โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส

Real-time object detection program to help automate and provide service for gym

ชื่อนิสิต นายนิวิตถ์ นันทนิวัตติกุล **เลขประจำตัว** 5833638023

นางสาวกชพร วัชรประภาพงศ์ **เลขประจำตัว** 5833601823

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2561

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของโครงงานทางวิชาการที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงงานทางวิชาการที่ส่งผ่านทางคณะที่สังกัด

The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)

are the senior project authors' files submitted through the faculty.

โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส

นายนิวิตถ์ นันทนิวัตติกุล
นางสาวกชพร วัชรประภาพงศ์

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Real-time object detection program to help automate and provide service for gym

Nivit Nantanivattikul

Kotchaporn Watcharaprapapong

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2018

Copyright of Chulalongkorn University

หัวข้อโครงงาน	โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส
โดย	นาย นิวิตถ์ นันทนิวัตติกุล
	นางสาวกชพร วัชรประภาพงศ์
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงานหลัก	รองศาสตราจารย์ ดร.พีระพนธ์ โสพัศสถิตย์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

..... หัวหน้าภาควิชาคณิตศาสตร์

(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

..... อาจารย์ที่ปรึกษาโครงงานหลัก

(รองศาสตราจารย์ ดร.พีระพนธ์ โสพัศสถิตย์)

..... กรรมการ

(รองศาสตราจารย์ ดร.นกุล คูหะโรจนานนท์)

..... กรรมการ

(อาจารย์ ดร.ธีรพงษ์ พงษ์พัฒนเจริญ)

นิวิตถ์ นันทนิวัตติกุล, กชพร วัชรประภาพงศ์: โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส. (Real-time object detection program to help automate and provide service for gym) อ.ที่ปรึกษาโครงงานหลัก: รองศาสตราจารย์ ดร.พีระพนธ์ โสพัศสถิตย์, 60 หน้า.

ปัจจุบันผู้คนเริ่มให้ความสนใจในการดูแลสุขภาพและนิยมการออกกำลังกายมากขึ้น ทำให้ระบบฟิตเนสเริ่มเข้ามาเป็นส่วนหนึ่งในการออกกำลังกายในชีวิตประจำวัน ที่พักอาศัยหลายแห่งรวมถึงคอนโดมีห้องให้บริการสิ่งอำนวยความสะดวกหรือฟิตเนสติดตั้งอยู่ภายในเพื่ออำนวยความสะดวกให้แก่ผู้พักอยู่อาศัยภายในโครงการได้เข้ามาใช้บริการ อย่างไรก็ตามเนื่องจากผู้พักอาศัยมีจำนวนมากทำให้สิ่งอำนวยความสะดวกไม่เพียงพอต่อการใช้งานทำให้การใช้งานต้องสลับเปลี่ยนกันใช้ ผู้จัดทำได้พัฒนาโครงงานโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนสเพื่ออำนวยความสะดวกต่อผู้ใช้งาน โดยการวิเคราะห์การตรวจจับภาพวัตถุจากไฟล์วิดีโอภาพเคลื่อนไหวจากกล้องวงจรปิดแบบเรียลไทม์ด้วยวิธีการประมวลผลภาพโดยการใช้การเรียนรู้เชิงลึกตรวจจับภาพคนที่เข้ามาใช้บริการสิ่งอำนวยความสะดวกในพื้นที่ให้บริการภายในห้องฟิตเนส และทำการแสดงผลสถานะเครื่องออกกำลังกายแต่ละเครื่องภายในห้องฟิตเนสผ่านแอปพลิเคชันมือถือโดยแสดงสถานะว่าขณะนั้นมีผู้ใช้งานอยู่หรือไม่ ทำให้ผู้ใช้งานสามารถทราบถึงสถานการณ์ใช้งาน ณ ขณะนั้นโดยไม่ต้องเข้าไปดูด้วยตนเองในพื้นที่บริการและสามารถตัดสินใจเข้าไปใช้บริการต่อไปได้ด้วยตนเองได้อย่างสะดวกรวดเร็วมากยิ่งขึ้น

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต กชพร วัชรประภาพงศ์

ลายมือชื่อนิสิต นิวิตถ์

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก

ปีการศึกษา 2561

5833638023, 5833601823: MAJOR COMPUTER SCIENCE

KEYWORDS : OBJECT DETECTION / DEEP LEARNING / REAL TIME / APPLICATION

NIVIT NANTANIVATTIKUL, KOTCHAPORN WATCHARAPRAPAPONG: REAL-TIME OBJECT DETECTION PROGRAM TO HELP AUTOMATE AND PROVIDE SERVICE FOR GYM. ADVISOR: ASSOC. PROF. PERAPHON SOPHATSATHIT, Ph.D., 60 pp.

Nowadays, people are becoming more health concern and focus on healthy lifestyles, resulting in high demands for fitness facilities as a part of their daily exercise. Many residential establishments, such as condominium, have fitness facilities on their premises to facilitate their residents. In this project, we develop a program to detect objects from CCTV to manage fitness services to residents. The principal technique is analyzing the image files obtained from real-time CCTV using image processing methods. These image files are further processed with the help of deep learning technique to capture images of people who using the facilities in the fitness room. The availability of each fitness equipment in the room is displayed via the mobile application. This will enable the users to see up-to-minute status of equipment usage without having to constantly check its availability in person, thereby they can decide when to use the equipment at their own pace.

Department: Mathematics and Computer Science Student's Signature กชพร วัชรประภาพงศ์

Student's Signature นิวิทก์

Field of Study: Computer Science Advisor's Signature Peraphon Sophatsathit

Academic Year: 2018

กิตติกรรมประกาศ

โครงงานเรื่อง โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์และช่วยเหลือจากคณาจารย์และบุคลากรหลายท่าน ในตลอดระยะเวลาการดำเนินการโครงงาน ทางคณะผู้ดำเนินโครงงานจึงใคร่ขอขอบพระคุณเป็นอย่างยิ่งสำหรับความช่วยเหลือต่างๆที่มอบให้ ดังต่อไปนี้

ขอขอบพระคุณรองศาสตราจารย์ ดร.พีระพนธ์ โสพัศสถิต อาจารย์ที่ปรึกษาโครงงาน ที่คอยให้คำปรึกษาแนะนา ชี้แนะแนวคิดในการทำโครงงาน ตลอดจนการปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่งจึงทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณรองศาสตราจารย์ ดร.นกุล คูหะโรจนานนท์ และ อาจารย์ ดร.ธีรพงษ์ พงษ์พัฒนเจริญ ผู้เป็นกรรมการคุมสอบ ที่คอยชี้แนะในสิ่งที่ควรนำไปปรับปรุงและแนะนำแนวทางที่เป็นประโยชน์ในการทำโครงงานนี้

ขอขอบคุณสถานที่ โครงการแอสปาย งามวงศ์วาน ที่ให้ความร่วมมือในการเก็บชุดข้อมูลจากสถานที่ภายในโครงการ รวมถึงให้ข้อมูลเกี่ยวกับสินค้าของทางบริษัท และข้อเสนอแนะจนทำให้โครงงานนี้ดำเนินการสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ทุกท่านในคณะวิทยาศาสตร์ที่ได้ถ่ายทอดวิชาความรู้ ทำให้ผู้จัดทำได้นำความรู้มาประยุกต์ใช้เพื่อพัฒนาโครงงานนี้ได้สาเร็จ

ขอขอบพระคุณบิดามารดาที่คอยให้กำลังใจและสนับสนุนการทำโครงงานนี้ คอยเอาใจใส่เป็นห่วงเป็นใย เป็นแรงบันดาลใจที่ทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณเพื่อนทุก ๆ คนที่คอยช่วยเหลือ ให้คำปรึกษาเมื่อเกิดปัญหาและยินดีให้ความช่วยเหลือเสมอมา ทำให้โครงงานนี้ประสบความสาเร็จ

ท้ายที่สุดนี้ ขอขอบคุณความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามไว้ข้างต้น รวมถึงบุคคลท่านอื่นที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ด้วย สำหรับความช่วยเหลือและคำแนะนำต่าง ๆ ที่คอยผลักดันให้โครงงานนี้ประสบผลสำเร็จไปได้ด้วยดี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ญ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและเหตุผลการวิจัย 1

1.2 วัตถุประสงค์ของการวิจัย 1

1.3 ขอบเขตการวิจัย 2

1.4 ขั้นตอนการวิจัย 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 5

1.6 โครงสร้างของรายงาน 6

บทที่ 2 งานวิจัยที่เกี่ยวข้อง 7

2.1 การตรวจจับวัตถุ 7

2.2 การจำแนกวัตถุ 8

2.3 แบบจำลองการตรวจจับวัตถุ 9

2.4 ฟังก์ชันซิกมอยด์ 10

2.5 โครงข่ายประสาทแบบคอนโวลูชั่น 11

2.6 ไลบรารี บริการและโปรแกรมที่ใช้สำหรับพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิด ..13

2.7 อุปกรณ์ที่เกี่ยวข้อง 16

บทที่ 3 การรวบรวม การเตรียมข้อมูลและการสร้างแบบจำลอง 18

3.1 การรวบรวมข้อมูล....18
3.2 การเตรียมข้อมูล....19
3.3 การสร้างและทดสอบแบบจำลอง....21
3.4 การพัฒนาแบบจำลองในอุปกรณ์....28
3.5 การนำไปใช้ในระบบ....29
บทที่ 4 การพัฒนาระบบและผลการทดสอบ....31
4.1 การพัฒนาระบบ....31
4.2 ผลของการทดสอบ....34
บทที่ 5 ข้อสรุปและข้อเสนอแนะ....37
5.1 ข้อสรุป....37
5.2 ปัญหาและอุปสรรค....37
5.3 วิธีแก้ไขปัญหา....38
5.4 ผลที่ได้รับจากการพัฒนาโครงงาน....38
5.5 ข้อเสนอแนะ....38
รายการอ้างอิง....39
ภาคผนวก ก แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal ปีการศึกษา 2561 44
ประวัติผู้เขียน....49

สารบัญตาราง

หน้า

ตารางที่ 1.1 แสดงผลลัพธ์ที่ได้จากแบบจำลองจำแนกบุคคล 35
ตารางที่ 1.2 แสดงผลลัพธ์ที่ได้จากการทดลองในหนึ่งครั้ง 35

สารบัญภาพ

หน้า

ภาพที่ 1.3 สถาปัตยกรรมของระบบ 2
ภาพที่ 2.1 การตรวจจับในงานวิจัย [3] 7
ภาพที่ 2.2 โครงสร้างเครือข่ายประสาทเทียมของงานวิจัย [5] 8
ภาพที่ 2.3 ภาพเปรียบเทียบโครงสร้างของ SSD กับ YOLO 9
ภาพที่ 2.4 ลักษณะการทำงานของ RPN 10
ภาพที่ 2.5 กราฟฟังก์ชันซิกมอยด์ 11
ภาพที่ 2.6 โครงข่ายประสาทเทียมแบบคอนโวลูชั่น 12
ภาพที่ 2.7 ตัวอย่างการทำแม็กซ์พูลลิ่งในเมทริกซ์ขนาด 4 x 4 13
ภาพที่ 2.8 สัญลักษณ์ Firebase 13
ภาพที่ 2.9 สัญลักษณ์ Keras 14
ภาพที่ 2.10 สัญลักษณ์ TensorFlow 15
ภาพที่ 2.11 สัญลักษณ์ React-Native 16
ภาพที่ 2.12 Raspberry PI 3b+ 16
ภาพที่ 2.13 Intel movidius ที่เชื่อมต่อกับ Raspberry Pi 17
ภาพที่ 3.1 ภาพรวมของขั้นตอนในการรวบรวม เตรียมข้อมูลและการสร้างแบบจำลอง 18
ภาพที่ 3.2 ชุดข้อมูลวีดีโอที่ผู้จัดทำรวบรวมมาจากกล้องวงจรปิดในฟิตเนส 19
ภาพที่ 3.3 วีดีโอจากกล้องวงจรปิดในพื้นที่บริเวณฟิตเนส 19
ภาพที่ 3.4 วีดีโอที่ได้ทำการลดขนาด 20
ภาพที่ 3.5 ตัวอย่างรูปภาพพื้นหลังที่ไม่มีคน 20
ภาพที่ 3.6 ตัวอย่างรูปภาพที่เป็นคน 21
ภาพที่ 3.7 ตัวอย่างการแบ่งภาพในเฟรมของวีดีโอ 22
ภาพที่ 3.8 ตัวอย่างของผลลัพท์ที่ได้จากการตรวจจับของแบบจำลอง 22
ภาพที่ 3.9 ตัวอย่างซอรส์โค้ดที่ใช้ในการดาวน์โหลดแบบจำลองที่ได้เรียนรู้ไว้แล้วและทำนาย 24
ภาพที่ 3.10 โครงสร้างของแบบจำลองจำแนกรูปภาพที่ผู้จัดทาสร้าง 25
ภาพที่ 3.11 ตัวอย่างซอร์สโค้ดสร้างแบบจำลอง 26
ภาพที่ 3.12 ตัวอย่างแสดงการทำการเพิ่มข้อมูล 27
ภาพที่ 3.13 ตัวอย่างผลลัพธ์ของแบบจำลองจำแนกบุคคล 28

ภาพที่ 3.14 ภาพแสดงกระบวนการแปลงแบบจำลองจาก Tensorflow เป็นโมวิดิอุสกราฟ..... 28
ภาพที่ 3.15 ตัวอย่างของคำสั่งที่ใช้แปลงกราฟ 29
ภาพที่ 3.16 ตัวอย่างซอร์สโค้ดที่รับวีดีโอถ่ายทอดสดจากกล้อง 30
ภาพที่ 3.17 สถาปัตยกรรมของระบบ 30
ภาพที่ 4.1 ตัวอย่างซอร์สโค้ดการส่งข้อมูลขึ้นฐานข้อมูลบนไฟร์เบส 32
ภาพที่ 4.2 ตัวอย่างของ JSON ที่ API คืนค่า 32
ภาพที่ 4.3 ตัวอย่างของหน้าที่แสดงสถานะของเครื่องเล่น 33
ภาพที่ 4.4 ตัวอย่างของวีดีโอเมื่อทดสอบบนอุปกรณ์ 36

บทที่ 1
บทนำ

บทนี้จะกล่าวถึงหลักการและเหตุผลในการจัดทำโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินงาน ตารางเวลาการดำเนินงาน ประโยชน์ที่ได้รับรวมไปถึงรายงานโดยสังเขป ซึ่งมีรายละเอียดดังนี้

1.1 ความเป็นมาและเหตุผลการวิจัย

ปัจจุบันผู้คนเริ่มให้ความสนใจในการดูแลสุขภาพและนิยมการออกกำลังกายมากขึ้น ทำให้ระบบฟิตเนสเริ่มเข้ามาเป็นส่วนหนึ่งในการออกลังกายในชีวิตประจำวันโดยทั่วไป เนื่องจากไม่ต้องใช้สถานที่กว้างขวางมากมาย สะดวกและราคาอัตราการใช้บริการไม่สูงมากนัก ดังนั้นในสถานที่ต่างๆเริ่มใช้ระบบฟิตเนส เข้ามาติดตั้งเพื่ออำนวยความสะดวกให้แก่ผู้คนที่เข้ามาใช้บริการ เช่น ที่ทำงาน บ้านพักอาศัย สโมสรต่างๆ ซึ่งคอนโดที่พักอาศัยก็เป็นส่วนหนึ่งที่มีบริการฟิตเนสเป็นสิ่งอำนวยความสะดวกหลักอยู่ภายในโครงการ

เนื่องจากภายในคอนโดมีผู้พักอาศัยร่วมกันจำนวนมาก และระยะเวลาการเข้าใช้งานฟิตเนสต่อบุคคลนั้นใช้ระยะเวลานานอย่างต่ำครึ่งชั่วโมงเป็นต้นไป ถ้าคอนโดใดมีผู้ใช้บริการเป็นจำนวนมากจะทำให้มีบางเวลาที่ผู้พักอาศัยที่ต้องการมาใช้บริการนั้นไม่ได้ออกกำลังกาย ต้องเสียเวลาในการรอคิวหรือเปลี่ยนใจเนื่องจากระยะเวลาที่นานเกินไปในการรอ ทำให้การออกกำลังกายไม่เป็นไปตามที่คาดหวังไว้ ดังนั้นทางผู้พัฒนาจึงทำโปรแกรมนี้ขึ้นมาเพื่อตอบสนองต่อความต้องการของผู้ที่ต้องการเข้ามาใช้งานฟิตเนส เพื่ออำนวยความสะดวกต่อผู้ใช้งาน ซึ่งผู้ใช้งานสามารถดูผู้เข้าใช้งานคนอื่นที่กำลังใช้งานฟิตเนสอยู่ผ่านทางระบบได้เลยโดยไม่ต้องเข้าไปดูด้วยตนเองที่พื้นที่ให้บริการ ทำให้ทางผู้ใช้งานเองสามารถใช้เวลาส่วนตนให้เกิดประโยชน์สูงสุด ไม่จำเป็นต้องเสียเวลาเข้ามารอที่หน้าฟิตเนสเพื่อรอใช้งาน สามารถใช้เวลาไปทำอย่างอื่นก่อน และเมื่อสิ่งอำนวยความสะดวกตามที่ผู้ใช้งานต้องการใช้งานนั้นว่างแล้ว ผู้ใช้งานสามารถเข้าใช้งานได้เลยทำให้ไม่เป็นการเสียอารมณ์ขุ่นมัวในใช้เวลาในการรอนานเกินไปซึ่งส่งผลเสียต่อสุขภาพจิตของผู้ใช้งาน

