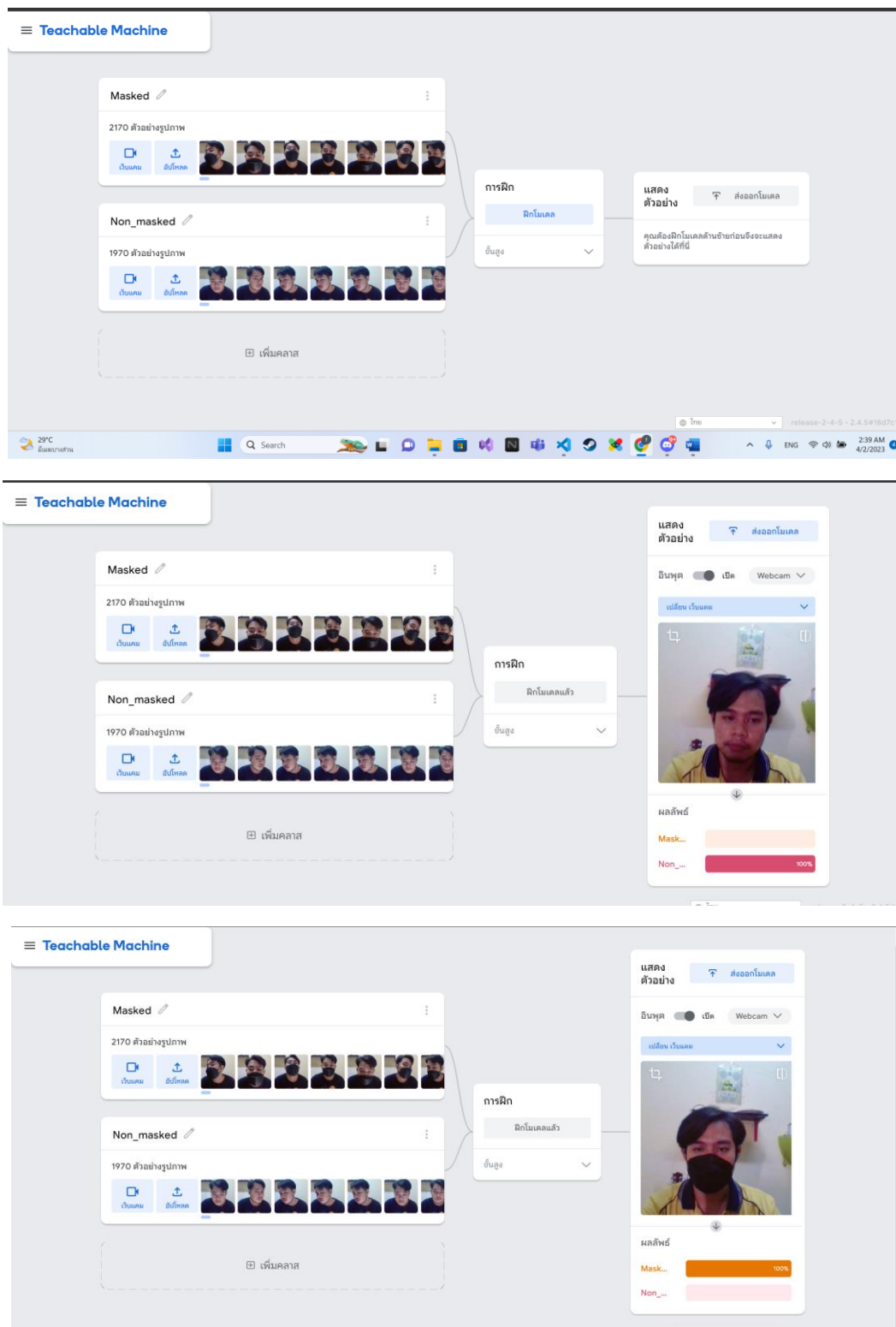
รูปที่ ข-11 ภาพใบหน้าที่สวมแมส

สร้าง Classification Model ผ่าน Teachable Machine ที่จะรับภาพเพื่อนให้ Model ระบุภาพที่ต้องการได้ เพื่อแบ่งประเภทภาพที่เห็นออกเป็นประเภทต่างๆ เช่น คนใส่แมส และคนไม่ใส่แมส

ซึ่งจะแบ่งข้อมูลเป็นกี่ประเภทก็ได้ ซึ่งจะเรียกว่า Class เราจะนำภาพของแต่ละ Class เข้าไปให้ Machine Learning เรียนรู้ และดาวน์โหลดโปรแกรมที่ Tune เสร็จแล้วมาใช้งานเป็นส่วนหนึ่งของโค้ดของเรา

