การจำแนกรถพยาบาลฉุกเฉินโดยใช้เทคโนโลยีปัญญาประดิษฐ์เพื่อแจ้งเตือนผู้ขับขี่ยานพาหนะส่วนบุคคล

นายถิรมนัส สาระภิรมย์

มหาวิทยาลัยเทคโนโลยีสุรนารี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2563

DRIVER ALERT: CLASSIFICATION AND NOTIFICATION FOR AMBULANCE ARRIVAL USING ARTIFICIAL INTELLIGENCE TECHNOLOGY

Tiramanat Sarapirom

มหาวิทยาลัยเทคโนโลยีสุรนารี

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Engineering in Telecommunication and Computer Engineering

Suranaree University of Technology

Academic Year 2020

การจำแนกรถพยาบาลฉุกเฉินโดยใช้เทคโนโลยีปัญญาประดิษฐ์
เพื่อแจ้งเตือนผู้ขับขี่ยานพาหนะส่วนบุคคล

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ร.อ. ดร. ประโยชน์ คำสวัสดิ์)
ประธานกรรมการ

(ผศ. ดร. เศรษฐวิทย์ ภูฉายา)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ผศ. ดร. ชิตพงศ์ เวชไธสงค์)
กรรมการ

(ผศ. ดร. พิชญา ชัยปัญญา)
กรรมการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

(รศ. ดร. พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

นายถิรมนัส สาระภิรมย์ : การจำแนกรถพยาบาลฉุกเฉินโดยใช้ปัญญาประดิษฐ์เพื่อแจ้งเตือนผู้ขับขี่ยานพาหนะส่วนบุคคล (DRIVER ALERT : CLASSIFICATION AND NOTIFICATION FOR AMBULANCE ARRIVAL USING ARTIFICIAL INTELLIGENCE TECHNOLOGY) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. เศรษฐวิทย์ ภูฉายา, 73 หน้า.

ปัจจุบันความแออัดของการจราจรเพิ่มขึ้นเนื่องจากมีการใช้ยานพาหนะส่วนบุคคลจำนวนมากบนถนน ส่งผลให้เกิดการกีดขวางทางจราจรของรถพยาบาลฉุกเฉิน หรือในบางกรณีผู้ขับขี่ยานพาหนะส่วนบุคคลไม่ทราบถึงการมาถึงของรถพยาบาลฉุกเฉิน ด้วยเหตุนี้จึงมีโอกาสที่ผู้ป่วยจะเสียชีวิตเพิ่มขึ้นเนื่องจากปัญหาดังกล่าว ปัจจุบันยานพาหนะส่วนบุคคลได้มีการติดตั้ง Car DVR โดยงานวิจัยนี้นำเสนอการใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) สำหรับการจำแนกรถพยาบาลจากคุณลักษณะของรถพยาบาล (Text Ambulance, ไฟไซเรน, สัญลักษณ์ Red cross และ สัญลักษณ์ Star of life) โดยใช้โมเดล Faster-RCNN ใน TensorFlow ขั้นต้นจะมีการใช้เทคนิคต่างๆก่อนการประมวลผลภาพ การตรวจจับจะใช้กล่องล้อมรอบคุณลักษณะของรถพยาบาล และแสดงความแม่นยำเป็นตัวเลข Intersection over Union (IoU) ใช้สำหรับตรวจสอบความถูกต้อง เพื่อแจ้งเตือนผู้ขับขี่ยานพาหนะส่วนบุคคลถึงการมาถึงของรถพยาบาลฉุกเฉิน

ผลลัพธ์ของงานวิจัยนี้แบ่งการทดสอบออกเป็นสองแบบคือ การทดสอบการใช้งานโมเดลโดยใช้คอมพิวเตอร์เป็นตัวประมวลผลเพื่อตรวจสอบว่าโมเดลสามารถนำไปใช้งานได้จริง ค่าความถูกต้องที่ได้ในการทดสอบนี้ไม่ต่ำกว่า 80% และการทดสอบที่สองคือการทดสอบที่นำโมเดลมาใช้ร่วมกับ Raspberry Pi พบว่าค่าความถูกต้องที่ได้ไม่ต่ำกว่า 75% จากผลการทดสอบพบว่างานวิจัยที่นำเสนอสามารถตรวจจับคุณลักษณะรถพยาบาลฉุกเฉิน และสามารถแจ้งเตือนผู้ขับขี่เพื่อให้ทางแก่รถพยาบาลฉุกเฉินได้ในสภาพแสงที่เหมาะสม

มหาวิทยาลัยเทคโนโลยีสุรนารี

สาขาวิชา วิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา _____

ปีการศึกษา 2563 ลายมือชื่ออาจารย์ที่ปรึกษา _____

TIRAMANAT SARAPIROM : DRIVER ALERT : CLASSIFICATION AND NOTIFICATION FOR AMBULANCE ARRIVAL USING ARTIFICIAL INTELLIGENCE TECHNOLOGY. THESIS ADVISOR : ASST. PROF. SETTAWIT POOCHAYA, Ph.D., 73 PP.

COMPUTER VISION/TENSORFLOW/FASTER R-CNN/EMERGENCY AMBULANCE

Traffic congestion increases due to number of vehicles on the road. Drivers cannot see and arrival of emergency ambulance. Then, personal vehicles block emergency ambulance. As a result, the chance of patient's dead increases due to such problems. Currently, personal vehicles equipped with Car DVR.

This article introduces the implementation of AI for ambulance classification from Feature of ambulance (Text Ambulance, Ambulance lights, Red Cross symbol, Star of life symbol) using Faster-RCNN by TensorFlow. Initially, various image pre-processing techniques are performed. The detection will be used bounding boxes which replace the feature of ambulance and show the accuracy in numbers. Intersection over Union (IoU) used for verify accuracy. To alert drivers of personal vehicles of ambulance approaching.

The results of this research were divided into two types of tests are Testing a model implementation using a computer as a processor and using Raspberry Pi as a processor. The first test to use a computer as a processor to verify that the model works. The result is a detection accuracy of not less than 80%. The second test was created by Raspberry Pi for accuracy of detection not less than 75%. Our results showed that

มหาวิทยาลัยเทคโนโลยีสุรนารี

proposed system can detect the emergency ambulance features and alert the driver to give way for emergency vehicle when notification lighting was appeared.

มหาวิทยาลัยเทคโนโลยีสุรนารี

School of Telecommunication Engineering Student's Signature _____

Academic year 2020 Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีเนื่องจากได้รับความกรุณาอย่างสูงจากคุณแม่และครอบครัวทุกคนที่อุปการะอบรมสั่งสอนและคอยให้กำลังใจเสมอมา

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.เศรษฐวิทย์ ภูฉายา อาจารย์ที่ปรึกษางานวิจัยที่กรุณาให้คำปรึกษาปัญหาในงานวิจัย และช่วยแนะนำและให้ข้อคิดจนสามารถปรับปรุงแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างดี จึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ในสาขาวิชาวิศวกรรมโทรคมนาคมทุกๆ ท่านที่ช่วยให้ความรู้ในด้านวิชาการ และให้คำแนะนำในการดำเนินงานต่างๆ รวมถึงให้ข้อคิดในการดำเนินชีวิต

ขอขอบคุณพี่ เพื่อน และน้องทุกคนที่มีส่วนร่วมในการให้ความช่วยเหลือในทุกๆ เรื่องและคอยให้กำลังใจจนงานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ผู้วิจัยหวังว่าวิทยานิพนธ์ฉบับนี้จะมีส่วนในการแก้ไขปัญหาในด้านการกีดขวางทางจราจรของรถพยาบาลฉุกเฉิน หรือมีประโยชน์ต่องานวิจัยที่เกี่ยวข้องไม่มากก็น้อย หากมีข้อบกพร่องประการใดผู้วิจัยก็ขออภัยไว้ ณ ที่นี้

ถิรมนัส สาระภิรมย์

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่

1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหาวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 ขอบเขตของการวิจัย 3
1.4 ประโยชน์ที่ได้รับจากการวิจัย 3

2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 4
2.1 ปัญญาประดิษฐ์ 4
2.2 วิธีการเรียนรู้เชิงลึก 5
2.3 การจรวจจับวัตถุ 6
2.4 การวัดผล 11
2.5 มาตรฐานรถพยาบาลฉุกเฉิน 13
2.6 TensorFlow 15
2.7 Raspberry Pi 16
2.8 งานวิจัยที่เกี่ยวข้อง 17
2.9 สรุปผลจากการศึกษา 20

3 วิธีดำเนินงานวิจัย 21
3.1 วิธีการดำเนินงาน 21
3.2 วิธีการเก็บรวบรวมข้อมูล 22

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ (ต่อ)

หน้า

3.3 การฝึกสอน AI ใน TensorFlow 22
3.4 การนำข้อมูลที่ฝึกสอนไปใช้กับ Raspberry Pi 32
4 ผลการดำเนินงานวิจัย 51
4.1 ผลลัพธ์การฝึกสอนโมเดล 51
4.2 ผลลัพธ์ของการทดสอบการใช้งานโมเดลโดยใช้คอมพิวเตอร์เป็นตัวประมวลผล 52
4.3 ผลลัพธ์ของการทดสอบการใช้งานโมเดลร่วมกับ Raspberry Pi ในสถานการณ์จริง 54
5 สรุปผล อภิปรายผลและข้อเสนอแนะ 58
5.1 สรุปผลการวิจัย 58
5.2 อภิปรายผล 58
5.3 ข้อเสนอแนะ 59
รายการอ้างอิง 60
ประวัติผู้เขียน 62

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตารางที่ **หน้า**

3.1 ตารางแสดงชื่อฉลากที่ใช้ในงานวิจัย 28

4.1 ผลการทดสอบการตรวจจับของโมเดลในแต่ละคลาสด้วยรูปภาพ 100 ภาพ 57

4.2 ตารางสรุปผลการทดสอบโมเดลร่วมกับ Raspberry Pi 57

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป

รูปที่ **หน้า**

1.1 รายงานสถิติการแพทย์ฉุกเฉินประจำปี 2562 2
1.2 การกีดขวางทางรถพยาบาลฉุกเฉินโดยรถยนต์ส่วนบุคคล 2
2.1 กระบวนการทำงานของ Deep learning 6
2.2 กระบวนการทำงานของ Sliding window 7
2.3 กระบวนการทำงานของ selective search 7
2.4 กระบวนการทำงานของ R-CNN 10
2.5 กระบวนการทำงานของ Faster R-CNN 11
2.6 ตัวอย่างการซ้อนทับของ Ground-truth และ Predicted bounding box 12
2.7 รูปแสดงการเปรียบเทียบผลเฉลยด้วยเมตริก 12
2.8 รูปแสดงผลเฉลยจากการเทียบด้วยเมตริก 13
2.9 ตัวอย่างรถพยาบาลฉุกเฉินที่ถูกต้องตามมาตรฐาน 15
2.10 รูปแสดงการทำงานของ TensorFlow 15
2.11 รูปแสดงองค์ประกอบของ Raspberry Pi 16
2.12 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 1 17
2.13 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 2 18
2.14 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 3 19
3.1 วิธีการดำเนินงาน 21
3.2 รูปตัวอย่างกล้อง DVR ที่ใช้ในการเก็บข้ออมูลภาพ 22
3.3 สัญลักษณ์ Star of life 23
3.4 สัญลักษณ์ Red cross 23
3.5 Text Ambulance 23
3.6 ไฟรถพยาบาลฉุกเฉิน 23
3.7 รูปแสดงการดาวโหลด folder model 24
3.8 รูปแสดง folder model ที่ใช้สำหรับการฝึกสอน 25

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.9 ตัวอย่างการเตรียมภาพที่ใช้ในการฝึกสอน 26
3.10 รูป folder train และ test 26
3.11 รูปแสดงการติดฉลากให้กับรูปภาพ 27
3.12 รูปแสดงการใช้คำสั่งเพื่อสร้างไฟล์ train และ test labels 28
3.13 รูปแสดงไฟล์ train และ test labels 29
3.14 รูปแสดงไฟล์ generate_tfrecord.py 29
3.15 รูปแสดงการกำหนดฉลากในไฟล์ generate_tfrecord.py 30
3.16 รูปแสดงการสร้างไฟล์ labelmap.pbtxt 30
3.17 รูปแสดงการใช้คำสั่งเพื่อฝึกสอนโมเดล 31
3.18 รูปแสดงการฝึกสอนโมเดล 31
3.19 รูปแสดงการเลือก Raspberry Pi OS 32
3.20 รูปแสดงการติดตั้ง Raspberry Pi OS ลงใน SD card 33
3.21 รูปแสดงการติดตั้ง Raspberry Pi OS สำเร็จ 33
3.22 รูปแสดงการใช้คำสั่งเช็คเวอร์ชั่นของโปรแกรม TensorFlow 36
3.23 รูปแสดงการตั้งค่าเพิ่มพื้นที่การแลกเปลี่ยนสำหรับการคอมไพล์กับคอร์ทั้งสี่ของ Raspberry Pi 38
3.24 รูปแสดงการตั้งค่าของ OpenCV 39
3.25 รูปแสดงการคอมไพล์ของ OpenCV 40
3.26 รูปแสดงการใช้คำสั่งเพื่อตรวจสอบเวอร์ชั่นของ OpenCV 41
3.27 รูปแสดงการใช้คำสั่งเพื่อแก้ไขสภาพแวดล้อม PYTHONPATH 43
3.28 รูปแสดงตัวอย่างการแปลงไฟล์ 44
3.29 รูปแสดงโฟลเดอร์ inference_graph และ training 45
3.30 รูปแสดงการแก้ไขไฟล์ Object_detection_webcam.py 45
3.31 รูปแสดงตำแหน่งการติดตั้งอุปกรณ์บนรถยนต์ส่วนบุคคล 46
3.32 รูปแสดงการติดตั้ง Raspberry Pi บนรถยนต์ส่วนบุคคล 47
3.33 รูปแสดงการติดตั้งกล้องบนรถยนต์ส่วนบุคคล 48

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.34 รูปแสดงการติดตั้งไฟแจ้งเตือนบนรถยนต์ส่วนบุคคล 49
3.35 รูปแสดงการติดตั้งจอมอนิเตอร์บนรถยนต์ส่วนบุคคล 50
4.1 รูปแสดงกราฟ total loss 51
4.2 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 1 52
4.3 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 2 ในสภาพแสง ที่ไม่ เหมาะสม 53
4.4 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 2 ในสภาพแสง ที่เหมาะสม 53
4.5 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 1 54
4.6 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 2 55
4.7 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 3 55
4.8 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 4 56
4.9 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 5 56

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

บริการการแพทย์ฉุกเฉิน (Emergency medical services) ในประเทศไทยได้กำหนดมาตรฐานรถพยาบาลฉุกเฉินต้องเดินทางถึงผู้ป่วยภายในเวลา 8 นาที หากเกินเวลาที่กำหนดจะถือว่าเป็นการให้บริการที่ต่ำกว่ามาตรฐาน จากการรายงานสถิติการแพทย์ฉุกเฉินประจำปี 2562 ดังรูปที่ 1.1 ทำให้ทราบว่า ในปีดังกล่าวมีจำนวนการปฏิบัติการของรถพยาบาลฉุกเฉินทั่วประเทศทั้งหมดประมาณ 238,364 ครั้ง ซึ่งจากการบริการทั้งหมดถูกแบ่งออกเป็นการบริการผู้ป่วยฉุกเฉินที่ได้รับบริการภายในเวลา 8 นาทีประมาณ 94,930 ราย และเป็นการบริการผู้ป่วยฉุกเฉินที่ได้รับบริการล่าช้าเกิน 8 นาที 143,434 ราย คิดเป็นร้อยละ 58 จากการบริการทั้งหมดที่ต่ำกว่ามาตรฐานบริการการแพทย์ฉุกเฉิน การบริการนี้ถูกแบ่งการให้บริการออกเป็น 12 โซน โดยจังหวัดนครราชสีมาถูกจัดอยู่ในโซนที่ 9 ซึ่งมีผู้ป่วยที่ได้รับบริการล่าช้าเกิน 8 นาทีถึงร้อยละ 62 [9]

จากการสำรวจพบว่า สาเหตุที่ทำให้การบริการการแพทย์ฉุกเฉินมีความล่าช้าเกิดได้จากหลายสาเหตุ เช่น ระยะทางในการเดินทางเพื่อให้บริการ เนื่องจากหากจุดให้บริการและจุดเกิดเหตุมีระยะทางที่ต่างกันมากเกินไปทำให้รถพยาบาลฉุกเฉินต้องใช้เวลาในการเดินทางเพิ่มขึ้น และอีกหนึ่งสาเหตุสำคัญที่พบได้บนถนน คือ การกีดขวางของรถยนต์ส่วนบุคคลในระหว่างการเดินทาง ในปัจจุบันนี้อัตราการใช้รถยนต์บนถนนเพิ่มมากขึ้นก่อให้เกิดปัญหาการจราจรติดขัดส่งผลให้การไปถึงที่หมายของรถพยาบาลฉุกเฉินล่าช้า อาจเนื่องด้วยรถยนต์ส่วนบุคคลมีการใช้เครื่องเสียงที่เสียงดังมากเกินไป หรือ รถยนต์ส่วนบุคคลไม่สามารถมองเห็นการมาถึงของรถพยาบาลฉุกเฉินได้ เป็นสาเหตุให้รถยนต์ส่วนบุคคลไม่สามารถหลีกทางให้รถพยาบาลฉุกเฉินได้ ดังรูปที่ 1

จากปัญหาข้างต้นจึงมีแนวคิดที่จะนำเทคโนโลยีปัญญาประดิษฐ์(AI)ซึ่งเป็นเทคโนโลยีการมองเห็นของคอมพิวเตอร์ ใช้ในการจำแนกรถพยาบาลฉุกเฉิน โดยการทำการสอนคุณลักษณะของรถพยาบาลฉุกเฉินให้อุปกรณ์ Digital Video Recorder บนรถยนต์ส่วนบุคคล เพื่อทำการตรวจจับ

การมาถึงของรถพยาบาลฉุกเฉิน เพื่อให้มีการแจ้งเตือนบนรถยนต์ส่วนบุคคลให้หลีกทางให้รถพยาบาลฉุกเฉิน

เครือข่าย	จำนวน			รวม	ร้อยละ		
	ภายใน 8 นาที	มากกว่า 8 นาที	ไม่ระบุ		ภายใน 8 นาที	มากกว่า 8 นาที	ไม่ระบุ
: รวม	94,930	143,434	1,855	238,364	42	58	1
: เครือข่าย 0	6	16	0	22	27	73	0
: เครือข่าย 1	11,461	15,596	16	27,057	45	55	0
: เครือข่าย 2	3,816	8,755	253	12,571	35	65	2
: เครือข่าย 3	3,523	6,495	5	10,018	35	65	0
: เครือข่าย 4	7,224	16,332	646	23,556	35	65	1
: เครือข่าย 5	7,604	14,945	104	22,549	35	65	0
: เครือข่าย 6	8,021	19,589	117	27,610	33	67	0
: เครือข่าย 7	10,350	9,425	4	19,775	55	45	0
: เครือข่าย 8	7,490	8,904	48	16,394	47	53	0
: เครือข่าย 9	8,525	15,146	167	23,671	38	62	1
: เครือข่าย 10	7,576	8,060	52	15,636	50	50	1
: เครือข่าย 11	9,328	8,403	192	17,731	48	52	1
: เครือข่าย 12	10,006	11,768	251	21,774	51	49	1

รูปที่ 1.1 รายงานสถิติการแพทย์ฉุกเฉินประจำปี 2562

(ที่มา: https://ws.niems.go.th/ITEMS_DWH/)

รูปที่ 1.2 การกีดขวางทางรถพยาบาลฉุกเฉินโดยรถยนต์ส่วนบุคคน

(ที่มา: http://www.mheemhee.com/หลีกทางให้รถฉุกเฉิน/)

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.2 วัตถุประสงค์ของการวิจัย

2.1 การนำเทคโนโลยี computer vision มาใช้ในการจำแนกรถพยาบาลฉุกเฉินโดยเทคนิคการประมวลผลภาพ

2.2 เพื่อแจ้งเตือนผู้ขับขี่รถยนต์ส่วนบุคคลให้ทราบการมาถึงของรถพยาบาลฉุกเฉิน

1.3 ขอบเขตของการวิจัย

3.1 ทดสอบค่าความแม่นยำของแบบจำลองที่ใช้กับการตรวจจับการมาถึงของรถพยาบาลฉุกเฉินในช่วงเวลากลางวันในสภาพแสงที่เหมาะสม

3.2 ติดตั้งแบบจำลอง ลงบนอุปกรณ์ DVR บนรถยนต์ส่วนบุคคล

1.4 ประโยชน์ที่ได้รับจากการวิจัย

4.1 การนำ computer vision มาใช้ในการจำแนกรถพยาบาลฉุกเฉินโดยเทคนิคการประมวลผลภาพเพื่อแจ้งเตือนผู้ขับขี่รถยนต์ให้ทราบถึงการมาถึงของรถพยาบาลฉุกเฉิน

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2

ปริทัศน์วรรณกรรมงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงวรรณกรรมที่เกี่ยวข้องกับงานวิจัย (Literature Review) และทฤษฎีที่ใช้ในงานวิจัยชิ้นนี้ เพื่อเป็นการศึกษารายละเอียดของงานวิจัยที่เกี่ยวข้องในการทำ Object detection และการทำ Classification เนื้อหาในบทนี้ประกอบด้วย

2.1 ปัญญาประดิษฐ์

2.2 วิธีการเรียนรู้เชิงลึก (Deep Learning)

2.3 การตรวจจับวัตถุ (Object detection)

2.4 การวัดผล (Intersection over Union)

2.5 มาตรฐานรถพยาบาลฉุกเฉิน

2.6 TensorFlow

2.7 Raspberry Pi

2.8 งานวิจัยที่เกี่ยวข้อง (Literature Review)

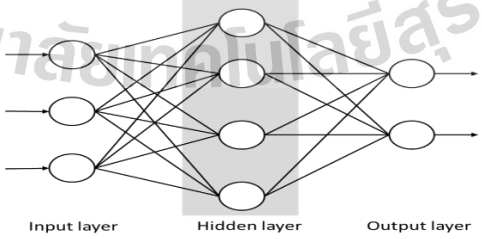
2.1 ปัญญาประดิษฐ์ (Artificial Intelligence)