1.2 วัตถุประสงค์ของการวิจัย

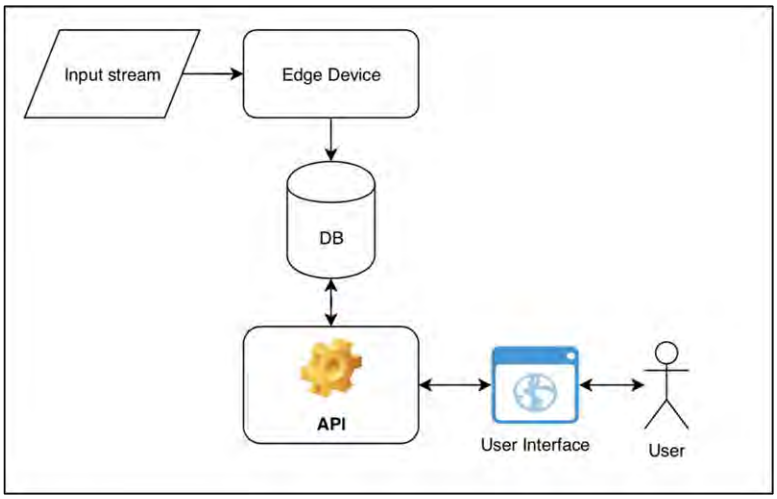
โครงงานนี้มีวัตถุประสงค์เพื่อ

1. เพื่อวิเคราะห์การตรวจจับวัตถุด้วยวิธีการประมวลผลภาพจากการใช้การเรียนรู้เชิงลึก
2. เพื่อวิเคราะห์การจำแนกวัตถุด้วยวิธีการประมวลผลภาพจากการใช้การเรียนรู้เชิงลึก

3. เพื่อพัฒนาระบบที่สามารถตรวจจับคนที่เข้ามาใช้บริการสิ่งอำนวยความสะดวกภายในพื้นที่สิ่งอำนวยความสะดวกในคอนโดแบบอัตโนมัติ
4. เพื่อทดสอบประสิทธิภาพของโปรแกรมที่นำเสนอ

1.3 ขอบเขตการวิจัย

1. ศึกษาข้อมูลภาพในการตรวจจับคนจากกล้องวงจรปิดภายในพื้นที่ให้บริการสิ่งอำนวยความสะดวกในที่อยู่อาศัย เช่นพื้นที่ฟิตเนส หรือภายในห้อง
2. บริเวณพื้นที่ที่ติดตั้งกล้องวงจรต้องมีแสงสว่างเพียงพอและสามารถเห็นคนได้อย่างชัดเจน
3. กล้องวงจรปิดต้องเป็นกล้องที่ไม่มีการเปลี่ยนทิศทางหรือก็คือภาพที่ได้จากกล้องจะต้องอยู่มุมเดิมเสมอ
4. ส่วนต่อประสานงานกับผู้ใช้ (user interface) เป็นแอปพลิเคชั่นมือถือบน IOS ที่เขียนด้วย react-native
5. สถาปัตยกรรมของระบบกำหนดดังในภาพที่ 1.3



ภาพที่ 1.3 สถาปัตยกรรมของระบบ

1.4 ขั้นตอนการวิจัย

โครงงานการพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส มีขั้นตอนการดำเนินการดังต่อไปนี้

1. เก็บรวบรวมข้อมูลและประมวลผลข้อมูล
    1.1 เก็บรวบรวมข้อมูลสำหรับการสร้างโมเดลการเรียนรู้เชิงลึก
    1.2 ประมวลผลข้อมูลและการระบุประเภทของรูป
2. การสร้างโมเดลในการเรียนรู้
    2.1 ตั้งค่า Google Cloud Instance สำหรับการเรียนรู้ของโมเดล
    2.2 สร้างการเรียนรู้ของโมเดล
    2.3 ปรับค่าโมเดล
    2.4 นำโมเดลลงอุปกรณ์และทดสอบ
3. การนำโมเดลไปใช้ในระบบหลังบ้าน
    3.1 ติดตั้งเซิร์ฟเวอร์สำหรับเรียกใช้ระบบหลังบ้าน
    3.2 พัฒนาระบบหลังบ้าน
4. พัฒนาส่วนต่อประสานผู้ใช้งาน
    4.1 ออกแบบส่วนต่อประสานผู้ใช้งานสำหรับแอพพลิเคชัน
    4.2 พัฒนาแอพพลิเคชัน
5. เชื่อมต่อส่วนต่อประสานผู้ใช้งานกับระบบหลังบ้าน
    5.1 เชื่อมต่อแอพพลิเคชันกับ API
    5.2 นำแอพพลิเคชันไปใช้งาน
6. การนำโมเดลไปทดสอบบนอุปกรณ์ในสถานที่ติดตั้งจริง
    6.1 ทดสอบการใช้งานของส่วนต่อประสานผู้ใช้งานกับอุปกรณ์
    6.2 การปรับแต่งโมเดล

ตารางระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลา															
	ปี 2560												ปี 2561			
	ต.ค.				พ.ย.				ธ.ค.				ม.ค.			
1. เก็บรวบรวมข้อมูลและประมวลผลข้อมูล																
1.1 เก็บรวบรวมข้อมูลสำหรับการสร้างโมเดลการเรียนรู้เชิงลึก																
1.2 ประมวลผลข้อมูลและการระบุประเภทของรูป																
2. การสร้างโมเดลในการเรียนรู้																
2.1 ตั้งค่า Google Cloud Instance สำหรับการสร้างโมเดล																
2.2 สร้างการเรียนรู้ของโมเดล																
2.3 ปรับค่าโมเดล																
2.4 นำโมเดลลงอุปกรณ์และทดสอบ																
3. การนำโมเดลไปใช้ในระบบหลังบ้าน																
3.1 ติดตั้งเซิร์ฟเวอร์สำหรับเรียกใช้ระบบหลังบ้าน																
3.2 พัฒนาระบบหลังบ้าน																
4. พัฒนาส่วนต่อประสานผู้ใช้งาน																
4.1 ออกแบบส่วนต่อประสานผู้ใช้งานสำหรับแอพพลิเคชัน																
4.2 พัฒนาแอพพลิเคชัน																
5. เชื่อมต่อส่วนต่อประสานผู้ใช้งานกับระบบหลังบ้าน																
5.1 เชื่อมต่อแอพพลิเคชันกับ API																
5.2 นำแอพพลิเคชันไปใช้งาน																
6. การนำโมเดลไปทดสอบบนอุปกรณ์ในสถานที่ติดตั้งจริง																
6.1 ทดสอบการใช้งานของส่วนต่อประสานผู้ใช้งานกับอุปกรณ์																
6.2 การปรับแต่งโมเดล																

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้รับจากการวิจัยในครั้งนี้มีดังนี้

ก. ประโยชน์ต่อนิสิต

1. สามารถประยุกต์โมเดลเชิงวิชาการ Convolutional neural network กับการแก้ปัญหาจริง
2. ได้พัฒนาทักษะการใช้ภาษา Python และการพัฒนาแอพพลิเคชันบนระบบ IOS โดยใช้ react-native
3. ได้ฝึกทักษะการทำงานเป็นกลุ่ม การคิดวิเคราะห์และแก้ปัญหา
4. ได้เรียนรู้และพัฒนาเทคนิคเกี่ยวกับการใช้การเรียนรู้เชิงลึกในการตรวจจับวัตถุจากภาพกล้องวงจรปิด
5. ได้เรียนรู้การพัฒนาระบบให้สอดคล้องกับอุปกรณ์ที่นำมาร่วมใช้ในการตรวจจับวัตถุจากกล้องวงจรปิด

ข. ประโยชน์ที่ได้จากโครงงานที่พัฒนาขึ้น

1. สามารถนำระบบไปใช้ในการจัดการและให้บริการในพื้นที่สิ่งอำนวยความสะดวกภายในคอนโดและสถานที่ต่างๆ
2. สามารถอำนวยความสะดวกสบายแก่ผู้เข้ามาใช้บริการสิ่งอำนวยความสะดวกภายในคอนโดมากยิ่งขึ้น
3. ช่วยลดปริมาณคน เวลาและต้นทุนที่จะเกิดขึ้นจากการให้บริการระบบตรวจจับปริมาณคนใช้งานเพื่ออำนวยความสะดวกผู้ใช้งานภายในพื้นที่สิ่งอำนวยความสะดวกภายในคอนโด
4. ผู้ให้บริการสิ่งอำนวยความสะดวกสามารถนำข้อมูลไปใช้ในการพัฒนาระบบและปรับปรุงคุณภาพในการให้บริการพื้นที่สิ่งอำนวยความสะดวก

1.6 โครงสร้างของรายงาน

บทที่ 2 จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุม และให้บริการฟิตเนส ขั้นตอนวิธีในการตรวจจับวัตถุ การจำแนกวัตถุ แบบจำลองการตรวจจับวัตถุ รวมถึงไลบรารีต่าง ๆ ที่ใช้ในโครงงานนี้

บทที่ 3 จะกล่าวถึงวิธีการดำเนินงาน ซึ่งจะประกอบไปด้วย การรวบรวมข้อมูล การเตรียมข้อมูล การสร้างและทดสอบแบบจำลอง การพัฒนาแบบจำลองบนอุปกรณ์ และการนำโมเดลไปใช้บนอุปกรณ์

บทที่ 4 จะกล่าวถึงการออกแบบพัฒนาระบบและผลการทดสอบ

บทที่ 5 จะกล่าวถึงข้อสรุป ปัญหาและอุปสรรค วิธีการแก้ไขปัญหา แนวทางในการพัฒนาระบบต่อไปในอนาคต และข้อเสนอแนะ

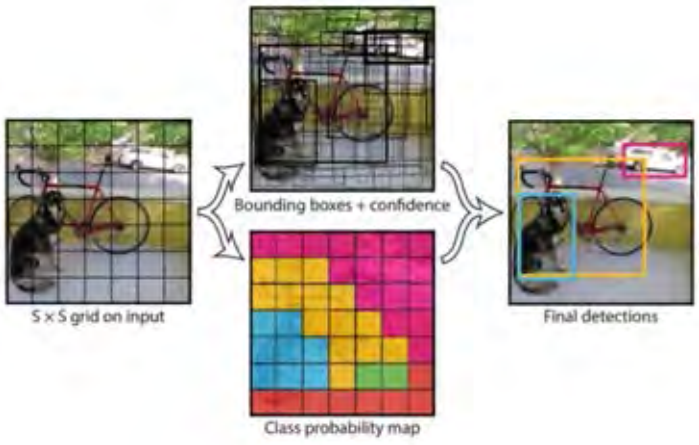
บทที่ 2
งานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการตรวจจับวัตถุ การจำแนกวัตถุ แบบจำลองการตรวจจับวัตถุ ทฤษฎีที่เกี่ยวข้อง รวมถึงไลบรารีที่ใช้ในการพัฒนาระบบการตรวจจับวัตถุและอุปกรณ์ที่ใช้ร่วมกับระบบ

2.1 การตรวจจับวัตถุ

การตรวจจับวัตถุ [1] คือกระบวนการที่ตรวจหาวัตถุซึ่งเป็นตัวอย่างเช่น รถ, จักรยาน, โทรศัพท์มือถือ และมนุษย์ ในภาพนิ่งหรือในวีดีโอ โดยสามารถตรวจจับได้ว่าวัตถุเป็นอะไรและอยู่ที่ตำแหน่งใดในรูปภาพ ซึ่งจะสามารถตรวจจับได้หลายวัตถุในรูป ๆ เดียว ปกติแล้วจะถูกใช้งานกันในทั่วไปในระบบค้นหารูปภาพ การรักษาความปลอดภัย หรือแม้กระทั่งระบบขับขี่อัตโนมัติ การตรวจจับวัตถุมีหลายเทคนิค เช่น การตรวจจับวัตถุโดยอาศัยฟีเจอร์, การตรวจจับวัตถุโดยการใช้ซัพพอร์ตเวคเตอร์แมชชีน, การตรวจจับวัตถุของวิโอลา โจนส์ และการตรวจจับวัตถุโดยการใช้เครือข่ายประสาทเทียม

จากงานวิจัย [2] การตวรจจับวัตถุโดยการใช้เครือข่ายประสาทเทียม ศึกษาการใช้คอนโวลูชั่นนอลในโครงสร้างของเครือข่ายประสาทเทียม โดยการนำเอาเฟรมรูปภาพมาแบ่งย่อยออกเป็นบริเวณเล็ก ๆ หลายส่วนและในแต่ละส่วนจะทำนาย 2 อย่าง หนึ่งคือขอบเขตของกล่อง (Bounding box) ที่วัตถุอยู่ (Localization) และทำนายว่าวัตถุนั้นเป็นอะไร (Class prediction) ดังภาพที่ 2.1



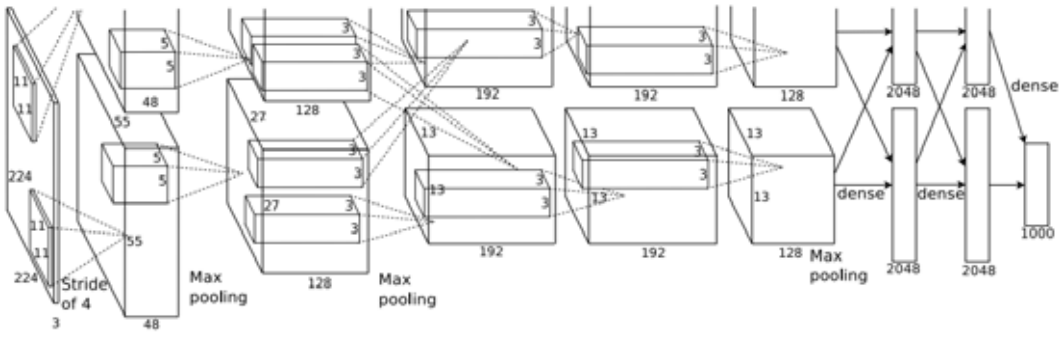
ภาพที่ 2.1 การตรวจจับวัตถุในงานวิจัย [3]

จากงานวิจัย [3] เป็นการพัฒนาขึ้นของงานวิจัย [2] โดยมีการเปลี่ยนแปลงโครงสร้างของระบบประสาทเทียมใหม่ เช่น เพิ่มชั้นเรซิดวล (Residual layer) คือ ชั้นที่นำเอาฟีเจอร์ของชั้นก่อนหน้ารวมเข้ากับชั้นล่าสุด รวมถึงการที่ปรับเปลี่ยนการให้อยู่บนสัดส่วนของ 3 สัดส่วน (Multi-scale Detection) ส่งผลให้การตรวจจับวัตถุที่มีขนาดเล็กมีประสิทธิภาพดีขึ้น นอกจากนั้นยังเปลี่ยนแปลงฟังก์ชันสูญเสียของแบบจำลองให้เป็น การถดถอยแบบโลจิสติก (Logistic Regression) ทำให้สามารถทำการทำนายวัตถุได้เป็นหลายคลาสในหนึ่งตัวอย่าง เช่น วัตถุที่เป็นคนและเป็นผู้หญิงก็จะสามารถทำนายออกมาโดยมีหลายคลาสได้ (Multi Label) ในขณะที่ในงานวิจัย [2] จะสามารถทำนายได้แค่เป็นบุคคลเนื่องจากใช้ฟังก์ชันซอฟท์แม็กในการให้คะแนนการทำนาย ในงานวิจัย [3] ยังกล่าวถึงสิ่งที่ได้ทำการทดลองไปและไม่ได้ประสิทธิภาพด้วย ซึ่งได้แก่การใช้ฟังก์ชันที่เป็นเชิงเส้นในการทำนาย x, y ของกล่องที่วัตถุอยู่

2.2 การจำแนกวัตถุ

การจำแนกวัตถุ [4] คือกระบวนการที่จะจำแนกวัตถุที่อยู่ในรูปภาพหรือวิดีโอ ว่าเป็นวัตถุประเภทใด (Class Predicton) แต่ไม่สามารถระบุตำแหน่งได้ว่าอยู่ในบริเวณใด ตัวอย่างของการใช้งานการจำแนกวัตถุหรือรูปภาพคือ การจดจำใบหน้า (Face Recognition) การค้นหารูปภาพ รวมถึงการรักษาความปลอดภัยเป็นต้น เทคนิคที่ใช้ในการจำแนกวัตถุมีหลายวิธีหนึ่งในนั้นประกอบด้วยการใช้เครือข่ายประสาทเทียมดังจะกล่าวในงานวิจัยต่อไปที่จะอ้างถึง

จากงานวิจัย [5] เป็นการใช้เครือข่ายประสาทเทียมโดยมีชั้นของคอนโวลูชันช่วยในการทำการสกัดลักษณะที่สำคัญ (Feature Extraction) โดยในงานวิจัยได้ทำการสร้างแบบจำลองที่เรียนรู้ชุดข้อมูล ImageNet LSVRC-2010 [6] ซึ่งมีคลาสต่าง ๆ ถึง 1000 ชนิด ซึ่งโครงสร้างของโครงข่ายประสาทเทียมในงานวิจัยนี้ ประกอบไปด้วยชั้นคอนโวลูชัน (Convolutional layer) 5 ชั้น เนื่องจากมิติของข้อมูลเข้ามีขนาดใหญ่ บางชั้นจะมีชั้นของแม็กพูลลิ่ง (Max Pooling layer) แทรกไว้ใช้ลดมิติของแมทริกซ์ ส่วนสุดท้ายจะเป็นชั้นของโครงข่ายประสาทเทียมที่เชื่อมต่อกันไปทั้งหมด (Fully-connected layer) 3 ชั้น ในชั้นสุดท้ายจะเป็นการใช้ฟังก์ชันซอฟท์แม็ก (Softmax) ในการทำนายคลาส ดังแสดงในภาพที่ 2.2



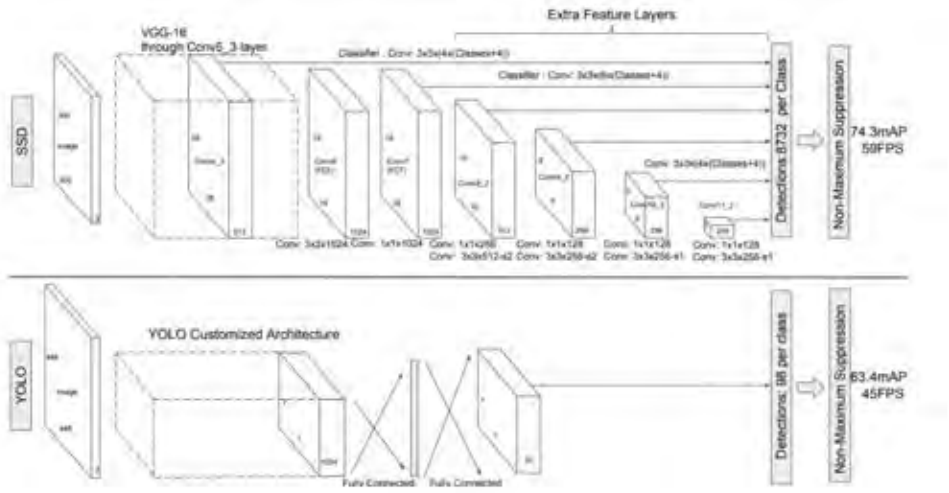
ภาพที่ 2.2 โครงสร้างเครือข่ายประสาทเทียมของงานวิจัย [5]