5. หน้าตา Teachable Machine

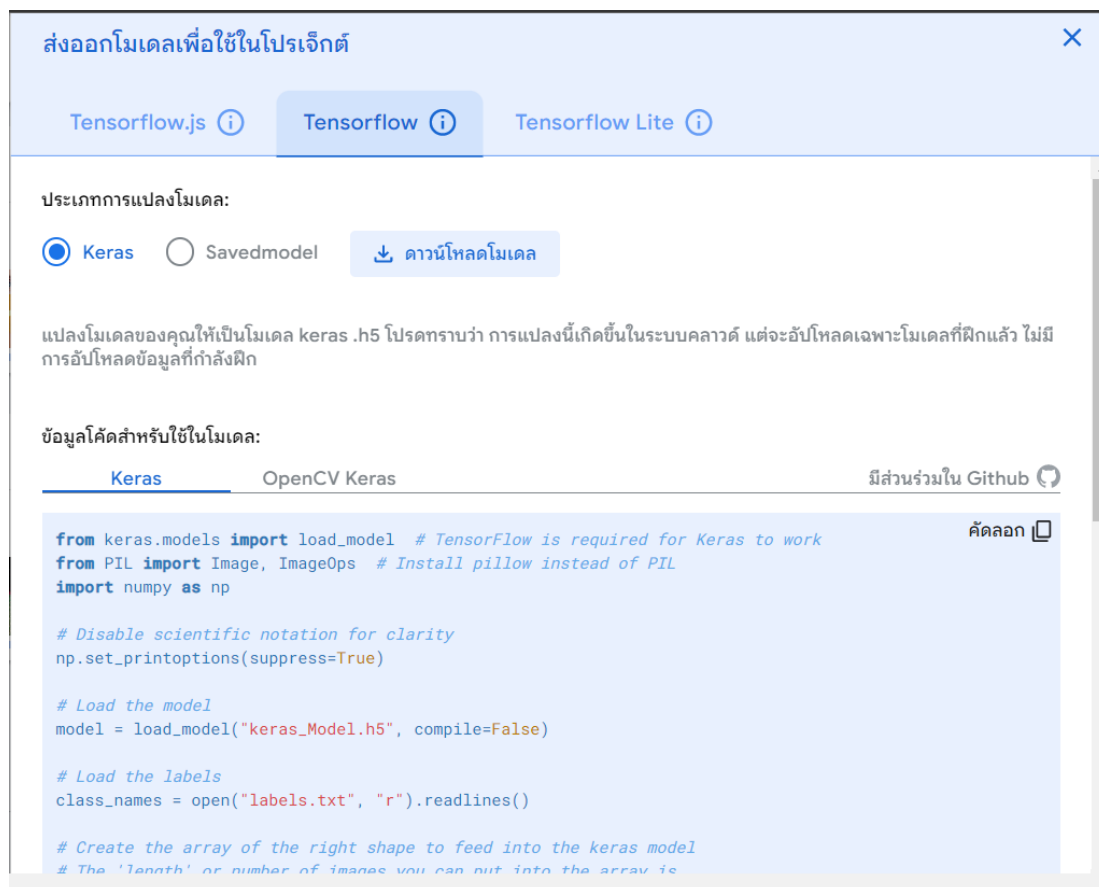
ให้เราอัปโหลดตัวอย่างในแต่ละ Class หรือกลุ่มที่ต้องการแยกได้เลย เมื่อเทรนเสร็จใช้ webcam ทดลองรันได้ real-time ทันที



รูปที่ ข-12 ภาพการบันทึกภาพแบบ real-time

Export Model

เมื่อเสร็จจก Export Model ด้านบนภาพของ webcam จะเป็นการดาวน์โหลดไฟล์ keras_model.h5 ไฟล์ประเภท h5 ที่เก็บข้อมูลโมเดลของเราเอาไว้ พร้อมกับโค้ดที่เตรียมให้เพื่อเรียกใช้ model keras_model.h5 ที่ดาวน์โหลดมา



รูปที่ ข-13 ภาพการส่งออกไฟล์

ส่วนนี้เป็นโค้ดที่ใช้เชื่อมโมเดลที่เทรนเสร็จแล้ว ที่ดาวน์โหลดมาจาก Teachable Machine จะใช้ได้ต่อเมื่อเรามีการดาวน์โหลดโมเดลมาจาก Teachable Machine หลังเทรนเสร็จแล้ว

```
In [17]:
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
# เปลี่ยนการแสดงผลเป็นทศนิยม
np.set_printoptions(suppress=True)

# Load the model
# นำโมเดลที่ Train แล้วมาใช้งาน
model = tensorflow.keras.models.load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
# สร้างข้อมูลที่ใช้รับในขนาดที่ Model ใช้
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
# สร้างข้อมูลที่ใช้รับในขนาดที่ Model ใช้
image = Image.open('test_photo.jpg')

#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the
center
# ปรับขนาดข้อมูลให้เป็นขนาดที่ต้องการ
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)

#turn the image into a numpy array
# เปลี่ยนข้อมูลให้เป็น array
image_array = np.asarray(image)

# display the resized image
# แสดงผลรูป
image.show()

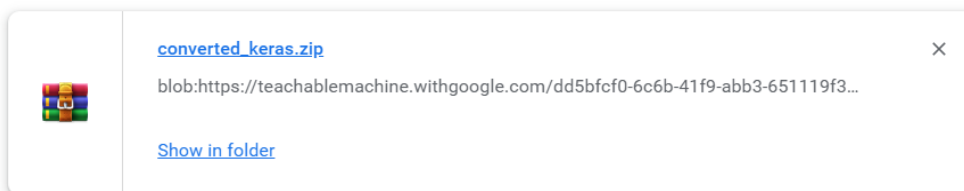
# Normalize the image
# ปรับข้อมูลสำหรับ Model
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# Load the image into the array
# นำข้อมูลใส่ data ในขนาดที่กำหนดเอาไว้
data[0] = normalized_image_array

# run the inference
# นำข้อมูลมาวิเคราะห์ด้วย model
prediction = model.predict(data)
print(prediction)
```