ปัญญาประดิษฐ์ (Artificial Intelligence) หรือเอไอ (AI) เป็นระบบประมวลผลที่มีต้นแบบมาจากโครงข่ายประสาทของมนุษย์สามารถเรียนรู้และเพิ่มประสิทธิภาพการประมวลผลได้ตามจำนวนข้อมูลที่เพิ่มขึ้นผ่านกระบวนการเรียนรู้ด้วยตนเอง ซึ่งสามารถจดจำ คิด วิเคราะห์เรียนรู้และเชื่อมโยงข้อมูลต่างๆที่ซับซ้อนได้อย่างรวดเร็ว (Deep Learning) เสมือนระบบสมองของมนุษย์ จึงอาจเรียกได้ว่า “สมองกลอัจฉริยะ” ดังนั้น AI จึงถือเป็นเทคโนโลยีที่นิยมที่สุดในปัจจุบันและเข้ามามีบทบาทสำคัญต่อการใช้ชีวิตการทำงาน รวมถึงการนำมาใช้ในการเสริมศักยภาพทางธุรกิจและอุตสาหกรรม ซึ่งจะสามารถส่งผลต่อการเจริญเติบโต ทางด้านเศรษฐกิจอย่างยั่งยืนของประเทศ

โดยทั่วไปแล้วศาสตร์ทางปัญญาประดิษฐ์จะเป็นสาขาในด้านวิทยาการคอมพิวเตอร์และวิศวกรรมเป็นหลัก แต่บางครั้งก็ยังรวมไปถึงศาสตร์ในด้านอื่นๆ อีกด้วย ไม่ว่าจะเป็นทางด้านจิตวิทยา ปรัชญา หรือแม้แต่ชีววิทยาศาสตร์ ทางด้านปัญญาประดิษฐ์นั้นไม่มีข้อกำหนดหรือรูปแบบที่ชัดเจนในการกำหนดความฉลาดของเครื่องจักร เนื่องจากเราไม่สามารถนำมาเปรียบเทียบกับความฉลาดของมนุษย์ที่มีการเปลี่ยนแปลงอยู่ตลอดเวลาได้ และสามารถเข้าใจกลไกเพียงบางส่วนเท่านั้น นอกจากนี้ศาสตร์ทางด้านปัญญาประดิษฐ์ถึงแม้ว่าจะเกี่ยวข้องกับความฉลาดของเครื่องจักร แต่ก็ไม่ได้หมายความว่า เป็นการจำลองความฉลาดของมนุษย์เสมอไป

2.2 วิธีการเรียนรู้เชิงลึก (Deep Learning)

วิธีการเรียนรู้เชิงลึกเป็นซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบให้คล้ายกับเครือข่ายเซลล์ประสาทในสมองของมนุษย์ (Network of Neuron) แสดงดังรูป 2.1 ซึ่ง Deep learning ถูกสร้างขึ้นโดยการนำเอา Neural network หลายๆ layer มาต่อกันโดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (input layer) และ layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (output layer) ส่วน layer ระหว่าง layer แรกสุดและ layer สุดท้ายจะถูกเรียกว่า Hidden layer โดยที่ Hidden layer ของแต่ละ layer จะเปรียบเสมือนว่าประกอบด้วยเซลล์ประสาทจำนวนมาก มีหน้าที่ในการประมวลผลรับข้อมูลจาก layer ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยัง layer ที่อยู่ล่างกว่า ข้อดีของการส่งข้อมูลแบบนี้ก็คือ layer แต่ละ layer สามารถที่จะมีค่าถ่วงน้ำหนัก (weight) ค่าความเอนเอียงของข้อมูล (bias) และวิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกันเมื่อมีข้อมูล input ให้กับ model มากเท่าไหร่ layer แต่ละ layer ก็จะสามารถสกัด feature ที่มีความซับซ้อนมากยิ่งขึ้น [4]



รูปที่ 2.1 กระบวนการทำงานของ Deep learning

(ที่มา: https://medium.com/@natthawatphongchit/ย้อนรอย-object-detection)

2.3 การตรวจจับวัตถุ (Object detection)

การที่จะให้คอมพิวเตอร์มีความสามารถจำแนกวัตถุที่อยู่ในรูปภาพออกมาแล้วทำนายว่าวัตถุเป็น Class อะไรต้องอาศัยหลักการของ Object classification และการที่จะให้คอมพิวเตอร์หาตำแหน่งของวัตถุต้องอาศัยหลักการของ Object Detection การทำ Object classification ซึ่งสามารถสรุปขั้นตอนการทำงานได้ดังนี้ Input Feature > Extractor > Classifier > Output Class นั่นคือนำข้อมูลเข้าเพื่อสกัดหาคุณลักษณะที่สำคัญแล้วทำ classifier เพื่อให้ได้ Output ของ class [3] ขั้นตอนถัดไปจะเข้าสู่กระบวนการ Object Detection ซึ่งมีหลายแบบดังนี้

2.3.1 Sliding Window

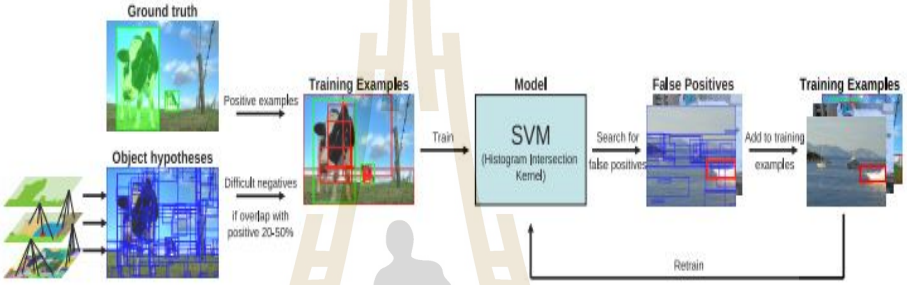
Sliding Window เป็นวิธีการทำ Object Detection ที่นิยมมากในสมัย10ปีที่ผ่านมา มีการทำงานจะนำเอาผลลัพธ์จากการทำ Object Classification มาทำการตีกรอบแล้วขยับไปเรื่อยๆ จากนั้นภาพที่ได้จากการขยับเข้าไปทำ Feature Extractor และทำการ classifier ว่าตรงกับ class หรือไม่อาจจะมีการขยับทีละ 1 pixel ไปเรื่อย ๆจนหมด [3] ซึ่งอาจจะทำให้เกิดปัญหาขึ้นได้ง่าย เพราะถ้าหากเรากำหนดกรอบ ขนาด (32X32) และมีภาพขนาด (512X512) เวลาทำ Sliding Window จะต้องใช้พลังงานของเครื่องในการประมวลผลอย่างมหาศาลถึงจะเสร็จ ซึ่งในปัจจุบันได้มีการแก้ไขปัญหานี้แล้วโดยมีการย่อขยายรูปทุกครั้งที่มีการทำงานเสร็จ

รูปที่ 2.2 กระบวนการทำงานของ Sliding window

(ที่มา: https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375)

2.3.2 Selective search

Selective search เป็นอัลกอริทึมข้อเสนอของ region proposal ที่ใช้ในการตรวจจับวัตถุโดยถูกออกแบบมาให้รวดเร็วพร้อมการเรียกคืนที่สูงมาก ขึ้นอยู่กับการคำนวณการกระจัดกลุ่มลำดับชั้นของพื้นที่ที่คล้ายกันซึ่งอาจจะขึ้นอยู่กับสีพื้นผิวขนาดและความเข้ากันได้ของรูปร่าง การค้นหาแบบเลือกเริ่มต้นด้วยการแบ่งส่วนภาพตามความเข้มของพิกเซลโดยใช้วิธีแบ่งกลุ่มตามกราฟ [4]



รูปที่ 2.3 กระบวนการทำงานของ selective search

(ที่มา: https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/)

2.3.3 Neural Network หรือ Artificial Neural Network (ANN)

ANN เป็นโมเดลทางคณิตศาสตร์ที่จำลองกระบวนการคิดจากสมองมนุษย์โดยสมองนั้นจะมีหน่วยประมวลผลขนาดเล็กอยู่เป็นจำนวนมาก และเชื่อมโยงกันด้วยโครงข่ายประสาทมากมายทำให้มนุษย์เรียนรู้และคิดวิเคราะห์ได้อย่างรวดเร็ว แต่คอมพิวเตอร์นั้นไม่ได้มีโครงข่ายที่ซับซ้อนเหมือนกับสมองของมนุษย์ คอมพิวเตอร์มีหน้าที่แค่รันโปรแกรมตามคำสั่งเท่านั้น เมื่อให้คอมพิวเตอร์ทำการเรียนรู้บางอย่างจึงเป็นเรื่องยากในรูปแบบปกติ จึงเกิดการจำลองแนวทางการเรียนรู้ของคนไปสู่คอมพิวเตอร์ด้วย ANN [4] ส่วนที่เล็กที่สุดคือ Neuron ซึ่งทำหน้าที่คำนวณ มีส่วนประกอบดังนี้

- Input เป็นค่าที่ส่งเข้ามาที่ Neuron โดยจะมีการส่งข้อมูลเข้ามาได้หลายครั้ง โดยทั่วไปมักจะเท่ากับจำนวน Class

- Weight เป็นการให้น้ำหนักของข้อมูลแต่ละข้อมูลที่ส่งเข้ามา โดยมีค่าตั้งแต่ 0 - 1 โดยที่ทำการสุ่มขึ้นมาจากนั้นเมื่อ Neuron ได้ทำการเรียนรู้เรื่อย ๆก็จะทำการปรับ weight เพื่อให้ได้คำตอบที่ใกล้เคียงที่สุด

- Bias คือค่าที่จะช่วยเข้ามาทำให้ค่าที่เข้ามาอยู่ระหว่าง 0 – 1 ได้โดยจะเป็นเลขที่สุ่มและปรับไปเรื่อย ๆทุกครั้งที่เรียนรู้

- Output คือผลลัพธ์

- Back Propagation คือการที่ Neuron นำค่า Error ของ Output ที่ได้ กับ Output ที่เรียนรู้นำไปปรับ Weight และ Bias ให้เกิดผลลัพธ์ที่ถูกต้องตามที่ได้เรียนรู้

เมื่อป้อน Input และ Output เพื่อทำการเรียนรู้ ตัว Neuron จะทำการบวกตัว Input ด้วย Weight คูณกับค่าของแต่ละข้อมูล จากนั้นจะบวกค่า Bias แล้วนำไปเข้าฟังก์ชันที่ตัว Neuron กำหนดไว้ก็จะเป็น Output โดยฟังก์ชันก็จะมีหลายแบบขึ้นอยู่กับรูปแบบที่จะนำไปใช้งาน

2.3.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) คือ โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นกระบวนการที่สกัดเอาลักษณะที่สำคัญของภาพออกมา โดยใช้ค่า Pixel ซึ่งได้จากข้อมูลอินพุต มีทั้งหมด 3 Channel ได้แก่ สีแดง, สีน้ำเงิน และสีเขียว สามารถใช้เลข 0 ถึง 255 เพื่อเป็นค่าแทนความเข้มของสี [3] เนื่องจาก CNN ไม่ค่อยเป็นที่นิยมเพราะต้องใช้กับคอมพิวเตอร์ที่มีประสิทธิภาพสูง แต่ในปัจจุบันมีการพัฒนาหน่วยประมวลผลกราฟฟิก (GPU) ให้รองรับการคำนวณ จึงทำให้ CNN ประสบความสำเร็จในการแก้ปัญหา Classification ที่เกี่ยวข้องกับรูปภาพ [4] อีกทั้งในปัจจุบันได้นำแนวคิดไปใช้ต่อยอดมากมาย โครงสร้างของ Convolutional Neural Network ประกอบด้วยดังนี้

2.3.3.1.1 Convolutional เป็น Layer หลักของ CNN ทำหน้าที่รับ Input เข้ามาแปลงภาพให้เป็น pixel ที่กำหนดให้เป็น 0 – 255 หลังจากนั้นใช้การดำเนินการทางคณิตศาสตร์แปลงเป็นข้อมูลเพื่อหา Feature Extraction เพื่อนำมาคูณกับตัวกรอง (filter) ที่ทำหน้าที่ดึงคุณลักษณะที่ใช้ในการจำแนกวัตถุออก โดยปกติตัวกรองอันหนึ่งจะดึงคุณลักษณะที่สนใจออกมาหรือที่เรียกว่า Feature Map ส่วนการที่จะขยับตัวกรองเป็นขนาดเท่าใดจะถูกกำหนดด้วย Stride ซึ่งสามารถกำหนดค่าของ Stride ให้มากขึ้นได้หากต้องการให้คำนวณหาคุณลักษณะที่มีพื้นที่ทับซ้อนกันน้อยลง แต่อย่างไรก็ตามการกำหนดค่าของ Stride ที่มากขึ้นจะทำให้ได้คุณลักษณะ (Feature Map) ที่มีขนาดเล็กลง

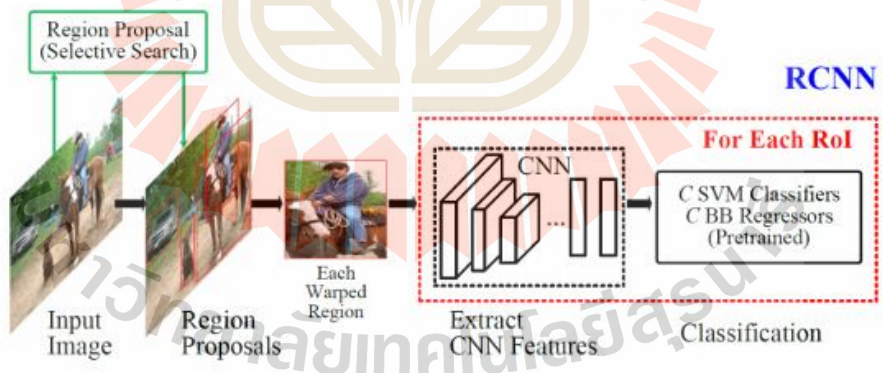
2.3.3.1.2 Padding เป็นการกำหนด Feature Map ให้มีขนาดเท่ากับ input ส่วนมากจะเติมค่า 0 (สีเทา) หรือค่าต่าง ๆเข้าไป

2.3.3.1.3 Pooling เป็นกระบวนการที่ทำหน้าที่ลดขนาดของ Feature Map ที่ได้มาจากการทำ CNN มีวัตถุประสงค์ในการลดจำนวนของพารามิเตอร์ที่มีมากเกินไป และลดระยะเวลาในการฝึกสอน

2.3.3.1.4 Fully Connected เป็น layer สุดท้ายของการทำ CNN ส่วนนี้ จะทำหน้าที่นำเอาคุณลักษณะสำคัญไปสร้างเป็น Neural Network สำหรับการเรียนรู้ จดจำรูปแบบ และการทำนายประเภท โดยใช้เทคนิคที่ชื่อว่า SoftMax Classification ต่อไป

2.3.3.2 R-CNN

R-CNN เป็นการนำข้อมูล region proposals มาแปรสภาพเป็นสี่เหลี่ยมจัตุรัสและป้อนเข้าสู่โครงข่ายประสาทเทียมที่สร้างเวกเตอร์ที่มีคุณลักษณะ 4096 มิติเป็นเอาต์พุต CNN ทำหน้าที่เป็นตัวแยกคุณลักษณะและเลเยอร์หนาแน่นของเอาต์พุต ซึ่งจะประกอบด้วยคุณลักษณะที่ดึงออกมาจากรูปภาพและฟีเจอร์ที่แยกออกมาแล้วจะถูกป้อนเข้าสู่ Support Vector Machine (SVM) ทำหน้าที่จำแนกการมีอยู่ของวัตถุภายใน [4][5]



รูปที่ 2.4 กระบวนการทำงานของ R-CNN

(ที่มา: https://medium.com/@natthawatphongchit/ย้อนรอย-object-detection)

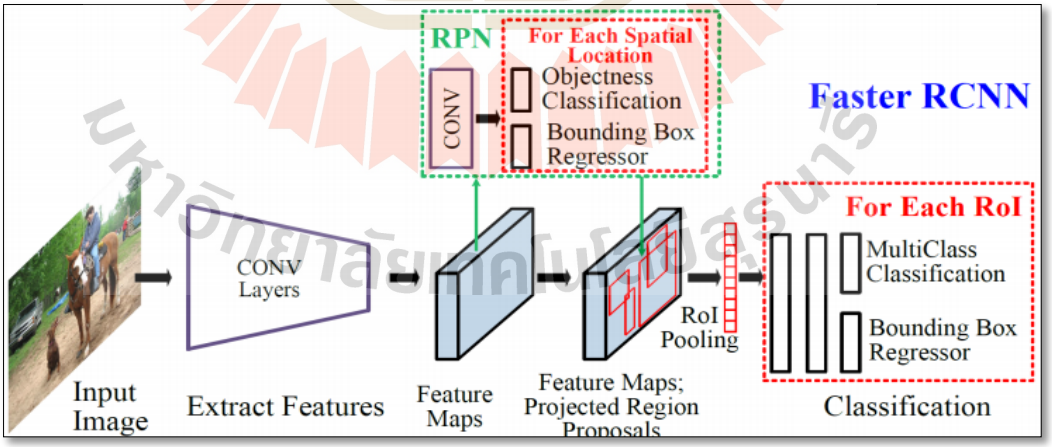
2.3.3.3 Faster R-CNN

Faster R-CNN เป็นการทำ Object Detection ที่ปรับปรุงข้อจำกัดหลายอย่างของ R-CNN เช่นการทำงานในการเสนอพื้นที่ที่น่าจะเป็นวัตถุแล้วสกัด feature เปลี่ยนเป็นการสกัดfeature ก่อนแล้วค่อยนำเสนอพื้นที่ที่น่าสนใจชุดใหญ่ เพื่อเพิ่มความแม่นยำและความรวดเร็วในการทำนาย [2] โดยโครงสร้างของ Faster R-CNN ประกอบด้วย

2.3.3.3.1 Convolutional Layer นำข้อมูลภาพ Input มา Process และส่งออกเป็น Feature Map แล้วแบ่งเป็นข้อมูลสองทาง คือ 1 ส่ง Feature Map ไป Region Proposal Network (RPN) และทางที่ 2 คือ ส่ง Feature Map ไปรอค่าจาก RPN ก่อนการ ROI Pooling (Region of Interest Pooling) [5]

2.3.3.3.2 RPN ทำหน้าที่สกัดเอาบริเวณที่น่าจะเป็นวัตถุจาก Feature Map แล้วส่งค่า Region ที่ได้สู่กระบวนการ ROI Pooling [4]

2.3.3.3.3 ROI Pooling เป็นขั้นตอนการนำค่า Region จาก RPN มา Pooling กับ Feature Map ของทางที่ 2 เพื่อสร้าง Feature Vector ที่มีขนาดคงที่ ส่ง Fully Connected Layers ทำการคำนวณผลลัพธ์แล้วแสดงผลการทำนายออกมาเป็น 2 ลักษณะ คือ Classification และ Bounding Box เพื่อตอบว่ามีวัตถุอยู่ในบริเวณนั้นหรือไม่ [6][7]



รูปที่ 2.5 กระบวนการทำงานของ Faster R-CNN

(ที่มา: https://medium.com/@natthawatphongchit/ย้อนรอย-object-detection)

2.4 การวัดผล Intersection over Union (IoU)

- Intersection over Union (IoU) เป็นการวัดประสิทธิภาพของโมเดลเป็นที่นิยมในการทำ Object detection ซึ่งหาได้จาก อัตราส่วนระหว่าง area ที่เป็น intersection ของ 2 bounding box หารด้วย area รวมของกรอบทั้งสอง หรืออาจจะเรียกว่าดัชนี jaccard เป็นหลักวิธีการในการหาจำนวนเปอร์เซ็นต์ที่ทับซ้อนกันระหว่างผลเฉลย (Ground Truth) และผลจากการทำนาย (Predict) ถ้าได้ค่า IoU ที่มากกว่า 0.5 ถือว่ายอมรับได้ [10]

- AP และ mean Average Precision (mAP) เป็นการหาค่าเฉลี่ยของความแม่นยำ (Precision) และความลึก (Recall) ของวัตถุในรูปนั้นๆ โดยที่ค่าความแม่นยำหาได้จาก

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

เมื่อ True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่าจริง

True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่าไม่จริง

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่าจริง แต่คนบอกว่าไม่จริง

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง แต่คนบอกว่าจริง

มหาวิทยาลัยเทคโนโลยีสุรนารี

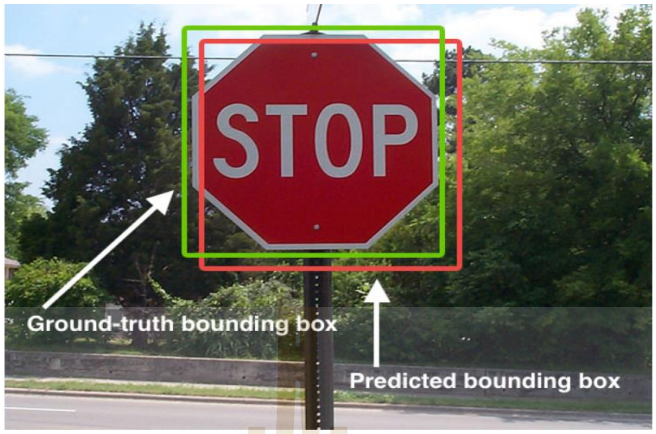


Figure 3: An example of detecting a stop sign in an image. The predicted bounding box is drawn

รูปที่ 2.6 ตัวอย่างการซ้อนทับของ Ground-truth และ Predicted bounding box

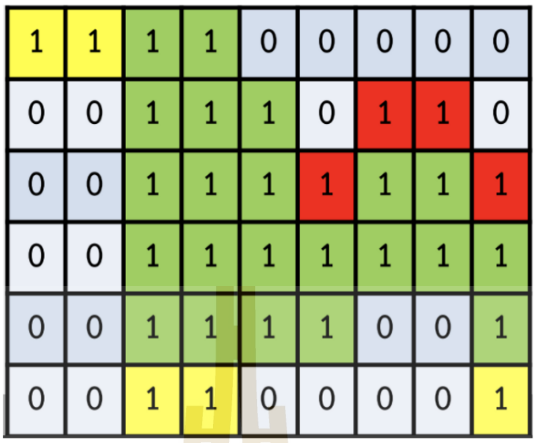
(ที่มา: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/)

1	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	1	1	0	0	0	0	1

0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	0	0	0	0	0	0	0

รูปที่ 2.7 รูปแสดงการเปรียบเทียบผลเฉลยด้วยเมตริก

(ที่มา: https://medium.com/@natthawatphongchit/ย้อนรอย-object-detection)



รูปที่ 2.8 รูปแสดงผลเฉลยจากการเทียบด้วยเมตริก

(ที่มา: https://medium.com/@natthawatphongchit/ย้อนรอย-object-detection)