2.3 แบบจำลองการตรวจจับวัตถุ

เนื่องจากปัญหาการตรวจจับวัตถุเป็นปัญหาที่ได้มีการค้นหาวิธีในการแก้ไขมานานแล้ว จึงมีการคิดค้นแบบจำลองการตรวจจับวัตถุขึ้นมามากมาย มีแบบจำลองจำนวนมากที่ได้ถูกทำการเรียนรู้เข้ากับชุดข้อมูลที่มีคลาสที่อาจจะนำมาใช้ได้กับปัญหาที่ต้องการผลลัพธ์ที่มีคลาสอยู่ในชุดข้อมูลที่เราได้ทดลองทำให้แบบจำลองนั้นเรียนรู้ เช่น SSD_Mobilenet และ faster_RCNN เพื่อที่จะดูผลลัพธ์ของแบบจำลองที่ได้ทำการเรียนรู้แล้วกับชุดข้อมูลวีดีโอในฟิตเนสของเราว่าผลลัพธ์มีประสิทธิภาพหรือไม่

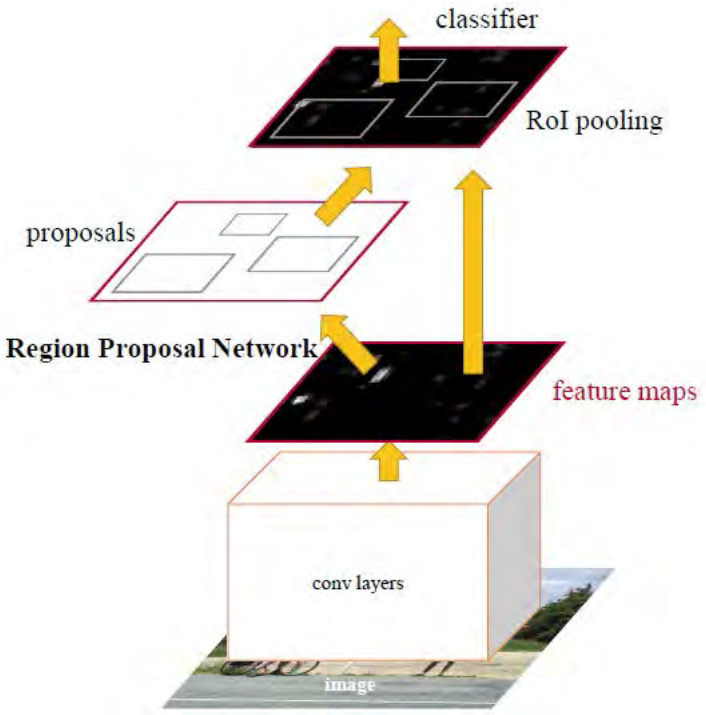
แบบจำลองการตรวจจับวัตถุ SSD_Mobilenet_COCO [7] คือแบบจำลองที่ใช้เทคนิคแบบซิงเกิ้ลช็อคมัลติบ็อกซ์ดีเทคเตอร์ (Single Shot Multibox Detector) [8] ซึ่งจะใช้วิธีในการกระจายขอบเขตของกล่องวัตถุไปในหลาย ๆ อัตราส่วนและขนาด จากนั้นในเวลาทำนายแบบจำลองจะทำการทำนายคะแนนสำหรับการมีอยู่ของวัตถุว่าเป็นประเภทใด ในอัตราส่วนและขนาดนั้น กล่าวได้ว่าแบบจำลองจะทำการรวมรวมการทำนายต่าง ๆ จากอัตราส่วนและขนาดที่ไม่เท่ากันมาเพื่อช่วยในการตรวจจับวัตถุที่มีขนาดแตกต่างกันไป กล่าวคือในชั้นแรก ๆ ของโครงข่ายประสาทเทียมจะตรวจจับวัตถุที่มีขนาดเล็กและ ชั้นที่ลึกลงไปจะตรวจจับวัตถุที่มีขนาดใหญ่ได้ โดยแบบจำลองนี้ได้ถูกนำไปเรียนรู้กับชุดข้อมูล COCO [9] ซึ่งมีคลาสของบุคคลที่เราต้องการตรวจจับในปัญหาที่ดำเนินการ ดังภาพที่ 2.3

แบบจำลองการตรวจจับวัตถุ YOLO [2] ดังที่ได้กล่าวไว้เบื้องต้นใน 2.1 สถาปัตยกรรมของแบบจำลอง YOLO นั้นจะนำรูปภาพที่รับเข้ามาแบ่งให้เป็นกริด (grid) ขนาด mxm และจากนั้นในแต่ละกริด จะคำนวณถึงขอบเขตที่วัตถุอยู่และความน่าจะเป็นที่จะเป็นวัตถุใด ๆ ส่งผลให้ YOLO นั้นมีข้อได้เปรียบหลายประการเช่น สามารถทำนายผลได้รวดเร็วกว่าเนื่องจากเรียนรู้ทั้งภาพ และสามารถจับเนื้อหาของภาพได้มากกว่ารวมถึง มีความสมดุล หรือก็คือแบบจำลองไม่จำเพาะเกินไปจนเรียนรู้ภาพใหม่ ๆ เวลานำไปใช้กับโดเมนอื่นไม่ได้



ภาพที่ 2.3 ภาพเปรียบเทียบโครงสร้างของ SSD กับ YOLO

แบบจำลองการตรวจจับวัตถุ faster_RCNN_COCO [10] คือแบบจำลองที่ใช้เทคนิคที่เรียกว่ารีเจียนโพรโพซอลเนตเวิร์ค (Region Proposal Network) ดังที่เห็นในภาพที่ 2.4 เป็นการแบ่งลักษณะสำคัญที่ได้มาจากชั้นคอนโวลูชั่นนอลไปคำนวณหาขอบเขตบริเวณของวัตถุและความน่าจะเป็นที่มีวัตถุอยู่ในบริเวณนั้นในเวลาเดียวกัน โดยแบบจำลองนี้ได้ถูกทำการเรียนรู้กับชุดข้อมูล COCO เช่นกันจึงสามารถนำมาใช้ได้



ภาพที่ 2.4 แสดงลักษณะการทำงานของ RPN

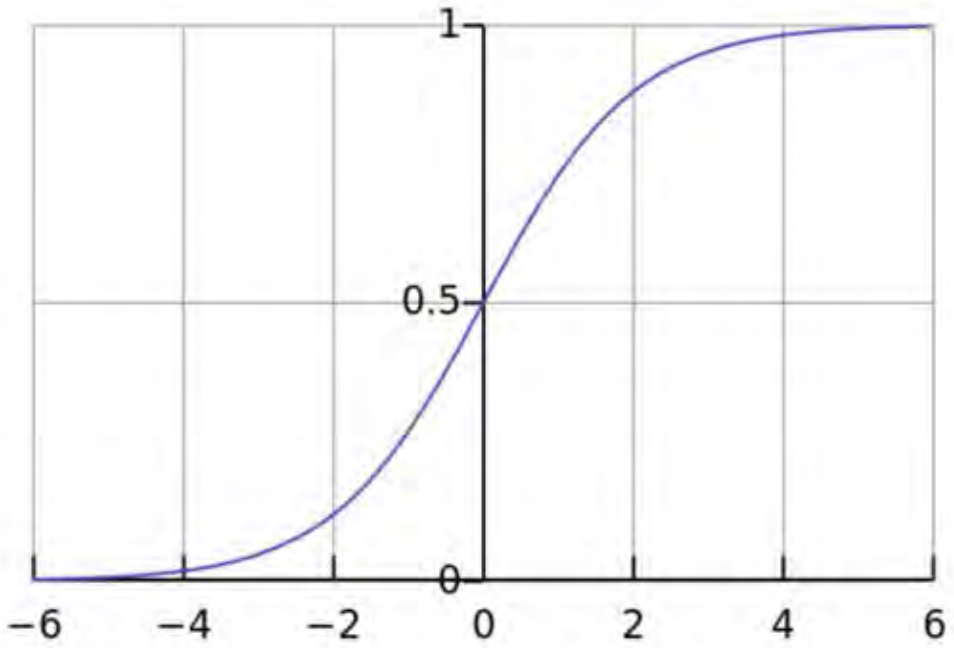
2.4 ฟังก์ชันซิกมอยด์

เนื่องจากในการตรวจจับวัตถุและการจำแนกวัตถุจะต้องการทำนายความน่าจะเป็นที่จะถูกต้องเป็นวัตถุนั้น ดังนั้นจึงมีการนำเอาฟังก์ชันซิกมอยด์ไปใส่ไว้ในชั้นสุดท้ายของโครงข่ายประสาทเทียมเพื่อคำนวณค่าที่ได้จากแบบจำลอง เป็นค่าที่วัดความน่าจะเป็นของคลาสนั้น ๆ โดยฟังก์ชันซิกมอยด์จะถูกนำไปใช้กับปัญหาที่เป็นไบนารี (Binary Classification) หรือก็คือมีแค่ 2 คลาส

ฟังก์ชันซิกมอยด์ (Sigmoid function) [11] เป็นฟังก์ชันที่นิยมนำมาใช้เป็นฟังก์ชันกระตุ้น ดังแสดงในสมการ 1

$$sig(x) = \frac{1}{1+e^{-x}} \qquad [1]$$

โดย x คือค่าที่รับเข้ามา (Input) และค่า $sig(x)$ คือความน่าจะเป็นที่จะเป็นวัตถุนั้น ซึ่งกราฟของฟังก์ชันซิกมอยด์จะมีลักษณะเป็นรูปตัว S ดังที่แสดงในภาพที่ 2.5



ภาพที่ 2.5 กราฟฟังก์ชันซิกมอยด์

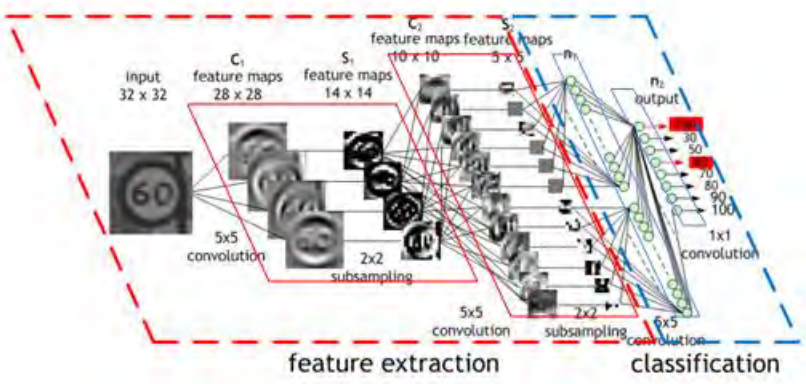
แหล่งที่มา: http://mathworld.wolfram.com/SigmoidFunction.html

จากภาพอธิบายได้ว่าการที่กราฟมีความชันอย่างมากบริเวณที่ -2 ถึง 2 หมายความว่าหากมีการเปลี่ยนแปลงของ x เพียงเล็กน้อย ค่าความน่าจะเป็นที่ได้ออกมาซึ่งเปรียบเทียบเท่ากับคลาสของเราจะมีค่าเปลี่ยนแปลงไปมาก ดังนั้นจึงเหมาะสมที่จะนำมาใช้ในการทำนายความน่าจะเป็นของวัตถุ

2.5 โครงข่ายประสาทแบบคอนโวลูชัน

โครงข่ายประสาทแบบคอนโวลูชัน [12] เป็นโครงข่ายประสาทเทียมที่ใช้กันอย่างแพร่หลายในการจำแนกรูปภาพ (Classify) รวมกลุ่มรูปภาพ (Clustering) การรู้จำวัตถุ (Object Recogniton) รวมไปถึงการเรียนรู้จำตัวอักษรจากภาพ (Optical Character Recognition) ทั้งจากตัวอักษรที่เป็นตัวพิมพ์หรือลายมือเขียน โดยถูกใช้เป็นตัวกลางในการสกัดลักษณะที่เด่น (Feature Extraction) ดั่งโครงสร้างในภาพที่ 2.6 อีกทั้งโครงข่ายประสาทคอนโวลูชั่นยังสามารถนำไปใช้กับคลื่นเสียงได้อีกด้วย

แนวคิดของโครงข่ายประสาทแบบคอนโวลูชั่นในการนำมาใช้กับข้อมูลจำพวกรูปภาพคือการประมวลผลรูปภาพให้เป็นเทนเซอร์ (Tensor) เนื่องจากรูปภาพคือตัวเลขระหว่าง 0 – 255 ในพิกเซลหนึ่งพิกเซล โดยจะรวมข้อมูลที่เข้ามากับคอนโวลูชั่นเคอร์เนล (Convolution Kernal) เพื่อให้ได้มาซึ่งฟีเจอร์แมพ หรือข้อมูลที่ได้ทำการรวมกันแล้ว จากนั้นจำนำข้อมูลที่ได้ไปซัพแซมปลิ้ง (Subsampling) ต่อเพื่อที่จะทำการลดขนาดมิติของข้อมูล



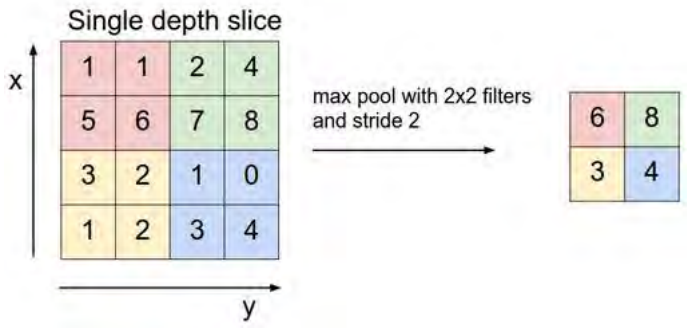
ภาพที่ 2.6 แสดงถึงโครงข่ายประสาทเทียมแบบคอนโวลูชัน

แหล่งที่มา: https://developer.nvidia.com/discover/convolutional-neural-network

โครงข่ายประสาทเทียมแบบคอนโวลูชันได้ชื่อมาจาก วิธีทางคณิตศาสตร์ [13] ที่เป็นการนำเอาฟังก์ชันสองฟังก์ชันมารวมกันให้เกิดฟังก์ชันใหม่ขึ้นมา เพื่อศึกษาว่าฟังก์ชันหนึ่งมีผลกระทบกับอีกฟังก์ชันหนึ่งอย่างไร โดยมีนิยามว่าเป็นอินทิกรัลของผลคูณทั้งสองสมการหลังจากที่มีสมการหนึ่งได้ทำการอินเวอร์สแล้ว ซึ่งคอนโวลูชั่นนอลของ f และ g สามารถเขียนได้ในรูปแบบของสมการดังในสมการ 2

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau. \qquad [2]$$

ซัพแซมปลิ้ง (Subsampling) [11] เป็นขั้นตอนในการลดขนาดข้อมูลขาเข้าในพื้นที่บริเวณหนึ่งให้ได้ค่า ๆ เดียวออกมาเช่นในกรณีของแม็กซ์พูลลิ่งจะทำการเลือกค่าที่มากที่สุดในพื้นที่นั้น ๆ นำมาเป็นค่าที่จะเป็นตัวแทนในชั้นต่อไปแทนการนำค่าทั้งหมดมาคิด เพื่อเป็นการลดขนาดมิติของเทนเซอร์และทำให้ประมวลได้เร็วขึ้น ภาพที่ 2.7 แสดงตัวอย่างการทำแม็กซ์พูลลิ่งในเมทริกซ์ขนาด 4 x 4 ด้วยฟิลเตอร์ขนาด 2 x 2 และใช้การขยับ (Stride) ทีละ 2 ซึ่งหมายถึงการเลื่อนฟิลเตอร์ไปทีละ 2 ช่องนั่นเอง



ภาพที่ 2.7 ตัวอย่างการทำแม็กซ์พูลลิ่งในเมทริกซ์ขนาด 4 x 4

แหล่งที่มา: https://developer.nvidia.com/discover/convolutional-neural-network

2.6 ไลบรารี บริการและโปรแกรมที่ใช้สำหรับพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิด

ผู้จัดทำได้ศึกษาไลบรารี (Library) เพื่อนำมาใช้สำหรับการพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดได้อย่างสะดวกและง่ายต่อการพัฒนาโปรแกรมยิ่งขึ้น โดยผู้จัดทำได้เลือกใช้ไลบรารีดังต่อไปนี้

2.6.1 Firebase

ภาพที่ 2.8 สัญลักษณ์ Firebase

Firebase คือ แพลตฟอร์มที่ให้บริการเครื่องมือต่างๆสำหรับใช้พัฒนา mobile และ web application ได้อย่างมีประสิทธิภาพ มีสัญลักษณ์ภาพที่ใช้ดังแสดงในภาพที่ 2.8 เริ่มพัฒนาโดยบริษัท Firebase และต่อมาพัฒนาโดยบริษัท Google บริการของ Firebase หลักเป็นฐานข้อมูลแบบเรียลไทม์ (Realtime Database) ซึ่งมี API ที่ช่วยให้นักพัฒนาในการสร้างแอพพลิเคชันที่ทำงานร่วมกันแบบเรียลไทม์ทางส่วนของฝั่งหลังบ้าน (Backend) กับฝั่งเซิร์ฟเวอร์ในด้านการจัดเก็บและซิงค์ข้อมูล Firebase ถูกออกแบบมาให้รองรับหลาย Platform ทั้ง IOS Application Android Application และ Web Application บริการของ Firebase ที่ให้บริการมีหลากหลายบริการ ตัวอย่างเช่น Firebase Analytics ให้บริการวิเคราะห์ข้อมูล Firebase Storage ให้บริการพื้นที่เก็บข้อมูล Firebase Authentication ใช้ในด้านการยืนยันตัวตน Firebase Realtime Database