รูปที่ ข-14 ภาพการดาวน์โหลดโมเดล

Today



	converted_keras	1/7/2023 2:00 AM	WinRAR ZIP archive	2,397 KB
	face Detection	1/7/2023 12:21 AM	Python Source File	1 KB
	face-detect-model	1/6/2023 7:45 PM	XML Source File	941 KB
	image	1/7/2023 12:32 AM	PNG File	1,847 KB
	import webcam	1/7/2023 2:07 AM	Python Source File	3 KB
	keras_model.h5	1/6/2023 7:00 PM	H5 File	2,396 KB
	labels	1/6/2023 7:00 PM	Text Document	1 KB
	Untitled-1222222	1/6/2023 11:12 PM	Jupyter Source File	1 KB

รูปที่ ข-15 ภาพการดาวน์โหลดและแตกไฟล์

ผลลัพธ์ของโมเดล

จะมี WARNING เกี่ยวกับการ compiled ซึ่งเราไม่ต้องสนใจเพราะเราไม่ต้องทำ เราทำมาจาก teachable machine แล้ว แต่ที่น่าสนใจ คือตัวเลขใน [[]] บรรทัดสุดท้าย ตัวนี้เป็นผลลัพธ์ที่ print มาจาก prediction ที่บอกถึง ความมั่นใจของ Model ว่าภาพที่เราใส่เข้าไปเป็นข้อมูล Class ไหน ซึ่งอ้างอิงตามตำแหน่ง ข้อมูล เลขด้านซ้าย บอกว่ามีความมั่นใจ 99.99985% ที่ข้อมูลจะเป็น Class แรก หรือคนใส่แมสก์นั่นเอง และ มั่นใจ 0.001496% ว่าข้อมูลจะเป็นรูปคนไม่ใส่แมสก์ หรือ Class ที่ 2

```
WARNING:tensorflow:No training configuration found in the save file, so
the model was *not* compiled. Compile it manually.
[[0.999985 0.0001496]]
```

รูปที่ ข-16 ภาพผลลัพธ์ของโมเดล

6. นำ Model มาใส่ Webcam

ปัจจุบันข้อมูลจาก webcam อยู่ในรูปแบบของ object array ซึ่งจะต้องแปลงเป็น Image object ในรูปแบบของ object ที่ได้จากการเปิดไฟล์จากรูป และต้องมีการแปลงสีจาก BGR เป็น RGB

```
In [19]:
webcam = cv2.VideoCapture(1)
success, image_bgr = webcam.read()
```

รูปที่ ข-17 ภาพโค้ดการนำเข้าภาพจาก webcam

```
In [20]:
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
```

รูปที่ ข-18 ภาพโค้ดการแปลงจาก BGR เป็น RGB

แปลง array เป็น ภาพ

และ crop ข้อมูลให้ได้เฉพาะหน้าโดยใช้ตำแหน่ง x, y ความกว้าง w, ความสูง h ที่ได้จากโปรแกรมตรวจจับใบหน้า มาตัดรูปให้ได้เฉพาะใบหน้าที่จับเจอ ก่อนที่จะนำไปตรวจว่าเป็นคนใส่ Mask หรือไม่

```
In [21]:
cface_rgb = Image.fromarray(image_rgb[y:y+h,x:x+w])
```

รูปที่ ข-19 ภาพการแปลง array เป็น ภาพ

7. Mask Detection บน webcam

หลังจากที่เราแปลงข้อมูลของ webcam และปรับให้เข้ากับข้อมูลที่ใช้ร่วมกับ Model ที่ Train มาแล้ว เราจะรวมคำสั่งเข้าด้วยกัน และแสดงผลเป็น real-time

แยกข้อมูลที่ไม่ต้องการทำซ้ำออกนอก loop

```
In [ ]:
face_classifier = cv2.CascadeClassifier(face_cascade)
np.set_printoptions(suppress=True)
model = tensorflow.keras.models.load_model('keras_model.h5')
size = (224, 224)
```