รูปที่ 2.7 แสดงถึงตัวอย่างของเมทริกซ์ Ground-truth bounding box (ฝั่งซ้าย) และ Predicted bounding box (ฝั่งขวา) เมื่อนำเมทริกซ์มาซ้อนทับกันแล้วจะได้ผลลัพธ์ดังรูปที่ 2.8 โดยสีเหลืองคือส่วนที่โปรแกรมทำนายว่าไม่จริงแต่คนบอกว่าจริง สีแดงคือส่วนที่โปรแกรมทำนายว่าจริงแต่คนบอกว่าไม่จริง สีเขียวคือส่วนที่โปรแกรมทำนายว่าจริงและคนบอกว่าจริง และสีเทาคือส่วนที่โปรแกรมทำนายว่าไม่จริงและคนบอกว่าไม่จริง

2.5 มาตรฐานรถพยาบาลฉุกเฉิน

เพื่อให้ประชาชนโดยทั่วไปและผู้ป่วยฉุกเฉิน ได้รับรู้ถึงพาหนะที่ใช้เพื่อการลำเลียงและการขนส่ง กรณีการเจ็บป่วยฉุกเฉินอย่างชัดเจน จึงกำหนดมาตรฐานการนำเครื่องหมายและตราสัญลักษณ์ เพื่อไปแสดงหรือติดไว้กับพาหนะเพื่อการลำเลียงหรือขนส่ง สำหรับการปฏิบัติการฉุกเฉิน [9]

เครื่องหมายหรือตราสัญลักษณ์ หมายถึง ตราเครื่องหมายของสถาบันการแพทย์ฉุกเฉินแห่งชาติ รวมถึงชื่อ ข้อความ หรือตัวอักษรที่บ่งบอกว่าเป็นการดำเนินการ หรือด้วยความยินยอมหรือการตรวจรับรองจากสถาบันการแพทย์ฉุกเฉินแห่งชาติ

ชื่อหน่วยปฏิบัติการ หมายถึง หน่วยงานหรือองค์กรที่ปฏิบัติการฉุกเฉินทั้งนี้รวมถึงสถานพยาบาลที่มีการปฏิบัติการฉุกเฉินด้วย

การแสดงเครื่องหมายหรือตราสัญลักษณ์บนรถบริการการแพทย์ฉุกเฉิน

1. กระจกหน้าด้านในฝั่งซ้าย ติดป้ายระบุผ่านการรับรองรถบริการการแพทย์ฉุกเฉิน โดยแสดงหมายเลขทะเบียน วันหมดอายุการรับรอง โดยมี QR code ประจำรถคันนั้นๆ แสดงอยู่ด้วย

2. กระจกหน้าด้านบน ติดชื่อหน่วยปฏิบัติการตัวอักษรสีน้ำเงินบนพื้นสติกเกอร์สีขาว หรือตัวอักษรสีขาวบนพื้นสีดำ หรือสีอื่นที่สามารถมองเห็นได้ชัดเจน

3. ใต้กระจกด้านหน้าติดแสดงตัวอักษรคำว่า “AMBULANCE” สีแดง เพื่อให้รถที่ขับอยู่ด้านหน้ามองกระจกหลังสามารถมองเห็นว่าเป็นรถบริการการแพทย์ฉุกเฉินได้อย่างชัดเจน

4. กระจกหลังติดข้อความ “ เจ็บป่วยฉุกเฉิน อุบัติเหตุโทร 1996 ”ด้วยตัวอักษรสีแดงขอบสีขาว หรือเพิ่มเติมคำอื่นๆ (ถ้ามี) เพื่อสื่อสารให้ประชาชนรับรู้ ประชาสัมพันธ์ การเข้าถึงบริการ

5. ใต้กระจกด้านหลัง ติดชื่อหน่วยปฏิบัติการ ตัวอักษรสีน้ำเงินขอบสีขาว

6. แสดงสัญลักษณ์หน่วยปฏิบัติการต้นสังกัดที่ประตูด้านหน้าทั้งสองข้าง

7. แสดงชื่อหน่วยปฏิบัติการ พร้อมหมายเลขโทรศัพท์ 1996 ด้วยตัวอักษรสีน้ำเงินขอบสีขาว ด้านข้างรถทั้งสองข้าง

8. ติดแถบสีสะท้อนแสง ด้านข้างรถตลอดแนว รอบคัน

9. ติดข้อความ “รถฉุกเฉินได้รับการอนุญาติแล้ว” เป็นตัวอักษรสีแดง

10. ด้านข้างช่วงหลังทั้งสองข้าง ติดสติกเกอร์ระบุผ่านการรับรองรถบริการการแพทย์ฉุกเฉินโดยสถาบันการแพทย์ฉุกเฉินแห่งชาติ โดยมี QR code ประจำรถคันนั้นๆแสดงอยู่ด้วย และติดสติ๊กเกอร์เครือข่ายรถบริการการแพทย์ฉุกเฉินระบุจังหวัดพื้นที่ให้บริการ

11. ให้รถที่ผ่านการรับรองและได้รับการอนุญาตเป็นรถบริการการแพทย์ฉุกเฉินแล้ว ให้ติดตั้งไฟสัญญาณวับวาบ ไว้บนหลังคารถ โดยวางตำแหน่งของสัญญาณไฟวับวาบสีแดงไว้ที่ด้านขวา เหนือคนขับ

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 2.9 ตัวอย่างรถพยาบาลฉุกเฉินที่ถูกต้องตามมาตรฐาน

(ที่มา: https://www.niems.go.th/1/upload/migrate/file/255712081310064271)

2.6 TensorFlow

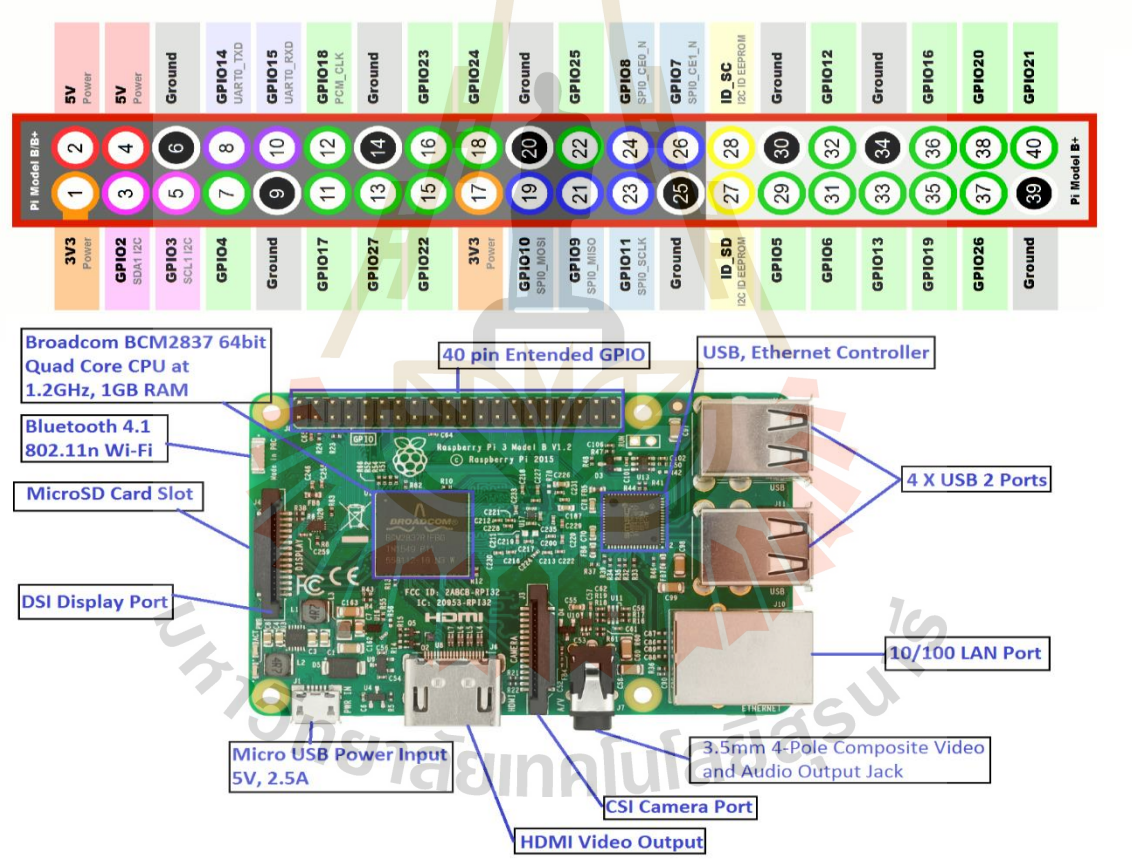
TensorFlow คือไลบรารีสำหรับการสร้างเครือข่ายประสาทเทียม ซึ่งรวม API ต่างๆไว้ด้วยกัน นำไปใช้เพื่อสร้างกระบวนการการเรียนรู้สถาปัตยกรรมเชิงลึกเช่น R-CNN, Fast-RCNN และ Faster-RCNN โดย TensorFlow ถูกพัฒนามาจากภาษา Python [6][7]

รูปที่ 2.10 รูปแสดงการทำงานของ TensorFlow

(ที่มา: https://www.tensorflow.org)

2.7 Raspberry Pi 3 Model B

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กสามารถเชื่อมต่อระบบเครือข่ายแบบใช้สายหรือไร้สายได้ตัวบอร์ดใช้ SoC ของ Broadcom เบอร์ BCM2837 สถาปัตยกรรม 64 บิต ตัว CPU ภายในเป็น ARM Cortex A53 มี 4 แกนสมอง ทำงานที่ความถี่สัญญาณนาฬิกา 1.2GHz แรม 1GB เชื่อมต่อกับอินเตอร์เน็ตผ่านพอร์ต Ethernet หรือ WiFi และยังมีบลูทูธ 4.1 มาในตัว ช่อง GPIO มีทั้งหมด 40 Pin ช่อง USB จำนวน 4 ช่อง มีช่องเสียบสัญญาณเสียงขนาด 3.5 มิลิเมตร มีช่อง HDMI สำหรับต่อจอภาพ ใช้กระแสไฟฟ้าสูงสุด 2.5A ที่แรงดันไฟฟ้า 5V ลง OS ผ่าน MicroSD Card และตัวบอร์ดมี OS เฉพาะชื่อ Raspbian



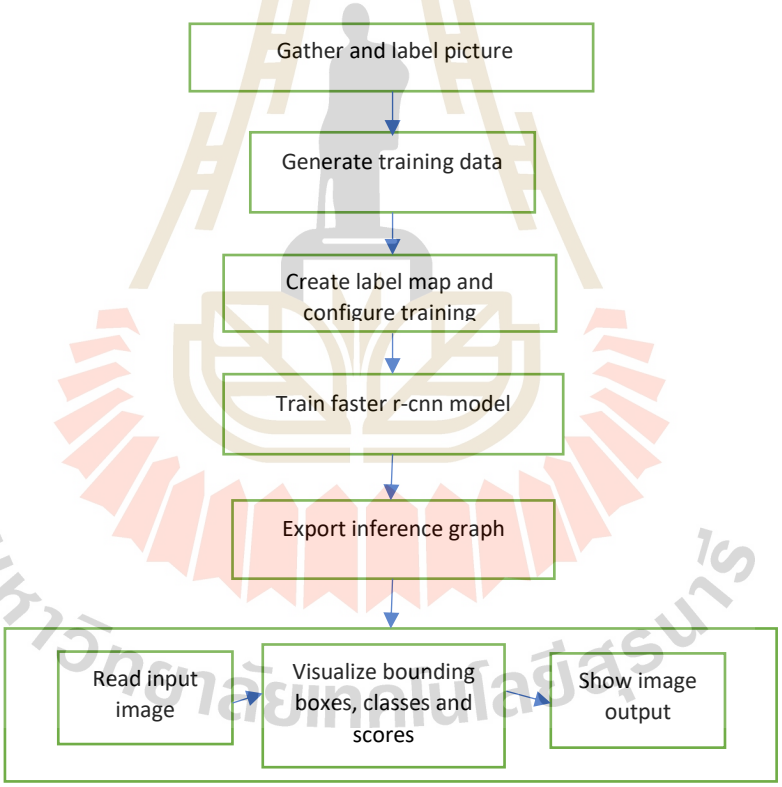
รูปที่ 2.11 รูปแสดงองค์ประกอบของ Raspberry Pi

(ที่มา: https://raspberrypi.stackexchange.com/)

2.8 งานวิจัยที่เกี่ยวข้อง (Literature Review)

2.8.1 งานวิจัยที่เกี่ยวข้องที่ 1

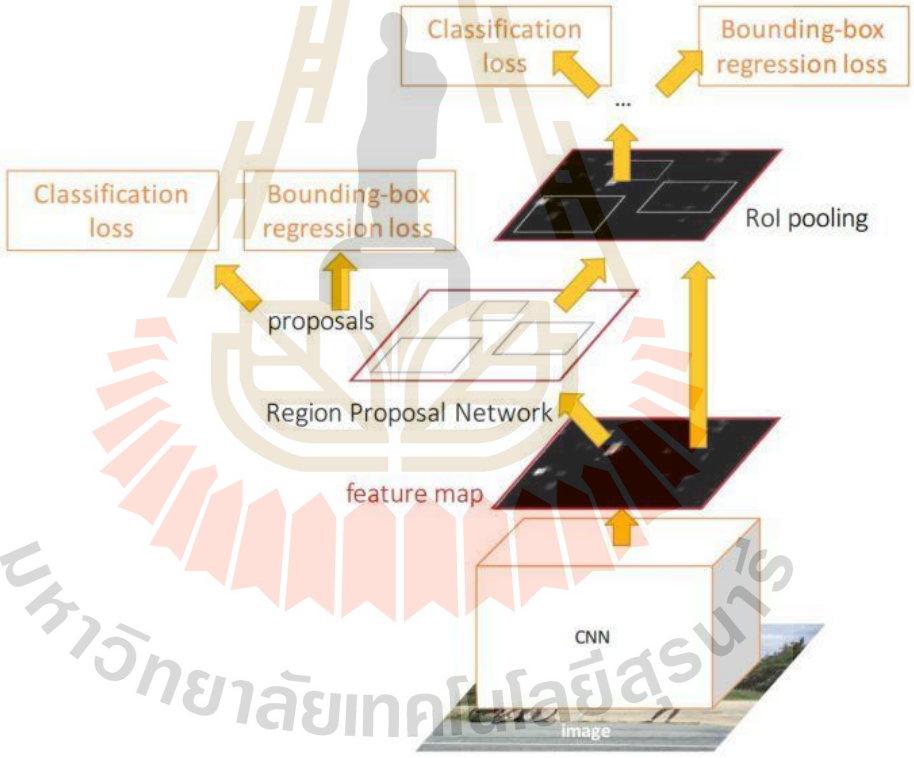
(Anuradha, Khelchandra (2020)) งานวิจัยได้กล่าวถึงการทำ Object detection โดยใช้ model จากการทำ Faster R-CNN เพื่อใช้จำแนกประเภทของฟันโดยได้ทำการแบ่งคลาสออกเป็น 6 คลาส คือ 1.Canine 2.Cap 3.Impacted 4.Incisor 5.Molar 6.Pre-molar ในการทำงานวิจัยนี้ใช้ชุดข้อมูลรูปภาพทั้งหมด 145 ภาพ โดยแบ่งออกเป็นภาพที่ใช้ในการ Train 96 ภาพ และภาพที่ใช้ในการ Test 49 ภาพ โดยในการวัดค่าความถูกต้องผู้วิจัยได้กำหนดค่า IoU ไว้ที่ 0.7 ผลจากการทดสอบพบว่าการตรวจจับวัตถุมีเปอร์เซนต์ความถูกต้องสูงกว่า 90%



รูปที่ 2.12 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 1

2.8.2 งานวิจัยที่เกี่ยวข้องที่ 2

(Hasan Basri, Iwan Syarif, Sritrustra Sukaridhoto (2018)) งานวิจัยได้กล่าวถึงการทำ Object detection โดยใช้ model จากการทำ Faster R-CNN เพื่อจำแนกประเภทของผลไม้โดยแบ่งคลาสออกเป็น 2 คลาส คือ 1. มะม่วง 2. แก้วมังกร ใช้โมเดล MobileNet บนแพลตฟอร์ม TensorFlow งานวิจัยนี้ใช้ข้อมูลภาพในการ Train ทั้งหมด 1400 ภาพ โดยแบ่งเป็นข้อมูลภาพของมะม่วง 700 ภาพ และ แก้วมังกร 700 ภาพ และใช้การทดสอบแบบสุ่มข้อมูล ใช้ตัวคูณความกว้างเพื่อตรวจสอบขนาดที่เหมาะสมในการเก็บเกี่ยว ผลจากการวิจัยพบว่าการตรวจจับวัตถุนี้มีค่าความถูกต้องที่ 99%

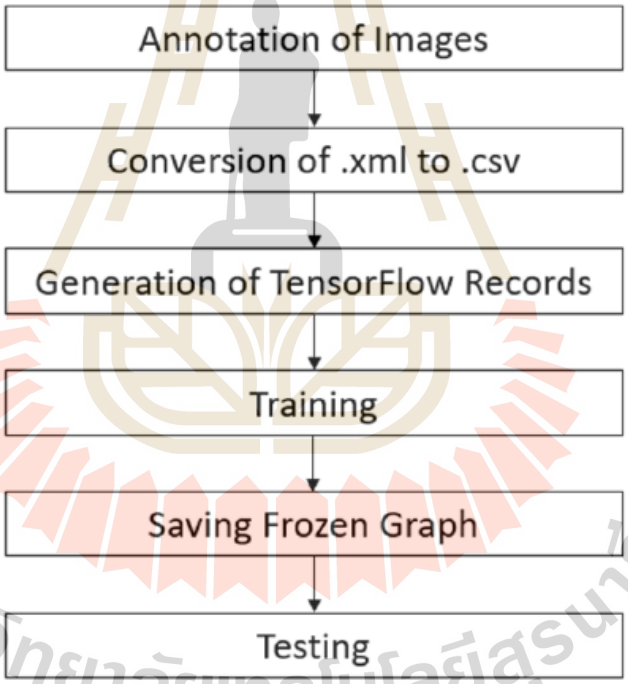


รูปที่ 2.13 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 2

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.8.3 งานวิจัยที่เกี่ยวข้องที่ 3

(Rogelio Ruzcko Tobias (2020)) งานวิจัยนี้ได้กล่าวถึงการทำ Object detection โดยใช้ model จากการทำ Faster R-CNN เพื่อใช้ในการจำแนกเซลล์เม็ดเลือดในร่างกายโดยแบ่งคลาสออกเป็น 5 คลาส คือ 1.RBC 2.Eosinophil 3.Lymphocyte 4.Monocyte 5.Neutrophil ในการวิจัยนี้ใช้ข้อมูลภาพสำหรับการ Train แต่ละคลาส แบ่งเป็นคลาสละ 1000 ภาพ ผลจากการทดสอบพบว่าในงานวิจัยนี้ยังมีค่าความถูกต้องในการตรวจจับวัตถุค่อนข้างน้อย โดยมีความถูกต้องสูงสุดที่ 99% และต่ำสุดที่ 66% ซึ่งสาเหตุอาจเกิดจากเซลล์เม็ดเลือดนั้นมีลักษณะที่ค่อนข้างซับซ้อนจึงจำเป็นต้องใช้ข้อมูลเพื่อทำการ Train มากขึ้นเพื่อให้การตรวจจับมีความแม่นยำมากขึ้น



รูปที่ 2.14 รูปแสดงกระบวนการศึกษาของงานวิจัยที่เกี่ยวของที่ 3

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.9 สรุปผลจากการศึกษา

จากการศึกษาค้นคว้าและทบทวนงานวิจัยที่เกี่ยวข้อง พบว่าในปัจจุบันมีการนำเทคโนโลยีปัญญาประดิษฐ์มาใช้กันอย่างแพร่หลาย ทั้งในด้านอุตสาหกรรม ด้านการแพทย์ หรือแม้แต่ในการใช้ชีวิตประจำวันเป็นต้น แต่ยังไม่มีการนำเทคโนโลยีปัญญาประดิษฐ์นี้มาใช้ในการตรวจจับรถพยาบาลฉุกเฉิน จึงเห็นสมควรทำการศึกษาวิจัยเกี่ยวกับ การนำเทคโนโลยีปัญญาประดิษฐ์นี้มาใช้ในการตรวจจับรถพยาบาลฉุกเฉิน เพื่อแจ้งเตือนผู้ขับขี่รถยนต์ส่วนบุคคล ให้ทราบถึงการมาถึงของรถพยาบาลฉุกเฉิน ซึ่งเป็นการแก้ไขปัญหาการกีดขวางของรถยนต์ส่วนบุคคล โดยการทำการสอนคุณลักษณะของรถพยาบาลในไรบลารี่ TensorFlow และนำโมเดลที่ได้จากการฝึกสอนไปใช้งานร่วมกับบอร์ด Raspberry Pi

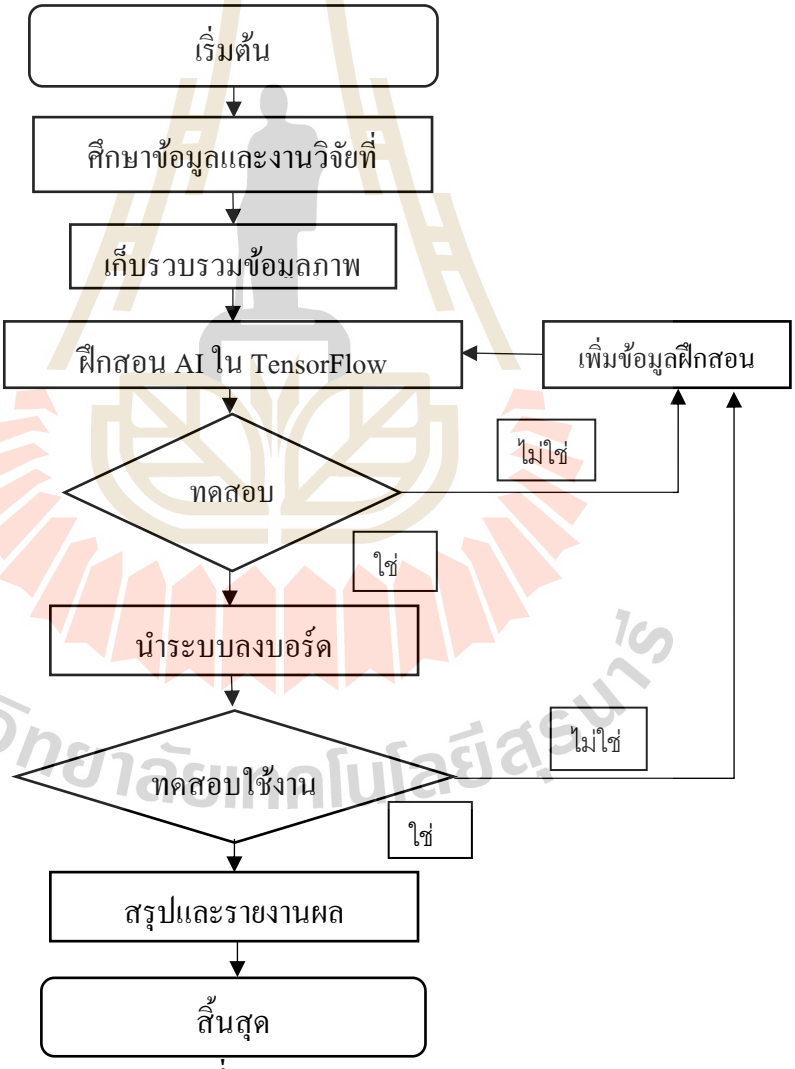
มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 3