ให้บริการด้านเก็บข้อมูลแบบเรียลไทม์ Cloud Messaging ให้บริการสำหรับการส่งข้อความและอื่น ๆ อีกหลายบริการ

2.6.2 Pyrebase

Pyrebase โดย James Childs-Maidment เป็นไลบรารี interface ภาษา Python ใช้ในการเขียนอ่านข้อมูลบน REST API ของ Firebase ในการจัดการฐานข้อมูลของ Firebase โดยใช้ภาษา Python รองรับ Python 3

2.6.3 OpenCV

OpenCV หรือ Open source Computer Vision ถูกพัฒนาโดยบริษัท Intel Corporation เป็นไลบรารีรวบรวมฟังก์ชันการเขียนโปรแกรมสำหรับการประมวลผลภาพ (Image Processing) คอมพิวเตอร์วิทัศน์ (Computer Vision) การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ และยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe สามารถประมวลผลภาพดิจิตอลได้ ทั้งภาพนิ่ง และภาพเคลื่อนไหว ตัวอย่างเช่น ภาพจากกล้องวิดีโอ มีฟังก์ชันสำหรับการจัดการข้อมูลภาพ และการประมวลผลภาพขั้นพื้นฐาน ซึ่งไลบรารี่ OpenCV นี้เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ open-source BSD ไลบรารี OpenCV มีอินเตอร์เฟสที่รองรับการพัฒนาโปรแกรมในหลายภาษา เช่น ภาษาC/C++ Python Java เป็นต้น และ OpenCV ยังสามารถนำไปต่อยอดพัฒนาโปรแกรมต่างๆได้ทั้งบนระบบปฏิบัติการ Window ระบบปฏิบัติการ Linux ระบบปฏิบัติการ Android และ ระบบปฏิบัติการ Mac

2.6.4 Keras

ภาพที่ 2.9 สัญลักษณ์ Keras

Keras คือ ไลบรารีการเรียนรู้เชิงลึก (deep learning library) ในภาษา Python ช่วยให้สร้างโมเดลการเรียนรู้เชิงลึกได้ง่ายยิ่งขึ้น มีสัญลักษณ์ภาพที่ใช้แทนดังแสดงในภาพที่ 2.9 ถูกพัฒนาโดย François Chollet อยู่ภายใต้ MIT license ซึ่ง Keras เป็น high-level neural networks API ที่ประมวลผลข้อมูลเกี่ยวกับ Deep Learning โดยเฉพาะ สามารถรันได้บน TensorFlow, CNTKs และ Theano และยังสามารถใช้งานได้กับทุกเวอร์ชั่นของภาษา Python

2.6.5 TensorFlow

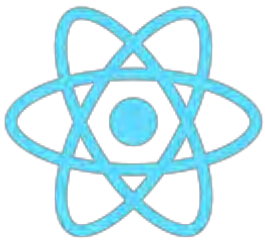
ภาพที่ 2.10 สัญลักษณ์ TensorFlow

TensorFlow คือ ไลบรารีการเรียนรู้เชิงลึก (deep learning library) ในภาษา Python แบบ Open Source ได้รับการพัฒนาโดยทีม Google Brain บริษัท Google มีสัญลักษณ์ภาพที่ใช้แทนดังแสดงในภาพที่ 2.10 ใช้สำหรับการเรียนรู้เชิงลึกและการพัฒนาแอปพลิเคชันโดยเฉพาะทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) สามารถใช้งานได้ฟรี รองรับภาษา Python 2 และ Python 3 สามารถทำงานได้บนหลายแพลตฟอร์มทั้งบนระบบปฏิบัติการ Windows ระบบปฏิบัติการ MacOS ระบบปฏิบัติการ Linux ระบบปฏิบัติการบนมือถือทั้ง iOS และ Android

2.6.6 Neural Compute Stick SDK

Intel® Movidius ™ Neural Compute Stick Development Kit หรือ NCS SDK คือ ชุดเครื่องมือพัฒนาซอฟต์แวร์สำหรับผู้ใช้งาน Intel® Movidius ™ Neural Compute Stick เพื่อให้นักพัฒนาสามารถสร้างซอฟต์แวร์ที่ใช้ประโยชน์จาก neural network ของตัวฮาร์ดแวร์ Intel® Movidius ™ Neural Compute Stick ซึ่งรองรับโมเดลของ TensorFlow และ Caffe ชุดพัฒนา NCS SDK ช่วยให้การ prototyping และการ deployment สำหรับ deep neural networks (DNNs) บนอุปกรณ์ประมวลผล neural compute device ที่ใช้งานด้วยกันได้ เช่น Intel® Movidius ™ Neural Compute Stick เป็นไปได้อย่างรวดเร็ว NCS SDK ประกอบด้วยชุดเครื่องมือซอฟต์แวร์เพื่อรวบรวมโปรไฟล์(profile) และตรวจสอบ(validate) DNNs รวมถึง Intel®Movidius ™ Neural Compute API (Intel® Movidius ™ NCAPI) สำหรับการพัฒนาแอปพลิเคชันในภาษา C / C ++ หรือภาษา Python

2.6.7 React-native



ภาพที่ 2.11 สัญลักษณ์ React-Native

React-Native ถูกพัฒนาขึ้นโดยทีมงานบริษัท Facebook เป็น JavaScript Framework แบบ Open Source มีสัญลักษณ์ภาพที่ใช้แทนดังแสดงในภาพที่ 2.11 ใช้สำหรับพัฒนาแอปพลิเคชันมือถือแบบข้ามแพลตฟอร์ม (Cross-Platform Mobile Application) ช่วยให้นักพัฒนาสามารถพัฒนาแอปพลิเคชันที่ใช้งานได้ทั้งบนระบบปฏิบัติการ Android และระบบปฏิบัติการ iOS โดยใช้ภาษา JavaScript เป็นหลักในการพัฒนา

2.7 อุปกรณ์ที่เกี่ยวข้อง

2.7.1 Raspberry Pi

เป็นอุปกรณ์ที่ใช้เพื่อเป็นเกตเวย์ (Gateway) หรือเอดจ์ดีไวซ์ (Edge device) หรืออุปกรณ์ที่ทำหน้าที่ประมวลผลข้อมูลดิบก่อนที่จะทำการส่งข้อมูลต่อไป สาเหตุที่ต้องใช้เนื่องจากการทำระบบที่ต้องการความรวดเร็วในการทำนายผลแบบเรียลไทม์ เช่น ระบบตรวจจับวัตถุหรือจำแนกวัตถุควรมีการทำนายผลที่อุปกรณ์ที่ได้รับข้อมูลมาเลย เพื่อให้ระยะเวลาการส่งข้อมูลไปยังจุดต่อไปลดน้อยลง

ราสพ์เบอรี่ไพ (Raspberry Pi) [14] เป็นอุปกรณ์ขนาดเล็กที่ทำหน้าที่เหมือนเป็นคอมพิวเตอร์เพียงแต่ว่ามีขนาดเล็กและมีประสิทธิภาพในการประมวลผลที่ต่ำกว่าและราคาถูกกว่าจึงถูกนำมาใช้ในการรับข้อมูลในงานนี้ ซึ่งราสพ์เบอรี่ไพนั้นใช้โปรเซสเซอร์ (Processor) Broadcom BCM2837 64bit ARMv8 QUAD Core 64bit Processor (1.2GHz) และมีเมมโมรี่ (RAM) 1 กิ๊กกะไบต์ ดังแสดงในภาพที่ 2.12

ภาพที่ 2.12 Raspberry PI 3b+

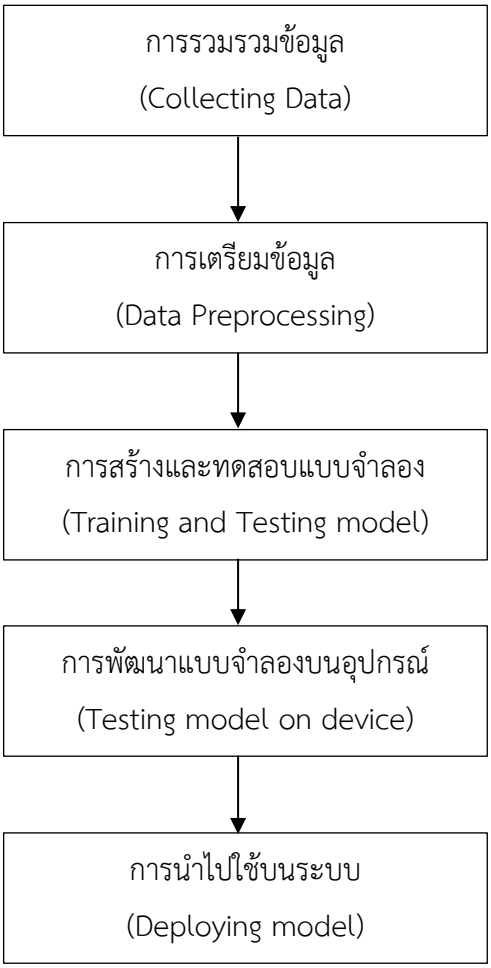
2.7.2 Intel Movidius: Neural Compute Stick

เป็นอุปกรณ์ที่มีรูปร่างเหมือนกับยูเอสบีขนาดเล็ก ดังภาพที่ 2.13 ซึ่งผู้ผลิตคืออินเทล ภายในประกอบด้วยหน่วยประมวลผล Intel® Movidius™ Vision Processing Unit (VPU) ซึ่งเป็นหน่วยประมวลผลที่อินเทลอ้างว่าผลิตออกมาเพื่อใช้ในการเรียนรู้เชิงลึกที่เป็นภาพโดยเฉพาะ ด้วยราคาที่ไม่แพง ขนาดที่เล็ก และไม่กินไฟเราจึงสามารถนำมาใช้เพื่อเพิ่มประสิทธิภาพในการคำนวณให้กับเกตเวย์ของเราได้

ภาพที่ 2.13 Intel movidius ที่เชื่อมต่อกับ Raspberry Pi

บทที่ 3
การรวบรวม การเตรียมข้อมูลและการสร้างแบบจำลอง

ในบทนี้จะกล่าวถึงวิธีการรวบรวมชุดข้อมูลภาพวิดีโอจากกล้องวงจรปิด การเตรียมข้อมูล การสร้าง การวิเคราะห์ข้อมูลด้วยวิธีที่กล่าวในบทก่อนหน้าและทดสอบแบบจำลองต่างๆ การพัฒนาแบบจำลองบนอุปกรณ์และการนำไปใช้ในระบบ โดยภาพรวมของขั้นตอนทั้งหมดนี้ สามารถสรุปได้เป็นดังภาพที่ 3.1



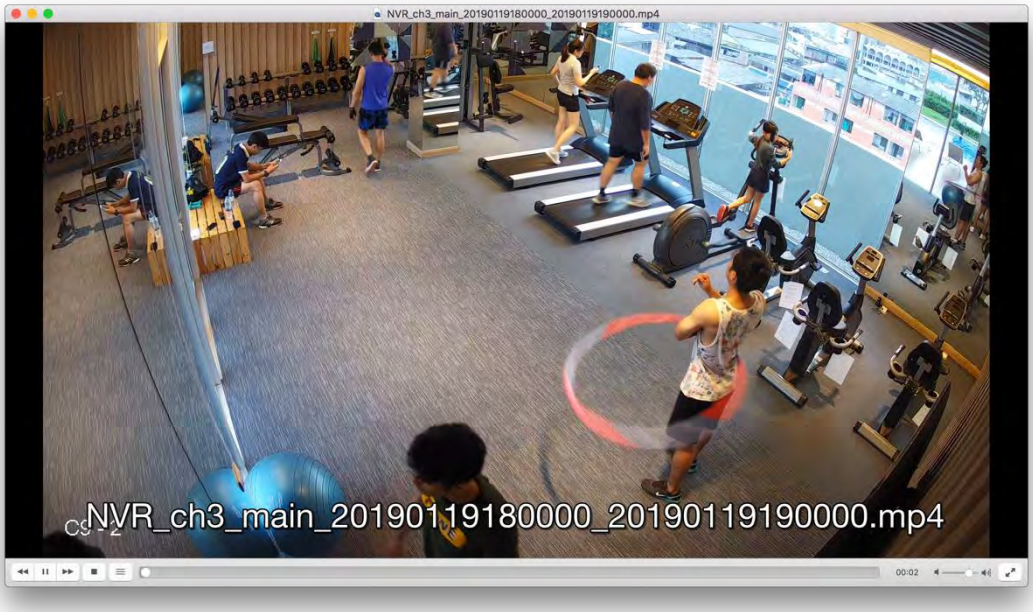
ภาพที่ 3.1 ภาพรวมของขั้นตอนในการรวบรวม เตรียมข้อมูลและการสร้างแบบจำลอง

3.1 การรวบรวมข้อมูล

ผู้จัดทำได้ทำการรวบรวมชุดข้อมูลซึ่งเป็นวีดีโอที่ได้จากกล้องวงจรปิดในฟิตเนสที่คอนโดแอสปายงามวงศ์วาน โดยชุดข้อมูลที่ได้มาเป็นกล้องวงจรปิดที่ไม่มีการเปลี่ยนทิศทาง และตั้งอยู่ที่มุมเดิมตลอดเวลา ซึ่งทางผู้จัดทำสนใจที่จะระบุตำแหน่งที่ตรงเครื่องเล่นว่ามีการใช้งานอยู่หรือไม่ ดังภาพที่ 3.2

ชุดข้อมูลวิดีโอจะถูกแบ่งแต่ละไฟล์ให้มีระยะเวลา 1 ชั่วโมงขนาด 2 กิกะไบต์ โดยนามสกุลของไฟล์คือ MPEG4 มีมิติ 2688 × 1520 ซึ่งถือว่าให้ความละเอียดระดับ HD ดังภาพที่ 3.3

ภาพที่ 3.2 ชุดข้อมูลวีดีโอที่ผู้จัดทำรวบรวมมาจากกล้องวงจรปิดในฟิตเนส

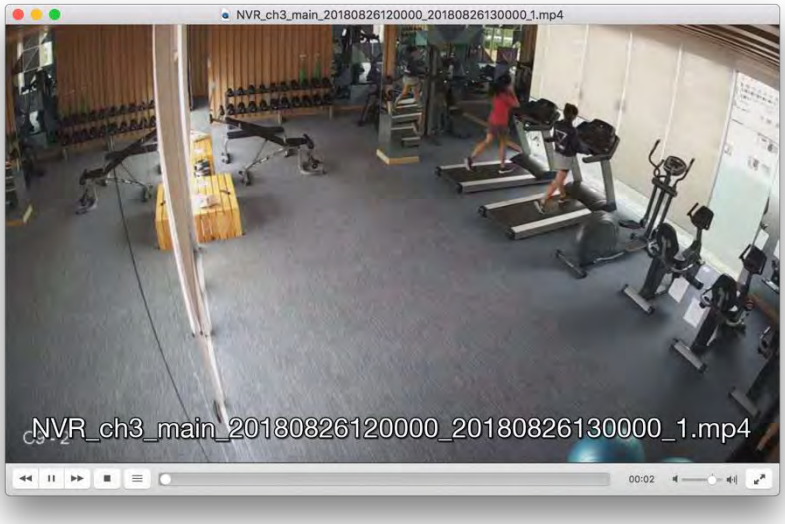


ภาพที่ 3.3 วีดีโอจากกล้องวงจรปิดในพื้นที่บริเวณฟิตเนส

3.2 การเตรียมข้อมูล

ข้อมูลที่ทางผู้จัดทำได้มาถือว่ามีจำนวนที่มากเพียงพอต่อการนำมาสร้างให้แบบจำลองเรียนรู้ แต่ข้อมูลวีดีโอที่ได้มานั้นมีขนาดใหญ่เกินไปและมีความละเอียดมากเกินความจำเป็นที่จะต้องใช้ อีกทั้งการที่วีดีโอมีมิติ ความละเอียด รวมถึงขนาดที่ใหญ่นั้นแสดงว่ามีจำนวนพิกเซลเยอะ แบบจำลองที่ได้

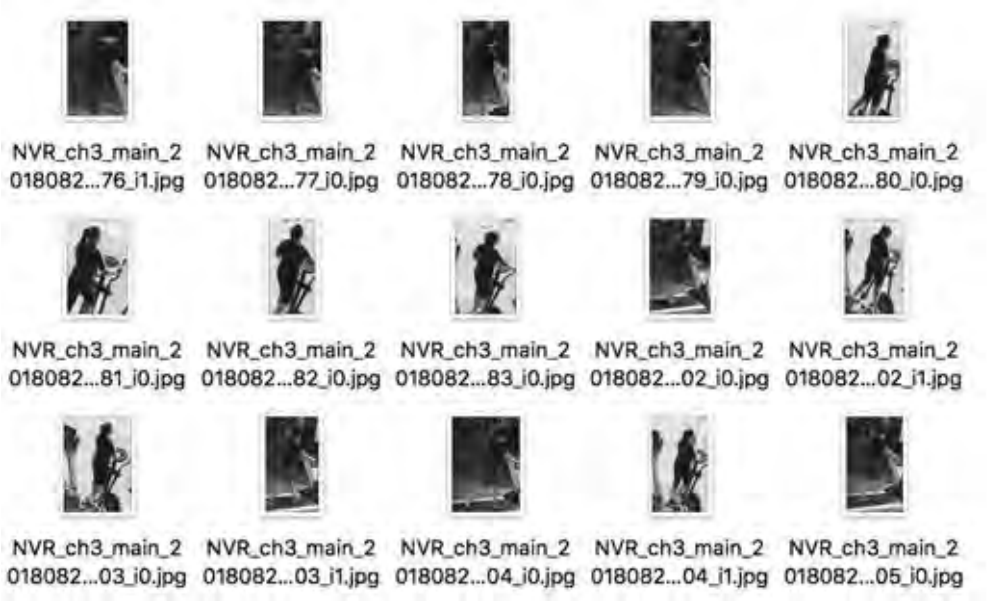
จะทำงานช้าส่งผลให้การทำนายการตรวจจับวัตถุหรือการจำแนกวัตถุไม่มีประสิทธิภาพ ทางผู้จัดทำจึงคิดว่าเป็นผลดีหากทำการลดขนาดความละเอียดลงเหลือ 848 x 480 ดังภาพที่ 3.4