รูปที่ ข-20 ภาพโค้ดการแยกข้อมูล

ส่วนการทำงานตรวจจับ Mask real-time บน webcam

```
In [24]:
while True:
    success, image_bgr = webcam.read()
    image_org = image_bgr.copy()
    image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
    image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
    faces = face_classifier.detectMultiScale(image_bw)
    for face in faces:
        x, y, w, h = face
        cface_rgb = Image.fromarray(image_rgb[y:y+h, x:x+w])
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        image = cface_rgb
        image = ImageOps.fit(image, size, Image.ANTIALIAS)
        image_array = np.asarray(image)
        normalized_image_array = (image_array.astype(np.float32) / 127.0) -
1
        data[0] = normalized_image_array
        prediction = model.predict(data)
        print(prediction)
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.imshow("Mask Detection", image_bgr)
    cv2.waitKey(1)
```

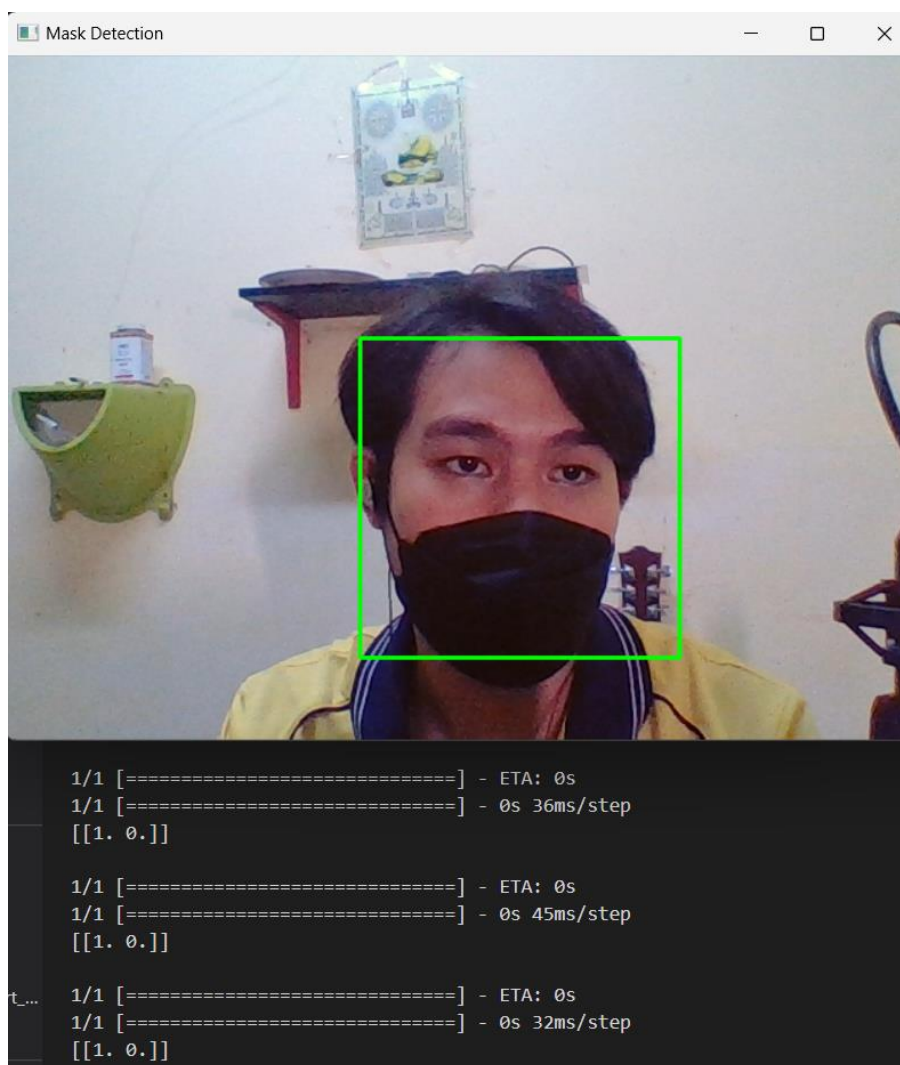
รูปที่ ข-21 ภาพโค้ดการตรวจจับแมส

ผลลัพธ์ที่ได้จากการ print(prediction) เป็นการความมั่นใจในการคัดแยกภาพไปยังแต่ละ Class อ้างอิงตามลำดับที่เราเทรนบน Teachable Machine

```
[[0.00000073 0.9999993 ]]
[[0.00000123 0.9999988 ]]
[[0.00000068 0.9999993 ]]
[[0.00000085 0.99999917]]
[[0.00000139 0.99999857]]
[[0.00000176 0.9999982 ]]
[[0.00000074 0.9999993 ]]
[[0.00000149 0.99999857]]
[[0.00000064 0.9999994 ]]
[[0.0000011 0.9999989]]
[[0.00000071 0.9999993 ]]
[[0.00000046 0.9999995 ]]
```