ระเบียบวิธีวิจัย

การศึกษาวิจัยในครั้งนี้เป็นการวิจัยเพื่อหาวิธีตรวจจับรถพยาบาลฉุกเฉินด้วยวิธีการเรียนรู้เชิงลึก (Deep learning) โดยการใช้โครงข่ายประสาทเทียม แบบคอนโวลูชัน โดยใช้โมเดลของ Faster R-CNN ซึ่งทำหน้าที่ในการตรวจจับรถพยาบาลจากกล้อง DVR ของรถยนต์ส่วนบุคคล

3.1 วิธีการดำเนินงาน

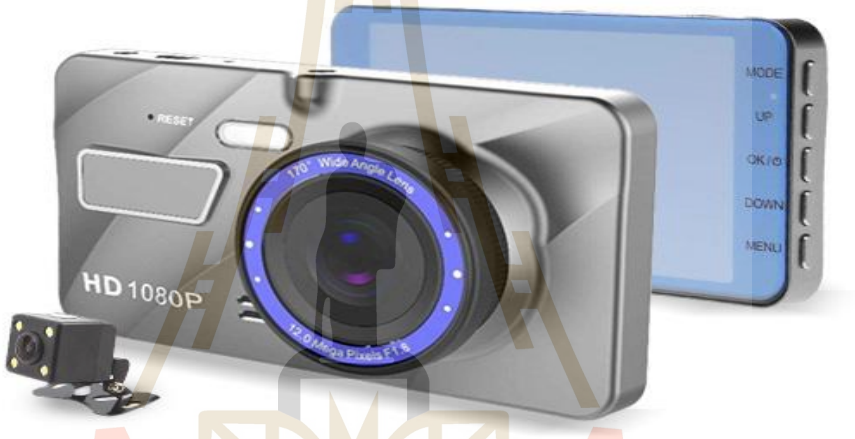


รูปที่ 3.1 วิธีการดำเนินงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.2 วิธีการเก็บรวบรวมข้อมูล (Data Collection)

ข้อมูลอินพุตเป็นไฟล์รูปภาพที่มีการบันทึกจากกล้อง DVR ของรถยนต์ส่วนบุคคล เป็นการเก็บรูปภาพรถพยาบาลฉุกเฉินจากโรงพยาบาลภายในจังหวัดนครราชสีมา ซึ่งเป็นการเก็บภาพรวมของรถพยาบาลฉุกเฉินและภาพแยกของสัญลักษณ์รถพยาบาลแต่ละแบบ เนื่องจากในการรวบรวมรูปรถพยาบาลฉุกเฉินเพื่อมาใช้งานนั้น จำเป็นต้องนำรถไปขวางทางรถพยาบาลซึ่งเป็นเรื่องยากในการรวบรวมข้อมูลของรถพยาบาล ดังนั้นจึงได้ทำการเก็บภาพมาจำนวนทั้งหมด 289 ภาพโดยแบ่งเป็นรูปที่ใช้ในการฝึกสอน 195 รูป และรูปที่ใช้ในการทดสอบ 94 รูป



รูปที่ 3.2 รูปตัวอย่างกล้อง DVR ที่ใช้ในการเก็บข้ออมูลภาพ

(ที่มา: https://www.yamaha-motor.co.th/privilege/best-deal/273165)

3.3 การฝึกสอน AI ใน TensorFlow

3.3.1 การเตรียมและแบ่งข้อมูล (Data Preparation)

การตรวจจับรถพยาบาลนั้นได้ทำการแบ่งคุณลักษณะที่ใช้ในการตรวจจับออกเป็น 4 แบบ คือ

1. Star of life symbol
2. Red Cross symbol
3. Text Ambulance
4. Ambulance light

รูปที่ 3.3 สัญลักษณ์ Star of life

(ที่มา: https://www.thaipng.com/png-lf1zw2/)

รูปที่ 3.4 สัญลักษณ์ Red cross

(ที่มา: https://www.pngegg.com/th/search?q=American+Red+Cross)

รูปที่ 3.5 Text Ambulance

(ที่มา: http://th.huangheindustry.com/other-vehicles/57691967.html)

รูปที่ 3.6 ไฟรถพยาบาลฉุกเฉิน

(ที่มา: http://th.huangheindustry.com/other-vehicles/57691967.html)

3.3.2 การสร้างแบบจำลอง (Model Training)

ประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) ซึ่งใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันโดยใช้เทคนิค Faster R-CNN ในการฝึกสอน

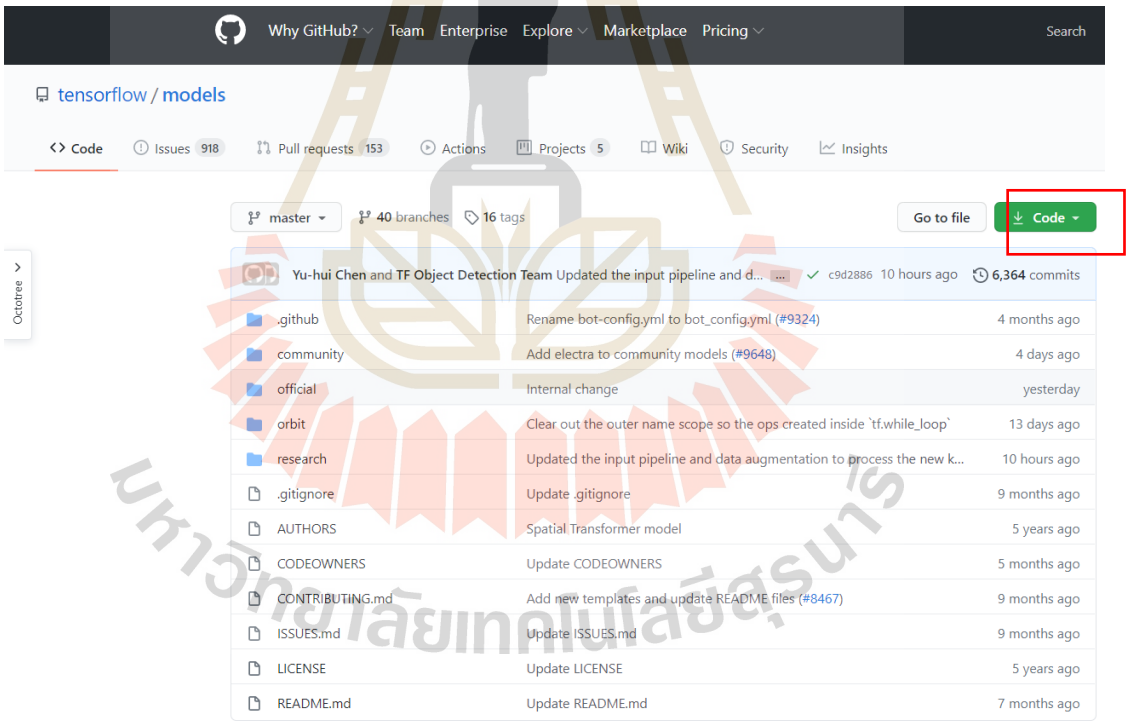
3.2.2.1 โปรแกรมที่จำเป็นสำหรับการสร้างแบบจำลอง

1. TensorFlow-GPU

2. Anaconda Python

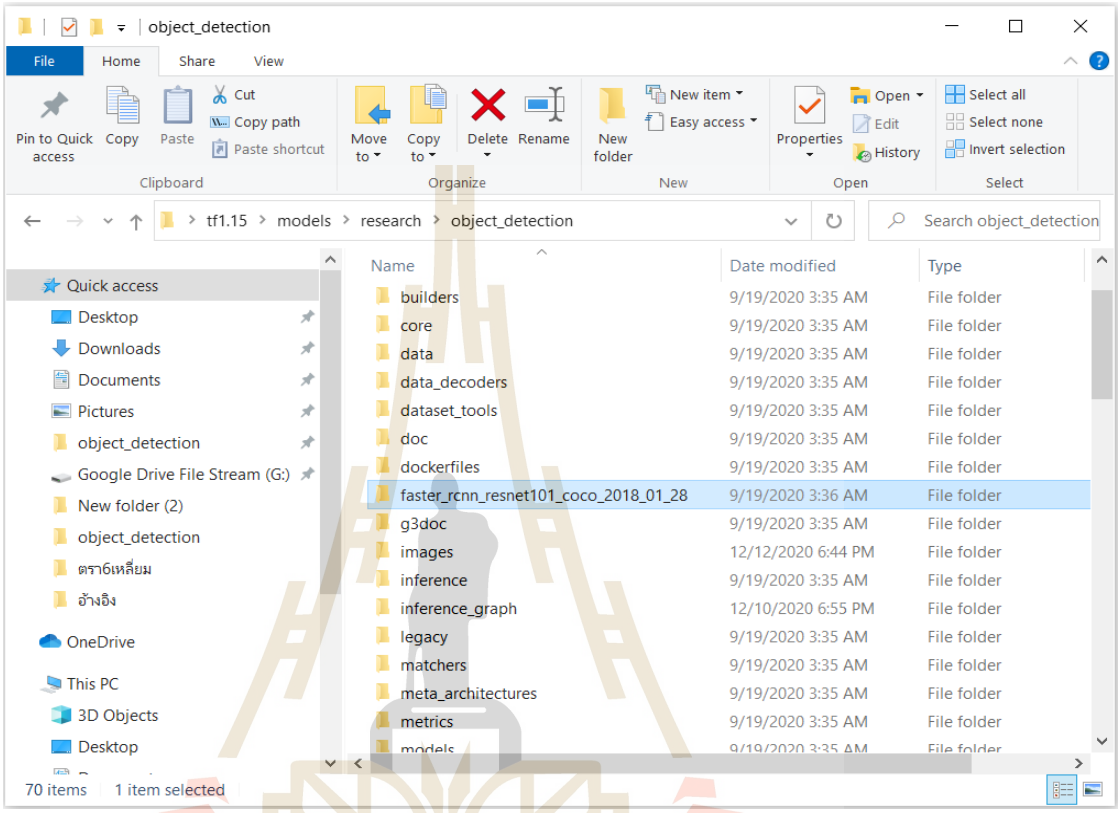
3.2.2.2 การทำ Object Detection

1. ทำการดาวน์โหลดและติดตั้ง TensorFlow model และทำการกดที่คำว่า Code ทำการแตกไฟล์จะได้ folder models-master ทำการเปลี่ยนชื่อเป็น tf1.15



รูปที่ 3.7 รูปแสดงการดาวโหลด folder model

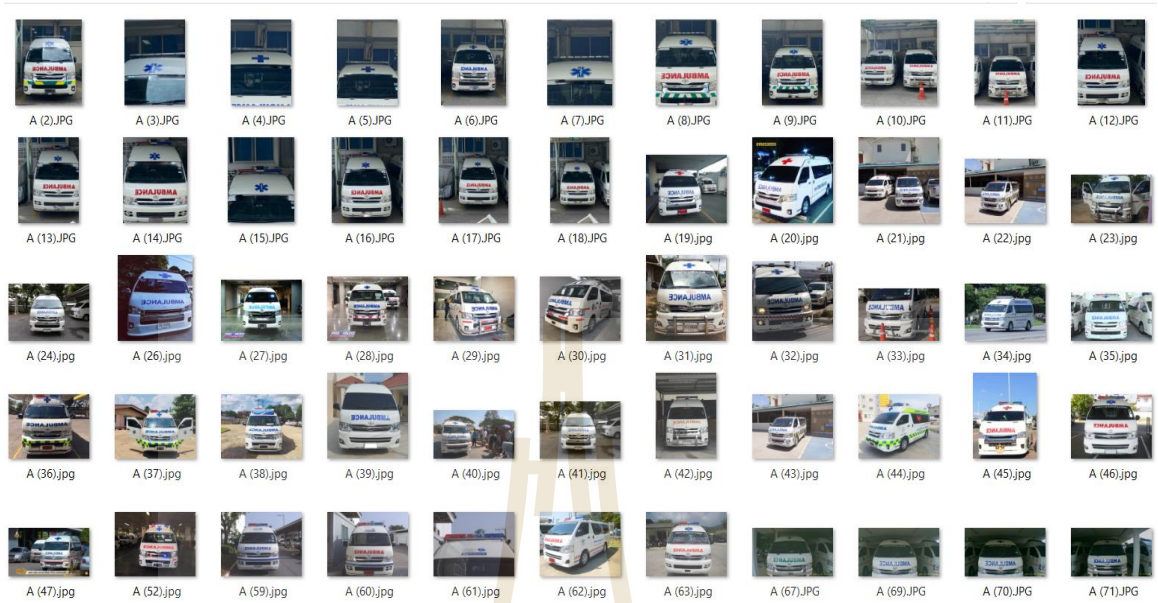
2. ทำการโหลด model ที่จะใช้ในการ train โดยในงานวิจัยนี้จะใช้ faster_rcnn_resnet101 ในการ train ทำการแตกไฟล์ไว้ที่ folder tf1.15\models\research\object_detection



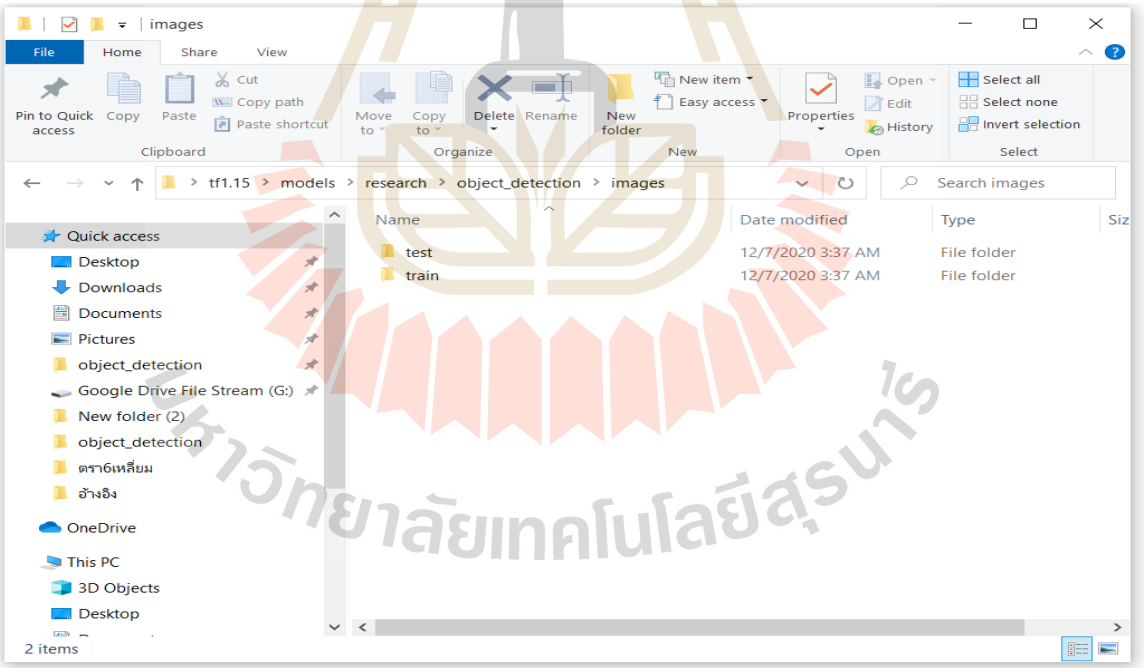
รูปที่ 3.8 รูปแสดง folder model ที่ใช้สำหรับการฝึกสอน

3. การรวบรวมรูปภาพ

TensorFlow ต้องการภาพของวัตถุหลายร้อยภาพเพื่อฝึกตัวจำแนกการตรวจจับที่ดี ในภาพควรมีวัตถุที่ต้องการและควรมีพื้นหลังและสภาพแสงที่หลากหลาย รูปในการทำโมเดล หาจากการลงพื้นที่เก็บข้อมูลจริงและอีกส่วนหนึ่งหาจากเว็ปไซต์ เก็บรูปที่ต้องการ train และ test ลงใน folder tf1.15\models\research\object_detection\images อัตราส่วน train:test ประมาณ 75:25 เนื่องจากในการทำการฝึกสอนต้องใช้รูปใน train ให้มากที่สุดส่วนตัว test เป็นเพียงข้อมูลที่ใช้ในการทดสอบจึงได้กำหนดอัตราส่วนดังนี้



รูปที่ 3.9 ตัวอย่างการเตรียมภาพที่ใช้ในการฝึกสอน



รูปที่ 3.10 รูป folder train และ test

มหาวิทยาลัยเทคโนโลยีสุรนารี

4. การติดฉลาก

ทำการโหลดโปรแกรม labelimg สำหรับติดฉลาก เมื่อดาวโหลดโปรแกรมสำเร็จต้องทำการแตกไฟล์และเปิดโปรแกรม labelimg เลือก open dir เพื่อเลือกที่อยู่ของภาพ จากนั้นเลือกที่ create rectbox เพื่อทำการตีกรอบสิ่งที่ต้องการ detect และทำการระบุคลาส กด Save จะได้ไฟล์ .xml ทำให้ครบทุกภาพทั้ง train และ test จำนวน Class ที่ใช้จะถูกกำหนดด้วยคุณสมบัติของรถพยาบาล (ได้แก่ Star of life symbol, Red Cross symbol, Text Ambulance, Ambulance light) รูปแบบและพารามิเตอร์ที่จะใช้ในการฝึกอบรมกำหนดโดยขั้นตอนนี้

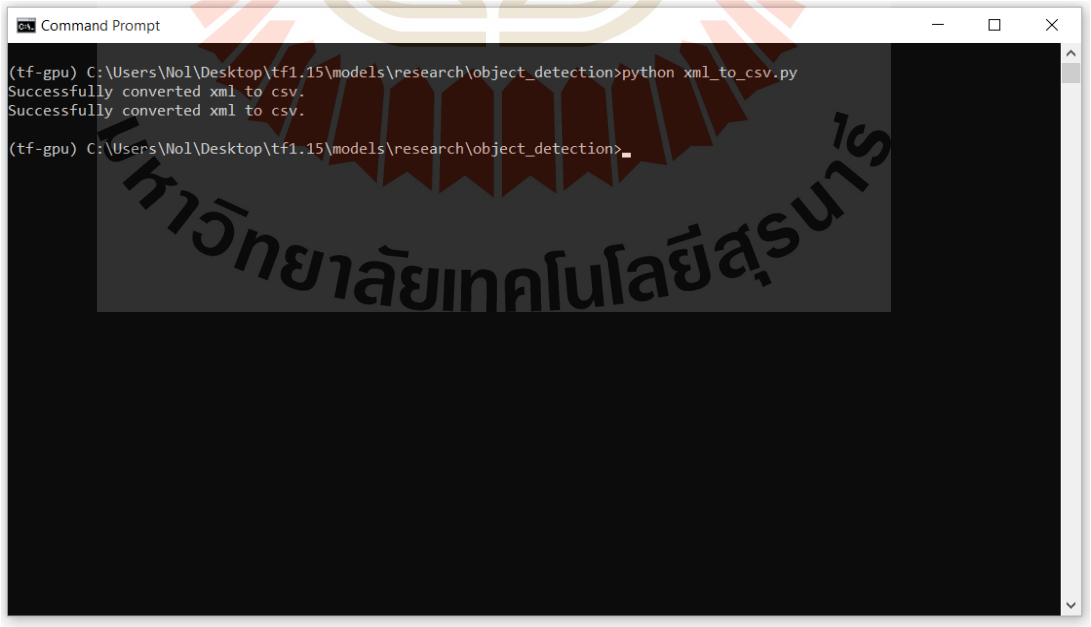
รูปที่ 3.11 รูปแสดงการติดฉลากให้กับรูปภาพ

ชื่อสัญลักษณ์	ชื่อ Label
Star of life symbol	Logo1
Red Cross symbol	Logo2
Text Ambulance	Ambulance
Ambulance light	emergency vehicles

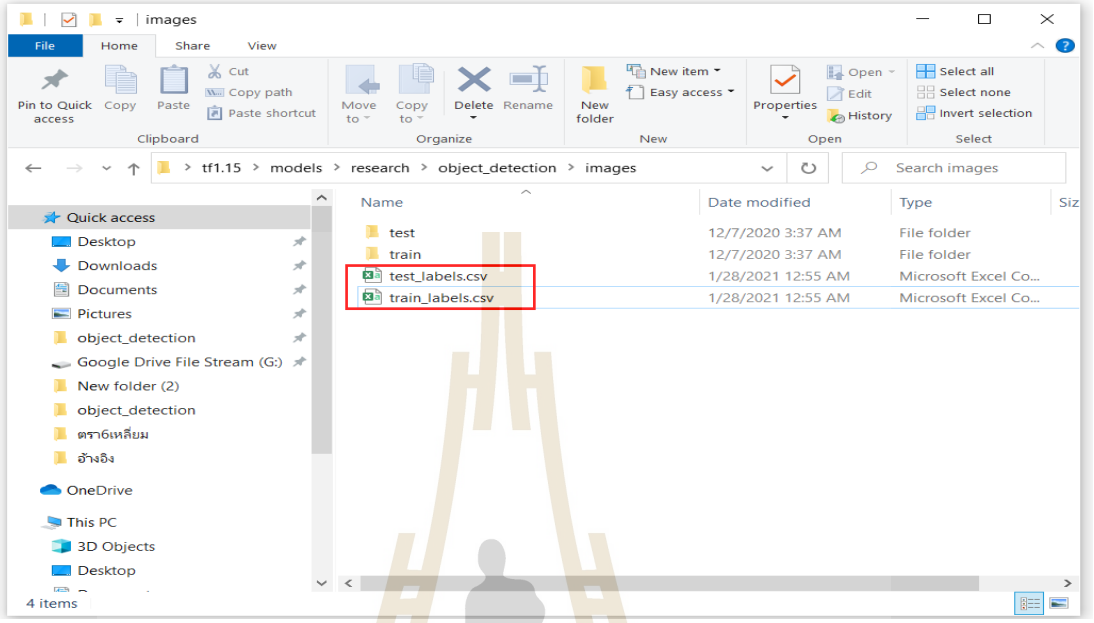
ตารางที่ 3.1 ตารางแสดงชื่อฉลากที่ใช้ในงานวิจัย