ภาพที่ 3.4 วีดีโอที่ได้ทำการลดขนาด

เนื่องจากผู้จัดทำได้วางแผนไว้ที่จะทดสอบทำแบบจำลองออกมา 2 รูปแบบคือแบบจำลองตรวจจับคน (Human Detection) และแบบจำลองจำแนกคน (Human Classification) สำหรับแบบจำลองในแบบที่ 2 ผู้จัดทำจึงจำเป็นที่จะต้องกำกับภาพที่จะให้แบบจำลองที่ผู้จัดทำจะจัดสร้างขึ้นนั้นเรียนรู้ โดยผู้จัดทำได้แบ่งชุดภาพออก 2 จำพวกได้แก่ พื้นหลังกับบุคคลโดยมีรูปพื้นหลังดังแสดงในภาพที่ 3.5 และ 3.6

ภาพที่ 3.5 ตัวอย่างรูปภาพพื้นหลังที่ไม่มีคน



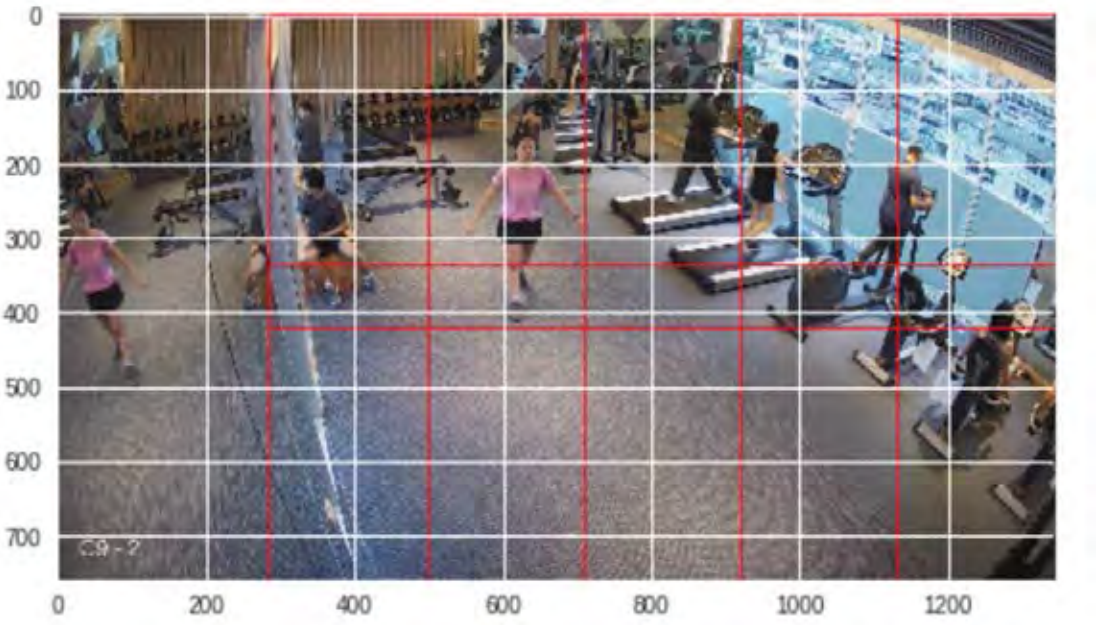
ภาพที่ 3.6 ตัวอย่างรูปภาพที่เป็นคน

3.3 การสร้างและทดสอบแบบจำลอง

ในบทนี้ผู้จัดทำจะขอกล่าวถึงการสร้างแบบจำลองทั้งแบบตรวจจับ (Detection) และแบบจำแนก (Classification) เพื่อนำมาทดสอบว่าแบบไหนได้ประสิทธิภาพที่ดีกว่าและเหมาะกับระบบมากกว่ากัน

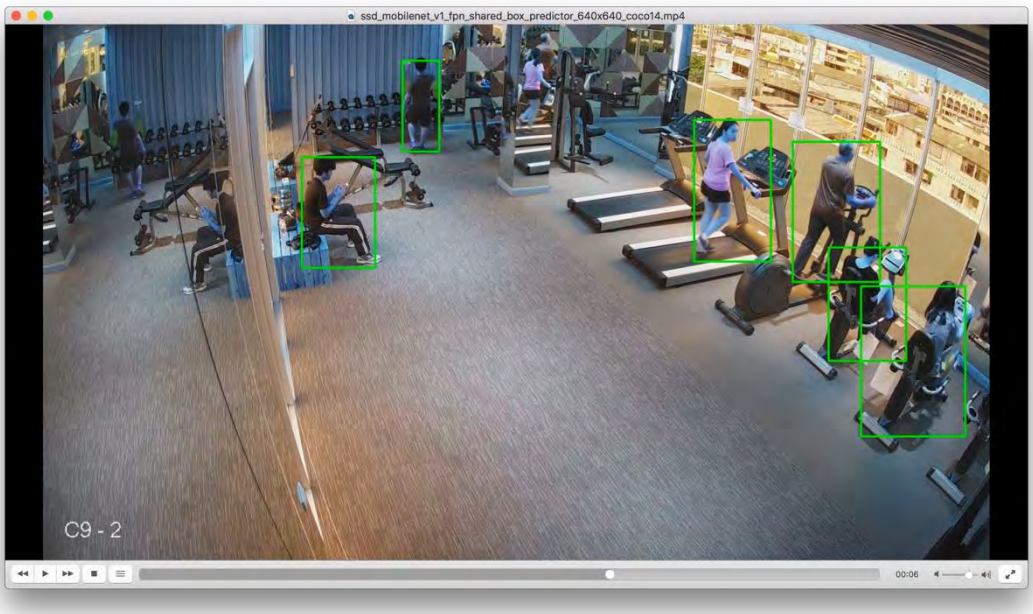
3.3.1 แบบจำลองตรวจจับคน

ในส่วนของแบบจำลองตรวจจับคนผู้จัดทำได้ทำการนำเอาแบบจำลองที่ได้มีการเรียนรู้ให้รู้จักคนจากชุดข้อมูลต่าง ๆ ที่มีตามที่ได้อ้างไว้ในบทที่ 2 ข้างต้น [7] [10] มาใช้เพื่อเป็นการวัดประสิทธิภาพของโมเดลกับชุดข้อมูลที่ผู้จัดทำได้ทำการหามาว่ามีประสิทธิภาพและสามารถนำมาแก้โจทย์ปัญหาที่ได้ตั้งไว้หรือไม่ เนื่องจากแบบจำลองที่ได้ทำการเรียนรู้ไว้แล้วที่ผู้จัดทำได้นำมาใช้ไม่สามารถรองรับรูปภาพที่มีความละเอียดที่มากได้ ทางผู้จัดทำจึงได้ทำปรับแต่งวิธีการส่งค่าอินพุธเข้าไปในแบบจำลองโดนทำการแบ่งรูปภาพในแต่ละเฟรมออกเป็น 10 ส่วนย่อย ๆ และละทิ้งบริเวณด้านข้างของเฟรมในวีดีโอไปเนื่องจากเป็นกระจกที่สะท้อนภาพในห้องฟิตเนส ดังในภาพที่ 3.7



ภาพที่ 3.7 ตัวอย่างการแบ่งภาพในเฟรมของวีดีโอ

จากนั้นผู้จัดทำได้ทำการเขียนชุดคำสั่ง ในภาพที่ 3.9 ที่จะไปดาวน์โหลดข้อมูลแบบจำลองที่ได้ทำการเรียนรู้ไว้แล้วทั้งโครงสร้างและค่าที่แบบจำลองได้เรียนรู้ไว้แล้วมาใช้ จากนั้นให้แบบจำลองได้ทำการตรวจจับบริเวณที่มีบุคคลอยู่จากนั้น ผู้จัดทำจึงได้ตีกรอบและเขียนวีดีโอที่เป็นผลสรุปจากการทำนายไว้ ดังภาพที่ 3.8 ด้านล่างนี้



ภาพที่ 3.8 ตัวอย่างของผลลัพท์ที่ได้จากการตรวจจับของแบบจำลอง

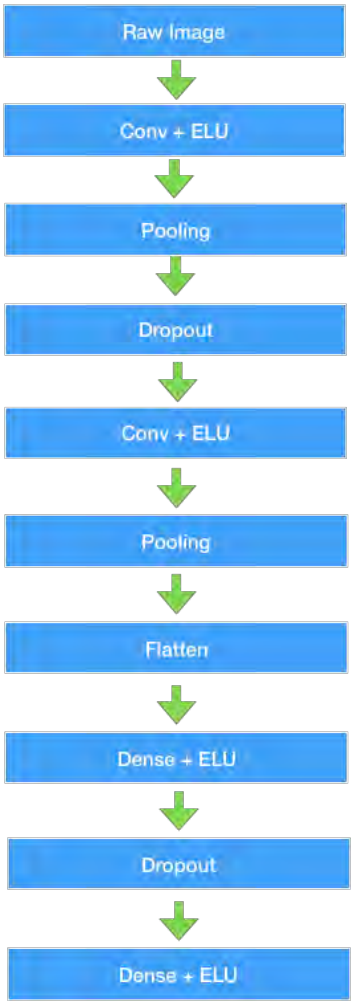
```
class TFAPI(object):
    def __init__(self, model_name):
        self._load_model(model_name)

    def _load_model(self, model_name):
        with tf.device("/device:GPU:0"):
            model_path = "nets/{}".format(model_name)
            model_file = "nets/{}.tar.gz".format(model_name)
            if not os.path.exists(model_path):
                print("model {} is not exists. Trying to download
model.".format(model_name))
                DOWNLOAD_URL =
'http://download.tensorflow.org/models/object_detection/{}.tar.gz'.fo
rmat(model_name)
                opener = urllib.request.URLopener()
                opener.retrieve(DOWNLOAD_URL, model_file)
                tar_file = tarfile.open(model_file)
                for file in tar_file.getmembers():
                    file_name = os.path.basename(file.name)
                    if 'frozen_inference_graph.pb' in file_name:
                        tar_file.extract(file, 'nets/')
            else:
                print("Loading model {} ...".format(model_name))
            GRAPH_PATH = os.path.join(model_path,
'frozen_inference_graph.pb')
            self._net = tf.Graph()
            with self._net.as_default():
                od_graph_def = tf.GraphDef()
                with tf.gfile.GFile(GRAPH_PATH, 'rb') as fid:
                    serialized_graph = fid.read()
                    od_graph_def.ParseFromString(serialized_graph)
                    tf.import_graph_def(od_graph_def, name='')

    def predict(self, image):
        if len(image.shape) == 3:
            image = np.expand_dims(image, 0)
        with self._net.as_default():
            with tf.Session() as sess:
                ops = tf.get_default_graph().get_operations()
                all_tensor_names = {output.name for op in ops for
output in op.outputs}
                tensor_dict = {}
                for key in [
                    'num_detections', 'detection_boxes',
'detection_scores',
                    'detection_classes', 'detection_masks'
                ]:
                    tensor_name = key + ':0'
                    if tensor_name in all_tensor_names:
                        tensor_dict[key] =
tf.get_default_graph().get_tensor_by_name(tensor_name)
                if 'detection_masks' in tensor_dict:
                    detection_boxes =
tf.squeeze(tensor_dict['detection_boxes'], [0])
                    detection_masks =
tf.squeeze(tensor_dict['detection_masks'], [0])
                    real_num_detection =
tf.cast(tensor_dict['num_detections'][0], tf.int32)
                    detection_boxes = tf.slice(detection_boxes, [0,
0], [real_num_detection, -1])
```

```
                    detection_masks = tf.slice(detection_masks, [0,
0, 0], [real_num_detection, -1, -1])
                    detection_masks_reframed =
utils_ops.reframe_box_masks_to_image_masks(
                      detection_masks, detection_boxes,
image.shape[0], image.shape[1])

                    detection_masks_reframed =
tf.cast(tf.greater(detection_masks_reframed, 0.5), tf.uint8)
                    tensor_dict['detection_masks'] =
tf.expand_dims(detection_masks_reframed, 0)

                image_tensor =
tf.get_default_graph().get_tensor_by_name('image_tensor:0')
                output_dict = sess.run(tensor_dict,
feed_dict={image_tensor: image})
                output_dict['num_detections'] =
output_dict['num_detections']
                output_dict['detection_classes'] = output_dict[
                    'detection_classes'].astype(np.uint8)
                output_dict['detection_boxes'] =
output_dict['detection_boxes']
                output_dict['detection_scores'] =
output_dict['detection_scores']
                if 'detection_masks' in output_dict:
                    output_dict['detection_masks'] =
output_dict['detection_masks']
            return output_dict
LABELS_PATH = os.path.join('/usr/local/lib/python3.6/dist-
packages/tensorflow/models/research/object_detection/data',
'mscoco_label_map.pbtxt')
category_index =
label_map_util.create_category_index_from_labelmap(LABELS_PATH,
use_display_name=True)
```

ภาพที่ 3.9 ตัวอย่างซอรส์โค้ดที่ใช้ในการดาวน์โหลดแบบจำลองที่ได้เรียนรู้ไว้แล้วและทำนาย

3.3.2 แบบจำลองการจำแนกคน

เนื่องจากปัญหาของผู้จัดทำคือการแยกแยะว่าบริเวณเครื่องเล่นนั้นได้มีคนใช้งานอยู่หรือไม่ จึงได้เล็งเห็นว่าเพื่อประสิทธิภาพและความรวดเร็วของการทำนายผลสามารถใช้การจำแนกว่ามีคนหรือเป็นเพียงพื้นหลังได้ ทำให้ผู้จัดทำลองสร้างแบบจำลองการจำแนกคน (Human Classifier) ขึ้นมาควบคู่ไปกับแบบจำลองตรวจจับคน โดยใช้รูปภาพที่ได้เตรียมไว้ในหัวข้อที่ 3.2 และได้ทำการสร้างแบบจำลองขึ้นมาโดยมีส่วนของการสกัดลักษณะสำคัญที่ใช้โครงข่ายประสาทแบบคอนโวลูชัน ดังแสดงในภาพที่ 3.10



ภาพที่ 3.10 โครงสร้างของแบบจำลองจำแนกรูปภาพที่ผู้จัดทำสร้าง

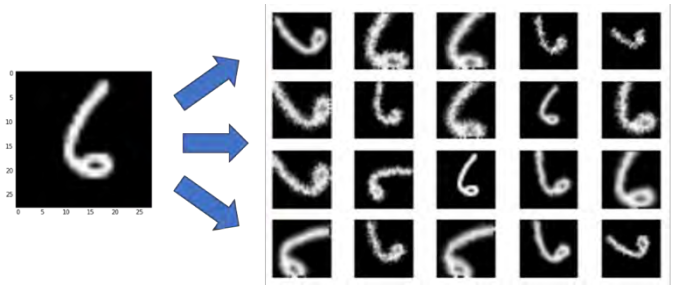
จากนั้นผู้จัดทำได้ทำนำเอาผลลัพธ์ที่ได้จากชั้นคอนโวลูชันไปเชื่อมต่อกับโครงข่ายประสาทแบบเชื่อมถึงทั้งหมด (Fully-Connected Neural Network) ที่มีชั้นสุดท้ายใช้ฟังก์ชันซิกมอย์ดเพื่อให้เป็นส่วนที่ทำนายผลของรูปภาพว่ามีคนอยู่หรือไม่ เนื่องจากว่าเป็นการจำแนกคน จึงไม่จำเป็นต่อการหากล่องขอบเขตที่อยู่ของวัตถุ (Localization) เหมือนกับการทำตรวจจับส่งผลให้การทำงานของแบบจำลองมีความรวดเร็วมากขึ้น ตัวอย่างของชุดคำสั่งในการสร้างแบบจำลองดังภาพที่ 3.11

```
def model_fn(features, labels, mode=TRAIN):
    drop_rate = 0.5 if mode == TRAIN else 0.
    device_type = "CPU:0" if mode == PREDICT else "GPU:0"
    reg = tf.keras.regularizers.l1_l2(l1=1e-3, l2=1e-3)

    def conv_block(x, ind, filters=32, patch_size=3, conv_strides=1,
drop_rate=0.5, reg=None, actv="elu"):
        x = Conv2D(filters, patch_size, conv_strides, padding="SAME",
kernel_regularizer=reg, activation=actv,
name="conv{}".format(ind)).apply(x)
```

```
        x = MaxPool2D(2, name="pool{}".format(ind)).apply(x)
        x = Dropout(rate=drop_rate,
name="drop{}".format(ind)).apply(x)
        return x

    with tf.device(device_type):
        with tf.variable_scope("cnn_model"):
            if mode == PREDICT:
                features = features["input"]
            x = features / 255. - 0.5
            x = conv_block(x, 1, 32, 5, 2, drop_rate, reg, "elu")
            x = conv_block(x, 2, 32, 5, 2, drop_rate, reg, "elu")
            x = Flatten(name="flat1").apply(x)
            pred = Dense(1, activation="sigmoid",
kernel_regularizer=reg, name="out").apply(x)
            if mode == PREDICT:
                return tf.estimator.EstimatorSpec( mode=PREDICT,
predictions=pred)
            loss =
tf.reduce_mean(tf.nn.weighted_cross_entropy_with_logits(targets=label
s, logits=pred, pos_weight=2))
            loss = tf.identity(loss, name="ce_loss")
            adam = tf.train.AdamOptimizer(learning_rate=1e-3)
            train_step = adam.minimize(loss,
global_step=tf.train.get_global_step())

    with tf.variable_scope("metric"):
        accuracy = tf.metrics.accuracy(labels, pred>0.5, name="acc")
        metrics = {'accuracy': accuracy}
    return tf.estimator.EstimatorSpec(
        mode=mode,
        predictions=pred,
        loss=loss,
        train_op=train_step,
        eval_metric_ops=metrics,
    )
```