รูปที่ ข-22 ภาพโค้ดการเก็บข้อมูลภาพ

7. ผลการเก็บภาพแบบ Real-time บน Webcam



รูปที่ ข-23 ภาพการเก็บข้อมูลภาพ

ใส่เงื่อนไขการแสดงผลเราจะใส่ if statement เพื่อรับการแสดงผลให้ขึ้นกับเงื่อนไข หากตรวจเจอ mask จะให้แสดงกรอบสีเขียว พร้อมข้อความ Masked หากไม่เจอให้แสดงกรอบสีแดง พร้อมข้อความ Non-Masked ซึ่งจะใช้การวัดค่าของการ prediction มาเทียบกันหาก model เชื่อว่าใส่ mask มากกว่าจะถือว่าใส่แมส

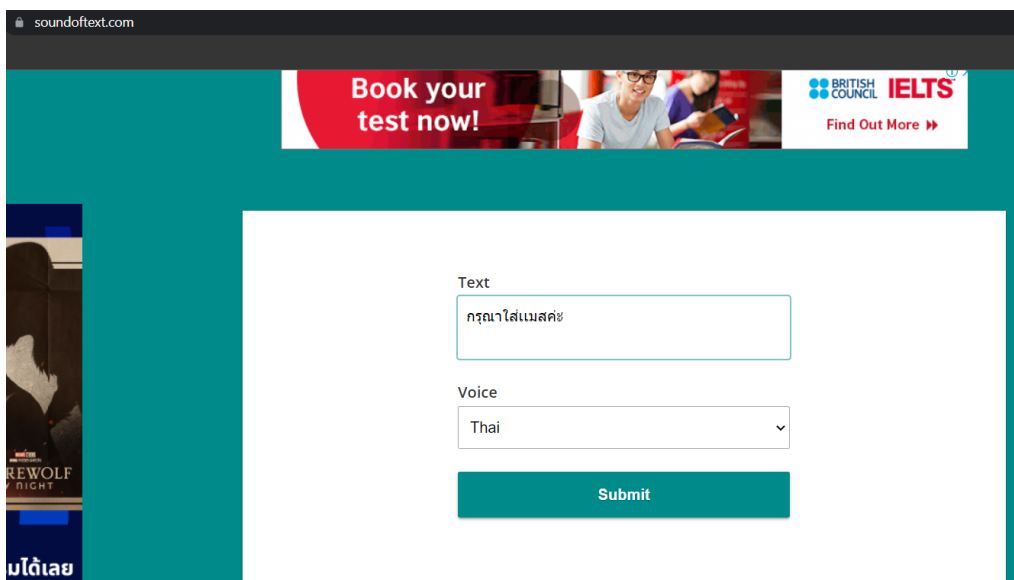
ใส่ text ใช้คำสั่ง cv2.putText() รับค่า ภาพที่ต้องการใส่ text ลงไป , ข้อความ , x,y ตำแหน่งซ้ายบน, font, ขนาดตัวหนังสือ, tuple ค่าสี และ ความหนาตัวอักษร

ภาคผนวก ค

ออกแบบจำลองระบบเสียง

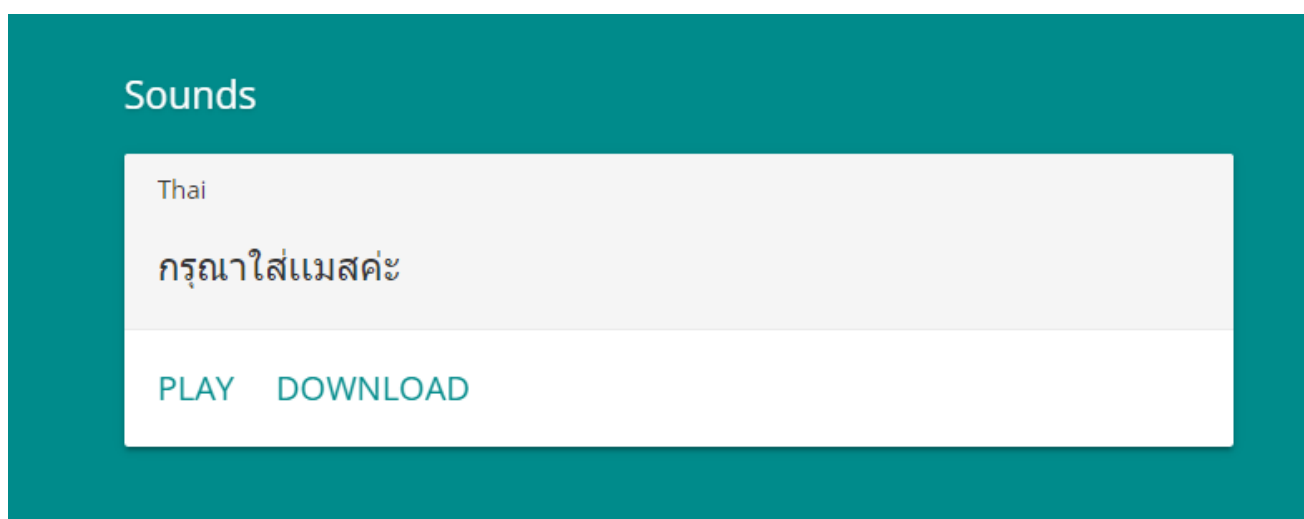
1. การใส่เสียงแจ้งเตือน

เข้าเว็บ sound of text เพื่อทำการพิมพ์ข้อความที่ต้องการให้แปลงจากข้อความให้เป็นเสียง