5. สร้างไฟล์สำหรับ training

สร้างไฟล์ .csv สำหรับใช้ในการ train จากคำสั่ง python xml_to_csv.py เพื่อให้ได้ไฟล์ test_labels.csv และ train_labels.csv

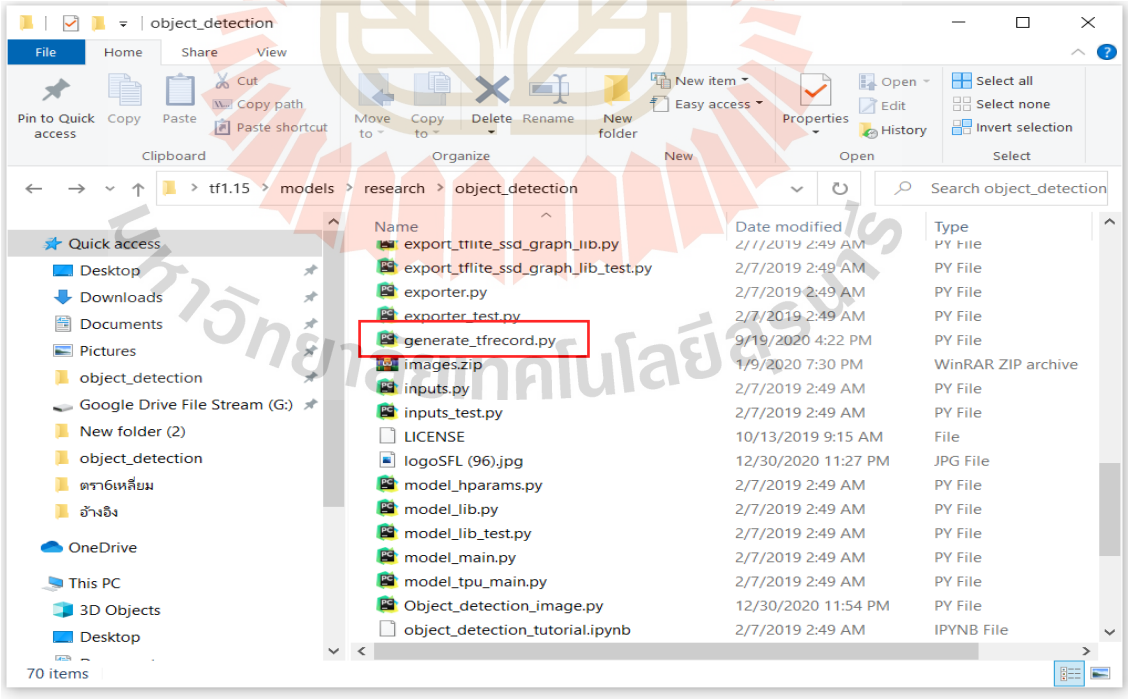


รูปที่ 3.12 รูปแสดงการใช้คำสั่งเพื่อสร้างไฟล์ train และ test labels

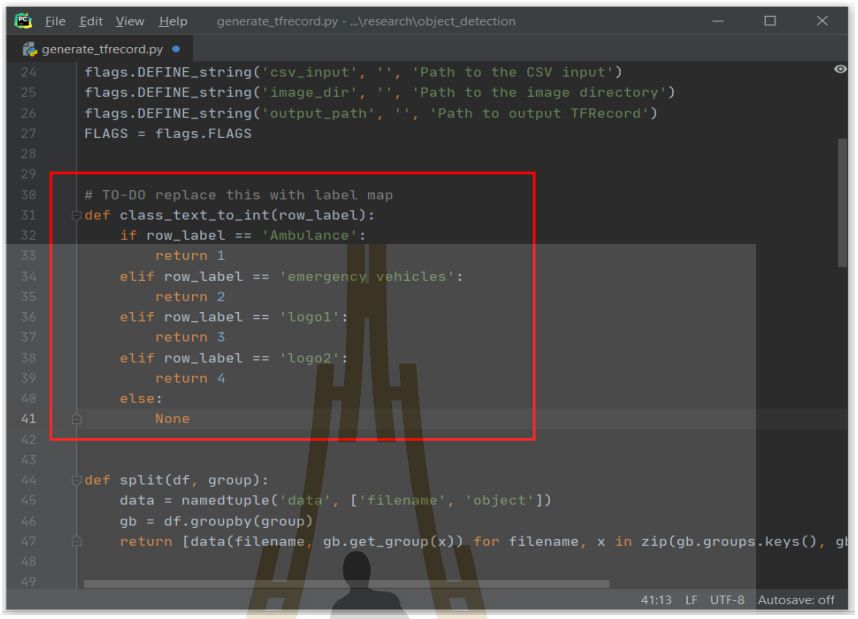


รูปที่ 3.13 รูปแสดงไฟล์ train และ test labels

แก้ไขไฟล์ generate_tfrecord.py ให้ตรงตาม class ที่จะทำการฝึกสอน

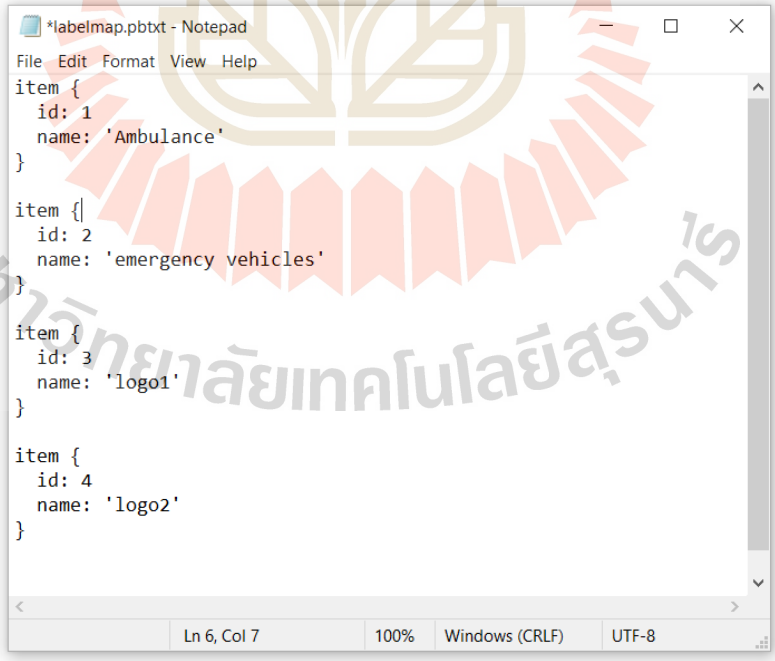


รูปที่ 3.14 รูปแสดงไฟล์ generate_tfrecord.py



รูปที่ 3.15 รูปแสดงการกำหนดฉลากในไฟล์ generate_tfrecord.py

ใช้ text editor สร้างไฟล์ labelmap.pbtxt ใส่ class ที่เราได้ทำการแบ่งไว้เข้าไป แล้วทำการ save ลงในfolder tf1.15\models\research\object_detection\training

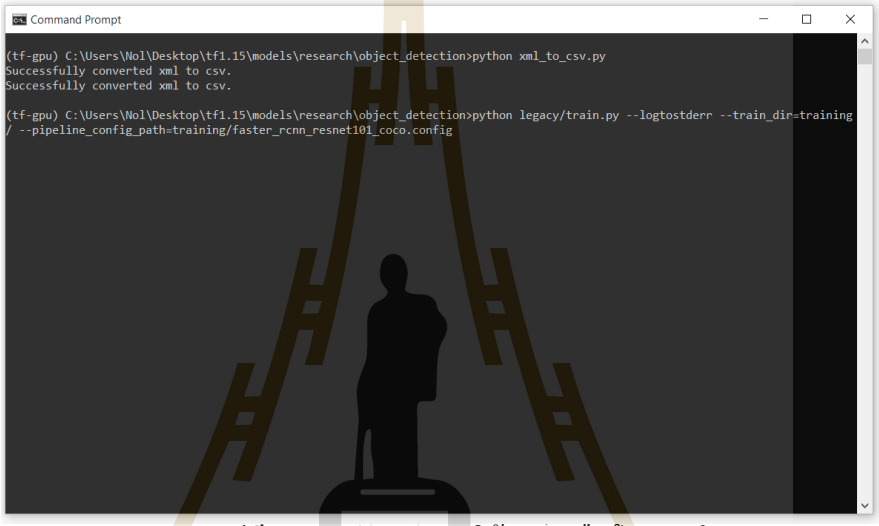


รูปที่ 3.16 รูปแสดงการสร้างไฟล์ labelmap.pbtxt

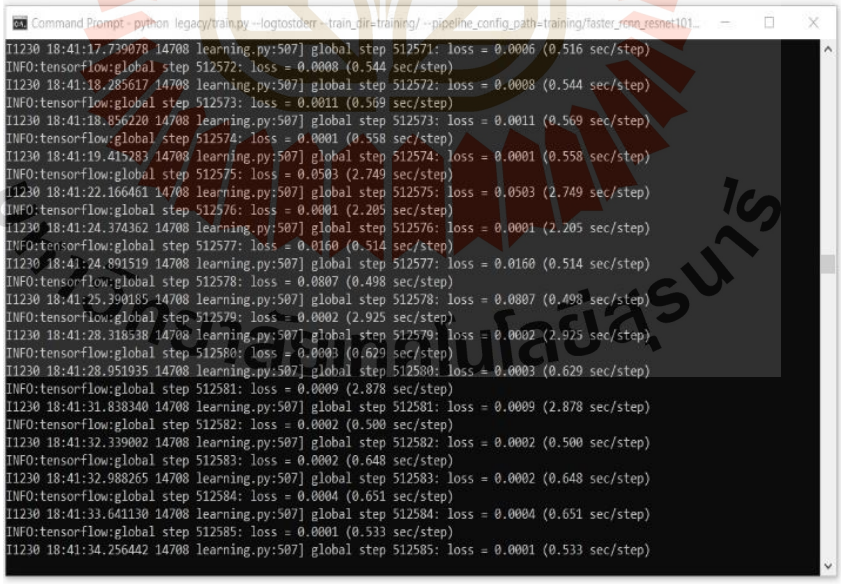
มหาวิทยาลัยเทคโนโลยีสุรนารี

6. ทำการ train model

เริ่มทำการ Training model บน CMD โดยใช้คำสั่ง python legacy/train.py logtostderr train_dir=training/ pipeline_config_path=training/faster_rcnn_resnet101_coco.config ปล่อยโมเดล run จนถึง loss ที่กำหนดหรือ step ที่ตั้งค่า และกด Ctrl+C เพื่อหยุดการทำงาน



รูปที่ 3.17 รูปแสดงการใช้คำสั่งเพื่อฝึกสอนโมเดล



รูปที่ 3.18 รูปแสดงการฝึกสอนโมเดล

7. การประเมินผล (Model Evaluation)

การประเมินผลในการตรวจจับนี้ได้ใช้ทฤษฎี Intersection over Union (IoU) ในการหาค่าความถูกต้องของการตรวจจับ โดยได้กำหนดค่าไว้ดังนี้

IoU >= 0.7, แสดงกรอบผลเฉลยพร้อมระบุคลาส

IoU < 0.7, ไม่แสดงกรอบผลเฉลยและไม่ระบุคลาส

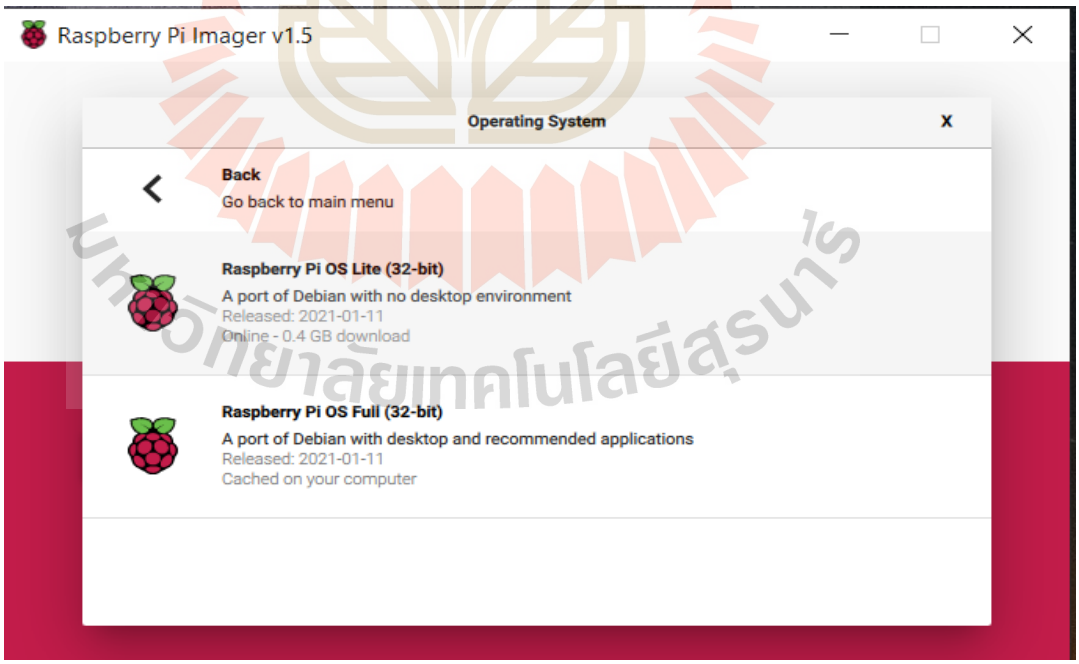
3.4 การนำข้อมูลที่ฝึกสอนไปใช้กับ Raspberry Pi

3.4.1 ซอร์ฟแวร์ที่จำเป็นในการใช้งานโมเดล Object detection บน Raspberry Pi

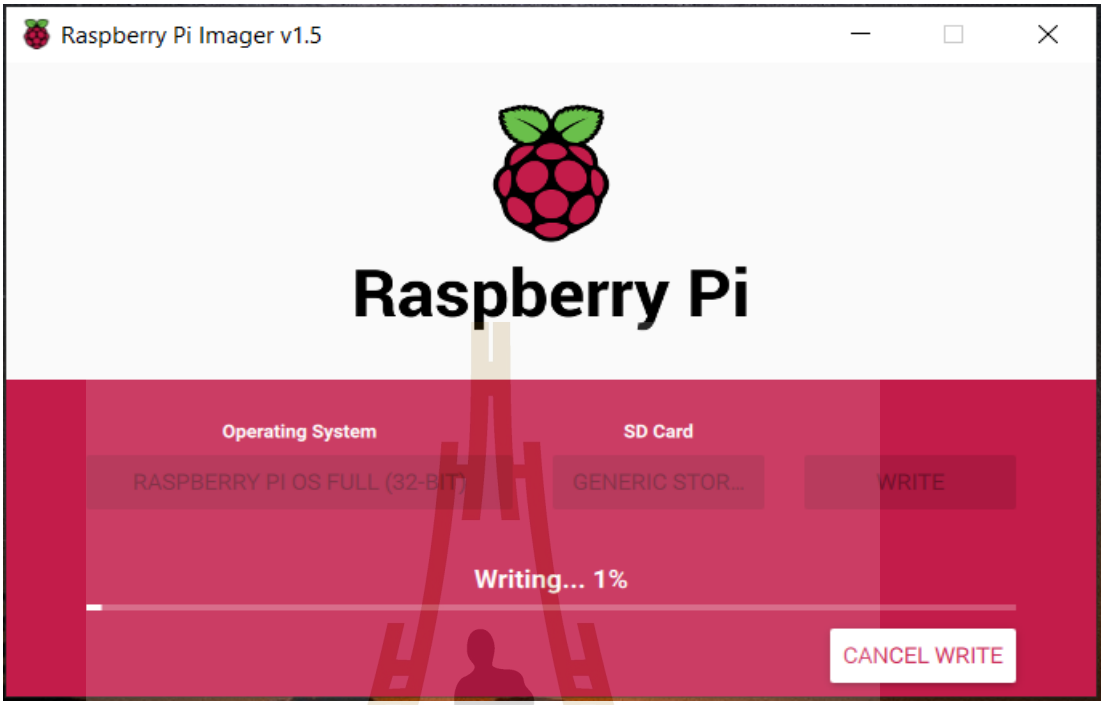
ในการทำ Object detection บน Raspberry Pi นั้นจำเป็นต้องมีซอร์ฟแวร์ที่ช่วยในการทำงานโดยมีขั้นตอนการติดตั้งดังนี้

3.4.1.1 การติดตั้ง Raspbian OS

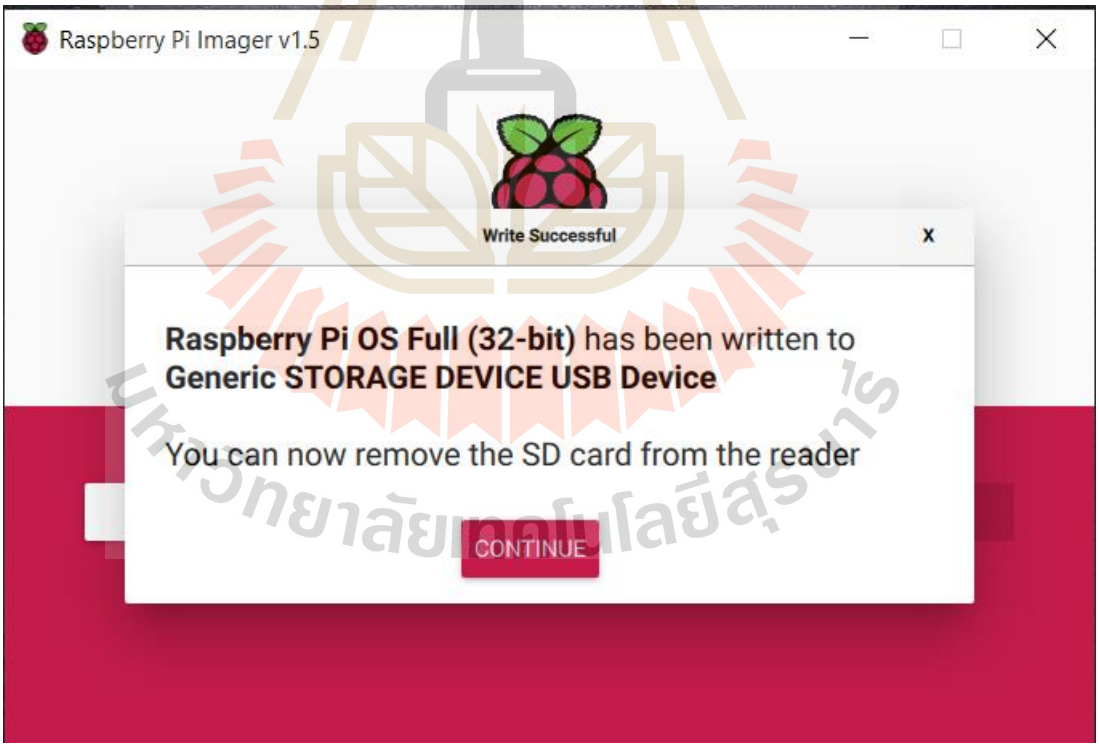
ทำการติดตั้ง Raspbian OS ลงใน SD card โดยเลือก Raspbian OS ตัว FULL เวอร์ชั่น แสดงดังรูปที่ 3.18-3.20



รูปที่ 3.19 รูปแสดงการเลือก Raspberry Pi OS



รูปที่ 3.20 รูปแสดงการติดตั้ง Raspberry Pi OS ลงใน SD card



รูปที่ 3.21 รูปแสดงการติดตั้ง Raspberry Pi OS สำเร็จ

3.4.1.2 อัปเดต Raspberry Pi

ใช้คำสั่งเพื่ออัปเดต Raspberry Pi ด้วยคำสั่งดังต่อไปนี้

- sudo apt-get update
- sudo apt-get upgrade

3.4.1.3 ติดตั้งซอร์ฟแวร์ที่จำเป็นอื่นๆ

ขั้นตอนนี้เป็นขั้นตอนการติดตั้งซอร์ฟแวร์ที่จำเป็นโดยใช้คำสั่งดังต่อไปนี้

- sudo apt-get install build-essential cmake pkg-config
- sudo apt-get install libjpeg-dev libtiff-dev libtiff5-dev libpng12-dev libjasper-dev
- sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
- sudo apt-get install libxvidcore-dev libx264-dev
- sudo apt-get install libgtk2.0-dev libgtk-3-dev
- sudo apt-get install libcanberra-gtk*
- sudo apt-get install gfortran libatlas-base-dev
- sudo apt-get install python-dev python3-dev
- sudo apt-get install python-tk

3.4.1.4 ขั้นตอนการกำหนดค่าสภาพแวดล้อมเสมือน Python3 สำหรับ TensorFlow และ OpenCV

ขั้นตอนนี้เป็นการติดตั้งและตั้งค่าสำหรับสภาพแวดล้อมเสมือนจริงเพื่อช่วยป้องกันการสับสนของการเรียกใช้งานฟังก์ชันต่างๆ ให้ดำเนินการตามคำสั่งต่อไปนี้

- sudo pip3 install virtualenv virtualenvwrapper
- exportWORKON_HOME=$HOME/.virtualenvs
- exportVIRTUALENVWRAPPER_PYTHON=/usr/bin/python3source/usr/local/bin/virtualenvwrapper.sh
- echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile

- echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
- echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.profile
- echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile

ต่อมาเป็นการกำหนดแหล่งไฟล์และสร้างสภาพแวดล้อมเสมือนจริง โดยในการทำในครั้งนี้ใช้ชื่อสภาพแวดล้อมเสมือนจริงว่า “tfcv”

- source ~/.profilemkvirtualenv tfcv -p python3

ใช้คำสั่งดังต่อไปนี้เพื่อเปิดใช้งานสภาพแวดล้อมที่ได้สร้างขึ้น

- workon tfcv

3.4.1.5 ติดตั้ง TensorFlow และแพ็คเกจพื้นฐาน

TensorFlow คือ Open-source Machine learning library พัฒนาโดย Google มีสถาปัตยกรรม 3 ส่วน คือ การเตรียมประมวลผลข้อมูล, การสร้างแบบจำลอง, ฝึกและประเมินแบบจำลองเมื่อเสร็จสิ้นกระบวนการเหล่านี้สามารถนำไปใช้งานได้หลายแพลตฟอร์ม เช่น คอมพิวเตอรระบบปฏิบัติการ Windows, macOS or Linux, คลาวด์หรือเว็บเซอร์วิซ, มือถือทั้ง iOS and Android เป็นต้น