ภาพที่ 3.11 ตัวอย่างซอร์สโค้ดสร้างแบบจำลอง

เพื่อเพิ่มประสิทธิภาพของแบบจำลองทางผู้จัดทำจึงได้พยายามที่จะทำให้ชุดข้อมูลที่เข้าไปมีความหลากหลายมากขึ้นโดยได้ใช้เทคนิคการเพิ่มข้อมูล (Data Augmentation) [14] ซึ่งเป็นเทคนิคที่ใช้ในกัน โดยจะนำรูปภาพที่ได้มาทำการหมุน ปรับขนาด ตัดบางส่วนหรือการใส่นอยซ์รบกวนเข้าไป ดังตัวอย่างในภาพที่ 3.12 โดยผู้จัดทำได้ทำการหมุนรูปและปรับแสงสว่างในรูป



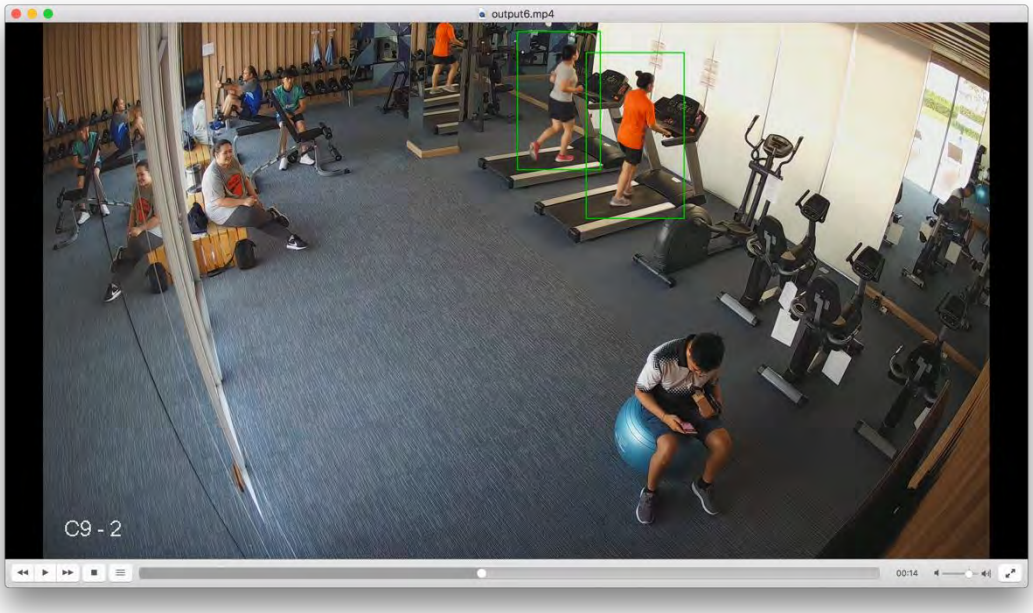
ภาพที่ 3.12 ตัวอย่างแสดงการทำการเพิ่มข้อมูล

แหล่งที่มา: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

เมื่อทำการวางโครงสร้างของโครงข่ายประสาทเทียมเสร็จแล้วจึงกำหนดให้แบบจำลองได้มีการเรียนรู้ทั้งหมด 100,000 รอบการเรียนรู้และได้ทำการเปลี่ยนไฮเปอร์พารามีเตอร์ต่าง ๆ เช่น ฟิวเตอร์ ดรอปเรต สไตร์ด เพื่อหาแบบจำลองการจำแนกบุคคลดังภาพที่ 3.13 ที่มีค่าความความถูกต้องและรวดเร็วมากที่สุด โดยค่าความถูกต้อง (Accuracy) [15] คำนวณได้จากสมการที่ 3

$$\text{ค่าความถูกต้อง} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \qquad [3]$$

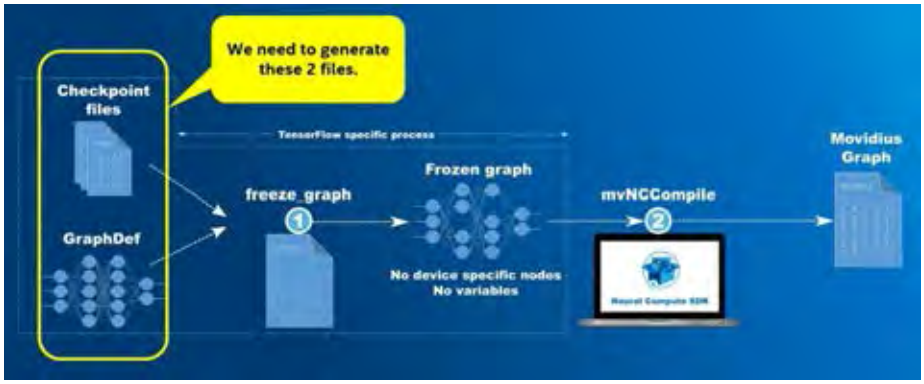
โดยที่ True Positive (TP) คือ แบบจำลองทำนายว่าเป็นขั้วบวก และขั้วจริงของข้อมูลเป็นขั้วบวก
True Negative (TN) คือ แบบจำลองทำนายว่าเป็นขั้วลบ และขั้วจริงของข้อมูลเป็นขั้วลบ
False Positive (FP) คือ แบบจำลองทำนายว่าเป็นขั้วบวก แต่ขั้วจริงของข้อมูลเป็นขั้วลบ
False Negative (FN) คือ แบบจำลองทำนายว่าเป็นขั้วลบ แต่ขั้วจริงของข้อมูลเป็นขั้วบวก



ภาพที่ 3.13 ตัวอย่างผลลัพธ์ของแบบจำลองจำแนกบุคคล

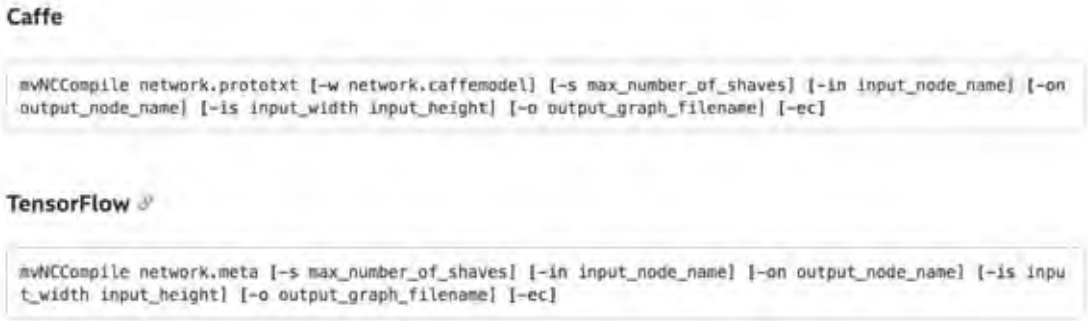
3.4 การพัฒนาแบบจำลองในอุปกรณ์

หลังจากที่ได้แบบจำลองที่มีประสิทธิภาพและความถูกต้องที่พอใจแล้ว ทางผู้จัดทำได้ทำการจัดเก็บข้อมูลของแบบจำลองเพื่อให้สามารถใช้เพื่อการทำนายได้โดยที่ไม่ต้องเรียนรู้ใหม่ โดยจะได้โมเดลที่จัดเก็บเป็นไฟล์นามสกุล protobuff ซึ่งทำให้สามารถโหลดไฟล์ที่เป็นโครงสร้างของแบบจำลองและค่าของแบบจำลองไปประมวลผลที่ไหนก็ได้ และเนื่องจากว่าทางผู้จัดทำต้องการใช้อินเทลโมวิดิอุสในการช่วยเพิ่มประสิทธิภาพของการทำนายผลจากแบบจำลอง เราจึงต้องใช้ NCSDK ซึ่งเป็นซอฟต์แวร์เดวิลอปเมนต์คิท ของตัวอินเทลโมวิดิอุสในการแปลงตัวโมเดลให้อยู่ในรูปแบบที่โมวิดิอุสจะอ่านและประมวลผลได้ ดังภาพที่ 3.14



ภาพที่ 3.14 ภาพแสดงกระบวนการแปลงแบบจำลองจาก Tensorflow เป็นโมวิดิอุสกราฟ

จากนั้นผู้จัดทำทำการใช้คำสั่งของ NCSDK เพื่อทำการแปลงไฟล์จากโฟรเซนกราฟให้เป็นโมวิดิอุสกราฟ ซึ่งตัว SDK นั้นสามารถแปลงได้ทั้งแบบจำลองที่สร้างด้วย Tensorflow และ Caffe คำสั่งที่ใช้แปลงกราฟอยู่ในภาพที่ 3.15



ภาพที่ 3.15 ตัวอย่างของคำสั่งที่ใช้แปลงกราฟ

แหล่งที่มา: https://movidius.github.io/ncsdk/tools/compile.html

3.5 การนำไปใช้ในระบบ

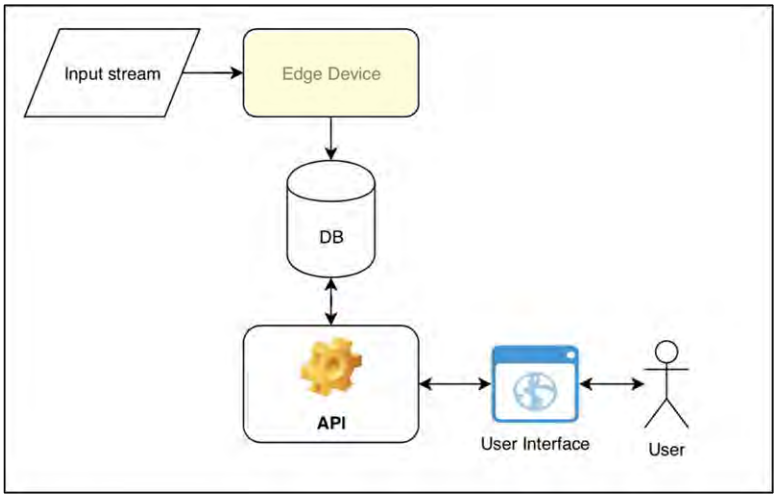
เมื่อทำการแปลงกราฟเสร็จสิ้น ทางผู้จัดทำได้ทำการเขียนชุดคำสั่งที่รับข้อมูลวีดีโอจาก IP camera ผ่าน RTSP protocol [16] ดังแสดงในภาพที่ 3.16 จากนั้นประมวลผลและทำการคืนรูปภาพเป็นเฟรม ๆ ให้ผ่านเข้าไปยังแบบจำลองที่ได้สร้างไว้และส่งผลลัพธ์ที่ได้จากการทำนายขึ้นไปบนระบบที่อยู่บนคลาวด์

```
class VideoConnection(object):
    def __init__(self, rtsp_path, scale=1.0):
        self._rtsp_path = rtsp_path
        self._cap = cv2.VideoCapture(rtsp_path)
        self._cap.set(3, 1)
        self._scale = scale

    def get_lastest(self):
        self._cap.set(1,-1)
        ret, frame = self._cap.read()
        if self._scale != 1.0:
            cv2.resize(frame, (0,0), fx=self._scale, fy=self._scale)
        return ret, frame

    def get(self, skip=0, wait_for_skip=0):
        while True:
            try:
                ts = time.time()
                img = self._cap.grab()
                for i in range(skip):
                    img = self._cap.grab()
                    if time.time() - ts > wait_for_skip:
                        break
                ret, frame = self._cap.retrieve()
                h, w, _ = frame.shape
                if self._scale != 1.0:
```

```
                    frame = cv2.resize(frame, (0,0), fx=self._scale,
fy=self._scale)
                break
            except:
                print("Video Connection lost. Trying to reconnect the
camera...")
                self._cap = cv2.VideoCapture(self._rtsp_path)
        return ret, frame

    def match_region(self, centers, regions):
        position_sets = []
        for c in centers:
            position_sets.append([])
            for r in range(len(regions)):
                rg = regions[r]
                if rg[0] < c[0] and c[0] < rg[2] and rg[1] < c[1] and
c[1] < rg[3]:
                    position_sets[-1].append(r)
        return position_sets
```

ภาพที่ 3.16 ตัวอย่างซอร์สโค้ดที่รับวีดีโอถ่ายทอดสดจากกล้อง

สถาปัตยกรรมของระบบมีโครงสร้างดังภาพที่ 3.17 แสดงถึงการใช้งานอุปกรณ์ที่รับภาพรวมกับฐานข้อมูลที่ตั้งอยู่บนคลาวด์ และระบบที่เป็นส่วนประสานผู้ใช้งานเข้าด้วยกันดังนั้นเมื่อได้ผลการทำนายตัวอุปกรณ์ที่เป็นเกตเวย์จะทำหน้าที่ส่งผลการทำนายขึ้นไปที่ฐานข้อมูลส่งผลให้เร็วกว่าไปประมวลผลบนระบบที่คลาวด์ และระบบ API สามารถคำนวณสถานะการใช้งานส่งให้ส่วนประสานงานได้ในทันที



ภาพที่ 3.17 สถาปัตยกรรมของระบบ

บทที่ 4
การพัฒนาระบบและผลการทดสอบ

ในบทนี้จะกล่าวถึง ขั้นตอนการพัฒนาระบบทั้งหมดทั้งส่วนของอุปกรณ์ที่รับข้อมูล ขั้นตอนการประมวลผล การจัดเก็บข้อมูล การพัฒนาแอปพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟซ การพัฒนาส่วนประสานผู้ใช้งาน และยังกล่าวถึงผลการทดสอบระบบตรวจจับคน ระบบจำแนกคน โดยแบบจำลองที่พัฒนาขึ้น

4.1 การพัฒนาระบบ

ในส่วนนี้จะอธิบายถึงการพัฒนาระบบที่ผู้จัดทำได้ทำขึ้นมาจุดประสงค์เพื่อให้ค่าที่ทำนายได้จากเกตเวย์ส่งต่อไปยังส่วนประสานผู้ใช้งาน ผู้ใช้งานจึงจะสามารถอ่านสถานะของเครื่องเล่นที่อยู่ในฟิตเนสและสามารถรับรู้ได้ว่าเครื่องเล่นพร้อมใช้งานหรือไม่ จะขอแบ่งเป็นส่วนของการติดตั้งเกตเวย์ การพัฒนาแอปพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟซ (API) ฐานข้อมูลที่ใช้รวมถึงการพัฒนาระบบส่วนประสานผู้ใช้งานบนโทรศัพท์มือถือ

4.1.1 การติดตั้งอุปกรณ์รับข้อมูล

เนื่องจากผู้จัดทำได้เลือกใช้ราสพ์เบอรี่ไพ ในการรับข้อมูลมาจากกล้องวงจงปิดผ่าน RTSP โปรโตคอลดังที่ได้กล่าวไว้ในหัวข้อ 3.5 ดังนั้นผู้จัดทำจึงได้นำเอาตัวอุปกรณ์ราสพ์เบอรี่ไพไปต่อในบริเวณที่วง LAN (Local Area Network) ของกล้องเข้าถึงได้เพื่อที่จะนำที่อยู่ของ RTSP จากกล้องมาใส่ไว้ในชุดคำสั่งของผู้จัดทำดังภาพที่ 4.1 และได้ทำการสั่งให้ราสพ์เบอรี่ไพทำการส่งค่าขึ้นไฟร์เบส (Firebase) ซึ่งเป็นบริการฐานข้อมูลแบบทันที (Realtime Database) ดังนั้นจึงรองรับการส่งข้อมูลแบบรายวินาทีจากการทำนายผลของแบบจำลองได้

```
class FirebaseConnection(object):
    def __init__(self, refresh_period=1800):
        self._config = {
          "apiKey": "apikeyhere",
          "authDomain": "authDomainhere",
          "databaseURL": "https://database.url",
          "storageBucket": "databasebucket"
        }
        self._firebase = pyrebase.initialize_app(self._config)
        self._auth = self._firebase.auth()
        self._user =
self._auth.sign_in_with_email_and_password("admin", "password")
        self._db = self._firebase.database()
        self._refresh_period = refresh_period
        self._lastest_refresh = time.time()
```

```
    def refresh_connection(self):
        if time.time() - self._lastest_refresh >
self._refresh_period:
            self._user = auth.refresh(self._user['refreshToken'])
            self._lastest_refresh = time.time()

    def push_data(self, json_data):
        try:
            results = self._db.child("detections").push(json_data,
self._user['idToken'])
        except:
            self.refresh_connection()
            results = self._db.child("detections").push(json_data,
self._user['idToken'])
```

ภาพที่ 4.1 ตัวอย่างซอร์สโค้ดการส่งข้อมูลขึ้นฐานข้อมูลบนไฟร์เบส

โดยข้อมูลที่ส่งขึ้นไปจะเป็นตำแหน่งของวัตถุที่ตรวจจับได้ และประเภทของวัตถุที่ตรวจจับได้และเวลาเป็น UNIX Timestamp เพื่อนำไปคำนวณใน API เป็นอันต่อไป

4.1.2 การพัฒนาแอปพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟซ

ผู้จัดทำได้เลือกใช้ Django REST Framework ซึ่งเป็นไลบรารีหนึ่งของไพธอนในการสร้างแอปพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟซของระบบ เพื่อให้การทำงานเป็นไปในแนวทางเดียวกันทั้งหมด จึงประยุกต์ไลบรารีนี้ในการวางสถาปัตยกรรมของระบบเป็นแบบ RESTful ช่วยในการรองรับข้อมูลที่ส่งได้หลากหลายกว่าเนื่องจากข้อมูลที่ได้จากไฟร์เบสจะเป็นจาวาสคริปต์ออปเจ็คโนเทชั่น JSON (Javascipt Object Notation) ดังนั้นการเลือกใช้ REST จึงเอื้อประโยชต์กว่าแบบ SOAP ที่สามารถส่งข้อมูลได้เพียงแต่ XML (eXtension Markup Language)

จากนั้นผู้จัดทำได้ทำการดึงข้อมูลล่าสุด 30 หน่วยมาจากไฟร์เบสมาประมวลผลเพื่อยืนยันว่าช่วงเวลาสั้น ๆ นั้นแบบจำลองได้ทำนายว่าบนเครื่องเล่นไหนที่มีคนใช้อยู่และเครื่องเล่นไหนที่ว่าง และทำการตอบรับด้วยการส่ง JSON ที่มีค่าสถานะของเครื่องเล่นกลับคืนไปให้ส่วนประสานงานของผู้ใช้ ตัวอย่างของ JSON ดังภาพที่ 4.2