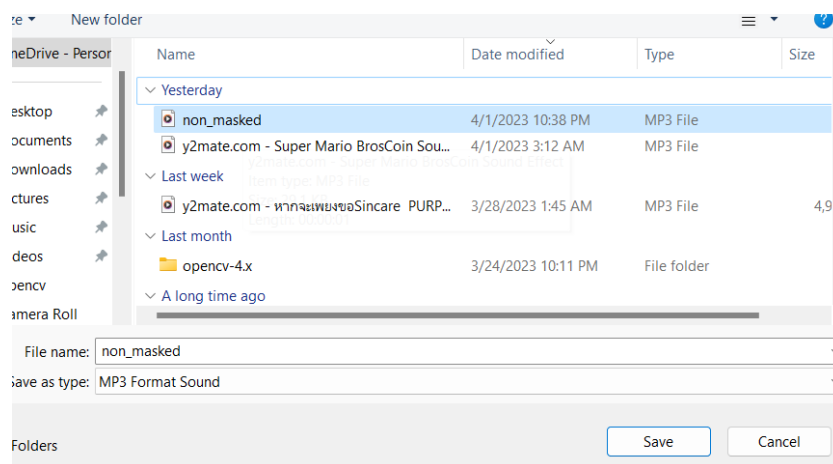
รูปที่ ค-1 ภาพการแปลงข้อความเป็นเสียง

เมื่อทำการ Submit แล้วให้เราลองฟังเสียงที่เราต้องการแล้วกดดาวโหลด



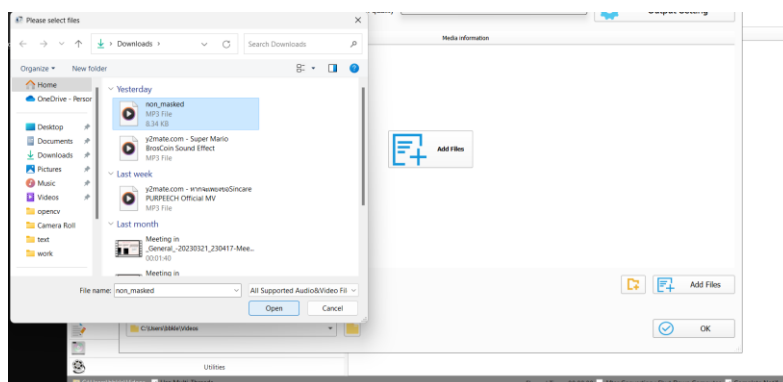
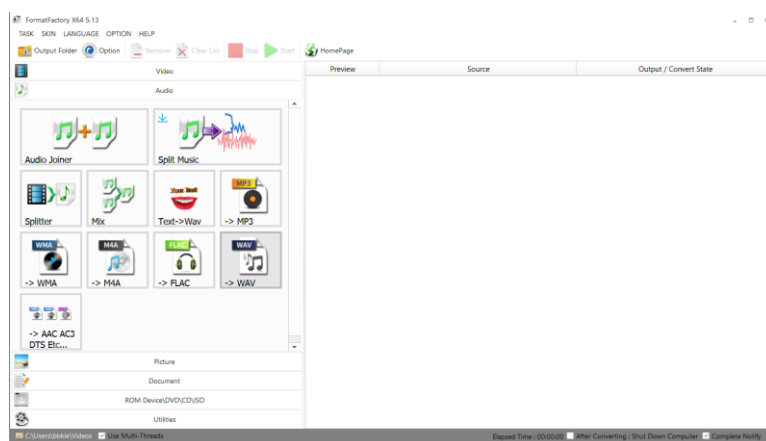
รูปที่ ค-2 ภาพการดาวโหลดไฟล์เสียง