ขั้นตอนนี้เป็นการติดตั้งไลบรารีที่จำเป็นสำหรับการใช้งาน TensorFlow มีคำสั่งดังนี้

- pip3 install Twisted
- pip3 install scrapy
- pip3 install pillow
- pip3 install lxml
- pip3 install cython
- pip3 install numpy
- pip3 install matplotlib
- pip3 install grpcio
- pip3 install h5py

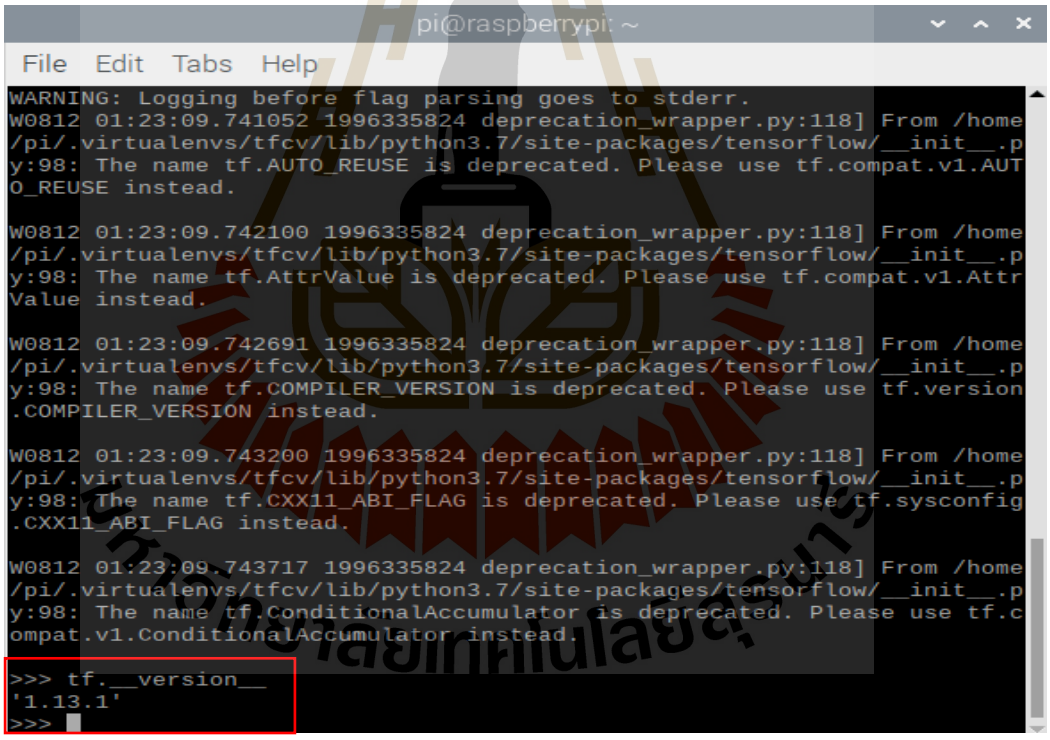
มหาวิทยาลัยเทคโนโลยีสุรนารี

ต่อมาจะเป็นขั้นตอนการติดตั้ง TensorFlow

- pip3 install tensorflow==1.13.1

เมื่อติดตั้งเสร็จแล้วให้ทำการทดสอบโดยการเรียกใช้งาน Python3 จากนั้นทำการเรียกใช้งานแพคเกจ TensorFlow ตามคำสั่งต่อไปนี้

- workon tfcv
- python3
- import tensorflow as tf
- tf.__version__



รูปที่ 3.22 รูปแสดงการใช้คำสั่งเช็คเวอร์ชั่นของโปรแกรม TensorFlow

3.4.1.6 Install OpenCV 4.0.0

OpenCV (Open Source Computer Vision) เป็นห้องสมุดของฟังก์ชั่นการเขียนโปรแกรมส่วนใหญ่มุ่งเป้าไปที่การมองเห็นคอมพิวเตอร์แบบเรียลไทม์ ถูกปล่อยภายใต้ใบอนุญาต BSD และเป็นแพตฟอร์มที่ใช้งานฟรีสำหรับการใช้งานเชิงวิชาการและเชิงพาณิชย์ มีอินเตอร์เฟซ C ++, Python และ Java และรองรับ Windows, Linux, Mac OS, iOS และ Android OpenCV ได้รับการออกแบบมาเพื่อประสิทธิภาพการคำนวณและให้ความสำคัญกับการใช้งานแบบเรียลไทม์

ดาวน์โหลดไฟล์

- wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.0.zip
- wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.0.0.zip

แตกไฟล์ที่ดาวน์โหลดมา

- unzip opencv.zip
- unzip opencv_contrib.zip

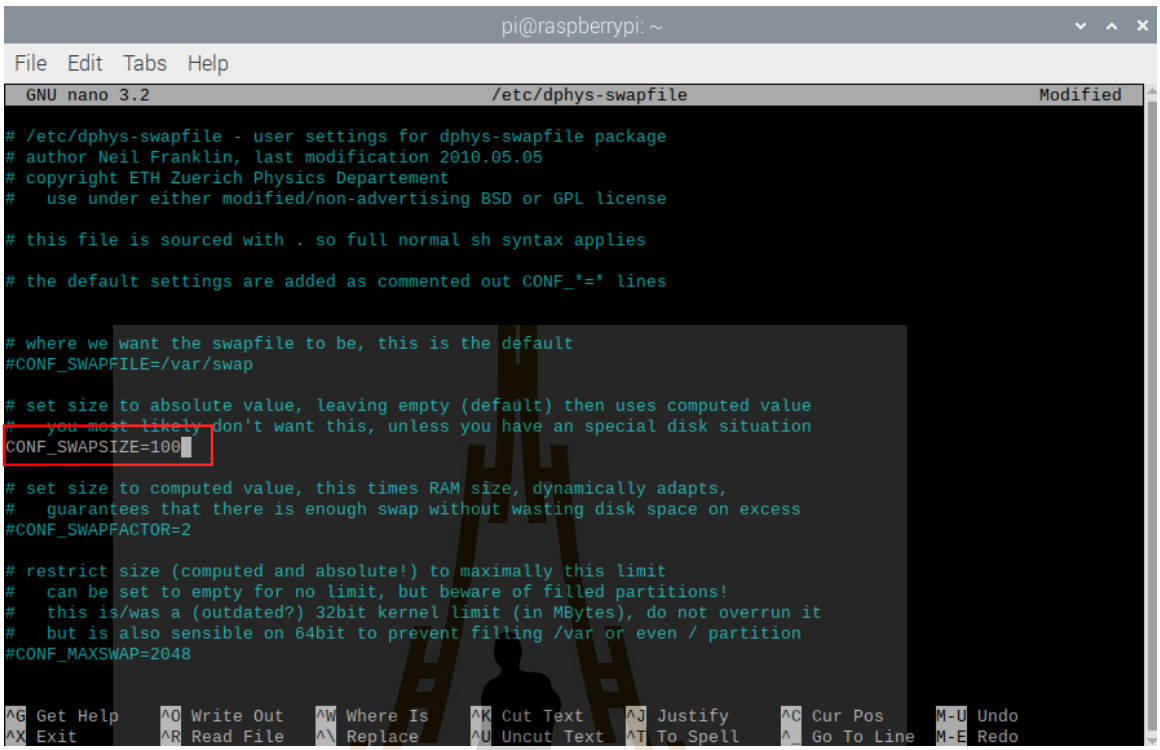
ลบไฟล์ที่ดาวน์โหลดมาเพื่อเพิ่มพื้นที่ให้หน่วยความจำ

- rm opencv.zip opencv_contrib.zip

ต่อมาให้ตั้งค่าเพิ่มพื้นที่การแลกเปลี่ยนสำหรับการคอมไพล์กับคอร์ทั้งสี่ของ Raspberry Pi โดยให้เปิดไฟล์ /etc/dphys-swapfile และทำการแก้ไขตัวแปร “ CONF_SWAPSIZE = 100” ให้เป็น “CONF_SWAPSIZE = 1024“

- sudo nano /etc/dphys-swapfile

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.23 รูปแสดงการตั้งค่าเพิ่มพื้นที่การแลกเปลี่ยนสำหรับการคอมไพล์กับคอร์ทั้งสี่ของ Raspberry Pi

จากนั้นให้ใช้คำสั่งต่อไปนี้ เพื่อเริ่มการทำงาน

- sudo /etc/init.d/dphys-swapfile stop
- sudo /etc/init.d/dphys-swapfile start

ขั้นตอนนี้เป็นการเซ็ตคอมไพล์ด้วย CMake

- cd opencv-4.0.0
- mkdir build
- cd buildcmake -D CMAKE_BUILD_TYPE=RELEASE \
  -D CMAKE_INSTALL_PREFIX=/usr/local \
  -D BUILD_opencv_java=OFF \
  -D BUILD_opencv_python2=OFF \
  -D BUILD_opencv_python3=ON \
  -D PYTHON_DEFAULT_EXECUTABLE=$(which python3) \

มหาวิทยาลัยเทคโนโลยีสุรนารี

-D INSTALL_C_EXAMPLES=OFF \

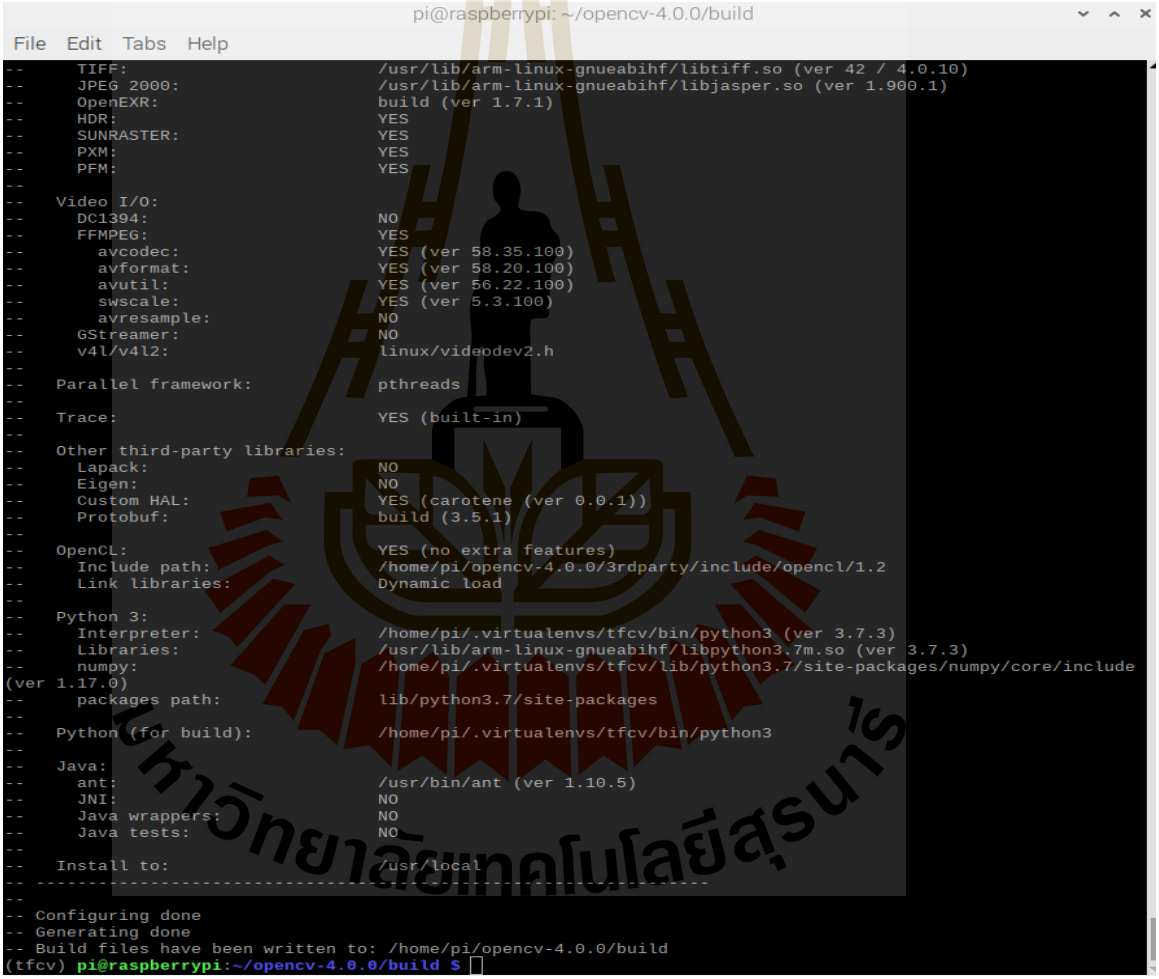
-D INSTALL_PYTHON_EXAMPLES=ON \

-D BUILD_EXAMPLES=ON\

-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.0.0/modules \

-D BUILD_TESTS=OFF \

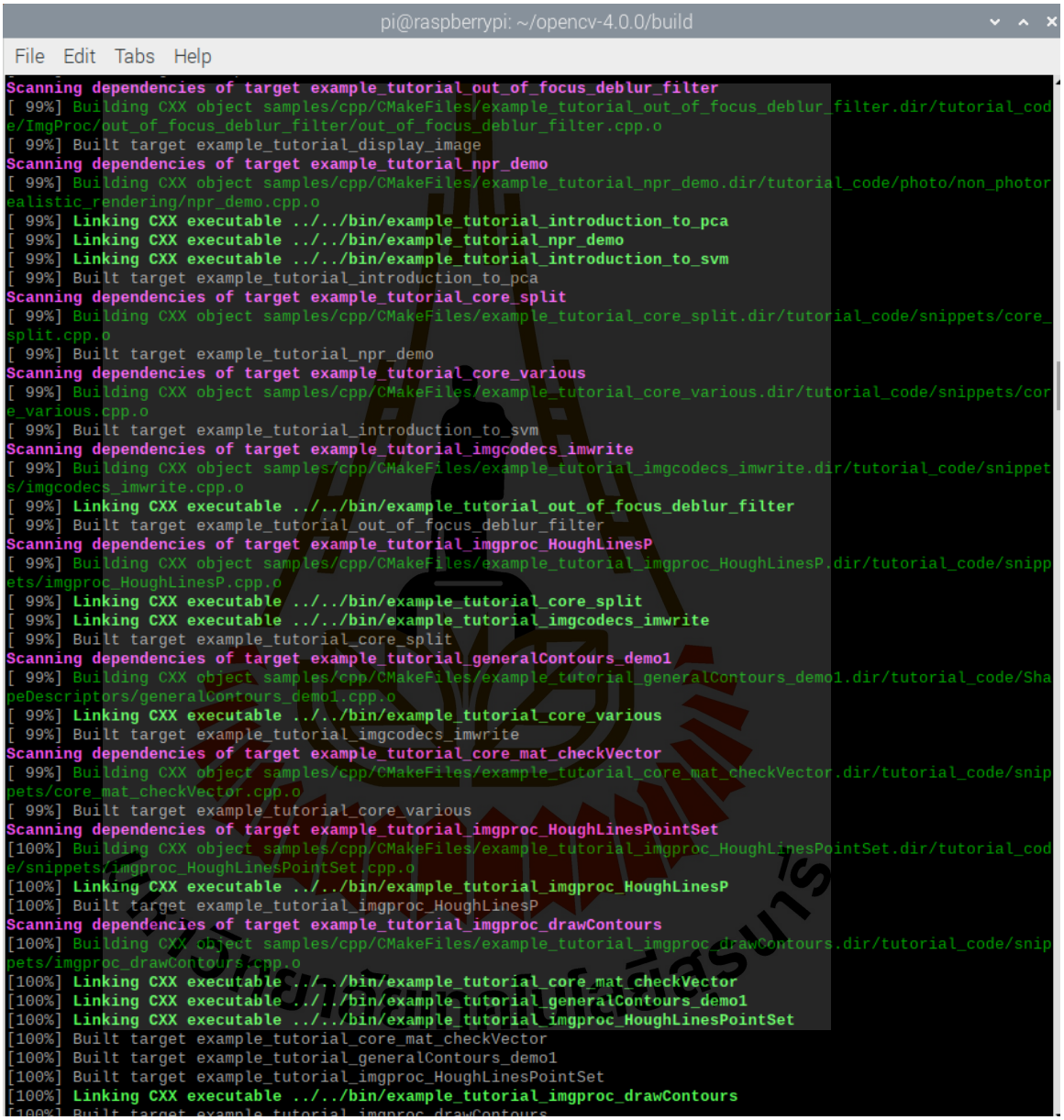
-D BUILD_PERF_TESTS= OFF ..



รูปที่ 3.24 รูปแสดงการตั้งค่าของ OpenCV

เริ่มทำการคอมไพล์ OpenCV ด้วยการ

- make -j4



รูปที่ 3.25 รูปแสดงการคอมไพล์ของ OpenCV

เริ่มทำการติดตั้ง OpenCV ใช้คำสั่งต่อไปนี้

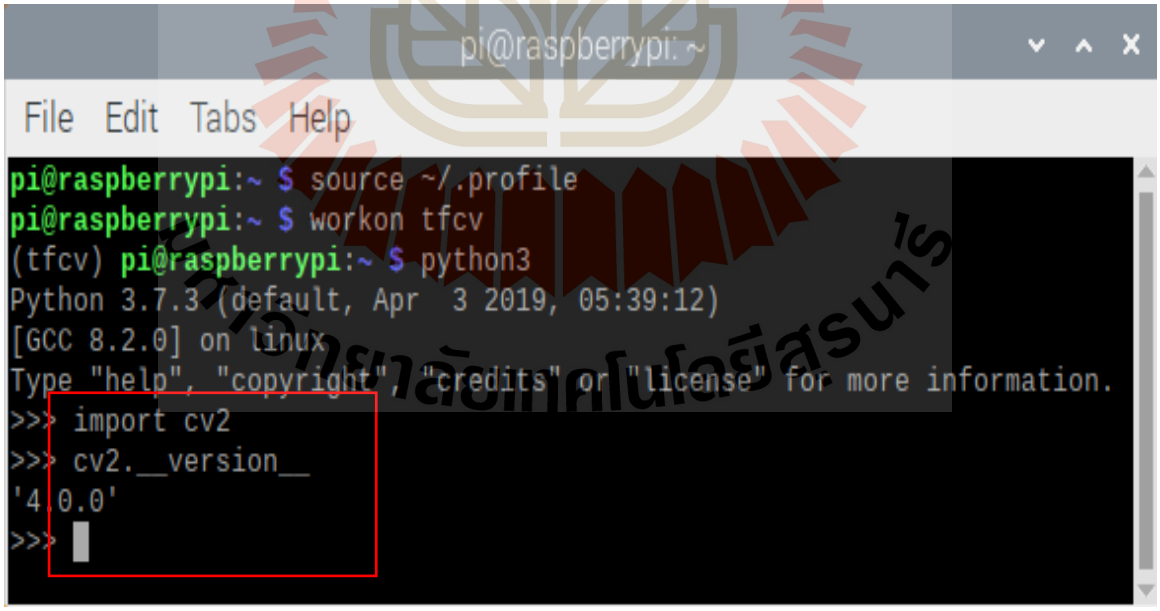
- sudo make install
- sudo ldconfig

เมื่อติดตั้งเสร็จแล้วให้ทำการเชื่อมโยง Opencv ไปยังสภาพแวดล้อมเสมือนที่สร้างขึ้นโดยใช้คำสั่งดังนี้

- cd ~/.virtualenvs/tfcv/lib/python3.7/site-packages/ln -s /usr/local/python/cv2/python-3.7/cv2.cpython-37m-arm-linux-gnueabihf.so cv2.so

เมื่อติดตั้งเสร็จแล้วให้ทำการทดสอบโดยการเรียกใช้งาน Python3 แล้วทำการเรียกใช้งานแพคเกจ OpenCV ตามคำสั่งต่อไปนี้

- workon tfcv
- python3
- import cv2
- cv2.__ version__



รูปที่ 3.26 รูปแสดงการใช้คำสั่งเพื่อตรวจสอบเวอร์ชั่นของ OpenCV

3.4.1.7 ติดตั้ง Protobuf

ทำการติดตั้ง Protobuf โดยใช้คำสั่งดังต่อไปนี้

- sudo apt-get install autoconf
- sudo apt-get install automake
- sudo apt-get install libtool
- sudo apt-get install curl
- wget https://github.com/google/protobuf/releases/download/v3.5.1/protobuf-all-3.5.1.tar.gz
- tar -zxvf protobuf-all-3.5.1.tar.gz
- cd protobuf-3.5.1
- ./configure
- Make
- make check
- sudo make install
- cd python
- export LD_LIBRARY_PATH=../src/.libs
- python3 setup.py build --cpp_implementation
- python3 setup.py test --cpp_implementation
- sudo python3 setup.py install --cpp_implementation
- export PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION=cpp
- export PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION=3
- sudo ldconfig
- protoc
- sudo reboot now

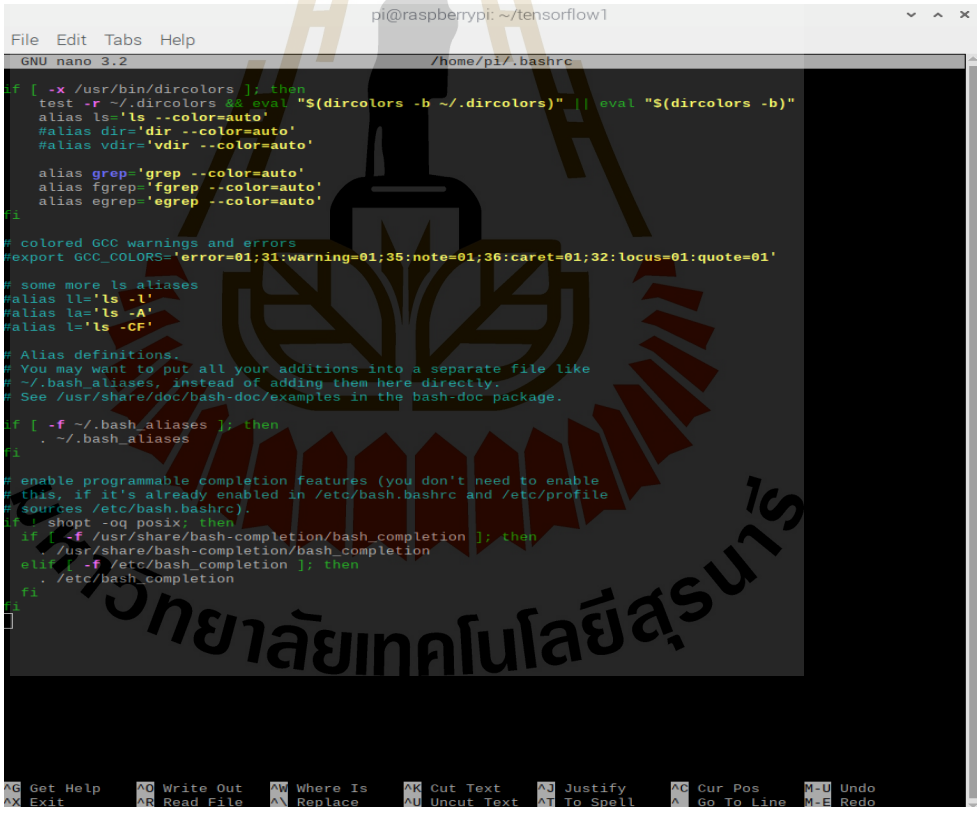
3.4.1.8 ตั้งค่า API การตรวจจับวัตถุ Tensorflow บน Raspberry Pi