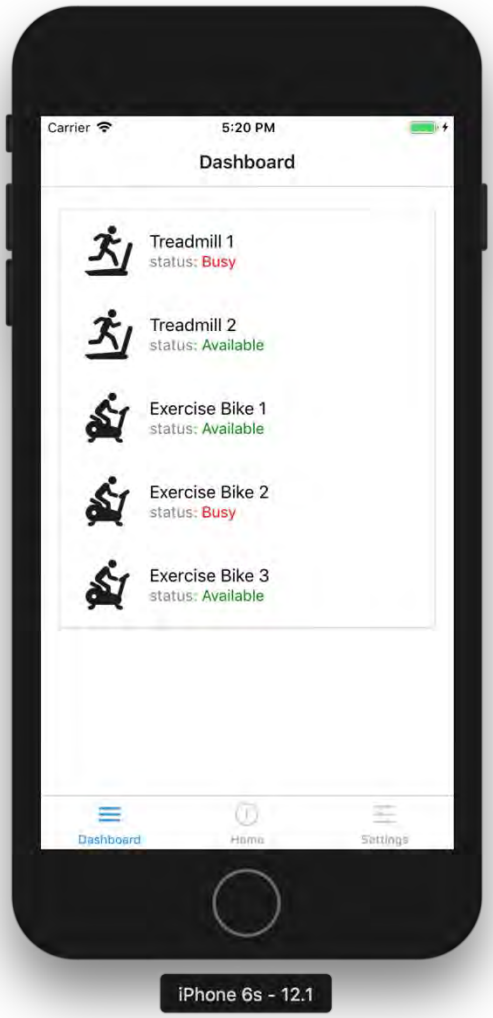
```
{
bicycle1: true,
bicycle2: false,
bicycle3: false,
treadmill1: true,
treadmill2: false
}
```

ภาพที่ 4.2 ตัวอย่างของ JSON ที่ API คืนค่า

4.1.3 การพัฒนาส่วนติดต่อประสานผู้ใช้งาน

ผู้จัดทำได้ใช้ไลบรารี React Native ซึ่งเป็นไลบรารีไว้สำหรับเขียนแอปพลิเคชั่นโทรศัพท์มือถือบนระบบไอโอเอส (IOS) และแอนดรอยด์ (Android) ที่พัฒนาโดยบริษัทเฟซบุ๊ค ตัวภาษาเป็นจาวาสคริปต์ แต่สามารถคอมไพล์เป็นภาษาสวิฟท์ (Swift) ของไอโอเอสและภาษาจาวา (Java) ซึ่งเป็นภาษาพื้นฐานในการพัฒนาแอปพลิเคชั่นบนแอนดรอยด์ได้ แต่ทางผู้จัดทำได้ทดสอบบนอุปกรณ์ไอโอเอสเท่านั้นเนื่องด้วยข้อจำกัดทางอุปกรณ์ โดยอุปกรณ์ที่เลือกมาทดสอบคือ iPhone 6s ดังแสดงในภาพที่ 4.3

โดยแอปพลิเคชั่นออกแบบมาให้สามารถใช้งานได้ง่ายไม่ซับซ้อนมีเพียงหน้าที่บอกสถานะของเครื่องเล่นเนื่องด้วยต้องการเพียงแค่ให้ผู้ใช้งานสามารถตรวจสอบสถานะของเครื่องเล่นได้ และต้องการทดสอบระบบว่าการทำงานโดยรวมทั้งหมดของระบบนั้นเป็นไปได้



ภาพที่ 4.3 ตัวอย่างของหน้าที่แสดงสถานะของเครื่องเล่น

4.2 ผลของการทดสอบ

ในส่วนนี้จะอธิบายถึงผลทดสอบของการสร้างแบบจำลองที่มีการเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ภายในแบบจำลอง เช่น ขนาดของฟิลเตอร์ในแต่ละชั้นของแบบจำลอง รวมไปถึงผลทดสอบของแบบจำลองที่สร้างกับวีดีโอ และการนำแบบจำลองไปใช้งานบนอุปกรณ์

4.2.1 ผลของการทดสอบของแบบจำลอง

ผู้จัดทำได้วัดผลทดสอบของแบบจำลองด้วยค่าความถูกต้องดังที่ได้กล่าวไว้ในบทที่ 3 โดยเบื้องต้นได้ทดสอบว่าแบบจำลองที่ได้นั้นหากแบ่งรูปภาพจากที่ได้เตรียมข้อมูลไว้โดยไม่นำไปเรียนรู้ทั้งหมดกับแบบจำลองและให้แบบจำลองทำการทำนายดูว่ารูปภาพที่ส่งเข้าไปนั้น มีคนอยู่หรือไม่ ซึ่งจำนวนรูปทั้งหมดที่มีถูกแบ่งดังนี้

- รูปที่เป็นพื้นหลัง 7,221 รูป ถูกแบ่งเป็นรูปสำหรับทดสอบ 2,168 รูป และรูปสำหรับให้แบบจำลองเรียนรู้ 5,053 รูป
- รูปที่เป็นบุคคล 4,523 รูป ถูกแบ่งเป็นรูปสำหรับทดสอบ 1,357 รูป และรูปสำหรับเรียนรู้ 3,166 รูป

กล่าวได้ว่าทำการแบ่งรูปทดสอบและรูปสำหรับเรียนรู้ออกจากกันโดยใช้อัตราส่วน 30% จากรูปทั้งหมด

ในชั้นสุดท้ายของแบบจำลองแบบจำแนกบุคคล จะเป็นฟังก์ชันซิกมอยด์ที่จะคืนค่าความน่าจะเป็นออกมา โดยทางผู้จัดทำได้เลือกค่าความมั่นใจ (confidence) ไว้ที่ 0.5 เพื่อเป็นขีดแบ่ง (threshold) ว่าหากมีค่าความความมั่นใจเกิน 0.5 ให้ตัดสินว่ามีคนอยู่ในรูปภาพนั้น แต่หากว่าน้อยกว่า 0.5 จึงจะตัดสินใจว่าไม่มี จากนั้นได้ทำการวัดโดยแบ่งแบบจำลองออกเป็น 2 จำพวกใหญ่และค่าพารามิเตอร์ของในแต่ละชั้นไม่เท่ากันโดยขอเขียนอยู่ในรูป (32, 5, 2) ซึ่งหมายถึง เลขฟิลเตอร์ ค่าขนาดของแพทช์ และการเลื่อนสไตร์ดของเคอร์เนลเป็นลำดับ

- แบบจำลองที่มีการนำเข้ารูปขนาด 45x75
  - ก. แบบจำลอง (32, 5, 2) ในทั้งสองชั้น
  - ข. แบบจำลอง (32, 5, 2) ในชั้นแรกและ (16, 5 ,2) ในชั้นที่สอง
  - ค. แบบจำลอง (32, 7, 2) ในชั้นแรกและ (16, 7, 2) ในชั้นที่สอง
- แบบจำลองที่มีการนำเข้ารูปขนาด 50x50
  - ง. แบบจำลอง (32, 5, 2) ในทั้งสองชั้น
  - จ. แบบจำลอง (32, 5, 2) ในชั้นแรกและ (16, 5 ,2) ในชั้นที่สอง

ซึ่งแบบจำลองทั้งหมดได้เรียนรู้ทั้งหมด 3 ครั้งเพื่อให้ได้ค่าที่เป็นกลางสำหรับผลลัพธ์ที่ได้ สามารถดูได้จากตารางที่ 1.1 และ ตารางที่ 1.2

ตารางที่ 1.1 แสดงผลลัพธ์ที่ได้จากแบบจำลองจำแนกบุคคล

แบบจำลอง	ค่าความถูกต้องบนข้อมูลที่ได้เรียนรู้	ค่าความถูกต้องบนชุดข้อมูลทดสอบ	ระยะเวลา* (วินาที)
ก.	0.997	0.922	36
ข.	0.994	0.918	36
ค.	0.992	0.922	41
ง.	0.998	0.934	31
จ.	0.995	0.918	30

*ระยะเวลาที่แบบจำลองเรียนรู้ต่อ 1,000. รอบ

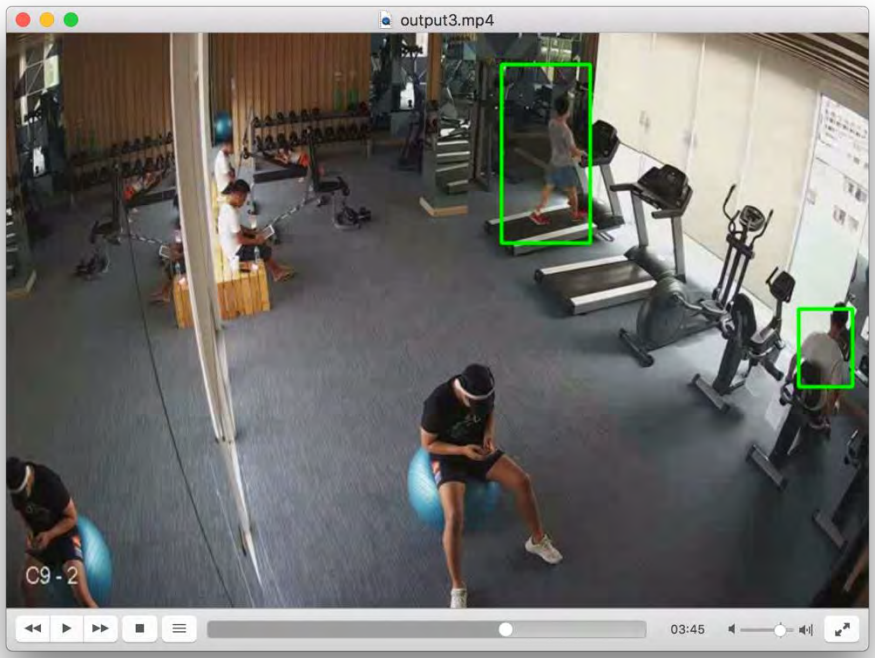
ตารางที่ 1.2 แสดงผลลัพธ์ที่ได้จากการทดลองในหนึ่งครั้ง

แบบจำลอง	ค่าความถูกต้องบนชุดข้อมูลเรียนรู้	ค่าความถูกต้องบนชุดข้อมูลทดสอบ	ผลบวกจริง	ผลลบจริง	ผลบวกปลอม	ผลลบปลอม
ก.	0.994	0.908	1225	1897	271	51
ข.	0.993	0.895	1306	1877	1877	79
ค.	0.995	0.916	1278	1966	202	96
ง.	0.998	0.940	1261	2038	130	81
จ.	0.995	0.933	1321	1969	199	36

ผลจากการทดสอบพบว่าแบบจำลองมีค่าความถูกต้องบนชุดทดสอบใกล้เคียงกัน แต่ที่มีประสิทธิภาพและค่าความถูกต้องดีที่สุดคือแบบจำลอง ง. แบบจำลอง (32, 5, 2) ในทั้งสองชั้น และมีขนาดรูปภาพ 50x50 เนื่องด้วยค่าความถูกต้องและระยะเวลาต่อการเรียนรู้ 1,000 รอบที่เร็วที่สุดนั่นหมายถึงสามารถทำนายผลได้เร็วที่สุดด้วย

4.2.2 ผลของการทดสอบของแบบจำลองบนอุปกรณ์

ผู้จัดทำได้นำแบบจำลองที่ดีที่สุด ทำการบันทึกค่าที่ได้จากแบบจำลองและทำตามขั้นตอนที่ได้กล่าวไว้ในหัวข้อที่ 3.4 ข้างต้นและนำไปใช้ โดยในเบื้องต้นทางผู้จัดทำได้ทำการทดลองกับวีดีโอที่มีอยู่แล้วที่ได้แยกไว้เพื่อทำการทดสอบ โดยเปลี่ยนจากการรับค่าจากกล้องวงจรปิดให้เป็นการอ่านค่าจากวีดีโอแทนแล้วทำการวาดกรอบเมื่อแบบจำลองทำนายว่ามีบุคคลอยู่บนเครื่องเล่น จากนั้นจึงทำการบันทึกวีดีโอที่ได้ทำการให้แบบจำลองทำนายลงมา เพื่อวิเคราะห์ถึงความถูกต้องเบื้องต้นว่ามีการคลาดเคลื่อนมากหรือไม่ ดังภาพตัวอย่าง 4.4 และยังทดสอบการส่งข้อมูลขึ้นไฟร์เบสว่ามีการคลาดเคลื่อนหรือไม่



ภาพที่ 4.4 ตัวอย่างของวีดีโอเมื่อทดสอบบนอุปกรณ์

บทที่ 5
ข้อสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง สรุปผลจากการพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส ปัญหาและอุปสรรคที่พบระหว่างการดำเนินงาน รวมถึงแนวการพัฒนาระบบต่อไปในอนาคต โดยมีรายละเอียดดังต่อไปนี้

5.1 ข้อสรุป

โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนสมีการทำงานโดยวิเคราะห์การตรวจจับภาพวัตถุจากไฟล์วิดีโอภาพเคลื่อนไหวจากกล้องวงจรปิดแบบเรียลไทม์ด้วยวิธีการประมวลผลภาพโดยการใช้การเรียนรู้เชิงลึกตรวจจับภาพคนที่เข้ามาใช้บริการสิ่งอำนวยความสะดวกในพื้นที่ให้บริการภายในห้องฟิตเนสและทำการแสดงผลสถานะเครื่องออกกำลังกายภายในห้องฟิตเนสผ่านแอปพลิเคชันถึงสถานะว่าขณะนั้นมีผู้ใช้งานอยู่หรือไม่ โดยขั้นตอนการพัฒนาเริ่มจากการเก็บรวบรวมข้อมูลที่ใช้ในการสร้างแบบจำลองการเรียนรู้เชิงลึก สร้างแบบจำลองการเรียนรู้เชิงลึกที่ใช้ในการประมวลผลภาพติดตั้งกับอุปกรณ์กล้องวงจรปิดกับส่วนการประมวลผลคือ Intel Movidius Neural Network Compute Stick และ Raspberry PI3 ส่งผ่านข้อมูลที่ได้จากแบบจำลองขึ้นไปยังบริการฐานข้อมูลบนคลาวด์ และเชื่อมต่อการพัฒนาแอพพลิเคชันส่วนต่อประสานผู้ใช้งานที่แสดงผลสถานะการใช้งานของสิ่งอำนวยความสะดวกต่อผู้ใช้งานซึ่งพัฒนาด้วย React-Native โดยระบบที่พัฒนาขึ้นส่งผลให้ผู้ใช้งานสามารถทราบถึงสถานะการใช้งานขณะนั้นและสามารถตัดสินใจเข้าไปใช้บริการต่อไปได้ด้วยตนเองได้อย่างสะดวกรวดเร็วมากยิ่งขึ้น รวมไปถึงช่วยให้ผู้ประกอบการธุรกิจสามารถนำไปต่อยอดพัฒนาและประยุกต์ใช้ในการตรวจจับภาพในให้บริการสิ่งอำนวยความสะดวกในห้องฟิตเนสได้

5.2 ปัญหาและอุปสรรค

1. กล้องวงจรปิดที่นำมาใช้หากมีฝุ่นหรือสิ่งกีดขวางกำบัง ส่งผลให้ซอฟต์แวร์อาจตรวจจับภาพคลาดเคลื่อนไปจากความเป็นจริง
2. พื้นที่บริเวณที่ติดตั้งต้องมีแสงสว่างเพียงพอให้ซอฟต์แวร์สามารถตรวจจับวัตถุในภาพทำงานได้อย่างมีประสิทธิภาพ ถูกต้องและแม่นยำ ซึ่งหากแสงสว่างภายในพื้นที่มืดเกินไปหรือสว่างเกินไป อาจะทำให้ซอฟต์แวร์ทำงานผิดพลาดได้
3. กล้องวงจรปิดที่ติดตั้งในพื้นที่ทดลองเปลี่ยนทิศทางจากตำแหน่งเดิมทำให้ต้องจัดวางตำแหน่งทิศทางใหม่

4. การเตรียมข้อมูลสำหรับแบบจำลองแบบจำแนกวัตถุยังต้องใช้เวลาถึงแม้จะพัฒนาซอฟต์แวร์ที่ช่วยในการกำกับแล้วก็ตาม
5. ตารางเวลาของทางผู้จัดทำโครงงานไม่ค่อยตรงกัน ทำให้เกิดปัญหาในการพัฒนาโปรแกรม

5.3 วิธีแก้ไขปัญหา

1. ตรวจสอบแสงสว่างในพื้นที่ติดตั้งกล้องภายในพื้นที่ห้องอำนวยความสะดวกให้มีแสงสว่างเพียงพอต่อการทำงานของซอฟต์แวร์
2. ต้องทำความสะอาดกล้องวงจรปิดที่ติดตั้งอยู่เป็นประจำเพื่อไม่ให้มีสิ่งบดบังหรือกีดขวางในบริเวณตรวจจับภาพ

5.4 ผลที่ได้รับจากการพัฒนาโครงงาน

1. ผู้ประกอบการธุรกิจสามารถนำข้อมูลการพัฒนาที่ได้จากโครงงานการพัฒนาโปรแกรมตรวจจับภาพจากกล้องวงจรปิดไปประยุกต์ใช้กับการพัฒนาระบบตรวจจับภาพต่างๆในพื้นที่ให้บริการต่างๆได้
2. ผู้ให้บริการสิ่งอำนวยความสะดวกสามารถนำข้อมูลได้จากโครงงานไปปรับปรุงคุณภาพในการให้บริการพื้นที่สิ่งอำนวยความสะดวก

5.5 ข้อเสนอแนะ

ในหัวข้อนี้จะกล่าวถึงข้อเสนอแนะที่ได้จากการพัฒนาโปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนสเพื่อเป็นแนวทางในการพัฒนาโปรแกรมต่อไปในอนาคต ดังนี้