เข้าไปตรวจสอบไฟล์ที่เราได้ดาวน์โหลดมา



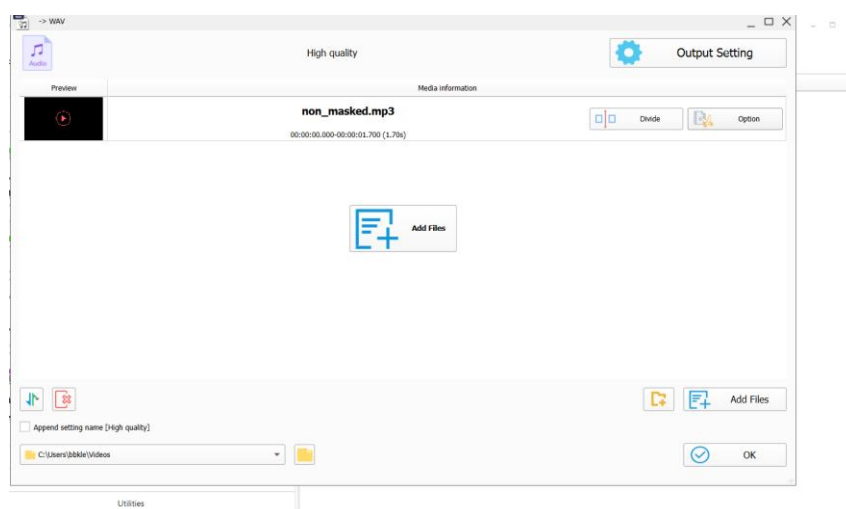
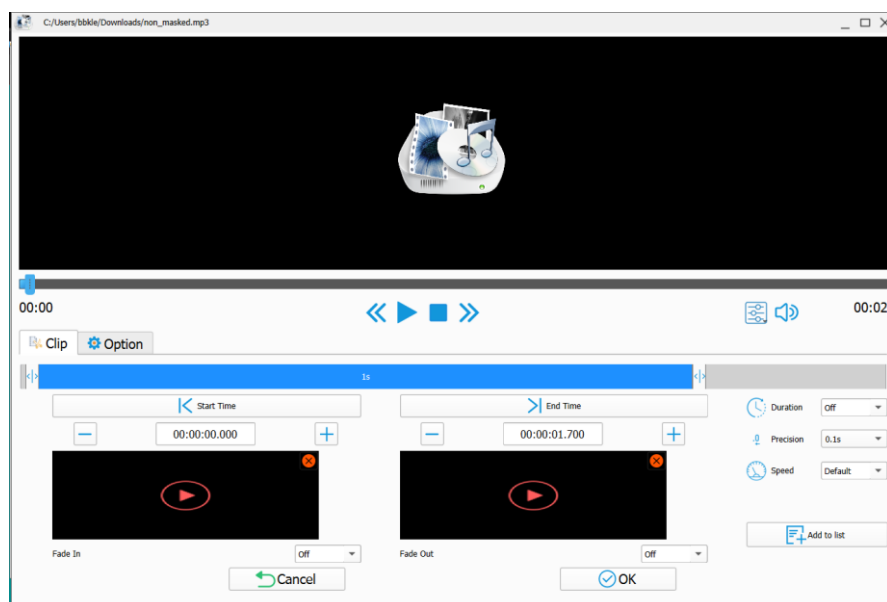
รูปที่ ค-3 ภาพการตรวจสอบไฟล์เสียง

2.ให้เราเข้าโปรแกรม Format Factory แล้วแปลงไฟล์จาก Mp3 เป็น .Wav



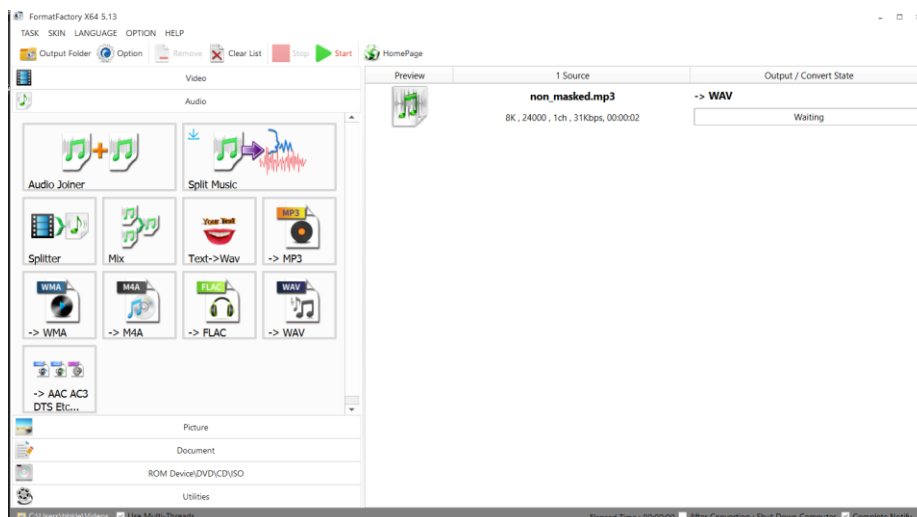
รูปที่ ค-4 ภาพการแปลงไฟล์เสียง

ตรวจสอบเสียงแล้วตัดต่อตามที่เรต้องการ



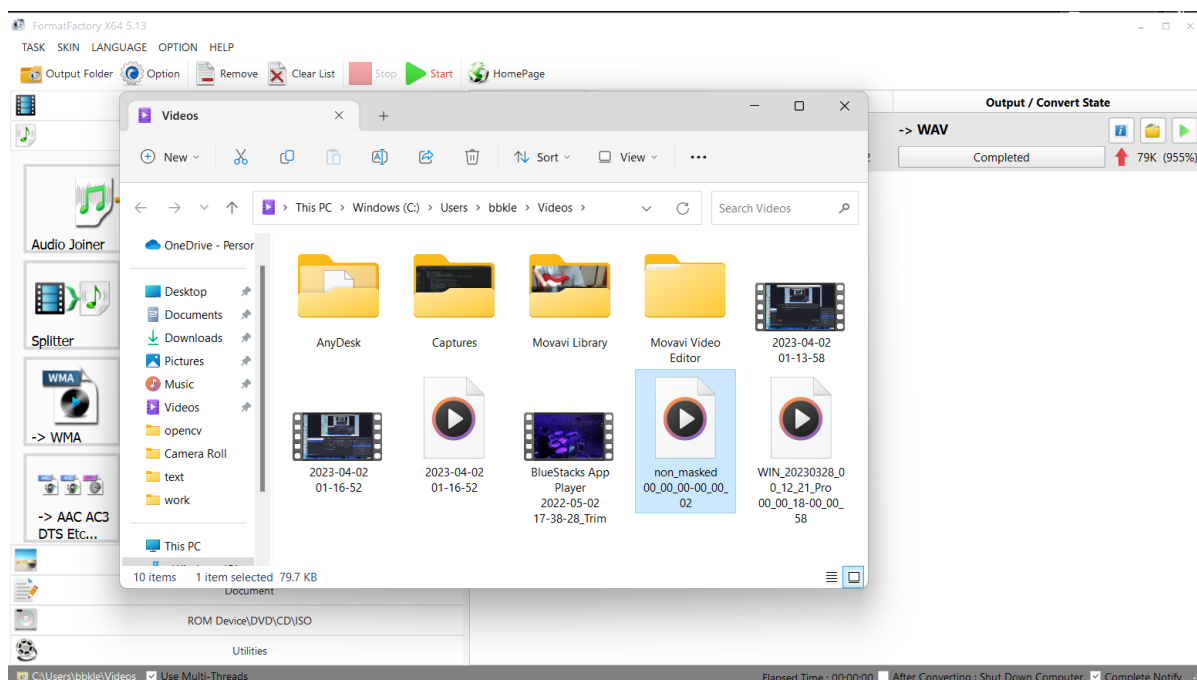
รูปที่ ค-5 ภาพการตรวจสอบและตัดต่อไฟล์เสียง

เมื่อกด OK แล้วให้ไปที่กดปุ่ม Start เพื่อทำการแปลงไฟล์



รูปที่ ค-6 ภาพการแปลงไฟล์

ตรวจสอบไฟล์ที่เราต้องการแล้วแก้ไขชื่อตามที่เราต้องการ



รูปที่ ค-7 ภาพการแปลงไฟล์เสร็จสมบูรณ์

3. ในส่วนของ Code ให้ไปแกะตัวแปรที่เล่นเสียงการแจ้งเตือน

```
non_masked_alert_sound_path = "non_masked.wav"
```

```
else:  
    cv2.putText(Image_bgr, 'Non-Masked', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)  
    cv2.rectangle(Image_bgr, (x, y), (x+w, y+h), (0, 0, 255), 2)  
    winsound.PlaySound(non_masked_alert_sound_path, winsound.SND_ASYNC | winsound.SND_ALIAS | winsound.SND_LOOP)  
    time.sleep(1.7)
```

รูปที่ ค-8 ภาพโค้ดการนำไปใช้จริง

ภาคผนวก ง

ประวัติผู้จัดทำ

ประวัติผู้จัดทำ



หัวข้อโครงการวิจัย : ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

ชื่อ – นามสกุล นาย ธนพัฒน์ วงคะสุ่ม

รหัสนักศึกษา 116210906093-6

อาจารย์ที่ปรึกษา ดร.อนุชา ตุงค์ขันธ์ฐาน

สาขาวิชา/ภาควิชา สาขาวิชาเทคโนโลยีสารสนเทศ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ปีการศึกษา 2565

ประวัติผู้จัดทำ



หัวข้อโครงงานวิจัย : ระบบตรวจจับการใส่แมสและส่งสัญญาณแจ้งเตือน

The system detects wearing a mask and sends an alert signal

ชื่อ – นามสกุล นาย วิระพล แสงภา

รหัสนักศึกษา 116210906060-5

อาจารย์ที่ปรึกษา ดร.อนุชา ตุงค์ช้วน

สาขาวิชา/ภาควิชา สาขาวิชาเทคโนโลยีสารสนเทศ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ปีการศึกษา 2565