ขั้นตอนนี้เป็นการตั้งติด Model Object Detection API โดยสร้างไดเรกทอรี่ใช้ชื่อ “tensorflow1” มีคำสั่งดังต่อไปนี้

- mkdir tensorflow1
- cd tensorflow1
- git clone --recurse-submodules https://github.com/tensorflow/models.git

ต่อไปเป็นการแก้ไขสภาพแวดล้อม PYTHONPATH ให้ชี้ไปที่ไดเรกทอรี่ที่เก็บ ข้อมูล TensorFlow ที่ได้ทำการดาวน์โหลดมา โดยใช้คำสั่งแก้ไขไฟล์. bashrc ดังนี้

- sudo nano ~/.bashrc



รูปที่ 3.27 รูปแสดงการใช้คำสั่งเพื่อแก้ไขสภาพแวดล้อม PYTHONPATH

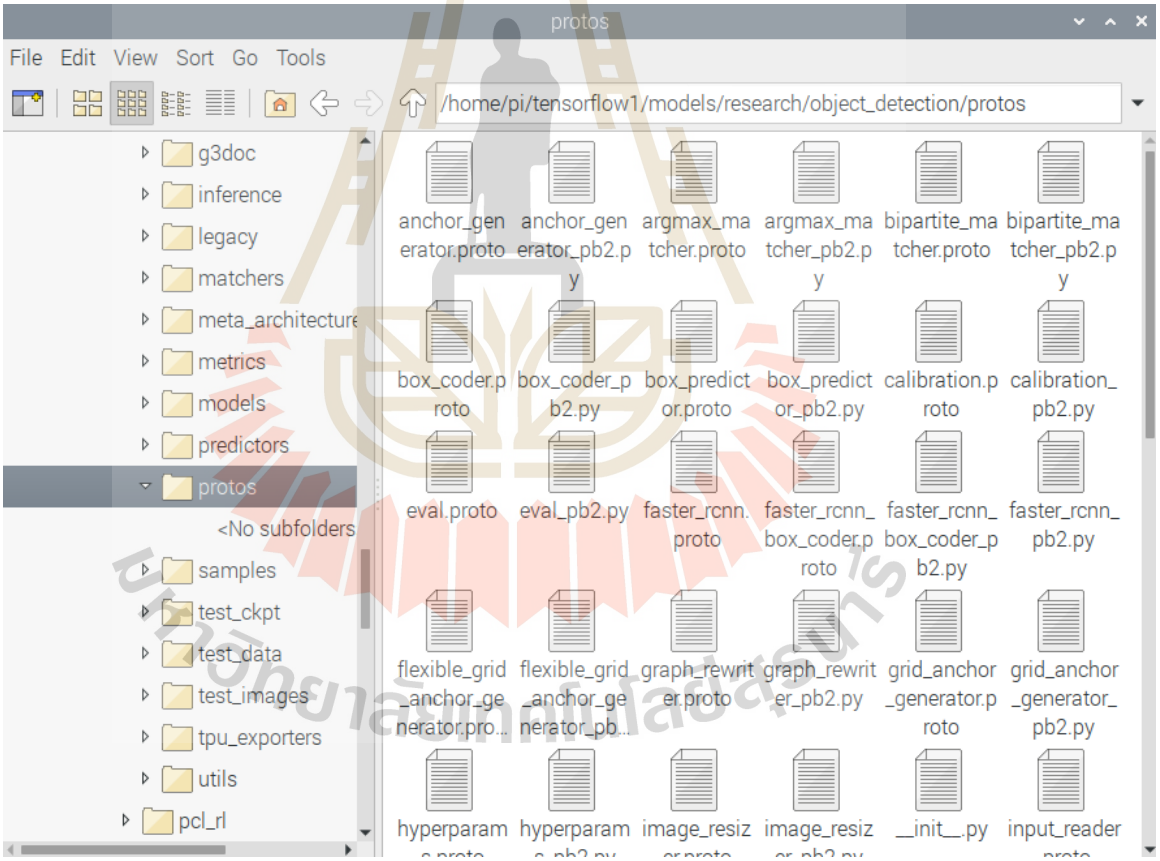
ให้ทำการตั้งค่า “PYTHONPATH” โดยเพิ่มคำสั่งดังต่อไปนี้

- export PYTHONPATH=$PYTHONPATH:/home/pi/tensorflow1/models/research:/home/pi/tensorflow1/models/research/slim

เมื่อเสร็จให้ save โดยกด ctrl+x และเลือก yes โดยกด y ตามด้วย Enter

ต่อมาให้ใช้คำสั่งต่อไปนี้เพื่อแปลงไฟล์ “name” .proto ทั้งหมดให้เป็น “name_pb2” .py

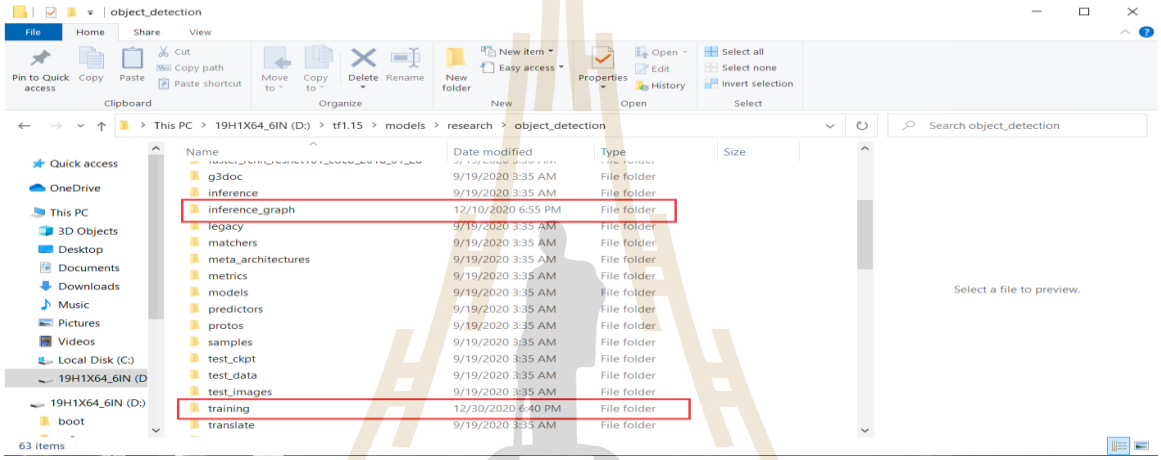
- cd /home/pi/tensorflow1/models/research
- protoc object_detection/protos/*.proto --python_out=.



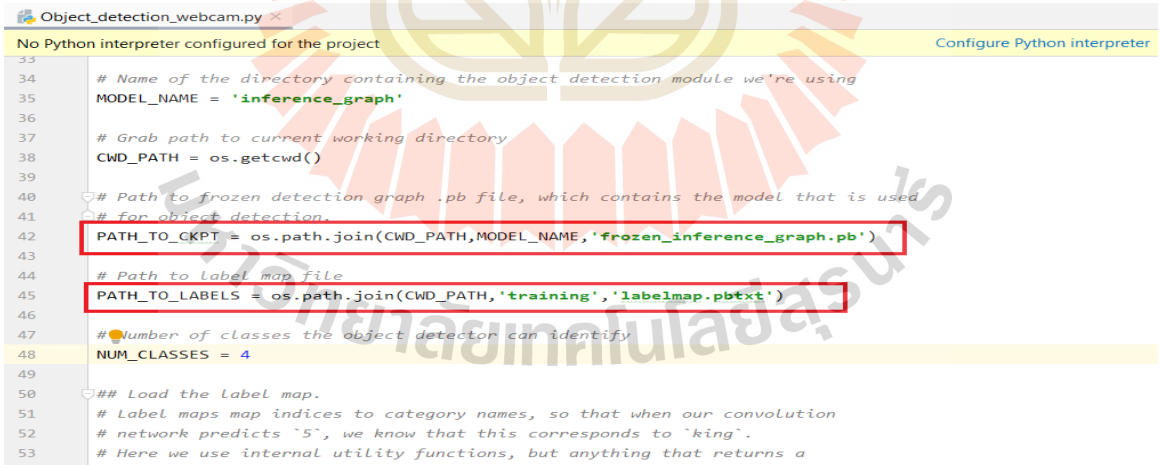
รูปที่ 3.28 รูปแสดงตัวอย่างการแปลงไฟล์

3.4.2 การรันโมเดลบน Raspberry Pi

ในการรันโมเดลนั้นจำเป็นจะต้องมีไฟล์ frozen_inference_graph.pb และไฟล์ labelmap ที่ได้ทำการฝึกสอนไว้ดังนั้นจึงได้ทำการคัดลอกโฟลเดอร์ inference_graph และ training ที่มีไฟล์ frozen_inference_graph.pb และไฟล์ labelmap เพื่อนำไปลงใน Raspberry Pi และตั้งค่าไฟล์ Object_detection_webcam.py ให้ดึงข้อมูลจากไฟล์ frozen_inference_graph.pb และไฟล์ labelmap แสดงดังรูปที่ 3.29



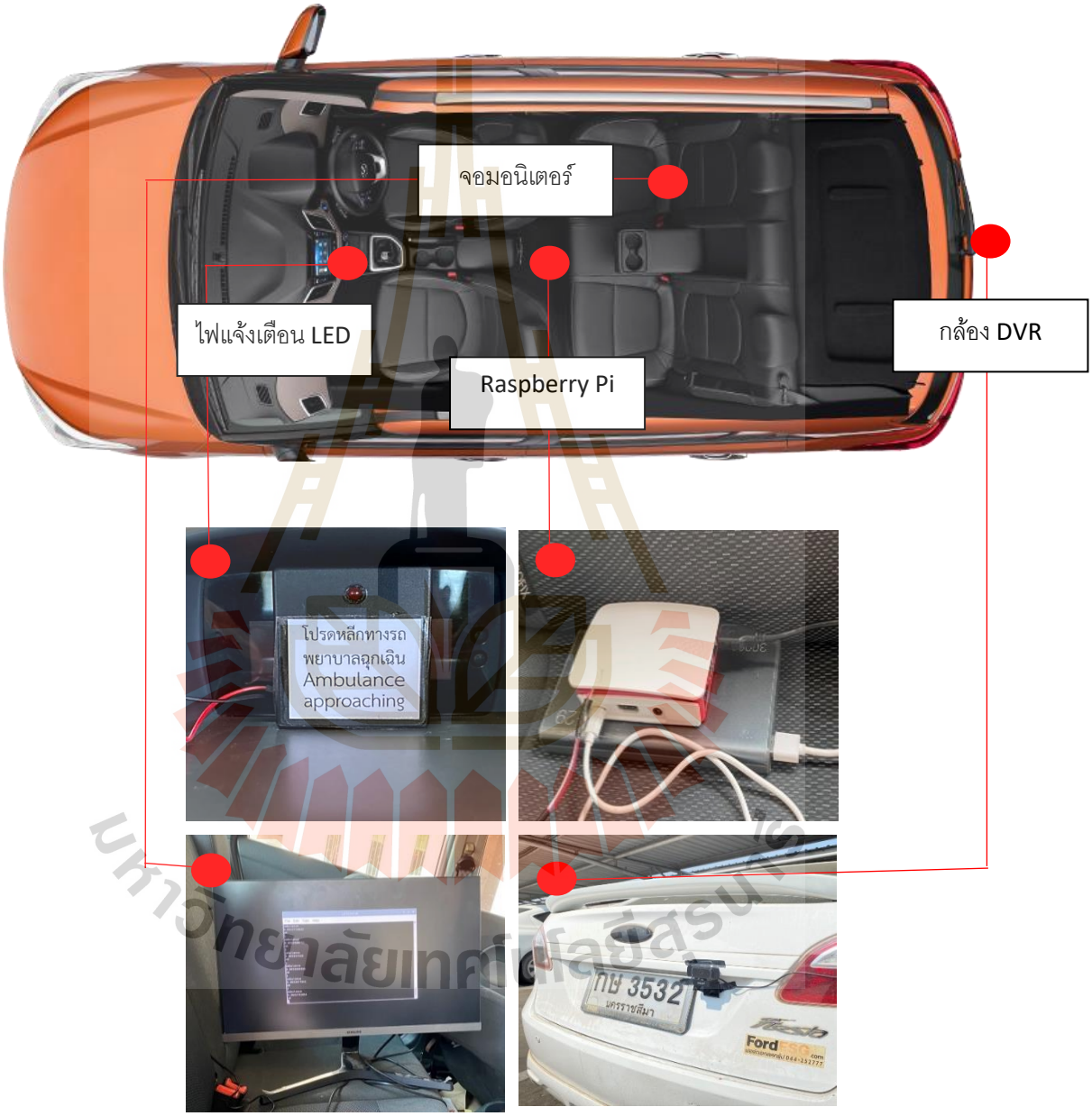
รูปที่ 3.29 รูปแสดงโฟลเดอร์ inference_graph และ training



รูปที่ 3.30 รูปแสดงการแก้ไขไฟล์ Object_detection_webcam.py

3.4.3 การติดตั้งอุปกรณ์การทดสอบบนรถยนต์ส่วนบุคคล

ในขั้นตอนนี้คือขั้นตอนที่นำอุปกรณ์ที่ใช้ในการทดสอบการตรวจจับรถพยาบาลฉุกเฉินมาติดตั้งบนรถยนต์ส่วนบุคคล โดยมีการติดตั้งอุปกรณ์ดังนี้



รูปที่ 3.31 รูปแสดงตำแหน่งการติดตั้งอุปกรณ์บนรถยนต์ส่วนบุคคล

3.4.3.1 Raspberry Pi

Raspberry Pi ถูกใช้เพื่อเป็นตัวประมวลผลในการทดสอบนี้จึงได้ทำการติดตั้งตัวเครื่องไว้ภายในรถยนต์ส่วนบุคคล โดยในงานวิจัยนี้ได้ใช้ Raspberry Pi รุ่นที่ 3 model B v1.2 ในการทำงาน ซึ่งแสดงดังรูปที่ 3.32

รูปที่ 3.32 รูปแสดงการติดตั้ง Raspberry Pi บนรถยนต์ส่วนบุคคล

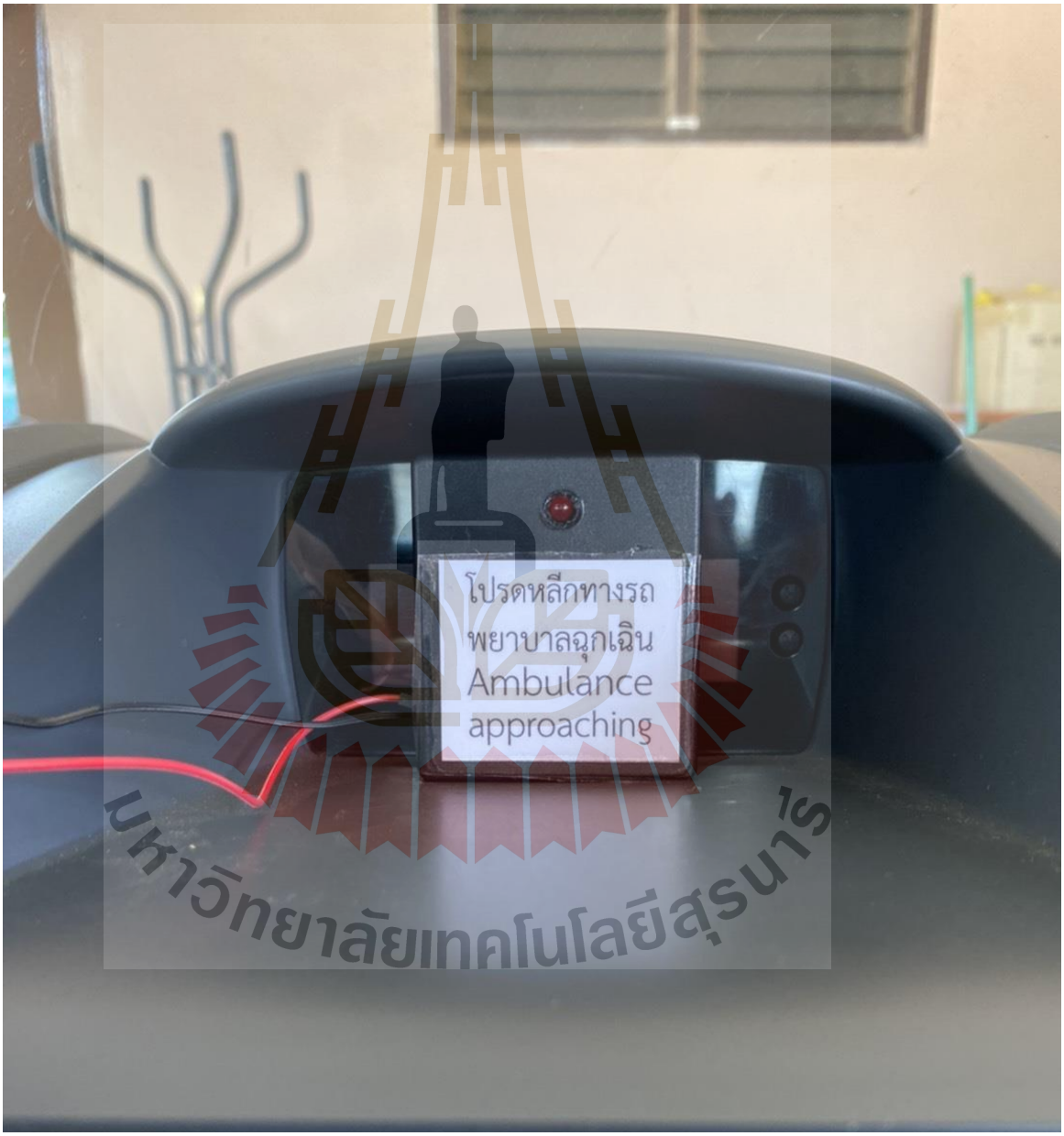
3.4.3.2 กล้องท้ายรถ

ในการทดสอบนี้มีวัตถุประสงค์เพื่อทำการตรวจจับรถพยาบาลฉุกเฉินโดยให้มีการส่งข้อมูลแบบเรียลไทม์ ดังนั้นจึงจำเป็นต้องใช้กล้องเพื่อทดสอบจริงโดยได้ทำการติดตั้งไว้ที่บริเวณท้ายของรถ ดังรูปที่ 3.33

รูปที่ 3.33 รูปแสดงการติดตั้งกล้องบนรถยนต์ส่วนบุคคล

3.4.3.3 ติดตั้งไฟแจ้งเตือน

ขั้นตอนนี้เป็นขั้นตอนในการติดตั้งไฟเพื่อใช้ในการแจ้งเตือนผู้ขับขี่รถยนต์ส่วนบุคคลถึงการมาถึงของรถพยาบาลฉุกเฉิน โดยทำการติดตั้งไว้ที่บริเวณคอนโซลรถยนต์ส่วนบุคคลเพื่อให้ง่ายต่อการสังเกตุเห็น แสดงดังรูปที่ 3.34



รูปที่ 3.34 รูปแสดงการติดตั้งไฟแจ้งเตือนบนรถยนต์ส่วนบุคคล

3.4.3.4 จอมอนิเตอร์

ในขั้นตอนการทดสอบได้ติดตั้งจอมอนิเตอร์ไว้ภายในรถยนต์ส่วนบุคลด้วย เพื่อใช้ในการตรวจสอบความถูกต้องในการตรวจจับรถพยาบาลฉุกเฉินว่ามีการตรวจจับที่ผิดพลาดหรือไม่ โดยการติดตั้งได้แสดงไว้ดังรูปที่ 3.35

รูปที่ 3.35 รูปแสดงการติดตั้งจอมอนิเตอร์บนรถยนต์ส่วนบุคคล

มหาวิทยาลัยเทคโนโลยีสุรนารี

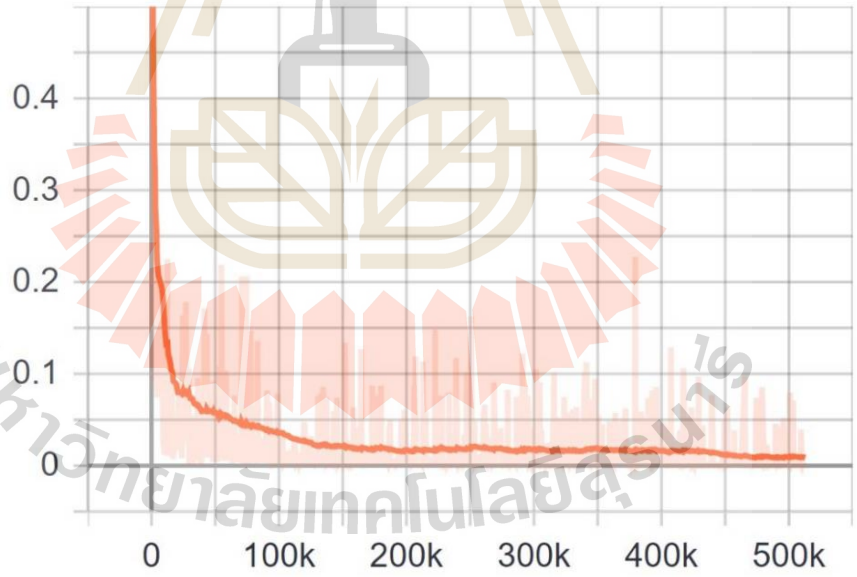
บทที่ 4

ผลการดำเนินงานวิจัย

จากการวิเคราะห์ศึกษาและประยุกต์ใช้เทคโนโลยีการเรียนรู้เชิงลึกตามกระบวนการเทคนิค และขั้นตอนต่างๆที่ใช้โปรแกรมที่กล่าวมาในงานวิจัยบทที่ 3 วิธีดำเนินการวิจัย ได้นำเทคนิคและ กระบวนการต่างๆมาปรับใช้ในงานวิจัยในหัวข้อการประยุกต์ใช้เทคโนโลยีการเรียนรู้เชิงลึกในการตรวจจับวัตถุโดยแบ่งการทดสอบออกเป็นสองแบบดังนี้

4.1 ผลลัพธ์ของการฝึกสอนโมเดล

การฝึกสอนโมเดลในงานวิจัยนี้ได้ตั้งค่าการฝึกสอนไว้ที่ 500,000 Step โดยผลลัพธ์จากการฝึกสอนนี้จะถูกแสดงด้วยรูปแบบของกราฟ total loss ซึ่งจะแสดงดังรูปที่ 4.1



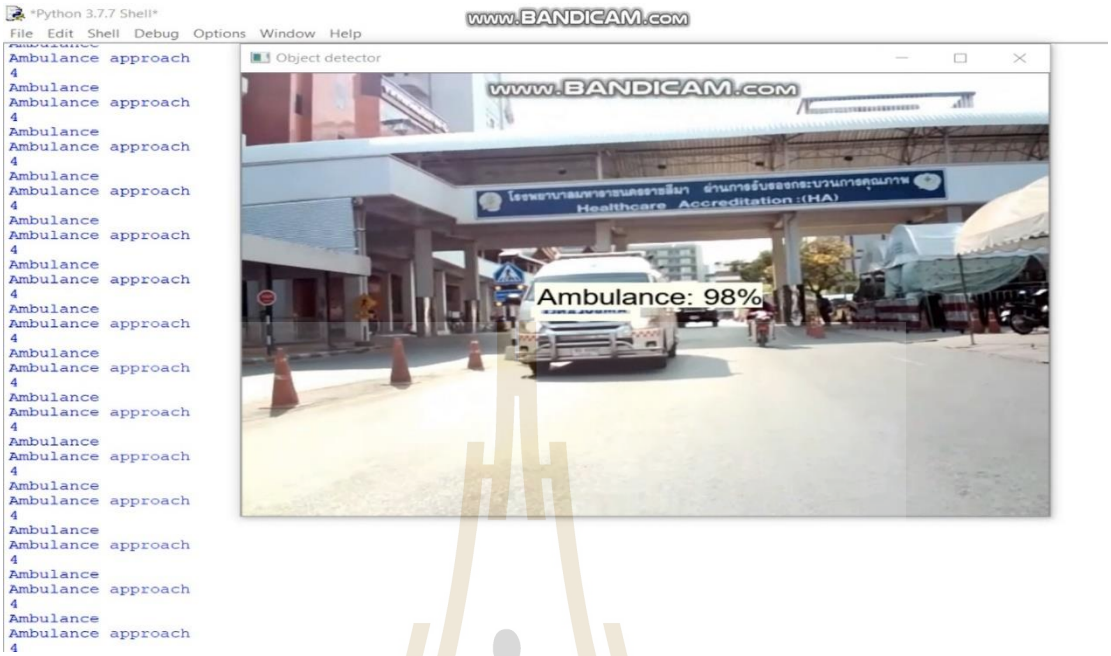
รูปที่ 4.1 รูปแสดงกราฟ total loss

จากรูปที่ 4.1 แสดงให้เห็นว่าค่า total loss นั้นมีค่ามากกว่า 0.5 ในขณะที่เริ่มต้นการฝึกสอนและจากนั้นจะมีค่าลดลงเรื่อยๆตามเวลาที่ใช้ในการฝึกสอนจนมีค่าเข้าใกล้ 0

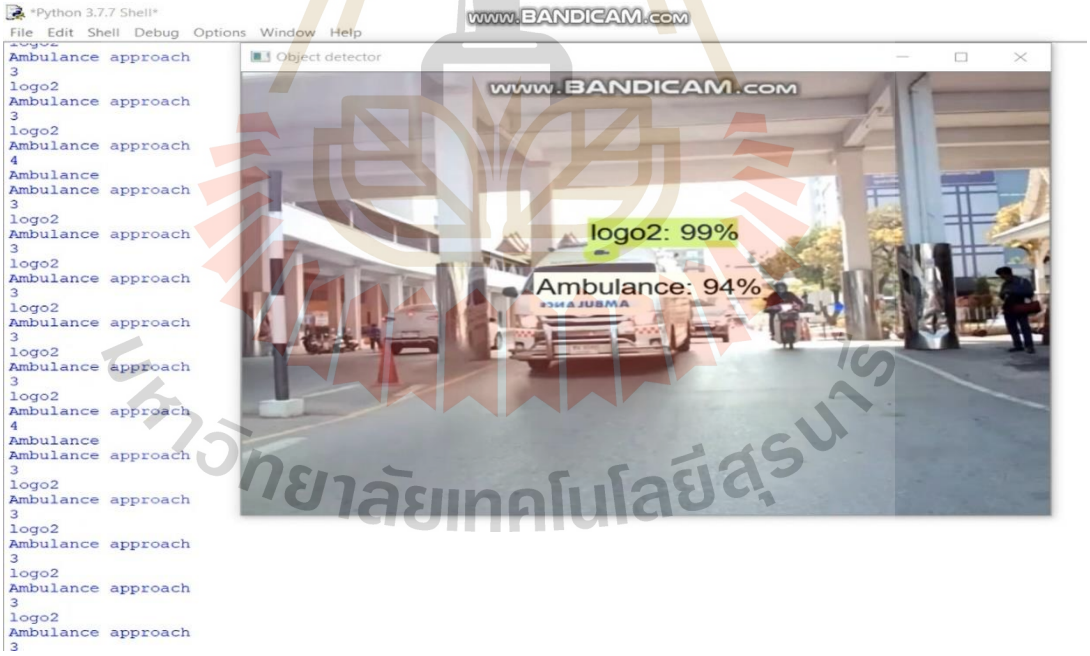
4.2 ผลลัพธ์ของการทดสอบการใช้งานโมเดลโดยใช้คอมพิวเตอร์เป็นตัวประมวลผล

ในขั้นตอนนี้ได้ทำการเก็บข้อมูลวีดีโอของรถพยาบาลบนท้องถนนจริงเพื่อนำมาทำการทดสอบการตรวจจับรถพยาบาลฉุกเฉินในโมเดลที่ฝึกสอนไว้โดยได้เก็บข้อมูลทั้งหมด 2 ตัวอย่าง โดยผลลัพธ์ที่ได้พบว่าการตรวจจับรถพยาบาลฉุกเฉินนั้น มีค่าความถูกต้องไม่น้อยกว่า 90% โดยตัวอย่างการทดสอบนี้เมื่อมีสภาพแสงที่เหมาะสมจะสามารถตรวจจับได้ 2 คุณลักษณะ และที่สภาพแสงไม่เหมาะสมจะสามารถตรวจจับได้เพียง 1 คุณลักษณะ ค่าความถูกต้องนั้นได้มาจาก IoU * 100% แสดงดังรูปที่ 4.2-4.4

รูปที่ 4.2 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 1



รูปที่ 4.3 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 2 ในสภาพแสงที่ไม่เหมาะสม

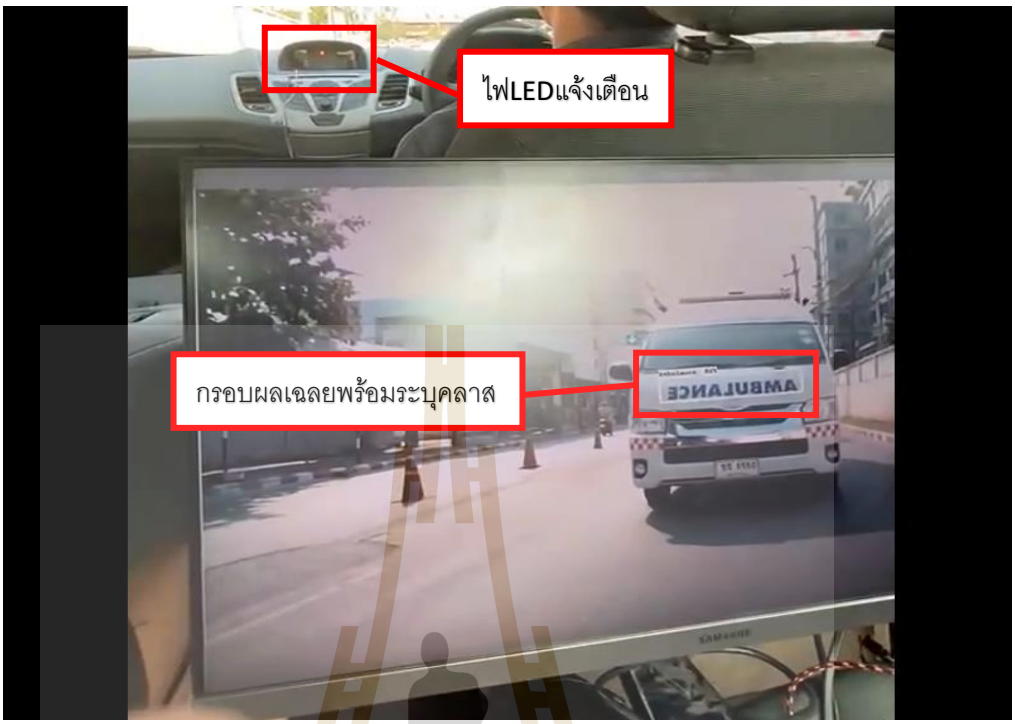


รูปที่ 4.4 รูปแสดงการทดสอบการตรวจจับรถพยาบาลฉุกเฉิน ตัวอย่างที่ 2 ในสภาพแสงที่เหมาะสม

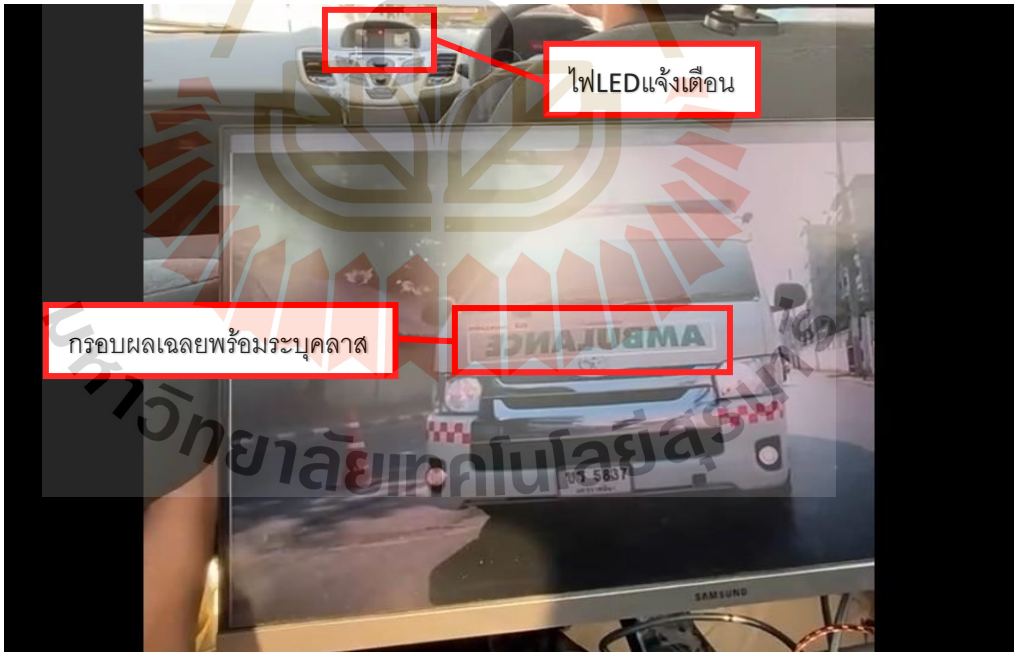
4.3 ผลลัพธ์ของการทดสอบการใช้งานโมเดลร่วมกับ Raspberry Pi ในสถานะการจริง

ในขั้นตอนนี้ได้ทำการติดตั้งอุปกรณ์ Raspberry Pi ที่ทำการลงโปรแกรมการตรวจจับรถพยาบาลฉุกเฉินไว้แล้วดังบทที่ 3 โดยต่อเข้ากับกล้อง DVR และได้ทำการทดสอบการใช้งานจริง โดยการนำไปทดสอบบนท้องถนนเพื่อตรวจจับรถพยาบาลฉุกเฉิน ทำการทดสอบทั้งหมด 5 ตัวอย่าง โดยผลลัพธ์ที่ได้พบว่าการตรวจจับรถพยาบาลฉุกเฉินนั้น มีค่าความถูกต้องไม่น้อยกว่า 75% และในการตรวจจับรถพยาบาลฉุกเฉินทั้ง 5 กรณี มีการตรวจจับเจอคุณลักษณะของรถพยาบาลฉุกเฉินเพียง 1 คุณลักษณะนั่นคือ Text Ambulance ค่าความถูกต้องนั้นได้มาจาก IoU * 100% แสดงดังรูปที่ 4.5-4.9

รูปที่ 4.5 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 1

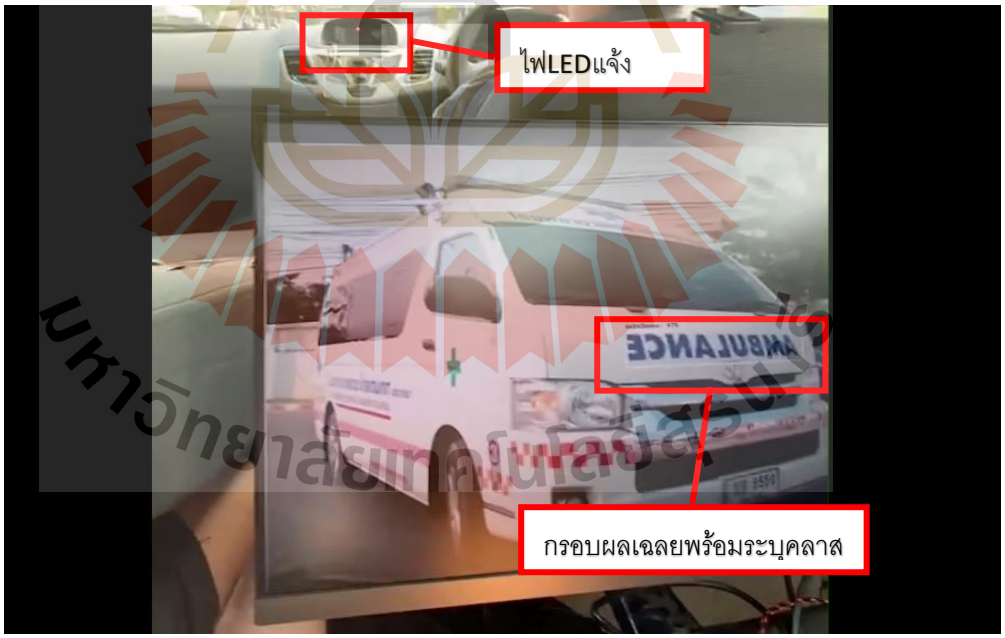


รูปที่ 4.6 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 2



รูปที่ 4.7 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 3

รูปที่ 4.8 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 4



รูปที่ 4.9 รูปแสดงผลลัพธ์การตรวจจับรถพยาบาลฉุกเฉินตัวอย่างที่ 5

CLASS	DETECTION FOUND	DETECTION NOT FOUND	DETECTION WRONG	AVERAGE ACCURACY
Star of life	85%	14%	1%	98.0%
Red Cross	96%	4%	0%	99.0%
Text Ambulance	89%	11%	0%	95.6%
ไฟไซเรน	48%	52%	0%	97.9%

ตารางที่ 4.1 ผลการทดสอบการตรวจจับของโมเดลในแต่ละคลาสด้วยรูปภาพ 100 ภาพ

การทดสอบ	Red cross (%)	Star of life (%)	Text Ambulance (%)	ไฟไซเรน (%)
ครั้งที่1	-	-	83%	-
ครั้งที่2	-	-	75%	-
ครั้งที่3	-	-	82%	-
ครั้งที่4	-	-	98%	-
ครั้งที่5	-	-	97%	-

ตารางที่ 4.2 ตารางสรุปผลการทดสอบโมเดลร่วมกับ Raspberry Pi

จากตารางที่ 4.2 โมเดลนั้นสามารถตรวจจับเจอคุณลักษณะของรถพยาบาลฉุกเฉินเพียงหนึ่งคุณลักษณะสาเหตุเกิดจากในตอนทำการทดสอบเป็นช่วงเวลาที่มีแสงสว่างมากทำให้มีแสงสะท้อนที่คุณลักษณะที่เหลือจึงทำให้ไม่สามารถตรวจจับได้

บทที่ 5

สรุปผล อภิปรายผลและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้ได้ทำการศึกษาการนำเทคโนโลยีปัญญาประดิษฐ์ (AI) มาใช้ในการตรวจจับรถพยาบาลฉุกเฉินเพื่อแจ้งเตือนให้ผู้ขับขี่รถยนต์ส่วนบุคคลทราบถึงการมาถึงของรถพยาบาลฉุกเฉินโดยการฝึกสอนคุณลักษณะของรถพยาบาลฉุกเฉินดังนี้

1. สัญลักษณ์ Star of life
2. สัญลักษณ์ Red cross
3. Text Ambulance
4. ไฟรถพยาบาลฉุกเฉิน

ในการฝึกสอนนั้นได้ทำการใช้โมเดล Faster R-CNN ในการทำงานวิจัยนี้ ซึ่งตั้งค่าการฝึกสอนไว้ที่ 500,000 STEP และค่าความถูกต้องไม่ต่ำกว่า 70% (IoU = 0.7) เมื่อนำมาทดสอบในการใช้งานจริง พบว่าเมื่อใช้คอมพิวเตอร์เป็นตัวประมวลผลในการตรวจจับรถพยาบาลฉุกเฉินจะได้ค่าความถูกต้องไม่ต่ำกว่า 80% แต่เมื่อนำมาทดสอบการใช้งานร่วมกับ Raspberry Pi พบว่าค่าความถูกต้องในการตรวจจับรถพยาบาลฉุกเฉินที่ได้มีค่าไม่ต่ำกว่า 75% และในการตรวจจับรถพยาบาลฉุกเฉินบน Raspberry Pi ทั้ง 5 กรณี มีการตรวจจับเจอคุณลักษณะของรถพยาบาลฉุกเฉินเพียง 1 คุณลักษณะนั่นคือ Text Ambulance

5.2 อภิปรายผล

จากผลการทดสอบที่ผ่านมาพบว่าเมื่อนำโมเดลที่ได้ทำการฝึกสอนไว้ไปใช้ร่วมกับ Raspberry Pi จะได้ค่าความถูกต้องน้อยกว่าการใช้คอมพิวเตอร์ในการประมวลผล เนื่องจากการใช้งานโมเดลคู่กับ Raspberry Pi จะให้ผลลัพธ์ที่มีค่า Frames per second (Fps) ที่ค่อนข้างน้อยมาก สาเหตุอาจเกิดจาก Raspberry Pi ไม่สามารถจัดการกับข้อมูลภาพขนาดใหญ่ได้ทัน จึงส่งผลให้ในการตรวจจับรถพยาบาลฉุกเฉินได้ผลลัพธ์ที่มีค่าความถูกต้องน้อยลงกว่าที่ควร และสาเหตุที่ตรวจจับคุณลักษณะของรถพยาบาลฉุกเฉินได้เพียง 1 คุณลักษณะเนื่องจากในขณะที่ทำการทดสอบ

นั้นสภาพแสงไม่เหมาะสมกับการทดสอบส่งผลให้คุณลักษณะอื่นๆ ถูกแสงสะท้อนจึงทำให้ไม่สามารถตรวจจับได้

5.3 ข้อเสนอแนะ

5.3.1 ควรใช้งานร่วมกับตัวประมวลผลที่มีความสามารถในการประมวลผลที่สูงขึ้น

5.3.2 ควรเลือกภาพที่ใช้ในการฝึกสอนให้มีขนาดภาพเท่ากันเพื่อไม่ให้เกิดปัญหาในการทำงานของโมเดล

5.3.3 ภาพที่ใช้ในการฝึกสอนควรมีขนาดและความละเอียดที่ใกล้เคียงกับกล้องที่ใช้งานจริงเพื่อให้ได้ผลการทำงานที่ถูกต้องมากขึ้น

มหาวิทยาลัยเทคโนโลยีสุรนารี

บรรณานุกรม

[1] Dhatrak, A.S. and Gandhe, S.T., 2018, August. Automatic Traffic Signals in Smart Cities for Speedy Clearance of Emergency Vehicles. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-6). IEEE.

[2] Jin, H., Yan, M., Lu, J., Zhu, L., Wang, K. and Bai, S.,2019, December. Performance Comparison of Moving Target Recognition between Faster R-CNN and SSD. In *2019 International Joint Conference on Information, Media and Engineering (IJCIME)* (pp. 42-46). IEEE.

[3] Jang, H., Yang, H.J., Jeong, D.S. and Lee, H., 2015, January. Object classification using CNN for video traffic detection system. In 2015 21st Korea-*Japan Joint Workshop on Frontiers of Computer Vision (FCV)* (pp. 1-4). IEEE.

[4] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*(2), pp.261-318.

[5] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[6] Basri, Hasan, Iwan Syarif, and Sritrustra Sukaridhoto. "Faster R-CNN Implementation method for multi-fruit detection using Tensorflow platform." 2018 international electronics symposium on knowledge creation and intelligent computing (IES-KCIC). IEEE, 2018.

[7] Laishram, Anuradha, and Khelchandra Thongam. "Detection and Classification of Dental Pathologies using Faster-RCNN in Orthopantomogram Radiography Image." *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2020.

[8] Goldsborough, P., 2016. A tour of tensorflow. arXiv preprint arXiv*:1610.01178*.

[9] National Institute for Emergency Medicine 2020 Available <https://ws.niems.go.th/ITEMS_DWH>

[10] Rezatofighi, Hamid, et al. "Generalized intersection over union: A metric and a loss for bounding box regression." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

ประวัติผู้เขียน

นายถิรมนัส สาระภิรมย์ เกิดเมื่อวันที่ 5 มีนาคม พุทธศักราช 2539 ที่จังหวัดฉะเชิงเทรา สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนอนุบาลวัดปิตุลาธิราชรังสฤษฎิ์ จังหวัดฉะเชิงเทรา ระดับมัธยมศึกษาจากโรงเรียนบางปะกง “บวรวิทยายน” จังหวัดฉะเชิงเทรา และสำเร็จการศึกษาระดับปริญญาตรี สาขาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จากนั้นได้ศึกษาต่อในระดับปริญญาโท สาขาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ระหว่างการศึกษาได้นำเสนอบทความเข้าร่วมประชุม International Virtual Conference on Science and Technology (SUT-IVCST 202) ณ มหาวิทยาลัยเทคโนโลยีสุรนารี วันที่ 28 สิงหาคม 2563

มหาวิทยาลัยเทคโนโลยีสุรนารี