1. ปรับปรุงให้การตรวจจับภาพจากกล้องวงจรปิดสามารถตรวจจับภาพจากพื้นที่ให้บริการได้โดยให้มีข้อจำกัดทางแสงสว่างน้อยลง เช่น สามารถตรวจจับภาพที่บริเวณสว่างจ้าได้
2. พัฒนาข้อมูลที่ใช้ในการประมวลภาพจากการใช้งานสิ่งอำนวยความสะดวกให้มีความถูกต้องแม่นยำมากยิ่งขึ้น
3. พัฒนาปรับปรุงส่วนต่อประสานผู้ใช้งานให้มีความสวยงามน่าใช้งานมากขึ้น
4. พัฒนาให้แอปพลิเคชันสามารถจองสถานะการใช้งานของสิ่งอำนวยความสะดวกในห้องฟิตเนสต่อผู้ใช้งานที่ประสงค์ขอใช้งานได้
5. พัฒนาให้แอปพลิเคชันสามารถเก็บสถิติการใช้งานเครื่องออกกำลังกายแต่ละเครื่องเพื่อเก็บเป็นสถิติได้สำหรับการซ่อมบำรุงเครื่องออกกำลังกายและการลงทุนในการบริการสิ่งอำนวยความสะดวก

รายการอ้างอิง

[1] Kurt. Object Detection Tutorial in TensorFlow: Real-Time Object Detection [online]. Available from: https://www.edureka.co/blog/tensorflow-object-detection-tutorial/ [Accessed 30 Mar. 2019]

[2] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection [online]. Available from: https://pjreddie.com/media/files/papers/yolo.pdf [Accessed 30 Mar. 2019]

[3] Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement [online]. Available at: https://pjreddie.com/media/files/papers/YOLOv3.pdf [Accessed 30 Mar. 2019]

[4] Image Classification [online]. Available from: http://www.sc.chula.ac.th/courseware/2309507/Lecture/remote18.htm [Accessed 30 Mar. 2019]

[5] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks [online]. Available from: http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf [Accessed 30 Mar. 2019]

[6] ImageNet [Image Database]. Stanford University: Available from: http://www.image-net.org/ [Accessed 30 Mar. 2019]

[7] SSD_Mobilenet_COCO [Computer Software]. Available from: http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v1_coco_2018_01_28.tar.gz [Accessed 30 Mar. 2019]

[8] Wei L., and others. SSD: Single Shot MultiBox Detector [online]. Available from: https://arxiv.org/pdf/1512.02325.pdf [Accessed 30 Mar. 2019]

[9] COCO Common Objects in Context [Computer Software]. Available from: http://cocodataset.org/ [Accessed 30 Mar. 2019]

[10] Ren, S., and others. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [online]. Available from: https://arxiv.org/pdf/1506.01497.pdf [Accessed 30 Mar. 2019]

[11] Weisstein, E. Sigmoid Function [online]. Wolfram Research: Available from: http://mathworld.wolfram.com/SigmoidFunction.html [Accessed 30 Mar. 2019]

[12] Convolutional Neural Network (CNN) [online]. Available from: https://developer.nvidia.com/discover/convolutional-neural-network [Accessed 30 Mar. 2019]

[13] Weisstein, E. Convolution [online]. Available from: http://mathworld.wolfram.com/Convolution.html [Accessed 30 Mar. 2019]

[14] Virdee, A. 2018. Data Augmentation Experimentation [online]. Available from: https://towardsdatascience.com/data-augmentation-experimentation-3e274504f04b [Accessed 30 Mar. 2019]

[15] Simple guide to confusion matrix terminology [online]. 2014. Available from: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ [Accessed 30 Mar. 2019]

[16] Rouse, M. Real Time Streaming Protocol (RTSP) [online]. Available from: https://searchvirtualdesktop.techtarget.com/definition/Real-Time-Streaming-Protocol-RTSP [Accessed 30 Mar. 2019]

[17] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[18] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[19] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[20] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[21] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint (2017).

[22] Jedsada Saengow. 2018. คืออะไร มาดูวิธีสร้าง Project และทำความรู้จักกับ Firebase [online]. Available from: https://medium.com/jed-ng/firebase-คืออะไร-มาดูวิธีสร้าง-project-และทำความรู้จักกับ-firebase-d48bfac67b14 [Accessed 27 Mar. 2019]

[23] Firebase คืออะไร และมีข้อดีอย่างไรบ้าง [online]. Available from: https://www.softmelt.com/article.php?id=588 [Accessed 27 Mar. 2019]

[24] Firebase [online]. Available from: https://en.m.wikipedia.org/wiki/Firebase [Accessed 27 Mar. 2019]

[25] Pyrebase [online]. Available from: https://firebase.google.com/docs/database/rest/start [Accessed 25 Mar. 2019]

[26] Heating up With Firebase: Tutorial on How to Integrate Firebase Into Your App [online]. 2016. Available from: https://blog.devcolor.org/heating-up-with-firebase-tutorial-on-how-to-integrate-firebase-into-your-app-6ce97440175d [Accessed 25 Mar. 2019]

[27] OpenCV [online]. Available from: https://en.m.wikipedia.org/wiki/OpenCV [Accessed 25 Mar. 2019]

[28] Keras: The Python Deep Learning library [online]. Available from: https://keras.io/ [Accessed 27 Mar. 2019]

[29] Keras [online]. Available from: https://en.wikipedia.org/wiki/Keras [Accessed 29 Mar. 2019]

[30] Intel® Movidius™ Neural Compute SDK [online]. Available from: https://github.com/movidius/ncsdk [Accessed 30 Mar. 2019]

[31] NCSDK Documentation [online]. Available from: https://movidius.github.io/ncsdk/ [Accessed 30 Mar. 2019]

[32] React Native [online]. Available from: https://en.wikipedia.org/wiki/React_Native [Accessed 30 Mar. 2019]

[33] React Native [online]. 2019. Available from: https://github.com/facebook/react-native [Accessed 30 Mar. 2019]

[34] TensorFlow [online]. 2019. Available from: https://en.wikipedia.org/wiki/TensorFlow [Accessed 30 Mar. 2019]

ภาคผนวก

ภาคผนวก ก
แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal
ปีการศึกษา 2561

ชื่อโครงงาน (ภาษาไทย)	โปรแกรมตรวจจับวัตถุจากกล้องวงจรปิดเพื่อช่วยในการควบคุมและให้บริการฟิตเนส
ชื่อโครงงาน (ภาษาอังกฤษ)	Real-time object detection program to help automate and provide service for gym
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.พีระพนธ์ โสพัศสถิตย์
ผู้ดำเนินการ	1. ชื่อ นิวิตถ์ นันทนิวัตติกุล เลขประจำตัวนิสิต 5833638023 2. ชื่อ กชพร วัชรประภาพงศ์ เลขประจำตัวนิสิต 5833601823 สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หลักการและเหตุผล

ปัจจุบันผู้คนเริ่มให้ความสนใจในการดูแลสุขภาพและนิยมการออกกำลังกายมากขึ้น ทำให้ระบบฟิตเนสเริ่มเข้ามาเป็นส่วนหนึ่งในการออกลังกายในชีวิตประจำวันโดยทั่วไป เนื่องจากไม่ต้องใช้สถานที่กว้างขวางมากมาย สะดวกและราคาอัตราการใช้บริการไม่สูงมากนัก ดังนั้นในสถานที่ต่างๆเริ่มใช้ระบบฟิตเนส เข้ามาติดตั้งเพื่ออำนวยความสะดวกให้แก่ผู้คนที่เข้ามาใช้บริการ เช่น ที่ทำงาน บ้านพักอาศัย สโมสรต่างๆ ซึ่งคอนโดที่พักอาศัยก็เป็นส่วนหนึ่งที่มีบริการฟิตเนสเป็นสิ่งอำนวยความสะดวกหลักอยู่ภายในโครงการ

เนื่องจากภายในคอนโดมีผู้พักอาศัยร่วมกันจำนวนมาก และระยะเวลาการเข้าใช้งานฟิตเนสต่อบุคคลนั้นใช้ระยะเวลานานอย่างต่ำครึ่งชั่วโมงเป็นต้นไป ถ้าคอนโดใดมีผู้ใช้บริการเป็นจำนวนมากจะทำให้มีบางเวลาที่ผู้พักอาศัยที่ต้องการมาใช้บริการนั้นไม่ได้ออกกำลังกาย ต้องเสียเวลาในการรอคิว หรือเปลี่ยนใจเนื่องจากระยะเวลาที่นานเกินไปในการรอ ทำให้การออกกำลังกายไม่เป็นไปตามที่คาดหวังไว้ ดังนั้นทีมผู้จัดทำจึงพัฒนาโปรแกรมนี้ขึ้นมาเพื่อตอบสนองต่อความต้องการของผู้ที่ต้องการเข้ามาใช้งานฟิตเนส เพื่ออำนวยความสะดวกต่อผู้ใช้งาน ซึ่งผู้ใช้งานสามารถดูผู้เข้าใช้งานคนอื่นที่กำลังใช้งานฟิตเนสอยู่ผ่านทางระบบได้เลยโดยไม่ต้องเข้าไปดูด้วยตนเองที่พื้นที่ให้บริการ ทำให้ทางผู้ใช้งานเองสามารถใช้เวลาส่วนตนให้เกิดประโยชน์สูงสุด ไม่จำเป็นต้องเสียเวลาเข้ามารอที่หน้าฟิตเนสเพื่อรอใช้งาน สามารถใช้เวลาไปทำอย่าง

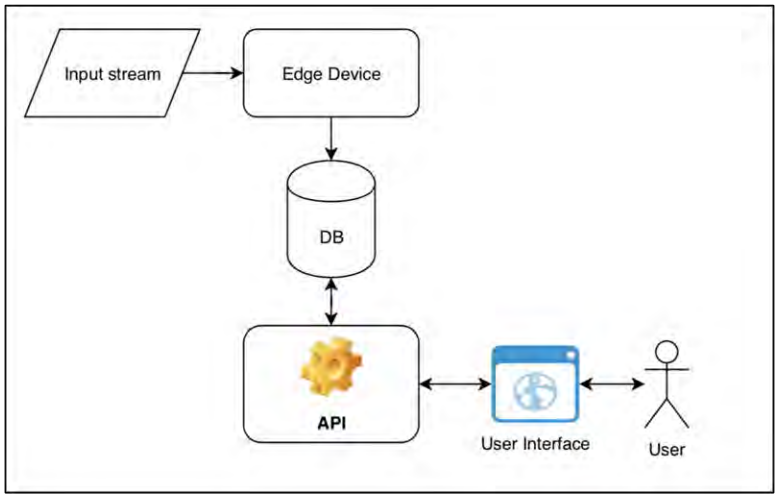
อื่นก่อน และเมื่อสิ่งอำนวยความสะดวกตามที่ผู้ใช้งานต้องการใช้งานนั้นว่างแล้ว ผู้ใช้งานสามารถเข้าใช้งานได้เลยทำให้ไม่เป็นการเสียอารมณ์ขุ่นมัวในใช้เวลาในการรอนานเกินไปซึ่งส่งผลเสียต่อสุขภาพจิตของผู้ใช้งาน

วัตถุประสงค์

1. เพื่อวิเคราะห์การตรวจจับวัตถุด้วยวิธีการประมวลผลภาพจากการใช้การเรียนรู้เชิงลึก
2. เพื่อพัฒนาระบบที่สามารถตรวจจับคนที่เข้ามาใช้บริการสิ่งอำนวยความสะดวกภายในพื้นที่สิ่งอำนวยความสะดวกในคอนโดแบบอัตโนมัติ
3. เพื่อทดสอบประสิทธิภาพของโปรแกรมที่นำเสนอ

ขอบเขตของโครงงาน

1. ศึกษาข้อมูลภาพในการตรวจจับคนจากกล้องวงจรปิดภายในพื้นที่ให้บริการสิ่งอำนวยความสะดวกในที่อยู่อาศัย เช่นพื้นที่ฟิตเนส หรือภายในห้อง
2. บริเวณพื้นที่ที่ติดตั้งกล้องวงจรต้องมีแสงสว่างเพียงพอและสามารถเห็นคนได้อย่างชัดเจน
3. กล้องวงจรปิดต้องเป็นกล้องที่ไม่มีการเปลี่ยนทิศทางหรือก็คือภาพที่ได้จากกล้องจะต้องอยู่มุมเดิมเสมอ
4. ส่วนต่อประสานงานกับผู้ใช้ (user interface) เป็นแอปพลิเคชั่นมือถือบน iOS ที่เขียนด้วย react-native
5. สถาปัตยกรรมของระบบถูกกำหนดดังในภาพที่ 1



ภาพที่ 1

ตารางระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลา															
	ปี 2560												ปี 2561			
	ต.ค.				พ.ย.				ธ.ค.				ม.ค.			
1. เก็บรวบรวมข้อมูลและประมวลผลข้อมูล																
1.1 เก็บรวบรวมข้อมูลสำหรับการสร้างโมเดลการเรียนรู้เชิงลึก																
1.2 ประมวลผลข้อมูลและการระบุประเภทของรูป																
2. การสร้างโมเดลในการเรียนรู้																
2.1 ตั้งค่า Google Cloud Instance สำหรับการสร้างโมเดล																
2.2 สร้างการเรียนรู้ของโมเดล																
2.3 ปรับค่าโมเดล																
2.4 นำโมเดลลงอุปกรณ์และทดสอบ																
3. การนำโมเดลไปใช้ในระบบหลังบ้าน																
3.1 ติดตั้งเซิร์ฟเวอร์สำหรับเรียกใช้ระบบหลังบ้าน																
3.2 พัฒนาระบบหลังบ้าน																
4. พัฒนาส่วนต่อประสานผู้ใช้งาน																
4.1 ออกแบบส่วนต่อประสานผู้ใช้งานสำหรับแอพพลิเคชัน																
4.2 พัฒนาแอพพลิเคชัน																
5. เชื่อมต่อส่วนต่อประสานผู้ใช้งานกับระบบหลังบ้าน																
5.1 เชื่อมต่อแอพพลิเคชันกับ API																
5.2 นำแอปพลิเคชันไปใช้งาน																
6. การนำโมเดลไปทดสอบบนอุปกรณ์ในสถานที่ติดตั้งจริง																
6.1 ทดสอบการใช้งานของส่วนต่อประสานผู้ใช้งานกับอุปกรณ์																
6.2 การปรับแต่งโมเดล																

ประโยชน์ที่คาดว่าจะได้รับ

ก. ประโยชน์ต่อนิสิต

1. สามารถประยุกต์โมเดลเชิงวิชาการ Convolutional neural network กับการแก้ปัญหาจริง
2. ได้พัฒนาทักษะการใช้ภาษา Python และการพัฒนาแอพพลิเคชันบนระบบ IOS โดยใช้ react-native

3. ได้ฝึกทักษะการทำงานเป็นกลุ่ม การคิดวิเคราะห์และแก้ปัญหา
4. ได้เรียนรู้และพัฒนาเทคนิคเกี่ยวกับการใช้การเรียนรู้เชิงลึกในการตรวจจับวัตถุจากภาพกล้องวงจรปิด
5. ได้เรียนรู้การพัฒนาระบบให้สอดคล้องกับอุปกรณ์ที่นำมาร่วมใช้ในการตรวจจับวัตถุจากกล้องวงจรปิด

ข. ประโยชน์ที่ได้จากโครงงานที่พัฒนาขึ้น

1. สามารถนำระบบไปใช้ในการจัดการและให้บริการในพื้นที่สิ่งอำนวยความสะดวกภายในคอนโดและสถานที่ต่างๆ
2. สามารถอำนวยความสะดวกสบายแก่ผู้เข้ามาใช้บริการสิ่งอำนวยความสะดวกภายในคอนโดมากยิ่งขึ้น
3. ช่วยลดปริมาณคน เวลาและต้นทุนที่จะเกิดขึ้นจากการให้บริการระบบตรวจจับปริมาณคนใช้งานเพื่ออำนวยความสะดวกผู้ใช้งานภายในพื้นที่สิ่งอำนวยความสะดวกภายในคอนโด
4. ผู้ให้บริการสิ่งอำนวยความสะดวกสามารถนำข้อมูลไปใช้ในการพัฒนาระบบและปรับปรุงคุณภาพในการให้บริการพื้นที่สิ่งอำนวยความสะดวก

อุปกรณ์และเครื่องมือที่ใช้

Hardware

- กล้องวงจรปิด (IP camera)
- Macbook Pro: Processor 2.7 GHz Intel Core i5 Memory 8 Gb เป็น gateway
- Raspberry PI3 [Optional] เป็น gateway
- Intel Movidius Neural Network Compute Stick
- iPhone 6s เป็นอุปกรณ์ในสำหรับใช้ทดสอบส่วนติดต่อผู้ใช้งาน

Software

- Library
    - Keras
    - Tensorflow
    - OpenCV
- OS: Ubuntu 16.xx

งบประมาณ

ก. ค่าใช้สอย

1. ค่าพิมพ์โปสเตอร์และค่าเดินทางไปนำเสนอผลงาน 1,000 บาท

ข. ค่าวัสดุ

1.	กล้องวงจรปิด (IP Camera)	2,000 บาท
2.	Micro SD Card	600 บาท
3.	raspberry pi 3+ (mainboard)	2,700 บาท
4.	Intel Movidius NCS	3,600 บาท
5.	กระดาษ A4	100 บาท

เอกสารอ้างอิง

[1] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[2] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

[4] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.

[5] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." *arXiv preprint* (2017).

ประวัติผู้เขียน

Mr. Nivit Nantanivattikul
นายนิวิตถ์ นันทนิวัตติกุล
รหัสนิสิต 5833638023
วัน เดือน ปี เกิด : 17 พฤษภาคม 2540
สถานที่เกิด : กรุงเทพมหานคร
อีเมล : nivit.nan@gmail.com
การศึกษา : สาขาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

Miss Kotchaporn Watcharaprapapong
นางสาวกชพร วัชรประภาพงศ์
รหัสนิสิต 5833601823
วัน เดือน ปี เกิด : 4 มกราคม 2539
สถานที่เกิด : เชียงราย
อีเมล : yellykotchaporn@gmail.com
การศึกษา สาขาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย