ระบบตรวจจับรถไฟโดยกระบวนการประมวลผลภาพ
TRAIN DETECTION IN RAILWAY TRACK USING IMAGE PROCESSING

ธิปก สาหร่าย
ปรัชญา จันทร์ทรัพย์
วรรณฤดี เมืองราช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมเครื่องกล สาขาวิศวกรรมขนส่งทางราง
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบตรวจจับรถไฟโดยกระบวนการประมวลผลภาพ

TRAIN DETECTION IN RAILWAY TRACK USING IMAGE PROCESSING

ธิปก สาหร่าย

ปรัชญา จันทร์ทรัพย์

วรรณฤดี เมืองราช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมเครื่องกล สาขาวิศวกรรมขนส่งทางราง

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRAIN DETECTION IN RAILWAY TRACK USING IMAGE PROCESSING

TIPOK SARAI

PRACHAYA JANSUP

WANRUDEE MUNGRAD

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2017

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

สาขาวิศวกรรมขนส่งทางราง ภาควิชาเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับรถไฟด้วยกระบวนการประมวลผลภาพ

TRAIN DETECTION IN RAILWAY TRACK USING IMAGE PROCESSING

ผู้จัดทำ

1.นาย ธิปก	สาหร่าย	รหัสประจำตัว	57010621
2.นาย ปรัชญา	จันทร์ทรัพย์	รหัสประจำตัว	57010744
3.นางสาว วรรณฤดี	เมืองราช	รหัสประจำตัว	57011111

_____อาจารย์ที่ปรึกษา

(ผศ. ดร. เอกพจน์ ตันตราภิวัฒน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจจับรถไฟโดยกระบวนการประมวลผลภาพ

นาย ธิปก สาหร่าย 57010621
นาย ปรัชญา จันทร์ทรัพย์ 57010744
นางสาววรรณฤดี เมืองราช 57011111
ผศ. ดร. เอกพจน์ ตันตราภิวัฒน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2560

บทคัดย่อ

ปัจจุบันการเดินทางด้วยรถไฟมีบทบาทที่สำคัญมากขึ้นจากในอดีต ผู้คนจึงต่างหันมาให้ความสนใจในระบบขนส่งทางราง ทำให้การป้องกันอุบัติเหตุที่เกิดขึ้นเกี่ยวกับรถไฟก็ได้รับพัฒนาตามมาเช่นกัน แต่การพัฒนาดังกล่าวก็ยังไม่สามารถป้องกันได้ครอบคลุมทุกพื้นที่ ซึ่งบริเวณพื้นที่อุบัติเหตุที่เกิดขึ้นเกี่ยวกับรถไฟมักจะเกิดตรงบริเวณจุดตัดทางรถไฟ โดยเฉพาะตามพื้นที่ห่างไกลนอกตัวเมืองที่ยังไม่มีอุปกรณ์ป้องกันติดตั้ง โครงงานนี้จึงได้จัดทำชุดต้นแบบของระบบตรวจจับรถไฟโดยใช้กระบวนการประมวลผลภาพ หรือ image processing ใช้ภาษา Python ในการเขียนโค้ดลงบนบอร์ด Raspberry Pi ที่ติด Module กล้อง แล้วนำไปติดตั้งบริเวณรางรถไฟ ซึ่งจะพิจารณาจาก 2 วิธี คือ การใช้ Haar Cascade ซึ่งเป็นฟังก์ชั่นของชุดเครื่องมือ OpenCV โดยเป็นวิธีที่ให้ตัวโปรแกรมทำการเรียนรู้จากรูปภาพรถไฟจำนวนมาก กับอีกวิธีคือ การใช้ Background Subtraction ซึ่งจะตรวจจับการเคลื่อนไหวที่แตกต่างจากพื้นหลังที่ตั้งไว้ และเมื่อกล้องตรวจเจอรถไฟตามวิธีต่าง ๆ ข้างต้น ก็จะทำการส่งสัญญาณไปยังไฟแจ้งเตือนได้

จากการทดลองระบบตรวจจับรถไฟด้วยการประมวลผลภาพโดยการติดตั้งกล้องไว้บริเวณสถานีรถไฟหัวตะเข้ ชุดทดลอง Haar Cascade จะสามารถเริ่มตรวจจับขบวนรถไฟได้เมื่อบริเวณโดยรอบมีค่าความเข้มแสงอยู่ในช่วงประมาณ 20,000 LUX ขึ้นไป ซึ่งมีระยะการตรวจจับที่มีความแม่นยำอยู่ที่ 99 เปอร์เซ็นต์ ที่ระยะ 33 เมตร และจะมีระยะการตรวจจับสูงที่สุดที่ระยะ 60 เมตร และ ชุดทดลอง Background Subtraction สามารถตรวจจับขบวนรถไฟได้ทั้งวันโดยสามารถตรวจจับขบวนรถไฟที่มีความแม่นยำที่ระยะ 22 เมตร และระยะสูงสุดที่ตรวจจับได้ 50 เมตรและในช่วงเวลากลางคืนจะตรวจจับได้ที่ระยะทางตรวจจับที่น้อยกว่า 30 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRAIN DETECTION IN RAILWAY TRACK USING IMAGE PROCESSING

TIPOK	SARAI	57010621
PRACHAYA	JANSUP	57010744
WANRUDEE	MUNGRAD	57011111
Asst.Prof.Dr.Akapot	Tantrapiwat	Advisor

Year 2017

ABSTRACT

Nowadays, railway transportation has a major role in the fact that it allows people to travel easier than ever before. The way it revolutionized transportation made more people interest in rail transportation system, and that lead to system which provide more safety have been improve a lot too. Safety and alarm system at rail crossing is a main point that need to be consider. Of course, this development cannot reach all of area especially urban area which safety systems have not been install yet. This project is to invent the prototype of train detection device using image processing. Base on Python programming language, then writing a code in Raspberry Pi with camera module. This detection system can be using 2 different ways. First, using Haar Cascade in OpenCV which let program learning itself by giving it a lot of positive and negative picture. Another way is using background subtraction technique that detect difference between static background and motion. When trains have been detected, the signal will be send to light alarm.

From the experimental, Haar Cascade technique starting to detect the train more than 20,000 LUX with accuracy 99 percents at 33 meter with maximum detecting range up to 60 meter, Background Subtraction technique, the result is accuracy at 22 meter, maximum detect range up to 50 meter at day time and maximum detecting range lower than 30 meter at night time.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำโครงงาน เรื่อง ระบบตรวจจับรถไฟโดยกระบวนการประมวลผลภาพ ได้รับความช่วยเหลือจากเพื่อนสมาชิกในกลุ่มทุกคนที่ได้ตั้งใจปฏิบัติงานตามหน้าที่ที่ได้รับมอบหมายที่ได้รับจนกระทั่งสามารถทำให้งานสำเร็จลุล่วงได้ตามระยะเวลาที่กำหนด

ขอขอบคุณ อาจารย์เอกพจน์ ตันตราภิวัฒน์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน และเป็นผู้ให้คำแนะนำปรึกษาเกี่ยวกับการวางแผนการดำเนินงาน รวมถึงรับฟังปัญหาและให้คำชี้แนะในการแก้ไขปัญหาต่างๆในการทำงานตลอดระยะเวลาในการปฏิบัติงาน ขอขอบคุณ คณะอาจารย์ภาควิชาวิศวกรรมเครื่องกลทุกท่านที่ช่วยตรวจสอบชี้แนะแนวทางข้อบกพร่องของโครงงานรวมถึงช่วยชี้แนะวิธีการแก้ไขปัญหาต่างๆจนทำให้โครงงานชิ้นนี้สำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำโครงงานหวังว่า ระบบตรวจจับรถไฟโดยกระบวนการประมวลผลภาพ จะเกิดประโยชน์กับผู้ใช้เส้นทางสันจรไปมาบริเวณทางข้ามรถไฟ นักศึกษา และผู้ที่สนใจศึกษาและนำไปต่อยอดพัฒนาระบบต่อไปไม่มากก็น้อย

นายธิปก สาหร่าย

นายปรัชญา จันทร์ทรัพย์

นางสาววรรณฤดี เมืองราช

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VI
สารบัญรูป.....VII

บทที่ 1 บทนำ.....1
1.1 ความสำคัญและที่มาของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....2
1.3 ขอบเขตของโครงงาน.....2
1.4 ผลที่คาดว่าจะได้รับ.....2
1.5 ขั้นตอนการดำเนินงาน.....3

บทที่ 2 ข้อมูลและทฤษฎีที่เกี่ยวข้อง.....4
2.1 ข้อมูลที่เกี่ยวข้อง.....4
2.2 ทฤษฎีที่เกี่ยวข้อง.....12
2.3 งานวิจัยที่เกี่ยวข้อง.....25

บทที่ 3 หลักการออกแบบและสร้างระบบ.....26
3.1 เครื่องมือที่ใช้ออกแบบระบบ.....26
3.2 หลักการออกแบบโปรแกรม.....26

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ขั้นตอนการดำเนินงาน.....30
4.1 ขั้นตอนการสร้างระบบ.....30
4.2 ขั้นตอนการต่อวงจรหลอดไฟ LED.....31
4.3 ขั้นตอนการทดสอบและอธิบายโค้ด Background subtraction 33
4.4 ขั้นตอนการทดสอบและอธิบายโค้ด Haar Cascade.....39

บทที่ 5 ผลการดำเนินงานและข้อเสนอแนะ.....41
5.1 ผลการดำเนินงาน 41
5.2 สรุปผลการทดลอง 56
5.3 ข้อเสนอแนะ 57

ภาคผนวก
ภาคผนวก ก
ภาคผนวก ข
ภาคผนวก ค

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 การเดินรถไฟ “เที่ยวออกจากกรุงเทพ” สายตะวันออก ของรถไฟโดยสาร.....8

2.2 การเดินรถไฟ “เที่ยวกลับเข้ากรุงเทพ” สายตะวันออก ของรถไฟโดยสาร.....9

2.3 การเดินรถไฟ “เที่ยวออกจากกรุงเทพ” สายตะวันออก ของรถไฟขนส่งสินค้า.....10

2.4 การเดินรถไฟ “เที่ยวกลับเข้ากรุงเทพ” สายตะวันออก ของรถไฟขนส่งสินค้า.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 รถไฟโดยสารยี่ห้อ Alsthom 4

2.2 รถไฟโดยสารยี่ห้อ Tokyu 5

2.3 รถไฟโดยสารยี่ห้อ General Electric 5

2.4 รถไฟโดยสารยี่ห้อ Hitachi 6

2.5 รถไฟโดยสารยี่ห้อ CSR Qishuyan....6

2.6 รถไฟโดยสารยี่ห้อ General Electric....7

2.7 ภาพแสดงแบบ Haar Like Feature 4 แบบที่ถูกเลือกใช้(A – B – C – D)....15

2.8 ภาพแสดงพื้นที่บริเวณที่ถูกอินทริกรัล....16

2.9 ภาพแสดงพื้นที่บริเวณที่ถูกอินทริกรัลเพื่อหาพื้นที่ที่เราต้องการ 16

2.10 ภาพแสดงความแตกต่างของพิกเซล (Threshold) 17

2.11 ภาพแสดงส่วนต่างของภาพสองภาพ....18

2.12 ภาพแสดงระบบสี RGB 19

2.13 แสดงระบบสี HSV 20

2.14 แสดง GPIO ของ Raspberry Pi 21

2.15 สายที่เชื่อมต่อ 22

2.16 LED แสดงสถานะของบอร์ด 22

2.17 ภาพแสดง พอร์ต CSI 23

2.18 สาย HDMI 23

2.19 HDMI to VGA 23

2.20 พอร์ต DSI (Display Serial Interface) 24

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.21 Raspberry PI Camera Module V2....24

3.1 ภาพแสดง positive เฉพาะส่วนรถไฟ....27

3.2 ภาพแสดง negative 27

3.3 ภาพแสดงการตรวจจับรถไฟด้วย วิธี Haar Cascade....28

3.4 ภาพแสดงการแจ้งเตือนรถไฟ 28

3.5 ภาพแสดงการแจ้งเตือนจากหลอด LED 29

4.1 Flowchart ขั้นตอนการดำเนินงานของโปรแกรม....30

4.2 ภาพแสดงชื่ออุปกรณ์ต่างๆ 31

4.3 ภาพการต่อวงจรของหลอดไฟกับราสเบอร์รีพาย 32

4.4 ภาพการต่อวงจร LED เรียบร้อยแล้ว 32

4.5 อธิบายโค้ด Background subtraction ตัวที่1 33

4.6 อธิบายโค้ด Background subtraction ตัวที่ 2 34

4.7 อธิบายโค้ด Background subtraction ตัวที่3 35

4.8 ภาพแสดงการวาดกรอบกำหนดพื้นที่ ROI 36

4.9 อธิบายโค้ด Background subtraction ตัวที่4 36

4.10 อธิบายโค้ด Background subtraction ตัวที่5....38

4.11 ภาพแสดงการตรวจพบรถไฟ ในบริเวณพื้นที่ ROI....39

4.12 อธิบายโค้ด Haar Cascade ตัวที่1....39

4.13 ภาพแสดงการตรวจจับด้วย วิธี Haar Cascade....40

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

โดยปกติแล้วรถไฟไม่สามารถเบรกได้ทันทีในระยะกระชั้นชิด เนื่องจากรถไฟในแต่ละขบวนจะมีระยะในการเบรกของแต่ละขบวนที่แตกต่างกัน ดังนั้นเมื่อมีสิ่งกีดขวางอยู่ในระยะใกล้ หรือในระยะที่สายตาสามารถมองเห็นรถไฟจึงไม่สามารถเบรกได้ในทันที จนในบางครั้งอาจทำให้เกิดการสูญเสีย

อุบัติเหตุบริเวณจุดตัดทางรถไฟเป็นอุบัติเหตุที่เกิดขึ้นบ่อยครั้ง ส่วนใหญ่แล้วมักจะเกิดจากความประมาทของผู้ใช้เส้นทาง 2 ปัญหาหลักที่พบ คือ ผู้ขับขี่รถยนต์ไม่หยุดชะลอรถตรงบริเวณจุดตัดทางรถไฟ และบุคคลข้ามทางรถไฟด้วยความประมาท ในขณะที่รถไฟเข้าใกล้บริเวณจุดตัดทางรถไฟ ซึ่งปัญหาดังกล่าวเกิดขึ้นเป็นจำนวนมากบริเวณจุดตัดทางข้ามรถไฟที่ไม่มีไม้กั้น หรือสัญญาณไฟและป้ายเตือนให้ระวังรถไฟที่อาจวิ่งผ่านมา จากข้อมูลการรถไฟแห่งประเทศไทยใน พ.ศ. 2552 โครงข่ายทางรถไฟมีความยาวทั้งสิ้น 4,043 กิโลเมตร มีจุดตัดผ่านระหว่างถนนกับทางรถไฟทั้งสิ้น 2,463 แห่ง เป็นจุดตัดที่ได้รับอนุญาต จากการรถไฟแห่งประเทศไทยจำนวน 1,923 แห่ง ทางลักผ่านประมาณ 540 แห่ง ทั้งนี้การรถไฟแห่งประเทศไทยได้ติดตั้งระบบป้องกันอุบัติเหตุหรือเครื่องกั้นถนน(รวมทั้งประเภทไฟเตือนอัตโนมัติ) เพียงประมาณ 669 แห่ง จึงทำให้เห็นว่ายังมีจุดตัดผ่านอีกมากกว่า 1,000 แห่ง ที่ยังไม่มีการติดตั้งระบบป้องกันอุบัติเหตุหรือเครื่องกั้นถนน และจากข้อมูลของการรถไฟไทยในปี พ.ศ. 2559 พบว่ามีจุดตัดทางรถไฟทั่วประเทศ 2,624 แห่ง โดยมีจุดตัดทางรถไฟที่ได้รับอนุญาต 1,935 แห่ง และจุดตัดทางรถไฟที่ไม่ได้รับอนุญาตหรือทางลักผ่าน689แห่ง

เพื่อลดความเสี่ยงในการเกิดอุบัติเหตุดังกล่าว คณะผู้จัดทำโครงงานจึงมีแนวคิดที่จะออกแบบชุดต้นแบบระบบแจ้งเตือนรถไฟอัตโนมัติ โดยอุปกรณ์นี้จะทำให้การดำรงชีวิตของคนในท้องถิ่นรวมถึงผู้ที่ใช้รถในการสัญจรไปมาผ่านบริเวณจุดตัดทางรถไฟ สามารถรับรู้ได้เมื่อรถไฟกำลังเคลื่อนที่เข้าใกล้บริเวณจุดตัดทางรถไฟ ทำให้การสัญจรไปมามีความสะดวกสบายมากขึ้น และมีความปลอดภัยมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อออกแบบชุดต้นแบบระบบเตือนรถไฟสำหรับนำไปใช้งานบริเวณทางข้ามหรือบริเวณจุดตัดทางรถไฟในแถบชนบทที่ยังไม่มีไม้กั้น

1.2.2 เพื่อลดความเสี่ยงในการเกิดอุบัติเหตุบริเวณจุดตัดทางรถไฟ

1.2.3 เพื่อออกแบบและสร้างอุปกรณ์ตรวจจับรถไฟอัตโนมัติที่ใช้งบประมาณน้อย

1.2.4 ศึกษาการเขียนโปรแกรมด้วยกระบวนการ Image Processing เพื่อควบคุมการทำงานของชุดแจ้งเตือนรถไฟบริเวณทางข้ามรถไฟโดยใช้ Raspberry PI

1.2.5 เพื่อศึกษารูปร่างลักษณะต่างๆของรถไฟที่ใช้ในเส้นทางดังกล่าว

1.3 ขอบเขตของโครงงาน

1.3.1 ออกแบบชุดต้นแบบอุปกรณ์ตรวจจับรถไฟและส่งสัญญาณเตือนด้วยหลอดไฟ LED

1.3.2 ระบบสามารถแจ้งเตือนได้เมื่อมีรถไฟเคลื่อนที่ผ่านบริเวณที่กำหนด

1.3.3 ศึกษากระบวนการตรวจจับด้วยวิธีการประมวลผลภาพ (Image Processing)

1.3.4 ศึกษาวิธีการประมวลผลภาพด้วยวิธี Haar Cascade

1.3.5 ศึกษาวิธีการประมวลผลภาพด้วยวิธี Background subtraction

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 อุปกรณ์สามารถแจ้งเตือนได้ เมื่อมีรถไฟอยู่ภายในระยะที่กำหนด

1.4.2 ทำให้ความเสี่ยงในการเกิดอุบัติเหตุที่จะเกิดบริเวณจุดตัดและทางข้ามทางรถไฟลดลง

1.4.3 ทำให้คณะผู้จัดทำมีความรู้ในเรื่องการเขียนโค้ดด้วยภาษา Python มากขึ้นรวมถึงรู้วิธีการสร้างโปรแกรม เพื่อให้บรรลุตามวัตถุประสงค์ของโครงงาน

1.4.4 ทำให้คณะผู้จัดทำได้เรียนรู้การดำเนินงานที่เป็นระบบและมีแบบแผน และสามารถทำงานเป็นทีมได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.5 ทำให้คณะผู้จัดทำได้เรียนรู้วิธีการเลือกซื้ออุปกรณ์ และการคาดการณ์ล่วงหน้าในอนาคตว่าอุปกรณ์รุ่นไหนมีประสิทธิภาพเหมาะสมกับการใช้งาน

1.4.6 ผู้จัดทำสามารถนำความรู้ที่ได้จากการจัดทำโครงงานนี้ ไปใช้ประโยชน์ต่อยอดหรือเพื่อเป็นแนวทางในการประยุกต์ใช้ร่วมกับการประกอบอาชีพในอนาคต

1.4.7 เพื่อเป็นแนวทางให้แก่ผู้ที่สนใจหรือผู้ที่ต้องการพัฒนาต่อยอดชุดระบบตรวจจับรถไฟให้ดียิ่งขึ้นต่อไปในอนาคต

1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษากระบวนการตรวจจับด้วยการประมวลผลภาพ (Image Processing)

1.5.2 ศึกษาการใช้ภาษา Python ในการเขียนโค้ด

1.5.3 ศึกษาวิธีการใช้บอร์ดราสเบอรี่พาย และอุปกรณ์ต่างๆที่ใช้ประกอบ

1.5.4 ค้นคว้าหาข้อมูลเกี่ยวกับรูปร่างลักษณะของขบวนรถไฟที่ผ่านสถานีหัวตะเข้ และป้ายหยุดรถพระจอมเกล้า ฯ รวมถึงจำนวนเที่ยวที่ใช้เดินขบวน

1.5.5 ออกแบบและวางแผนการทำระบบการทำงานของชุดตรวจจับ

1.5.6 จัดซื้ออุปกรณ์ตามที่ได้ออกแบบไว้

1.) Raspberry Pi Board

2.) Raspberry Pi 2/3 Case box

3.) แผ่นระบายความร้อน (Heat sink)

4.) HDMI Cable for Raspberry Pi

5.) Raspberry Pi Camera Module Extension Cable 75 cm

6.) สายแพ ผู้-เมีย

7.) กล่องใส่อุปกรณ์ชุดออกแบบ

8.) แบตเตอรี่สำรอง

1.5.7 ทดสอบอุปกรณ์ต่างๆว่าสามารถใช้งานได้ตามปกติหรือไม่

1.5.8 ประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน ทำการเขียนโค้ดที่ใช้ตรวจจับรถไฟขึ้นมา

1.5.9 ทดสอบกับสถานที่จริงที่มีขบวนรถไฟวิ่งผ่าน

1.5.10 วิเคราะห์ผลการทดสอบ แก้ไขปรับปรุง พร้อมทั้งสรุปผลการทดสอบที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ข้อมูลและทฤษฎีที่เกี่ยวข้อง

2.1 ข้อมูลที่เกี่ยวข้อง

2.1.1 ลักษณะรถไฟที่วิ่งผ่านป้ายหยุดรถพระจอมเกล้า

ลักษณะรถไฟที่วิ่งผ่านป้ายหยุดรถพระจอมเกล้า แบ่งออกเป็น 2 ประเภท คือ รถโดยสาร และรถขนส่งสินค้า

2.1.1.1 รถโดยสารแบ่งเป็นยี่ห้อ

1.) ยี่ห้อ Alsthom รุ่น AD24C (ALS, AHK, ALD, ADD) ดังรูปที่ 2.1 ใช้ในขบวนที่ 275 281 367 368 277 391 371

รูปที่ 2.1 รถไฟโดยสารยี่ห้อ Alsthom

(ภาพจาก Facebook :Master Tong Apollo [7])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ยี่ห้อ Tokyu (THN,NKF) ดังรูปที่ 2.2 ใช้ในขบวนที่ 376 378 281 389 279 383

รูปที่ 2.2 รถไฟโดยสารยี่ห้อ Tokyu

(ภาพจาก Facebook :Master Tong Apollo [7])

3.) ยี่ห้อ General Electric รุ่น GE UM12C (GEK) ดังรูปที่ 2.3 ใช้ในขบวนที่283 285 281

รูปที่ 2.3 รถไฟโดยสารยี่ห้อ General Electric

(ภาพจาก Facebook :Master Tong Apollo [7])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.2 รถขนส่งสินค้าแบ่งเป็นยี่ห้อ

1.) ยี่ห้อ Hitachi รุ่น 8FA-36C (HID) ดังรูปที่ 2.4

รูปที่ 2.4 รถไฟขนส่งสินค้ายี่ห้อ Hitachi

(ภาพจาก Facebook :Master Tong Apollo [7])

2.) ยี่ห้อ CSR Qishuyan รุ่น sda3 (CSR-U20) ดังรูปที่ 2.5

รูปที่ 2.5 รถไฟขนส่งสินค้ายี่ห้อ CSR Qishuyan

(ภาพจาก Facebook :Master Tong Apollo [7])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) ยี่ห้อ General Electric รุ่น GE CM22-7i (GEA) ดังรูปที่ 2.6

รูปที่ 2.6 รถไฟขนส่งสินค้ายี่ห้อ General Electric
(ภาพจาก Facebook :Master Tong Apollo [7])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 จำนวนขบวนที่รถไฟที่วิ่งผ่านป้ายหยุดรถพระจอมเกล้า

ตารางที่ 2.1 การเดินรถไฟ “เที่ยวออกจากกรุงเทพ” สายตะวันออก ของรถไฟโดยสาร (ข้อมูลจาก ตารางเวลลาการเดินรถไฟแห่งประเทศไทย [8])

ขบวนรถ	ต้นทาง		หัวตะเข้	ปลายทาง		หมายเหตุ
	ชื่อสถานี	เวลาออก		ชื่อสถานี	เวลาถึง	
379	มักกะสัน	04:28	05:10	หัวตะเข้	05:10	ไม่มีเดินวันเสาร์-อาทิตย์และวันหยุดราชการ
376	รังสิต	05:35	07:40	หัวตะเข้	07:40	ไม่มีเดินวันเสาร์-อาทิตย์และวันหยุดราชการ
275	กรุงเทพ	05:55	07:02	อรัญประเทศ	11:35	
283	กรุงเทพ	06:55	08:13	บ้านพลูตาหลวง	11:20	เดินเฉพาะวันจันทร์-ศุกร์
285	กรุงเทพ	06:55	08:13	ชุมทางฉะเชิงเทรา	08:56	เดินรถเฉพาะวันเสาร์-อาทิตย์
281	กรุงเทพ	08:00	8:56	กบินทร์บุรี	11:35	
367	กรุงเทพ	10:10	11:08	ชุมทางฉะเชิงเทรา	11:30	
389	กรุงเทพ	12:10	13:02	ชุมทางฉะเชิงเทรา	13:30	
279	กรุงเทพ	13:05	13:47	อรัญประเทศ	17:35	
277	กรุงเทพ	15:25	16:17	กบินทร์บุรี	18:20	
391	กรุงเทพ	16:55	17:51	ชุมทางฉะเชิงเทรา	18:15	
371	กรุงเทพ	17:40	18:41	ปราจีนบุรี	20:55	
383	กรุงเทพ	18:25	19:23	ชุมทางฉะเชิงเทรา	20:00	เดินเฉพาะวันทำงานราชการ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การเดินรถไฟ “เที่ยวกลับเข้ากรุงเทพ” สายตะวันออก ของรถไฟโดยสาร (ข้อมูลจาก ตารางเวลลาการเดินรถไฟแห่งประเทศไทย [8])

ขบวนรถ	ต้นทาง		หัวตะเข้	ปลายทาง		หมายเหตุ
	ชื่อสถานี	เวลาออก		ชื่อสถานี	เวลาถึง	
380	หัวตะเข้	06:00	05:10	กรุงเทพ	07:10	ไม่มีเดินวันเสาร์-อาทิตย์และวันหยุดราชการ
384	ชุมทางฉะเชิงเทรา	05:45	06:26	กรุงเทพ	07:45	เดินเฉพาะวันทำงานราชการ
372	ปราจีนบุรี	05:00	07:01	กรุงเทพ	08:15	
388	ชุมทางฉะเชิงเทรา	07:05	07:35	กรุงเทพ	08:35	
378	หัวตะเข้	08:05	08:05	กรุงเทพ	09:20	ไม่มีเดินวันเสาร์-อาทิตย์และวันหยุดราชการ
278	กบินทร์บุรี	06:00	09:11	กรุงเทพ	10:15	
280	อรัญประเทศ	06:40	11:07	กรุงเทพ	12:05	
368	ชุมทางฉะเชิงเทรา	12:35	13:16	กรุงเทพ	14:10	
390	ชุมทางฉะเชิงเทรา	14:05	14:35	กรุงเทพ	15:25	
282	กบินทร์บุรี	13:25	16:09	กรุงเทพ	17:15	
284	บ้านพลูตาหลวง	13:35	17:06	กรุงเทพ	18:15	เดินเฉพาะวันจันทร์-ศุกร์
286	ชุมทางฉะเชิงเทรา	16:20	17:16	กรุงเทพ	18:25	เดินเฉพาะวันเสาร์ -อาทิตย์
276	อรัญประเทศ	13:40	18:42	กรุงเทพ	19:40	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การเดินรถไฟ “เที่ยวออกจากกรุงเทพ”สายตะวันออกของรถไฟขนส่งสินค้า (ข้อมูลจาก สถานีรถไฟหัวตะเข้ [9])

ขบวนรถ	ต้นทาง		หัวตะเข้	ปลายทาง		หมายเหตุ
	ชื่อสถานี	เวลาออก		ชื่อสถานี	เวลาถึง	
861	ไอซีดีลาดกระบัง	23:55	00:50	ท่าเรือแหลมฉบัง	03:05	
863	ไอซีดีลาดกระบัง	00:40	00:50	ท่าเรือแหลมฉบัง	03:45	เฉพาะวัน เสาร์-อาทิตย์
865	ไอซีดีลาดกระบัง	02:20	02:30	ท่าเรือแหลมฉบัง	05:20	
867	ไอซีดีลาดกระบัง	04:00	04:10	ท่าเรือแหลมฉบัง	07:45	
869	ไอซีดีลาดกระบัง	04:55	05:05	ท่าเรือแหลมฉบัง	08:40	
871	ไอซีดีลาดกระบัง	09:37	09:47	ท่าเรือแหลมฉบัง	12:40	เฉพาะวัน เสาร์-อาทิตย์
873	ไอซีดีลาดกระบัง	11:35	11:45	ท่าเรือแหลมฉบัง	15:10	
875	ไอซีดีลาดกระบัง	13:20	13:30	ท่าเรือแหลมฉบัง	16:05	เฉพาะวัน เสาร์-อาทิตย์
877	ไอซีดีลาดกระบัง	16:05	16:15	ท่าเรือแหลมฉบัง	19:30	
879	ไอซีดีลาดกระบัง	18:18	18:28	ท่าเรือแหลมฉบัง	21:40	
881	ไอซีดีลาดกระบัง	19:10	19:20	ท่าเรือแหลมฉบัง	22:10	
883	ไอซีดีลาดกระบัง	22:05	22:15	ท่าเรือแหลมฉบัง	01:10	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 เดินรถไฟ “เที่ยวกลับเข้ากรุงเทพ” สายตะวันออก ของรถไฟขนส่งสินค้า (ข้อมูลจาก สถานีรถไฟหัวตะเข้ [9])

ขบวนรถ	ต้นทาง		หัวตะเข้	ปลายทาง		หมายเหตุ
	ชื่อสถานี	เวลาออก		ชื่อสถานี	เวลาถึง	
862	ท่าเรือแหลมฉบัง	04:10	07:20	ไอซีดีลาดกระบัง	07:30	
864	ท่าเรือแหลมฉบัง	06:00	09:22	ไอซีดีลาดกระบัง	09:32	เฉพาะวันเสาร์- อาทิตย์
866	ท่าเรือแหลมฉบัง	06:40	10:20	ไอซีดีลาดกระบัง	10:30	
868	ท่าเรือแหลมฉบัง	08:15	13:27	ไอซีดีลาดกระบัง	13:37	
870	ท่าเรือแหลมฉบัง	10:00	14:55	ไอซีดีลาดกระบัง	15:05	
872	ท่าเรือแหลมฉบัง	13:10	15:45	ไอซีดีลาดกระบัง	15:55	เฉพาะวันเสาร์- อาทิตย์
874	ท่าเรือแหลมฉบัง	16:30	21:10	ไอซีดีลาดกระบัง	21:20	
876	ท่าเรือแหลมฉบัง	18:20	21:47	ไอซีดีลาดกระบัง	21:57	เฉพาะวันเสาร์- อาทิตย์
878	ท่าเรือแหลมฉบัง	20:25	23:37	ไอซีดีลาดกระบัง	23:47	
880	ท่าเรือแหลมฉบัง	22:30	01:20	ไอซีดีลาดกระบัง	01:30	
882	ท่าเรือแหลมฉบัง	22:55	01:57	ไอซีดีลาดกระบัง	02:07	
884	ท่าเรือแหลมฉบัง	02:40	05:35	ไอซีดีลาดกระบัง	05:45	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนขบวนรถไฟที่วิ่งผ่านสถานีหัวตะเข้ในแต่ละวัน

วันจันทร์ถึงวันศุกร์จำนวนขบวนรถไฟโดยสารเที่ยวไปและเที่ยวกลับที่ผ่านสถานีหัวตะเข้มีจำนวน 24 ขบวน ดังตารางที่ 2.1 และ ตารางที่ 2.2

วันเสาร์และวันอาทิตย์จำนวนขบวนรถไฟโดยสารเที่ยวไปและเที่ยวกลับที่ผ่านสถานีหัวตะเข้มีจำนวน 18 ขบวน ดังตารางที่ 2.1 และ ตารางที่ 2.2

วันจันทร์ถึงวันศุกร์จำนวนขบวนรถไฟขนส่งสินค้าเที่ยวไปและเที่ยวกลับที่ผ่านสถานีหัวตะเข้มีจำนวน 18 ขบวน ดังตารางที่ 2.3 และ ตารางที่ 2.4

วันเสาร์และวันอาทิตย์จำนวนขบวนรถไฟขนส่งสินค้าเที่ยวไปและเที่ยวกลับที่ผ่านสถานีหัวตะเข้มีจำนวน 24 ขบวน ดังตารางที่ 2.3 และ ตารางที่ 2.4

เฉลี่ย จำนวนขบวนรถไฟที่ผ่านสถานีหัวตะเข้ ในแต่ละวันคือ 42 ขบวน

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 การประมวลผลภาพ (Image processing)

เทคนิคการประมวลผลภาพ (image processing) ถูกพัฒนาขึ้นครั้งแรกเมื่อประมาณปี พ.ศ. 2503 โดยกลุ่มนักวิจัยจากมหาวิทยาลัย Maryland พัฒนาขึ้นเพื่ออำนวยความสะดวกในการใช้วิเคราะห์ภาพต่างๆ ต่อมากระบวนการประมวลผลภาพได้ถูกพัฒนามาอย่างต่อเนื่อง จนในปัจจุบันได้มีการนำเอาการประมวลผลภาพไปใช้กันอย่างแพร่หลาย เช่น การตรวจจับรถยนต์ การจำแนกประเภทรถยนต์ การจดจำใบหน้า การถ่ายภาพทางดาวเทียม ระบบตรวจจับความเคลื่อนไหวเพื่อรักษาความปลอดภัย รวมถึงการประยุกต์เทคนิคการประมวลผลภาพเข้ากับเทคนิคอื่นๆ จนกลายเป็นอีกเทคโนโลยีหนึ่งที่มีความแม่นยำและสามารถนำมาใช้ในกระบวนการทางการแพทย์ได้ ในส่วนทางด้านอุตสาหกรรม ก็ได้มีการนำกระบวนการประมวลผลภาพมาใช้ทำการตรวจสอบผลิตภัณฑ์ จากการตรวจสอบสีของผลิตภัณฑ์ การตรวจสอบตำแหน่ง และขนาดของผลิตภัณฑ์ ใช้ในการกำหนดตำแหน่งในการเชื่อม การกำหนดตำแหน่งในการขันสกรู เป็นต้น

การประมวลผลภาพ สามารถทำงานโดยการนำเอาภาพต่างๆมาใส่ฟังก์ชันและสมการทางคณิตศาสตร์ที่ต้องการเข้าไป เพื่อให้ได้ผลตามที่ต้องการหรือตามที่ได้วางแผนเอาไว้ ซึ่งการทำงานของกระบวนการประมวลผลภาพแบ่งออกเป็น 3 ส่วนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. อุปกรณ์รับภาพ ทำหน้าที่ ถ่ายภาพหรือตรวจจับภาพที่ต้องการเข้าสู่คอมพิวเตอร์

2. อุปกรณ์ในการประมวลผล ทำหน้าที่ประมวลผลภาพที่ได้รับมาจากอุปกรณ์รับภาพ

3. โปรแกรมการคำนวณ เป็นตัวจัดการวิธีการทำงานกับภาพที่ได้รับมา โดยจะมีการปรับปรุงคุณภาพก่อนทำการประมวลผล การใช้อัลกอริทึม และฟังก์ชันทางคณิตศาสตร์ เพื่อให้ได้ output ตามที่ต้องการ ซึ่งโปรแกรมที่ใช้ในการประมวลผลภาพมีหลายประเภท เช่น OpenCV, IMAQ, OpenVIDIA ,Scorpion และแบบที่เป็น FIRMWARE ที่ติดมากับ คอนโทรลเลอร์

2.2.2 ภาษาไพธอน (python)

ไพธอน เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวาง ภาษา ไพธอน (python) ถูกสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 ไพธอน เป็นภาษาที่ถูกออกแบบมาเพื่อทำให้โค้ดอ่านได้ง่ายขึ้นการเขียนโค้ดใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java เหมาะสำหรับงานด้านการดูแลระบบ (system administration) เป็นอย่างยิ่ง

โครงสร้างของภาษา ไพธอน (python)

1.) Module ประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกจากนี้เรายังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ใน package ซึ่งเป็นเหมือน directory ของโมดูล

2.) Comment คอมเมนท์ในภาษาไพธอน จะเริ่มด้วยเครื่องหมาย # โดยทั่วไปแล้วคอมเมนท์ที่สร้างขึ้นมักจะใช้อธิบายเกี่ยวกับโค้ดที่เขียนขึ้นมา ซึ่งมันไม่มีผลต่อการทำงานของโปรแกรม

3.) Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษาไพธอน (python) นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ แต่อย่างไรก็ตามเราสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมาย เซมิโคลอน (;)

4.) Indentation and while space การใช้ Whilespace และ Tab สำหรับกำหนดบล็อกของโปรแกรม เช่น คำสั่ง If Else หรือการประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านี้เป็นคำสั่งแบบบล็อก โดยจำนวนช่องว่างที่ใช้นั้นต้องเท่ากัน ในภาษาไพธอนจะเข้มงวดกับจำนวนช่องว่างภายในบล็อกมาก ดังนั้นทุกคำสั่งภายในบล็อกต้องมีจำนวนช่องว่างเท่ากันเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.) Literals คือเครื่องหมายที่ใช้แสดงค่าคงที่ ซึ่งในโปรแกรมภาษาไพธอน (python) นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating- point number และ string หรือแม้กระทั่งตัวอักษร และ Boolean การกำหนด literal ประเภทต่างๆให้กับตัวแปร ค่าที่เป็นตัวเลขสามารถกำหนดค่าลงไปโดยตรงได้ทันที และสามารถกำหนดในรูปแบบสั้น อย่างในตัวแปร b และสำหรับ Boolean นั้นจะเป็น ตัวอักษรส่วน String หรือ character แต่จะต้องอยู่ในเครื่องหมาย double quote ("...") หรือ single quote ('...') เสมอ

6.) Expressions เป็นการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน ในภาษาไพธอน (python) มี 2แบบ แบบที่1 คือ Boonlean Expression เป็นการกระทำกันของตัวแปรและตัวดำเนินการและจะได้ผลลัพธ์เป็นค่า Boonlean โดยทั่วไปแล้วมักจะเป็นตัวดำเนินการตรรกศาสตร์ และแบบที่2 คือ Expression ทางคณิตศาสตร์ เป็นการกระทำกันกับตัวดำเนินการและได้ค่าใหม่ที่ไม่ใช่ Boonlean

7.) Keywords เป็นคำที่สงวนไว้ในการเขียนโปรแกรมภาษา python เราไม่สามารถใช้คำสั่งเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส หรือ identifier ได้ รายการ keywords ในภาษา python ได้แก่ False ,None ,True ,and ,as ,assert ,break ,class ,continue ,def ,del ,elif ,else ,except ,finally ,for ,from ,global ,if ,import ,in ,is ,lambda ,nonlocal ,not ,or pass ,raise ,return ,try while ,with และ yield

2.2.3 โอเพนซีวี (OpenCV)

OpenCV ย่อมาจาก Open Source Computer Vision เป็น library ในภาษา C++ และ Python สำหรับการพัฒนา โปรแกรมที่เกี่ยวข้องกับ Image Processing และ Computer Vision โดยสามารถพัฒนาได้ทั้ง ในระบบปฏิบัติการวินโดว์ และระบบปฏิบัติการลีนุกซ์(Linux System) จุดเด่นในด้านความสามารถของไลบรารี่ของ OpenCV คือ สามารถประมวลผลภาพได้ทั้งภาพนิ่งและภาพเคลื่อนไหว ไลบรารี่ 4 ส่วน หลักของ OpenCV มีดังนี้

1.) ซีวี (CV) ใช้ในการประมวลผลและการวิเคราะห์รูปภาพต่างๆที่เราต้องการ

2.) ซีเอกซ์คอ (CXCORE) ใช้จัดการเกี่ยวกับจุด ขนาด อาร์เรย์ หน่วยความจำคำสั่ง

3.) Machine Learning เป็นไลบรารี่ที่รวมคลาสฟังก์ชันทางสถิติ การแยกคลาส และการแบ่งกลุ่มของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) HighGUI เป็นไลบรารี่ที่ใช้ในการดึงภาพ การบันทึกภาพ การเปลี่ยนขนาด และเคลื่อนย้ายหน้าต่าง

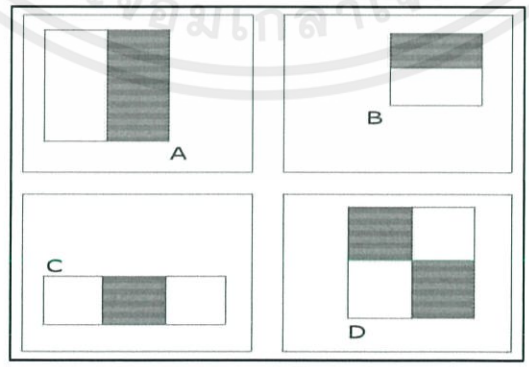
2.2.4 กระบวนการ Haar Like Feature

วิธีนี้ถูกนำเสนอเมื่อปี 2001 โดย Paul Viola และ Michael Jones ซึ่งพวกเขาใช้กระบวนการ Machine Learning ตามที่ฟังก์ชันคาสเคดได้ถูกสอนผ่านการวิเคราะห์รูป ซึ่งถูกแบ่งเป็นสองกลุ่มคือ กลุ่มรูปถูกต้อง ซึ่งคือรูปของสิ่งที่เราอยากจะตรวจจับ(Positive) ในรูปแบบต่างๆ แต่ต้องมีการตัดเอาส่วนอื่นออกให้เหลือเพียงส่วนของสิ่งที่เราต้องการเท่านั้น และกลุ่มของรูปทั่วไปที่ไม่มีสิ่งที่เราอยากจะตรวจจับอยู่ในรูปเลย (Negative) เพื่อที่จะสามารถนำข้อมูลที่ได้ไปตรวจจับในรูปอื่นๆต่อไป

วิธีที่ Paul Viola และ Michael Jones คิดค้นขึ้นมาใช้นั้น ช่วยให้ลดการใช้ทรัพยากร และเวลาในการประมวลลงได้อย่างมาก โดยพวกเขาเรียกว่า “Fast computation of Haar-like features” วิธีคือ การแบ่งพื้นที่ของภาพด้วยรูปแบบสำเร็จรูป 4 แบบที่พวกเขาได้กำหนดจาก Haar-like features ไว้ดังรูป (A – B – C – D) จากนั้นจึงคำนวณตารางสีของภาพโดยการ อินทิเกรต ตารางสีที่อยู่ในพื้นที่อยู่ในช่องสีขาวลบด้วยตารางสีที่อยู่ในพื้นที่อยู่ในช่องสีดำในกระบวนการสร้างรูอินทิกรัล แล้วนำผลที่ได้ไปใช้ใน Adaboost ที่เป็นอัลกอริทึมสำหรับการเรียนรู้แบบหนึ่ง (Adaptive Boost Learning Algorithm) ด้วยการใช้สมการดังนี้

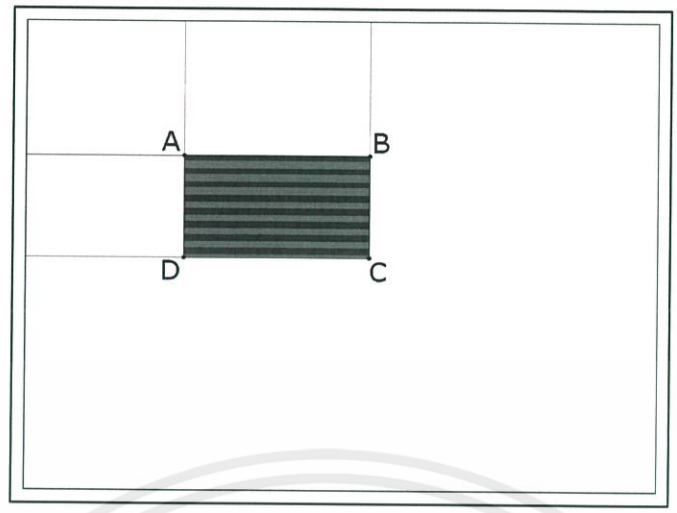
$$sum = c + a - b - d \tag{2.1}$$

ซึ่ง A, B, C และ D คือ พื้นที่ที่ถูก อินทริกรัล ดังรูปที่ 2.7 – รูปที่ 2.9

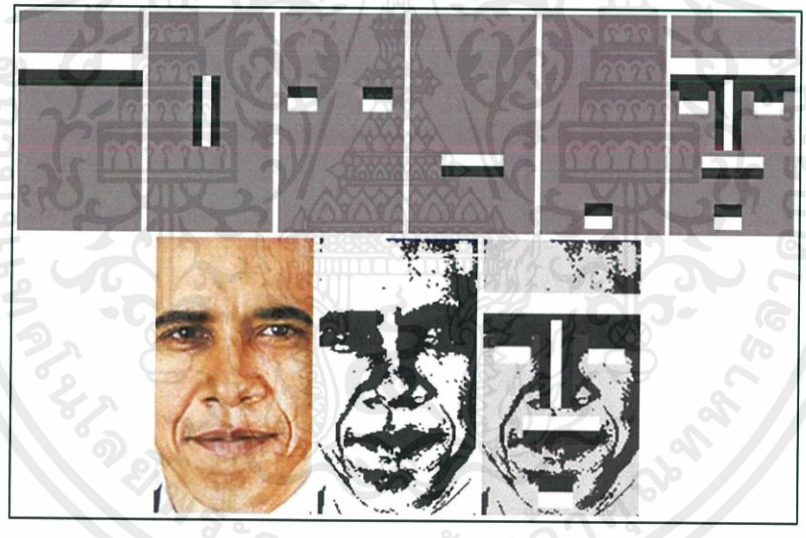


รูปที่ 2.7 ภาพแสดงแบบ Haar Like Feature 4 แบบที่ถูกเลือกใช้(A – B – C – D)
(ภาพจาก เว็บไซต์การสร้างไฟล์ Cascade ไว้ตรวจจับวัตถุ [13])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ภาพแสดงพื้นที่บริเวณที่ถูกอินทริกรัล
(ภาพจาก เว็บไซต์Background subtraction [15])



รูปที่ 2.9 ภาพแสดงพื้นที่บริเวณที่ถูกอินทริกรัลเพื่อหาพื้นที่ที่เราต้องการ
(ภาพจาก เว็บไซต์ Haar-Like for face detection [14])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 กระบวนการ Background subtraction

Background subtraction เป็นเทคนิคหนึ่งในกระบวนการ image processing โดยใช้วิธีการแยกฉากหลังด้วยการนำภาพ 2 ภาพภายในบริเวณเดียวกันในเวลาที่แตกต่างกัน แล้วนำมาลบกัน หากมีการเคลื่อนที่เข้ามาของวัตถุระบบจะนำภาพทั้ง 2 ภาพมาลบกันฉากหลังของภาพทั้ง 2 ภาพจะเกิดส่วนต่าง ซึ่งผลที่ได้ก็คือการเคลื่อนไหวของวัตถุที่เข้ามายังภาพ โดยกระบวนการนี้ใช้เทคนิคในการแยกพิกเซล(Pixel) ด้วยการตรวจสอบพิกเซล ณ ตำแหน่งเดียวกันว่าพบการเปลี่ยนแปลงหรือไม่ ซึ่งสามารถแสดงเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$P\left[F(t)\right] = P\left[I(t)\right] - P\left[B\right] \tag{2.2}$$

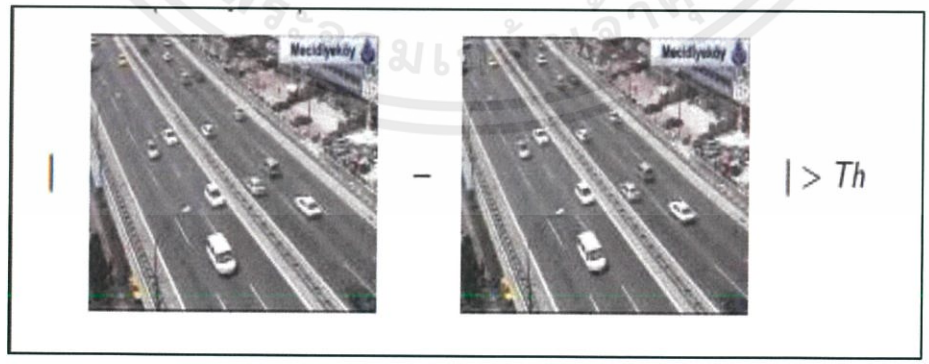
ซึ่ง $P[I(t)]$ เป็นฟังก์ชันของพิกเซล ณ เวลาใดๆ

$P\left[B\right]$ เป็นฟังก์ชันพิกเซลที่ใช้อ้างอิง (reference) ณ ตำแหน่งเดียวกัน

ภาพพื้นหลัง (Background) จะถูกนำมาใช้พิจารณาเป็นภาพอ้างอิง ณ เวลา t ใดๆ โดยฟังก์ชัน $P\left[F(t)\right]$ จะแสดงถึงความหนาแน่นของจำนวนพิกเซลที่เปลี่ยนไป

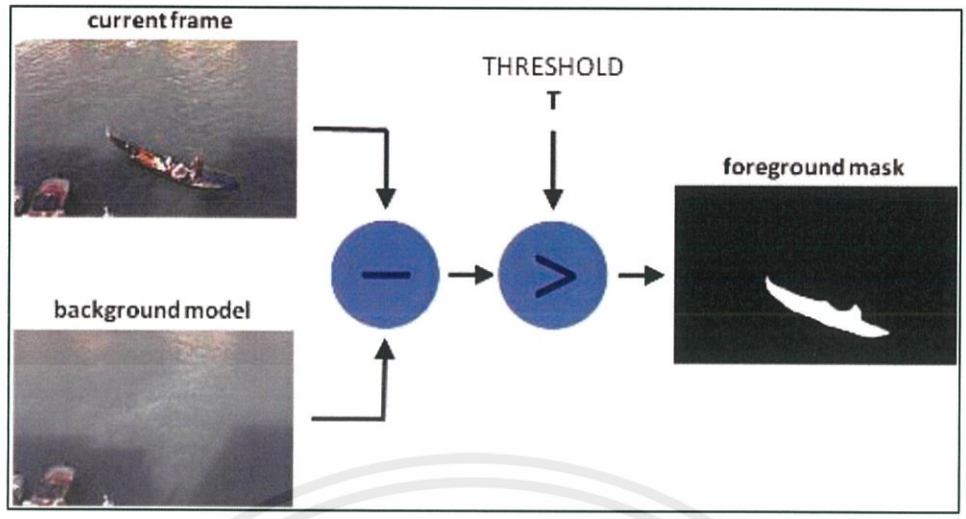
ฟังก์ชันที่ได้จะแสดงผลก็ต่อเมื่อ มีจำนวนพิกเซลเคลื่อนที่ต่างจากพิกเซลที่ใช้อ้างอิง ซึ่งเราจะเรียกความแตกต่างนี้ว่า Threshold โดยความแม่นยำจะขึ้นอยู่กับความเร็วของภาพที่เปลี่ยนไป หากภาพยิ่งมีความเคลื่อนไหวเร็วจะพบว่า ค่า Threshold จะมีค่ามาก ดังรูที่ 2.10 และรูปที่ 2.11

$$\left|P\left[F(t)\right] - P\left[F(t-1)\right]\right| > Threshold \tag{2.3}$$



รูปที่ 2.10 ภาพแสดงความแตกต่างของพิกเซล (Threshold)
(ภาพจาก เว็บไซต์ Open Source Computer (OpenCV) [16])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ภาพแสดงส่วนต่างของภาพสองภาพ
(ภาพจาก เว็บไซต์ Open Source Computer (OpenCV) [16])

โมเดลจัดเรียงตัวของพิกเซล ถูกนำเสนอโดย Stauffer and Grimson ซึ่งกระบวนการนี้แสดงได้จากสมการผลรวมของ Gaussion เรียกว่า Gaussion Mixture Model หรือ GMM กระบวนการนี้จำนวนพิกเซลจะถูกคำนวณโดยค่าเฉลี่ย(mean) μ_t และความแปรปรวน (variance) σ_t^2 โดยวิธีการคำนวณจะแสดงในสมการดังต่อไปนี้

$$\mu_t = \alpha x_t + (1+\alpha)\mu_{t-1} \tag{2.4}$$

$$\sigma_t^2 = \alpha(x_t - \mu_t)^2 + (1-\alpha)\sigma_{t-1}^2 \tag{2.5}$$

โดยที่ μ_t คือ ค่าเฉลี่ย ณ เวลา t ใดๆ

α คือ learning rate ($\alpha = 0.01$)

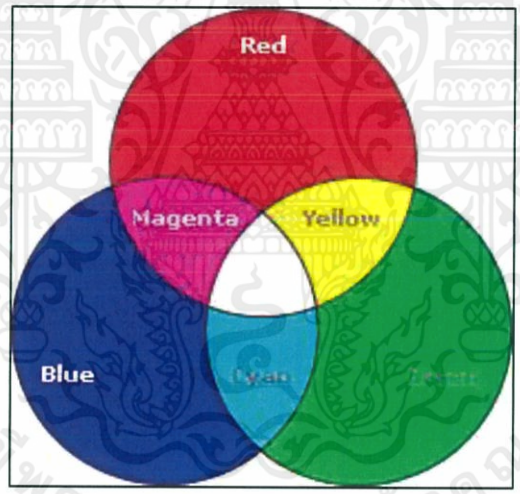
x_t คือ จำนวนพิกเซลในเฟรม ณ เวลา t ใดๆ

จากที่ได้กล่าวมาข้างต้น โมเดลการจัดเรียงตัวของพิกเซลนี้ แต่ละพิกเซลจะประกอบด้วย ตัวแปร 2 ตัวแปร คือ μ_t และ σ_t ซึ่งทำให้สามารถประมวลผลได้อย่างรวดเร็วและใช้หน่วยจำน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 ระบบสี RGB (Red, Green, Blue)

ระบบสี RGB ประกอบด้วยสีสามสี คือ สีแดง สีเขียว และสีน้ำเงิน ซึ่งการสร้างงานกราฟฟิคนั้นเราจะใช้โหมด RGB เป็นหลัก ซึ่งโหมด RGB นี้สีที่ได้จะเกิดจากการผสมแสงสามสีให้เกิดเป็นจุดสี ระบบสี RGB เป็นระบบสีของแสงผ่านแท่งแก้วปริซึม จะเกิดแถบสีที่เรียกว่า สีรุ้ง หรือ spectrum โดยแยกสีตามที่สายตาสามารถมองเห็นได้ คือ สีแดง สีแสด สีเขียว สีน้ำเงิน สีคราม สีม่วง ซึ่งเป็นพลังงานที่อยู่ในรูปของรังสี ที่มีช่วงคลื่นที่สายตาสามารถมองเห็นได้ แสงสีม่วงมีความถี่คลื่นสูงที่สุดคลื่นแสงที่มีความถี่สูงกว่าแสงสีม่วง เรียกว่า อัลตราไวโอเลต (Ultra Violet) และคลื่นแสงสีแดง มีความถี่คลื่นต่ำที่สุด คลื่นแสงที่ต่ำกว่าแสงสีแดงเรียกว่า อินฟราเรด (Infrared) คลื่นแสงที่มีความถี่สูงกว่าสีม่วงและต่ำกว่าแสงสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ และดังรูปที่ 2.12 พบว่าเมื่อศึกษาดูแล้วแสงสีทั้งหมดเกิดจาก แสง 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่อีก 3 สี คือ สีแดงมาเจนต้า สีฟ้าไซแอน และสีเหลืองและถ้าฉายแสงสีทั้งหมดรวมกันจะได้แสงสีขาว



รูปที่ 2.12 ภาพแสดงระบบสี RGB

(ภาพจาก เว็บไซต์ทฤษฎีสีของระบบสี RGB ,SARA DEE [17])

2.2.7 ระบบสี HSV color system

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก (แดง เขียว และน้ำเงิน) ในทางปฏิบัติค่าจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มุมมีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถเห็นได้ในรูปขององศาได้ ดังนี้คือ

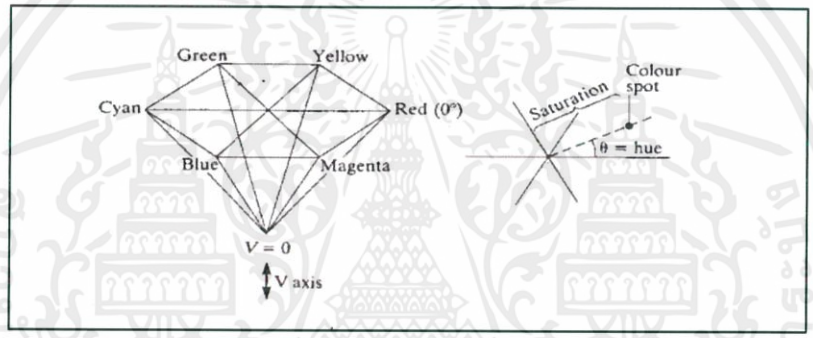
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สีแดง = 0 องศา สีเขียว เท่ากับ120 องศา สีน้ำเงินเท่ากับ 240 องศา Hue สามารถคำนวณได้จากระบบสี RGB ดังนี้

$$red_h = red - \min(red, green, blue) \tag{2.1}$$

$$green_h = green - \min(red, green, blue) \tag{2.2}$$

$$blue_h = blue - \min(red, green, blue) \tag{2.3}$$



รูปที่ 2.13 แสดงระบบสี HSV

(ภาพจาก เว็บไซต์การประมวลผลภาพดิจิตอล,ระบบสี HSV, [18])

จากโมเดลพบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี(ค่าสี) ดังรูปที่ 2.13 ซึ่งจะมีค่าเปลี่ยนไปตามสีที่สามและถ้าสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\frac{(240 \times blue_h) + (120 \times green_h)}{blue_h + green_h} \tag{2.4}$$

Saturation คือค่าความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีสีแสงสีขาวผสมอยู่เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณ Saturation

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)} \quad (2.5)$$

Value คือค่าความสว่างของสี ซึ่งความสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

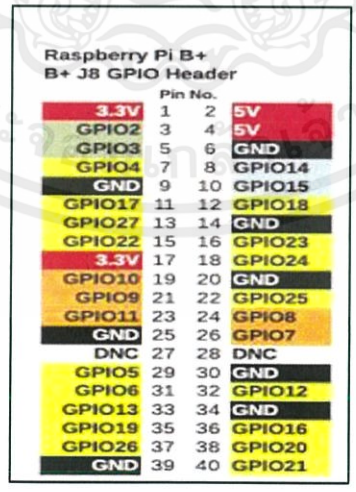
$$Value = \max(red, green, blue) \quad (2.6)$$

2.2.8 Raspberry Pi 3 Model B

Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อเข้ากับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาใช้งานในด้านการเขียนโปรแกรม หรือใช้เป็นคอมพิวเตอร์ขนาดเล็กได้ บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian), Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card

ส่วนประกอบของบอร์ด Raspberry Pi (ดังรูปที่ 2.14 ถึงรูปที่ 2.20)

1.) พอร์ต GPIO

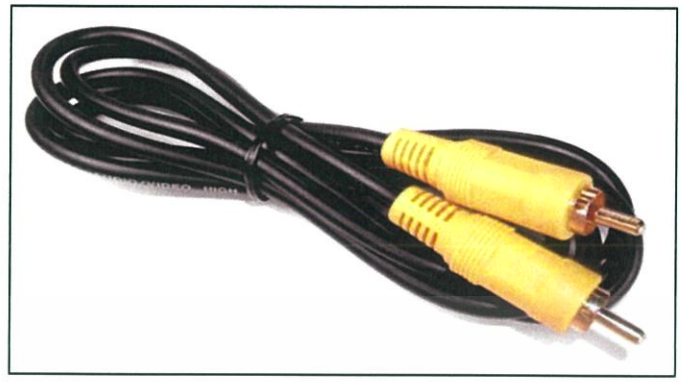


รูปที่ 2.14 แสดง GPIO ของ Raspberry Pi

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) พอร์ตเชื่อมต่อสัญญาณภาพออกแบบ RCA ตัวอย่างของสายที่เชื่อมต่อแสดง ดังรูป

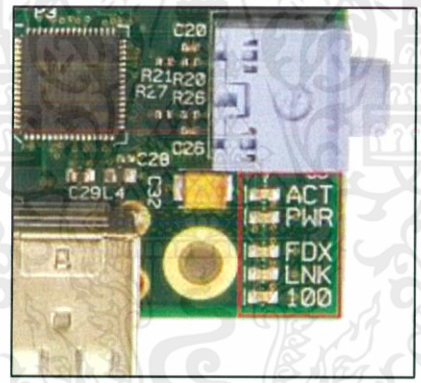


รูปที่ 2.15 สายที่เชื่อมต่อ

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

3.) จุดเชื่อมต่อสัญญาณเสียงขนาด 3.5 มิลลิเมตร

4.) LED แสดงสถานะของบอร์ด อยู่ภายในบริเวณกรอบสีแดง ดังภาพ



รูปที่ 2.16 LED แสดงสถานะของบอร์ด

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

ACT คือ ไฟสถานะ SD Card Access *(เขียว)*

PWR คือ ไฟสถานะ 3.3 V Power *(สีแดง)*

FDX คือ ไฟสถานะ Full Duplex LAN Model B *(สีเขียว)*

LNK คือ ไฟสถานะ Link/Activity LAN Model B (สีเขียว)

100 คือ ไฟสถานะ 10/100 Mbps LAN Model B (สีเหลือง)

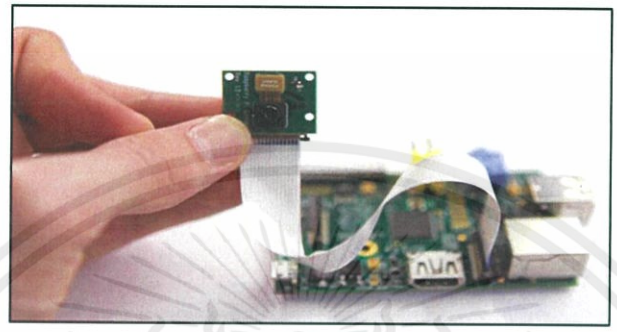
5.) ชิปควบคุม LAN (LAN Controller)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.) พอร์ต USB 2.0 จำนวน 2 พอร์ต

7.) พอร์ต RJ-45 Ethernet LAN 10/100 Mbps

8.) พอร์ต CSI (Camera Serial Interface) สำหรับเชื่อมต่อโมดูลกล้องดังภาพ แสดงตัวอย่างโมดูลกล้อง



รูปที่ 2.17 ภาพแสดง พอร์ต CSI

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

9.) พอร์ต HDMI สำหรับเชื่อมต่อสัญญาณภาพและเสียง ตัวอย่างสาย HDMI และตัวแปลง HDMI to VGA แสดงดังรูปด้านล่าง

รูปที่ 2.18 สาย HDMI

รูปที่ 2.19 HDMI to VGA

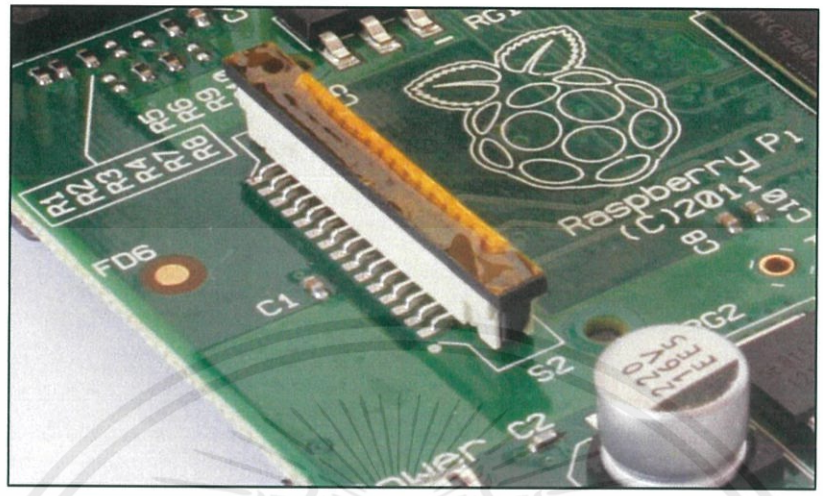
(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

10.) ชิพ Broadcom BCM2835 ARM11 700MHz

11.) พอร์ต Micro USB Power สำหรับเป็นไฟเลี้ยงวงจรบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.) พอร์ต DSI (Display Serial Interface) ใช้สำหรับต่อจอแสดงผล เช่น จอแสดงผลแบบ TFT Touch Screen เป็นต้น



รูปที่ 2.20 พอร์ต DSI (Display Serial Interface)

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

13.) ช่องเสียบ SD Card อยู่บริเวณด้านล่างของบอร์ด

2.2.9 Raspberry PI Camera Module V2

สเปคของกล้องที่ใช้ : 8 megapixels fixed focus

Supports 1080p, 720p, VGA

Sony IMX219PQ CMOS image sensor

Frame rate 30 fps

รูปที่ 2.21 Raspberry PI Camera Module V2

(ภาพจาก เว็บไซต์การพัฒนาโปรแกรมบน Raspberry Pi [19])

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 คมกฤช และ คณะ (2550) [1] ได้ทำการสร้างชุดเตือนรถไฟด้วยอุปกรณ์ตรวจจับแรงสั่นสะเทือน พบว่าความไวต่อแรงสั่นสะเทือนที่ค่าความต้านทาน 49-72 KΩ ตัวตรวจจับแรงสั่นสะเทือนสามารถทำงานได้ปกติในทุกขบวน แต่การส่งสัญญาณของเครื่องรับส่งข้อมูลไร้สายพบว่าไม่ควรมีสิ่งกีดขวางเพราะจะทำให้คลื่นไม่สามารถผ่านไปได้

2.3.2 บริพัตร และคณะ [2] ได้ทำการวิจัยเรื่อง ชุดระบบเตือนภัยทางข้ามรถไฟด้วยการใช้แสงตรวจจับการเคลื่อนที่ของรถไฟ พบว่าชุดตรวจจับด้วยแสงมีประสิทธิภาพการทำงานรวมของทั้งระบบเตือนภัยมีค่า 95% เนื่องจากขณะทำการทดลองมีคลื่นสัญญาณวิทยุอื่นรบกวน ทำให้เกิดปัญหาการสื่อสารในระบบ

2.3.3 ปุณณรัตน์ และคณะ [3] ได้ศึกษา ระบบแจ้งเตือนสิ่งกีดขวางสำหรับคนพิการทางสายตาด้วยกระบวนการ Image Processing พบว่า พื้นที่โล่งเป็นปัจจัยหนึ่งที่ทำให้ภาพที่ได้ไม่เกิดการทับซ้อน และระบบสามารถตรวจสอบข้อมูลที่ได้รับมาไปเปรียบเทียบกับฐานข้อมูลได้อย่างแม่นยำ

2.3.4 Rakesh [4] ได้ทำการทดลองตรวจนับจำนวนคนโดยใช้เทคนิค ROI แบบ Real Time พบว่าระบบมีความแม่นยำและมีเสถียรภาพมาก

2.3.5 M. Oliveira, V. Santos [5] ได้ทำการวิจัยเรื่องการตรวจจับรถบนท้องถนนโดยกระบวนการ Haar-like features พบว่า เมื่อขณะวัตถุมีการเคลื่อนที่ ระบบจะทำการตรวจจับได้น้อยกว่าวัตถุอยู่นิ่ง ถ้าต้องการการตรวจจับที่แม่นยำมาก กล้องจะต้องอยู่กับที่และกำหนดขอบเขตที่ต้องการจะตรวจจับด้วย

2.3.6 ศรราม และ คณะ [6] โครงงานนี้ใช้กล้อง IP Camera ซึ่งยึดติดอยู่กับที่ มาใช้ในระบบตรวจจับการเคลื่อนไหวของคนภายในห้องแบบอัตโนมัติ โดยใช้เทคนิคการลบภาพพื้นหลัง และ ทำการติดตามคนโดยการสั่งงานไปที่ Servo motor ไปยังตำแหน่งของคน และสามารถ เข้าดูภาพการเคลื่อนไหวภายในห้องผ่าน Web browser ในระบบ Internet ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและสร้างระบบ

3.1 เครื่องมือที่ใช้เขียนโปรแกรม

3.1.1 Raspberry Pi 3 model B

3.1.2 Python (Programming language)

3.1.3 OpenCV

3.2 หลักการออกแบบโปรแกรม

ในการออกแบบระบบตรวจจับรถไฟอาศัยแนวคิดจากกล้องตรวจจับพาหนะบนท้องถนน โดยการเปลี่ยนรูปแบบจากการตรวจจับรถยนต์มาเป็นรถไฟ ซึ่งใช้ภาษา Python ในการพัฒนาร่วมกับการใช้ชุดเครื่องมือ OpenCV ที่เป็นซอฟต์แวร์สำหรับใช้ในการประมวลผลภาพ (image-processing) ด้วยการนำ Background subtraction ซึ่งเป็นฟังก์ชันหนึ่งใน library ของชุดเครื่องมือ OpenCV ที่นำมาช่วยในการออกแบบ

เทคนิคแรกที่จะนำมาใช้งานคือเทคนิคการตรวจจับรถไฟด้วยฟังก์ชัน Haar Cascade เป็นการทำให้ตัวประมวลผลทำความรู้จักกับรถไฟ โดยวิธีการทำให้รู้จักรถไฟนั้นเราจะนำภาพรถไฟมาจำนวนหนึ่ง และแบ่งข้อมูลภาพเป็นสองส่วนคือ ภาพ positive (ดังรูปที่ 3.1) เป็นภาพที่เราตัดมาเฉพาะส่วนรถไฟเพียงส่วนเดียว เป็นภาพที่เราต้องการให้ระบบตรวจจับ และ ภาพ negative (ดังรูปที่ 3.2) เป็นภาพที่มีองค์ประกอบอยู่ภายในรูปที่มีรถไฟ เป็นภาพที่เราไม่ต้องการให้ระบบตรวจจับ เช่น ต้นไม้ ก้อนหิน รางรถไฟ ป้ายหยุดรถ สถานีรถไฟ คน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 ภาพแสดง positive เฉพาะส่วนรถไฟ

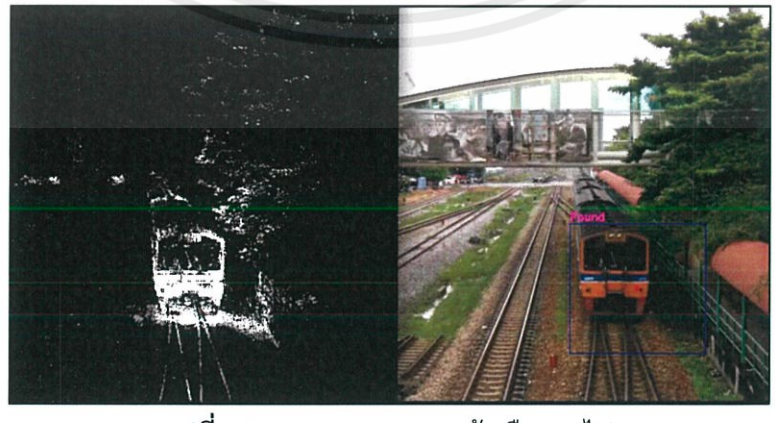
รูปที่ 3.2 ภาพแสดง negative

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีรถไฟวิ่งผ่านมา ระบบจะประมวลผลว่าภาพไหนเป็น positive ภาพไหนเป็น negative (ดังรูปที่3.3) และจะตรวจจับเฉพาะส่วนที่เป็น positive นั่นก็คือรูปรถไฟ

รูปที่ 3.3 ภาพแสดงการตรวจจับรถไฟด้วย วิธี Haar Cascade

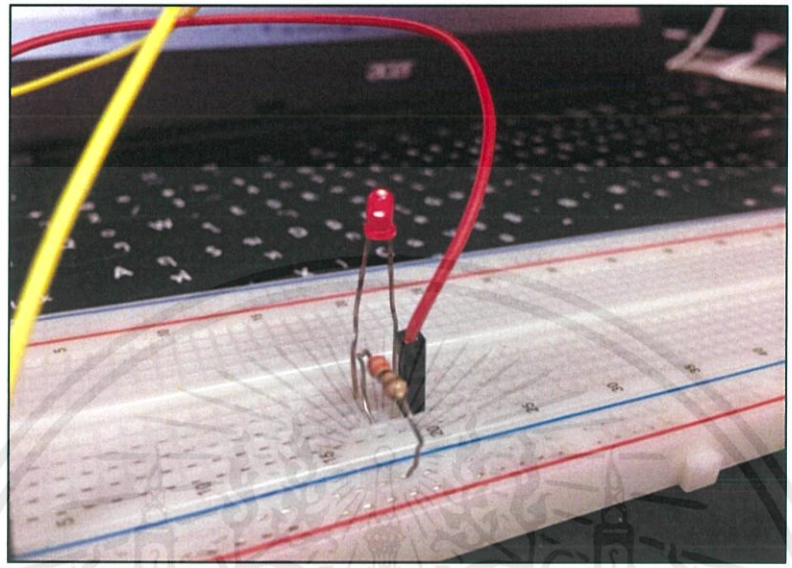
เทคนิคต่อมาเป็นเทคนิคในการตรวจจับรถไฟด้วยฟังก์ชัน Background subtraction เป็นเทคนิคที่ใช้ในการตรวจจับการเคลื่อนไหวภายในรูป มีกระบวนการคือ หาค่าความแตกต่างระหว่าง foreground กับ background โดยที่เราจะสามารถบอกได้ว่ามีวัตถุอะไรที่กำลังเคลื่อนไหวอยู่ในเฟรม (frame) ซึ่งเทคนิคนี้จะทำการบันทึกภาพ background ก่อนที่จะมีรถไฟวิ่งเข้ามา แล้วเมื่อมีรถไฟวิ่งเข้ามาภายในเฟรมระบบจะตรวจพบว่ามีความแตกต่างระหว่างภาพ background กับภาพขณะที่มีการเคลื่อนที่ของรถไฟ (ดังรูปที่ 3.4) จากนั้นระบบจึงแจ้งเตือนว่าพบรถไฟกำลังเคลื่อนที่ผ่านบริเวณกล้องของเรา



รูปที่ 3.4 ภาพแสดงการแจ้งเตือนรถไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกระบวนการที่กล่าวมาข้างต้น เมื่อระบบตรวจจับรถไฟตรวจพบรถไฟแล้ว จะทำการส่งสัญญาณไปที่ไฟแจ้งเตือน ซึ่งชุดการออกแบบตัวต้นแบบ ผู้จัดทำได้นำหลอดไฟ LED มาใช้แทนในการส่งสัญญาณไฟแจ้งเตือน ดังรูปที่ 3.5



รูปที่ 3.5 ภาพแสดงการแจ้งเตือนจากหลอด LED

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

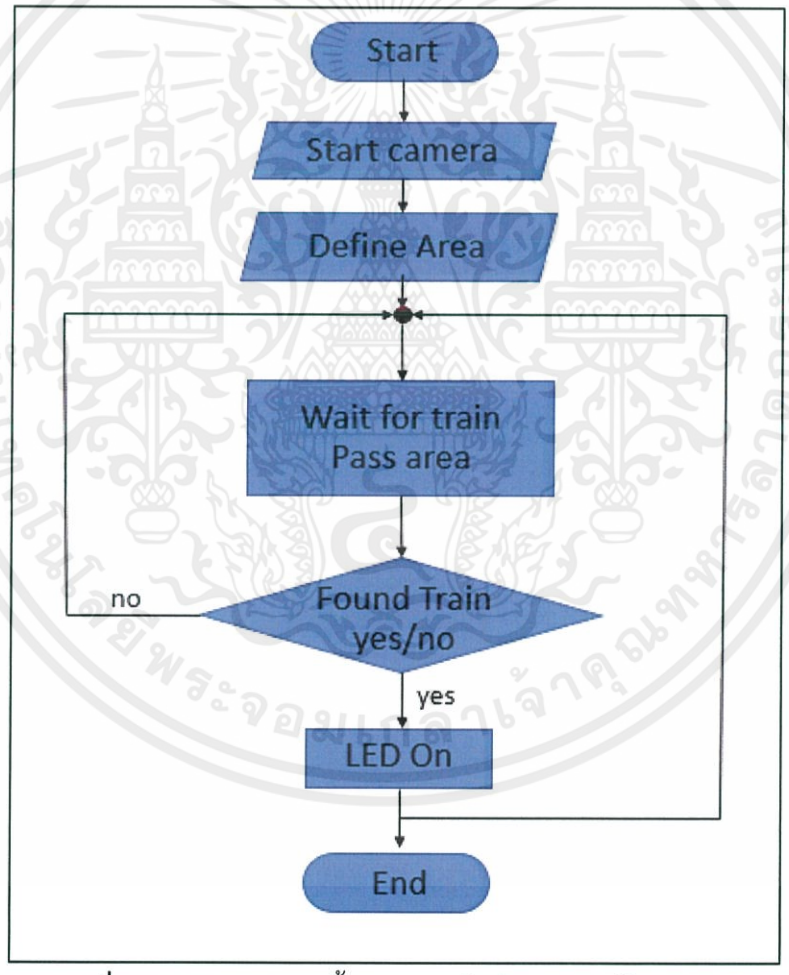
บทที่ 4

ขั้นตอนการดำเนินงาน

4.1 ขั้นตอนการสร้างระบบ

(1) ศึกษาเครื่องมือที่ใช้ในการพัฒนาโปรแกรม เช่น Python, OpenCV

(2) ออกแบบขั้นตอนการดำเนินงาน (Process) ของโปรแกรม พร้อมทั้งเขียน Flowchart ดังรูปที่ 4.1



รูปที่ 4.1 Flowchart ขั้นตอนการดำเนินงานของโปรแกรม

(3) เตรียมพร้อมทรัพยากรที่ต้องใช้งาน ติดตั้ง OS และซอฟต์แวร์ที่จำเป็นให้กับ Raspberry Pi พร้อมทั้งเชื่อมต่อเข้ากับกล้อง และชุด LED Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

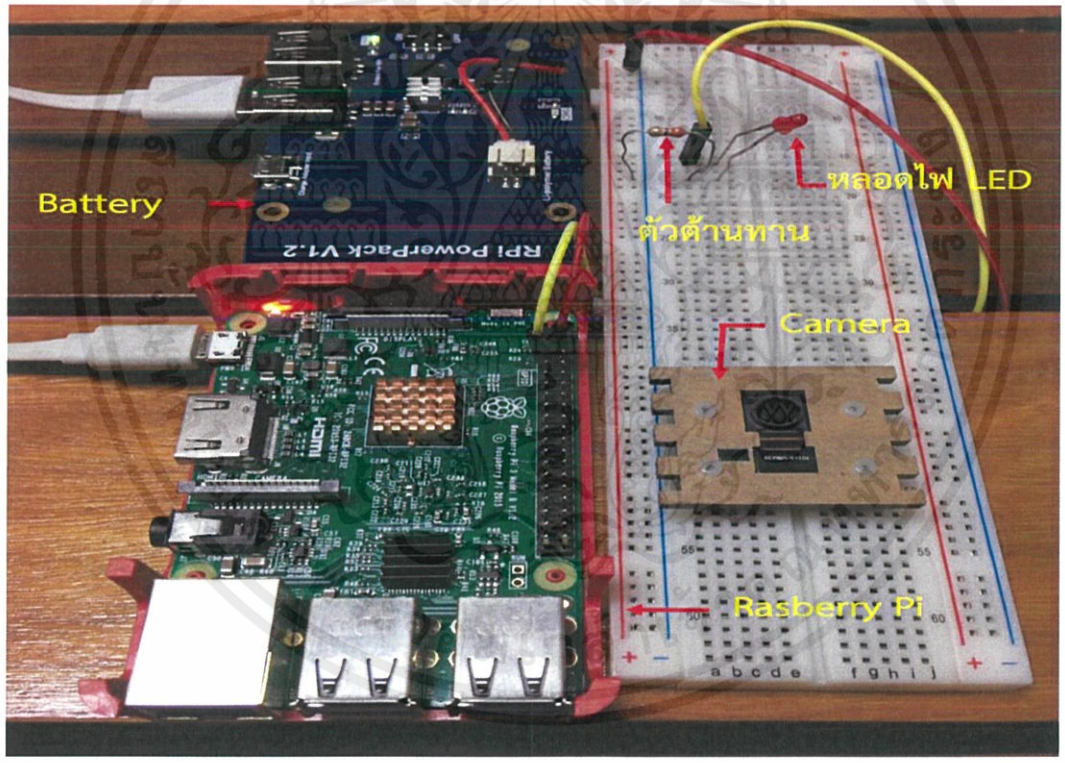
(4) ทดสอบการบันทึกวิดีโอโดยนำกล้องไปติดตั้งที่ตำแหน่งเดียวกับที่จะนำไปทดสอบ พร้อมทั้งบันทึกวิดีโอตามสภาพแวดล้อมต่าง ๆ เพื่อนำมาปรับตั้งค่ากล้อง เพื่อให้มีประสิทธิภาพการตรวจจับสูงสุด

(5) เขียนโค้ดให้ได้ตรงตามที่ได้ออกแบบ แล้วทำการทดสอบกับวิดีโอที่ได้บันทึกไว้ในข้อที่ผ่านมา ปรับแต่งโค้ดให้มีประสิทธิภาพตามสภาพแวดล้อมที่แตกต่างกัน

(6) ทดสอบโค้ดในสถานที่จริง นำอุปกรณ์ไปติดตั้งและเริ่มการทำงาน หากสามารถตรวจจับและแจ้งเตือนได้ตามที่เราต้องการ เป็นอันเสร็จสิ้นการเขียนโปรแกรม

4.2 ขั้นตอนการต่อวงจรหลอดไฟ LED

(1.) ทำการเตรียมอุปกรณ์ให้พร้อม



รูปที่ 4.2 ภาพแสดงชื่ออุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกขนาด ตัวต้านทาน จากสมการ

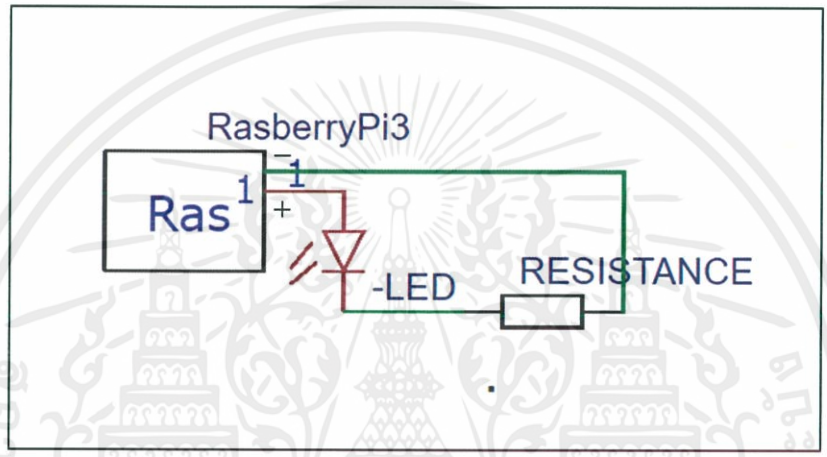
$$V = I \times R \tag{4.5}$$

จะได้

$$R = 3.3 \div 0.01$$

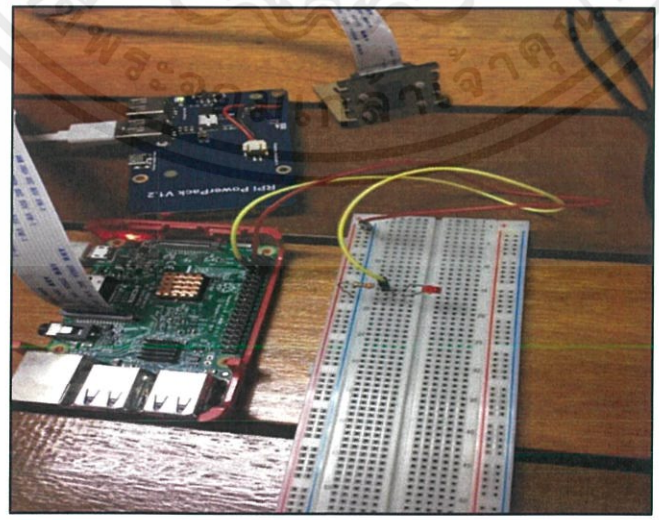
$$R = 330\Omega$$

(2.) ทำการต่อวงจรดังรูปที่ 4.3



รูปที่ 4.3 ภาพการต่อวงจรของหลอดไฟกับราสเบอร์รีพาย

โดยขั้วบวกต่อเข้ากับ Pin No.7 ของ RasberryPI ซึ่งเป็น port จ่ายไฟ และ ขั้วลบต่อเข้ากับ Pin No. 6 ของ RasberryPI ซึ่งเป็น Ground



รูปที่ 4.4 ภาพการต่อวงจร LED เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ขั้นตอนการทดสอบและอธิบายโค้ด (Coding) Background subtraction

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import numpy as np
import RPi.GPIO as GPIO

# ******************Camera Setup ************************

camera = PiCamera()
camera.resolution = (768, 576)
camera.framerate = 30
camera.vflip = True
camera.hflip = True
rawCapture = PiRGBArray(camera, size=camera.resolution)
time.sleep(0.9)
```

รูปที่ 4.5 อธิบายโค้ด Background subtraction ชุดที่1

ในส่วนแรกของโค้ด Background subtraction ตัวที่1 ดังรูปที่ 4.5 ในบรรทัดที่1 ถึงบรรทัดที่6 เป็นการนำเข้า Library ที่จำเป็นสำหรับตัวโปรแกรมของเรา ซึ่งส่วนหลักที่สำคัญของโปรแกรม คือ

1. Import PiCamera คือ Library ที่ใช้ควบคุมและปรับการตั้งค่าของกล้อง Pi
2. Import cv2 คือชุดคำสั่ง OpenCV ที่เกี่ยวกับ Images Processing ซึ่งเป็นหัวใจสำคัญของโปรแกรม
3. Import RPi.GPIO as GPIO คือชุดคำสั่งที่ควบคุมช่องสัญญาณ GPIO และในชุดคำสั่งนี้เราจะนำมาใช้ควบคุมคำสั่งการเปิดปิดการทำงานของหลอด LED

ในส่วนต่อไปคือ Camera Setup เป็นส่วนที่ใช้ปรับการตั้งค่าของกล้อง ให้ทำงานในสภาพแวดล้อมที่เรากำหนด โดยคำสั่งสำคัญแต่ละคำสั่งมีดังนี้

1. camera.resolution คือการกำหนดอัตราส่วนกว้างคูณยาวของเฟรม โดยเรากำหนดขนาดเป็น กว้าง 768 พิกเซล และยาว 576 พิกเซล
2. camera.framerate คือการกำหนดอัตราความเร็วของเฟรม (framerate) กำหนดไว้ที่ 30 เฟรม/วินาที
3. camera.vflip คือ คำสั่งหมุนกล้องในแนวตั้ง (Vertical)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. camera.hflip คือ คำสั่งหมุนกล้องในแนวนอน (Horizontal)

```
# *************************** Define Area ********************************

ix,iy = -1,-1
fx,fy = -1,-1
drawing = False
setup_complete = False
prompt = ''

def prompt_on_image(txt):
    global image
    cv2.putText(image, txt, (10, 35),
    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

def draw_rectangle(event,x,y,flags,param):
    global ix,iy,fx,fy,drawing,setup_complete,image, org_image, prompt

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        ix,iy = x,y

    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == True:
            image = org_image.copy()
            prompt_on_image(prompt)
            cv2.rectangle(image,(ix,iy),(x,y),(0,255,0),2)

    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False
        fx,fy = x,y
        image = org_image.copy()
        prompt_on_image(prompt)
        cv2.rectangle(image,(ix,iy),(fx,fy),(0,255,0),2)
```

รูปที่ 4.6 อธิบายโค้ด Background subtraction ชุดที่ 2

ในส่วนของ Define Area หรือเราจะเรียกว่า ROI (Region on interest) จะรวมชุดคำสั่งที่ใช้วาดรูปเพื่อกำหนดโซนพื้นที่ที่เราต้องการให้ตรวจจับ โดยใช้ฟังก์ชัน cv2.rectangle เพื่อกำหนดกรอบสี่เหลี่ยมบริเวณที่เราสนใจ และกำหนดตัวแปรพื้นที่ (ix, iy) และ (fx, fy) โค้ดสำคัญแต่ละส่วนมีดังนี้

จากรูปที่ 4.6 บรรทัดที่ 1 ถึงบรรทัดที่ 5 เป็นการกำหนดค่าพื้นฐานให้กับตัวแปรที่เรากำหนด โดยจะมีการเรียกใช้งานในขั้นตอนต่อไป

ฟังก์ชั่น prompt_on_image(txt) เป็นการกำหนดฟังก์ชั่นการแสดงข้อความขึ้นที่ด้านบนของหน้าจอ โดยให้แสดงผลที่ตำแหน่ง (10, 35) พิกเซล ใช้ฟ้อนต์ Hershey และใช้สี RGB(255,0,0)

ฟังก์ชั่น draw_rectangle ชุดคำสั่งการวาดรูปสี่เหลี่ยมโดยการคลิกซ้ายเพื่อเริ่มวาด โดยพิกัดจุดแรกเมื่อคลิก จะถูกบันทึกไว้ในตัวแปร ix และ iy และเมื่อลากได้ขนาดสี่เหลี่ยมที่ต้องการ พิกัดสุดท้ายจะบันทึกไว้ในตัวแปร fx และ fy และเราจะนำค่า ix,iy,fx,fy ไปใช้ในขั้นตอน Processing ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#1
camera.capture(rawCapture, format="bgr", use_video_port=True)
image = rawCapture.array
org_image = image.copy()
rawCapture.truncate(0)

#2
cv2.namedWindow("Camera")
cv2.moveWindow("Camera", 10, 40)

#3
cv2.setMouseCallback('Camera',draw_rectangle)

#4
prompt = "Draw detection area - press c to continue"
prompt_on_image(prompt)

#5
while not setup_complete:
    cv2.imshow("Camera",image)
    key = cv2.waitKey(1) & 0xFF

    #6
    if key == ord("c"):
        break
cv2.destroyAllWindows()
```

รูปที่ 4.7 อธิบายโค้ด Background subtraction ชุดที่3

ในโค้ด Background subtraction ชุดที่ 3 นี้ (ดังรูปที่ 4.7) เป็นส่วนที่เกี่ยวกับการบันทึกรูปจากกล้องมา 1 รูป แล้วทำการเปิดรูปขึ้นมาเพื่อให้เราได้ทำการวาดสี่เหลี่ยมบนตำแหน่งที่เราสนใจ โดยอธิบายแต่ละส่วนได้ดังนี้

ส่วนที่ #1 เป็นการเรียกฟังก์ชั่นการจับภาพนิ่งขึ้นมาจากกล้อง แล้วทำการบันทึกรูปไว้ในตัวแปร image

ส่วนที่ #2 จะเป็นคำสั่งเปิดหน้าต่างขึ้นมา ตั้งชื่อว่า Camera

ส่วนที่ #3 เป็นการเรียกใช้คำสั่ง draw_rectangle ซึ่งเป็นคำสั่งวาดรูปสี่เหลี่ยมที่เรากำหนดไว้

ส่วนที่ #4 คือคำสั่งแสดงตัวอักษรขึ้นบนหน้าจอ โดยเรากำหนดให้แสดงคำว่า Draw detection area - press c to continue

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ #5 และ #6 เป็นการดึงตัวแปร image ซึ่งคือรูปที่เราถ่ายไว้ในส่วนที่ #1 ให้ไปแสดงผลที่หน้าต่าง camera และตรวจจับว่ามีการกดแป้นอักษร c จากคีย์บอร์ดหรือไม่ ถ้ามี ให้หยุดการทำงานของหน้าต่าง และปิดหน้าต่างลงไป

รูปที่ 4.8 ภาพแสดงการวาดกรอบกำหนดพื้นที่ ROI

```
# ************************************ Processing ********************************************************

bgsMOG = cv2.createBackgroundSubtractorMOG2()
for fff in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):

            frame = fff.array
            fgmask = bgsMOG.apply(frame, None, 0.01)
            cv2.rectangle(frame,(ix,iy),(fx,fy),(0,255,0),2)

            #1
            _,contours, hierarchy = cv2.findContours(fgmask, cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

            try:
                hierarchy = hierarchy[0]

            except:
                hierarchy = []

            for contour, hier in zip(contours, hierarchy):
                (x, y, w, h) = cv2.boundingRect(contour)

                if w > 70 and h > 70:
                    cv2.rectangle(frame, (x,y), (x+w,y+h), (255, 0, 0), 1)

                    #2
                    x1 = w/2
                    y1 = h/2

                    cx = x+x1
                    cy = y+y1

                    centroid = (cx,cy)
```

รูปที่ 4.9 อธิบายโค้ด Background subtraction ชุดที่4

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Processing หัวข้อนี้คือส่วนที่สำคัญของโปรแกรมดังรูปที่ 4.9 เป็นส่วนที่เรียกโมดูลกล้องขึ้นมาใช้งาน แล้วเริ่มทำการประมวลผลภาพ โดยใช้ฟังก์ชัน Background Subtraction ตรวจสอบพื้นที่ที่มีการเคลื่อนไหวภายในเฟรม โดยเมื่อตรวจสอบพบวัตถุภายในพื้นที่มีการเคลื่อนไหวระบบจะทำการตีกรอบสี่เหลี่ยมขึ้นมาล้อมรอบและคำนวณหาจุดกึ่งกลาง(Centroid)ของวัตถุนั้น โดยส่วนสำคัญแต่ละส่วน คือ

ส่วนที่ #1 เป็นการเรียกคำสั่ง cv2.createBackgroundSubtractorMOG2() ขึ้นมาใช้งาน โดยกระบวนการของคำสั่งนี้คือ เมื่อเปิดกล้องขึ้นมา ฟังก์ชั่นนี้จะทำการบันทึกรูปแรกที่แสดงผลในกล้องเก็บไว้ แล้วเมื่อระยะเวลาผ่านไป วัตถุมีการเคลื่อนไหวหรือมีสิ่งแปลกปลอมเข้ามาในเฟรม เราจะเรียกใช้คำสั่ง cv2.findContours เพื่อตรวจหาความแตกต่างระหว่างภาพแรกและภาพปัจจุบัน เมื่อตรวจพบ จะทำการแจ้งเตือนโดยตีกรอบสี่เหลี่ยมขึ้นมาล้อมรอบ

ส่วนที่ #2 คือส่วนของการหาจุดกึ่งกลาง (Centroid) โดยเราจะในสูตรการคำนวณ จากสมการ

$$L_x = \frac{W_f}{2} \tag{4.1}$$

$$L_y = \frac{H_f}{2} \tag{4.2}$$

ดังนั้น พิกัดจุดกึ่งกลางแกน x คือ $c_x = x + L_x$ (4.3)

พิกัดจุดกึ่งกลางแกน y คือ $c_y = y + L_y$ (4.4)

โดยที่ L_x คือ ระยะครึ่งของกรอบ X_1

L_y คือ ระยะครึ่งของกรอบ Y_1

W_f คือ ความกว้างของกรอบ

H_f คือ ความสูงของกรอบ

สุดท้าย เราจะได้พิกัดจุด Centroid เก็บไว้ในตัวแปร (c_x, c_y)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    #3
                    cv2.circle(frame,(int(cx),int(cy)),2,(0,0,255),-1)

                    #4
                    GPIO.setmode(GPIO.BOARD)
                    GPIO.setup(7,GPIO.OUT)

                    #5
                    if (cx >= ix)and(cx <= fx):
                        if (cy > iy) and (cy < fy):
                            cv2.putText(frame, 'Found', (x,y-5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 255), 2)
                            GPIO.output(7,GPIO.HIGH)
                            time.sleep(2)
                            GPIO.output(7,GPIO.LOW)
                            time.sleep(2)
                            GPIO.output(7,GPIO.HIGH)
                            time.sleep(2)
                    GPIO.cleanup()

            #6
            cv2.imshow('Output', frame)
            key = cv2.waitKey(10)
            if key == 27:
                break
            rawCapture.truncate(0)

#7
cv2.destroyAllWindows()
```

รูปที่ 4.10 อธิบายโค้ด Background subtraction ชุดที่5

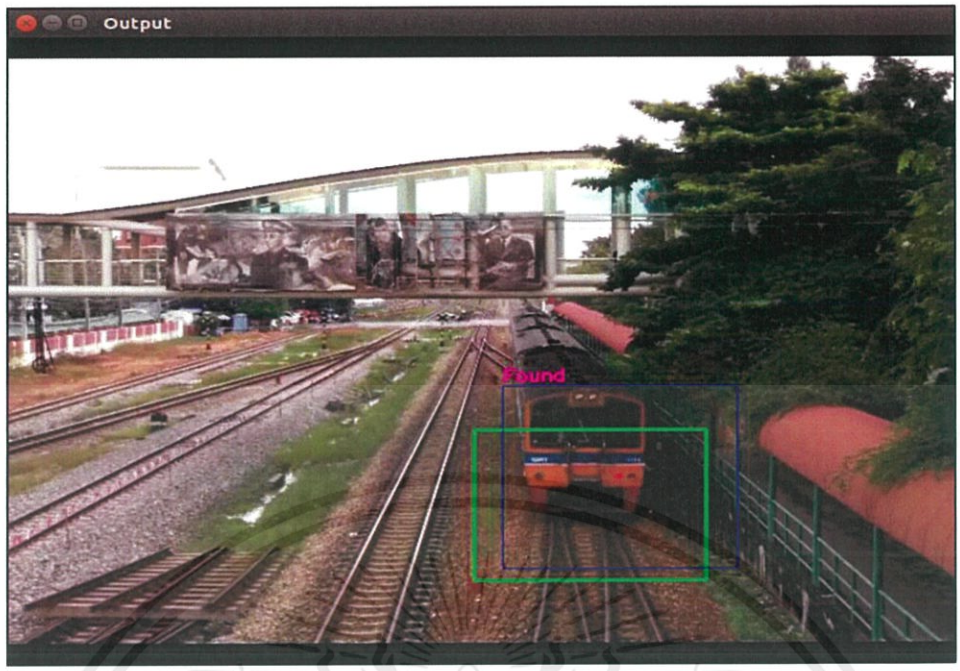
เมื่อเราได้พิกัดบริเวณจุดกึ่งกลาง(Centroid) มาแล้ว ขั้นตอนต่อไปเราจะทำการตรวจสอบพิกัด (c_x, c_y)ของจุดกึ่งกลาง ว่าได้เข้าไปอยู่ใน พื้นที่ ROI ที่เรากำหนดไว้ตั้งแต่ช่วงแรกหรือไม่ดังรูปที่ 4.10 ถ้าหากจุดกึ่งกลาง (centroid) ได้เข้าไปอยู่ภายในพื้นที่ ROI แล้ว ระบบจะทำการแสดงกรอบขึ้นมาเขียนว่า Found แล้วทำการส่งสัญญาณไปที่หลอด LED ของเราให้กระพริบเป็นระยะเวลา 6 วินาที โดยอธิบายโค้ดแต่ละส่วนได้ดังนี้

ส่วนที่ #3 เป็นคำสั่ง cv2.circle วาดรูปจุดวงกลมขึ้นมาที่ตำแหน่งจุดกึ่งกลาง cx,cy

ส่วนที่ #4 เป็นคำสั่งเปิดการใช้งานของ GPIO ช่องที่ 7 ซึ่งได้ต่อกับหลอด LED ที่ติดตั้งไว้

ส่วนที่ #5 เป็นเงื่อนไขการตรวจสอบพิกัดกึ่งกลาง cx,cy ว่าได้เข้าไปอยู่ในพิกัด ROI ix,iy,fx,fy ที่เราได้กำหนดไว้ในช่วงแรกหรือไม่ ถ้าพิกัดกึ่งกลางได้เข้าไปอยู่ในพื้นที่ ROI เราจะทำการเรียกใช้คำสั่ง cv2.putText ให้แสดงผลขึ้นว่า Found ที่กรอบ และส่งสัญญาณไปที่ LED ด้วยคำสั่ง GPIO.output(7,GPIO.HIGH) เพื่อให้หลอดไฟติดและค้างไว้ 2 วินาทีด้วยคำสั่ง time.sleep(2) และให้หลอดไฟดับอีก 2 วินาที ด้วยคำสั่ง GPIO.output(7,GPIO.LOW), time.sleep(2) และทวนซ้ำคำสั่งแรกอีก 1 รอบ ผลที่ออกมาเราจะเห็นได้ว่า หลอดไฟจะกระพริบเป็นระยะเวลา 6 วินาที เป็นอันเสร็จสิ้นกระบวนการเขียนโค้ดของโปรแกรม ผลลัพธ์ของโค้ด Background subtraction ชุดที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ภาพแสดงการตรวจพบรถไฟ ในบริเวณพื้นที่ ROI

4.4 ขั้นตอนการทดสอบและอธิบายโค้ด (Coding) Haar Cascade

```
#1
import cv2

#2
cap = cv2.VideoCapture("last.mp4")
faceCascade = cv2.CascadeClassifier("cascade.xml")

while(True):
        #3
        ret, frame = cap.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, scaleFactor=2.5, minNeighbors=12, minSize=(80, 80))
        #4
        print("Found {0} ".format(len(faces)))

        #5
        for (x, y, w, h) in faces:
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
                x1 = w/2
                y1 = h/2

                cx = x+x1
                cy = y+y1
                centroid = (cx,cy)
                cv2.circle(frame,(int(cx),int(cy)),2,(0,0,255),-1)

        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
                break

cap.release()
cv2.destroyAllWindows()
```

รูปที่ 4.12 อธิบายโค้ด Haar Cascade ชุดที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนการทำงานของ Haar Cascade นั้น เราจะนำไฟล์ที่ทำไว้เฉพาะตรวจจับรถไฟ นำมารวมกับตัวโค้ด แล้วนำมาประมวลผล โดยโค้ดหลักๆจะเหมือนกับของ Background subtraction แต่จะมีส่วนที่แตกต่าง คือ ส่วนที่ 2 และ 3 โดยที่

ส่วนที่ 2 เป็นการเรียกวิดีโอที่เรานำมาใช้ทดสอบขึ้นมา ชื่อไฟล์ว่า last.mp4 ดังรูปที่ 4.12 และเรียกใช้งานไฟล์ cascade ที่เรารวมรูปรถไฟเอาไว้เป็นไฟล์สำเร็จรูป ชื่อไฟล์ว่า cascade.xml

ส่วนที่ 3 เป็นส่วนการเรียกใช้ฟังก์ชั่นแปลงวิดีโอที่เราเปิดมาเป็นขาวดำ และทำการตรวจจับภาพ Positive

ซึ่งโค้ดที่ใช้ในส่วนอื่นๆของวิธีนี้ ตัวฟังก์ชันต่างๆที่ใช้จะเป็นคำสั่งที่คล้ายกันกับฟังก์ชันของ วิธี Background subtraction ซึ่งเมื่อทำการทดสอบตรวจจับได้แล้วจะแสดงผล ดังรูปที่ 4.13

รูปที่ 4.13 ภาพแสดงการตรวจจับด้วย วิธี Haar Cascade

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการดำเนินงานและข้อเสนอแนะ

5.1 ผลการดำเนินงาน

ตารางบันทึกผล โดยกระบวนการ Haar Cascade ชุดที่1

เป็นการเก็บข้อมูล โดยวิธี Haar Cascade หาระยะตรวจจับที่ไกลที่สุด ในรอบที่เวลาต่าง ๆ จำนวน 12 รอบต่อวัน บริเวณชานชาลาหัวตะเข้ ขาเข้าสถานีปลายทางกรุงเทพ ฯ โดยทดลองแต่ละครั้งในวันที่อากาศแจ่มใส และได้วัดค่าความเข้มแสงมาเฉลี่ย เพื่อหาความแม่นยำของอุปกรณ์ในสภาพแสงที่แตกต่างกัน

ในการเก็บข้อมูลนั้น ได้ทดลองในตัวแปรกรอบสี่เหลี่ยมการตรวจจับที่ 70x70 pixel, 50x50 pixel และ 30x30 pixel ได้ผลว่าจากการทดสอบในช่วงเวลา 6.00 น. 6.30 น. 7.01 น. 7.35 น. และ 8.05 น. ทำการทดสอบ10ครั้ง ได้ความเข้มแสงเฉลี่ย 9 LUX, 68 LUX, 1974 LUX, 5730 LUX และ 9100 LUX ตามลำดับ ในทุก ๆค่าตัวแปรกรอบสี่เหลี่ยม พบว่าอุปกรณ์ไม่สามารถตรวจจับรถไฟในสภาพแสงช่วงนี้ได้มากกว่าระยะ 10 เมตร อุปกรณ์จะเริ่มจับภาพได้ระยะไกลตั้งแต่รอบ 9.11 น. พล็อตค่าลงในตารางการตรวจจับที่ 70x70 pixel ได้ว่า

ตารางบันทึกผลการทดสอบที่เวลา 9.11 น. ความเข้มแสงเฉลี่ย 61,933 LUX ที่ 70x70 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ
		1	2	3	4	5	6	7	8	9	10	(ที่ 61933 LUX)
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	20%
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 9.39 น. ความเข้มแสงเฉลี่ย 75,425 LUX ที่ 70x70 pixel

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 75425 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
30	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
20	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	60%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 11.16 น. ความเข้มแสงเฉลี่ย 108,000 LUX ที่70x70 pixel

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 108000 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
40	✗	✓	✓	✗	✓	✓	✓	✓	✗	✗	60%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 12.30 น. ความเข้มแสงเฉลี่ย 121,500 LUX ที่70x70 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 121500 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	40%
	50	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	80%
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 16.09 น. ความเข้มแสงเฉลี่ย 52,825 LUX ที่70x70 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 52825 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	20%
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

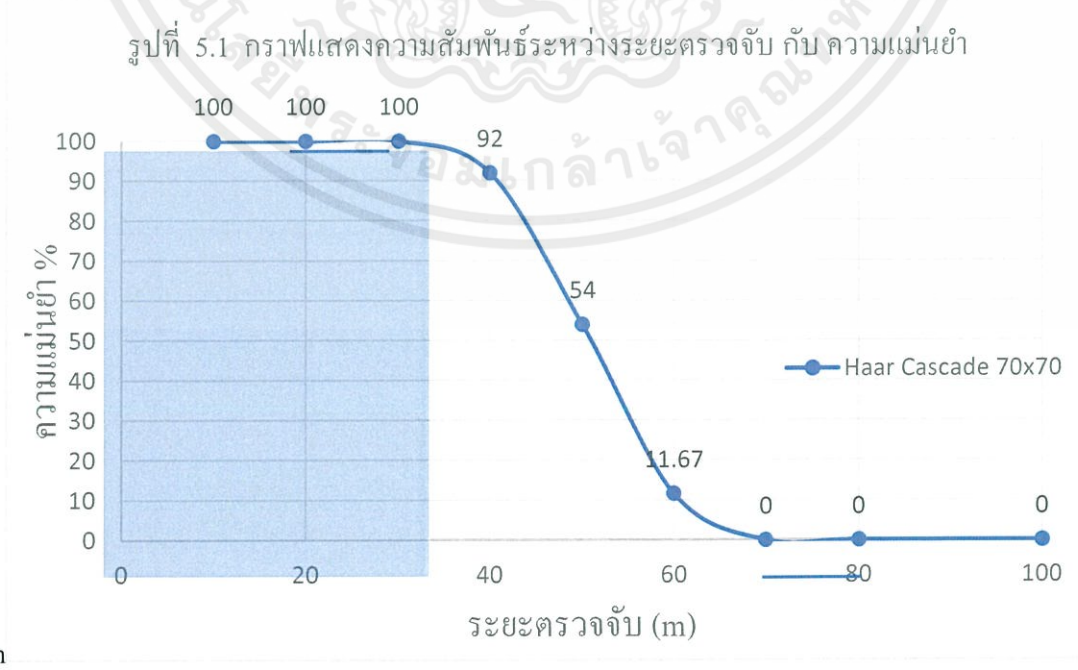
ตารางบันทึกผลการทดสอบที่เวลา 17.06 น. ความเข้มแสงเฉลี่ย 21,166 LUX ที่70x70 pixel

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 21166 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	10%
50	✓	✗	✓	✓	✗	✓	✗	✓	✓	✓	70%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

และผลการทดสอบรอบสุดท้าย ในช่วงเวลา 18.42 น. ความเข้มแสงเฉลี่ย 27 LUX พบว่าไม่สามารถตรวจจับขบวนรถไฟได้

จากข้อมูลข้างต้น เมื่อนำระยะตรวจจับ กับ ความแม่นยำมาเฉลี่ย แล้วพล็อตกราฟแสดงความสัมพันธ์ (ดังรูปที่5.1) จะพบว่าความแม่นยำของอุปกรณ์นั้นจะเริ่มมีค่าลดลง เมื่อตรวจจับที่ระยะเกิน 30m ดังนั้น เพื่อให้ได้ความแม่นยำที่ 99% จึงกำหนดให้อุปกรณ์ที่ 70x70 px มีระยะตรวจจับที่ 33

รูปที่ 5.1 กราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับ กับ ความแม่นยำ



m

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวแปรกรอบสี่เหลี่ยมการตรวจจับที่ 50x50 pixel ได้ผลว่าจากการทดสอบในช่วงเวลา 6.00 น. ถึง 8.05 น. พบว่า อุปกรณ์ไม่สามารถตรวจจับรถไฟในสภาพแสงช่วงนี้ได้ อุปกรณ์จะเริ่มจับภาพได้ตั้งแต่รอบ 9.11 น. เช่นเดียวกับ 70x70 pixel พล็อตค่าลงในตารางได้ว่า

ตารางบันทึกผลการทดสอบที่เวลา 9.11 น. ความเข้มแสงเฉลี่ย 61,933 LUX ที่ 50x50 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 61933 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	20%
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 9.39 น. ความเข้มแสงเฉลี่ย 75,425 LUX ที่ 50x50 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 75425 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	30	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	20	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	80%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 11.16 น. ความเข้มแสงเฉลี่ย 108,000 LUX ที่50x50 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ
		1	2	3	4	5	6	7	8	9	10	(ที่ 108000 LUX)
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 12.30 น. ความเข้มแสงเฉลี่ย 121,500 LUX ที่50x50 pixel

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ
		1	2	3	4	5	6	7	8	9	10	(ที่ 121500 LUX)
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	80%
	60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 16.09 น. ความเข้มแสงเฉลี่ย 52,825 LUX ที่50x50 pixel

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 52825 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	40%
50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 17.06 น. ความเข้มแสงเฉลี่ย 21,166 LUX ที่50x50 pixel

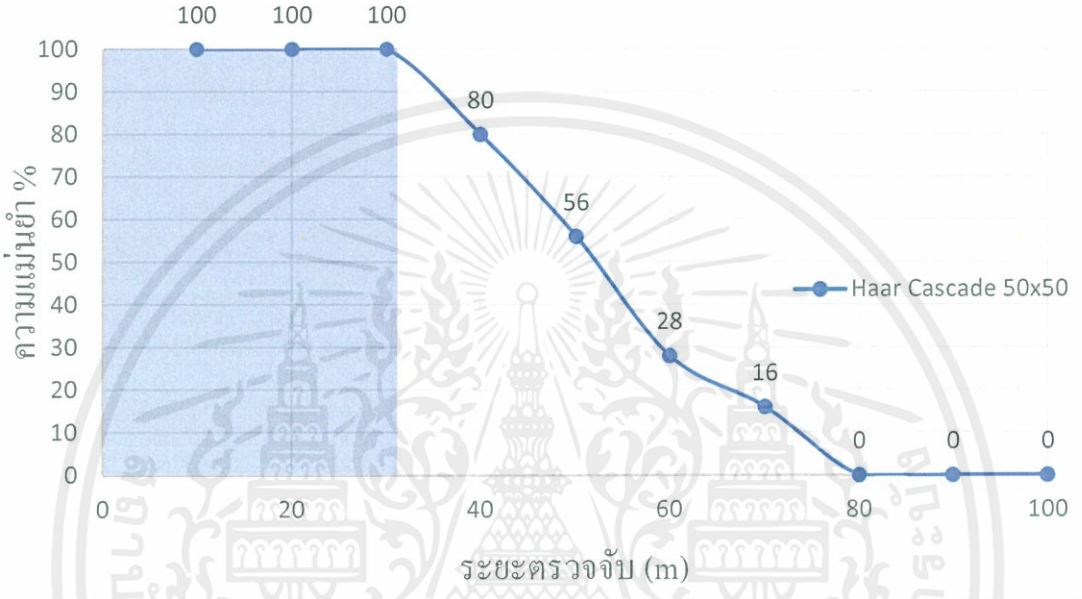
ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 21166 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓	60%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

และผลการทดสอบรอบสุดท้าย ในช่วงเวลา 18.42 น. ความเข้มแสงเฉลี่ย 15 LUX พบว่าไม่สามารถตรวจจับขบวนรถไฟได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อมูลข้างต้น เมื่อนำระยะตรวจจับ กับ ความแม่นยำมาเฉลี่ย แล้วพล็อตกราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับกับความแม่นยำ (ดังรูปที่ 5.1) เหมือนกับในกรณี 70x70 px จะพบว่าความแม่นยำของอุปกรณ์นั้นจะเริ่มมีค่าลดลง เมื่อตรวจจับที่ระยะเกิน 30m ดังนั้น เพื่อให้ได้ความแม่นยำที่ 99% จากตารางจึงกำหนดให้อุปกรณ์ที่ 50x50 px มีระยะตรวจจับที่ 31 m

รูปที่ 5.2 กราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับ กับ ความแม่นยำ



สำหรับตัวแปรกรอบสี่เหลี่ยมการตรวจจับที่ 30x30 pixel เมื่อทำการทดสอบแล้วพบว่า อุปกรณ์มีอาการกระตุกและใช้เวลาในการประมวลผลที่นาน อีกทั้งผลที่ได้ยังไม่ต่างจาก 50x50 px อย่างมีนัยสำคัญ

ดังนั้นจากข้อมูลทั้งหมดข้างต้น พบว่า การเปลี่ยนค่าตัวแปรกรอบสี่เหลี่ยมการตรวจจับที่ 70x70 pixel, 50x50 pixel และ 30x30 pixel มีผลต่อระยะการตรวจจับของอุปกรณ์น้อยมาก ถึงแม้ว่าที่ 50x50 pixel จะมีระยะตรวจจับสูงสุดมากกว่าถึง 10 เมตร แต่ก็เป็นช่วงที่ไม่มีความเสถียร ในทางกลับกันในจุดที่มีความเสถียรสูง ๆ เช่นช่วงเกิน 90% 70x70 pixel จะทำได้ดีกว่า และเป็นการตั้งค่าที่ทำให้ได้ frame rate ของอุปกรณ์ลื่นที่สุด ทำให้อุปกรณ์ไม่ต้องทำงานหนักมาก ทางกลุ่มจึงได้เลือกใช้โปรแกรมที่ 70x70 pixel ในการตั้งเป็นค่าเริ่มต้นของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผล โดยกระบวนการ Background Subtraction ชุดที่1

เป็นการเก็บข้อมูล โดยวิธี Background Subtraction หาระยะตรวจจับที่ไกลที่สุด ในรอบที่เวลาต่าง ๆ จำนวน 12 รอบต่อวัน บริเวณชานชาลาหัวตะเข้ ขาเข้าสถานีปลายทางกรุงเทพ ฯ สภาพแวดล้อมเช่นเดียวกับวิธี Haar Cascade

ตารางบันทึกผลการทดสอบที่เวลา 6.00 น. ความเข้มแสงเฉลี่ย 9 LUX

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 9 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	30	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	60%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 6.30 น. ความเข้มแสงเฉลี่ย 68 LUX

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 4160 LUX)
		1	2	3	4	5	6	7	8	9	10	
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	30	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	90%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 9.11 น. ความเข้มแสงเฉลี่ย 61,933 LUX

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 61933 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
30	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	40%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 9.39 น. ความเข้มแสงเฉลี่ย 75,425 LUX

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 75425 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 11.16 น. ความเข้มแสงเฉลี่ย 108,000 LUX

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	(ที่ 108000 LUX)
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

ตารางบันทึกผลการทดสอบที่เวลา 12.30 น. ความเข้มแสงเฉลี่ย 121,500 LUX

		ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	(ที่ 121500 LUX)
ระยะตรวจจับ (m)	100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	50	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
	40	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗	60%
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดสอบที่เวลา 16.09 น. ความเข้มแสงเฉลี่ย 52,825 LUX

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 52825 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	40%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

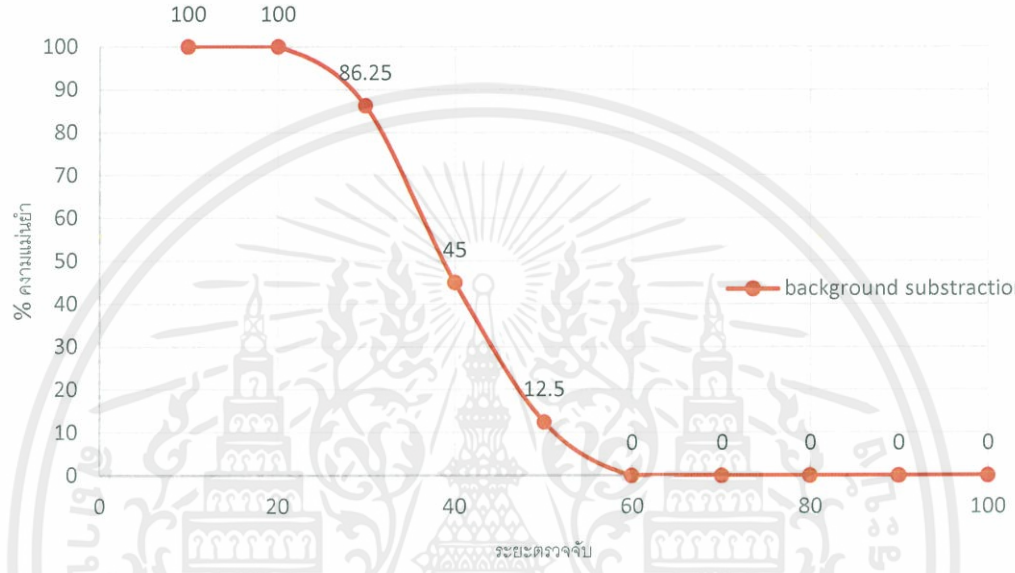
ตารางบันทึกผลการทดสอบที่เวลา 17.06 น. ความเข้มแสงเฉลี่ย 21,166 LUX

ระยะตรวจจับ (m)	ครั้งที่ทดลอง										เปอร์เซ็นความแม่นยำ (ที่ 21166 LUX)
	1	2	3	4	5	6	7	8	9	10	
100	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
80	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
70	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
60	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0%
50	✗	✗	✓	✓	✓	✗	✓	✗	✓	✓	60%
40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

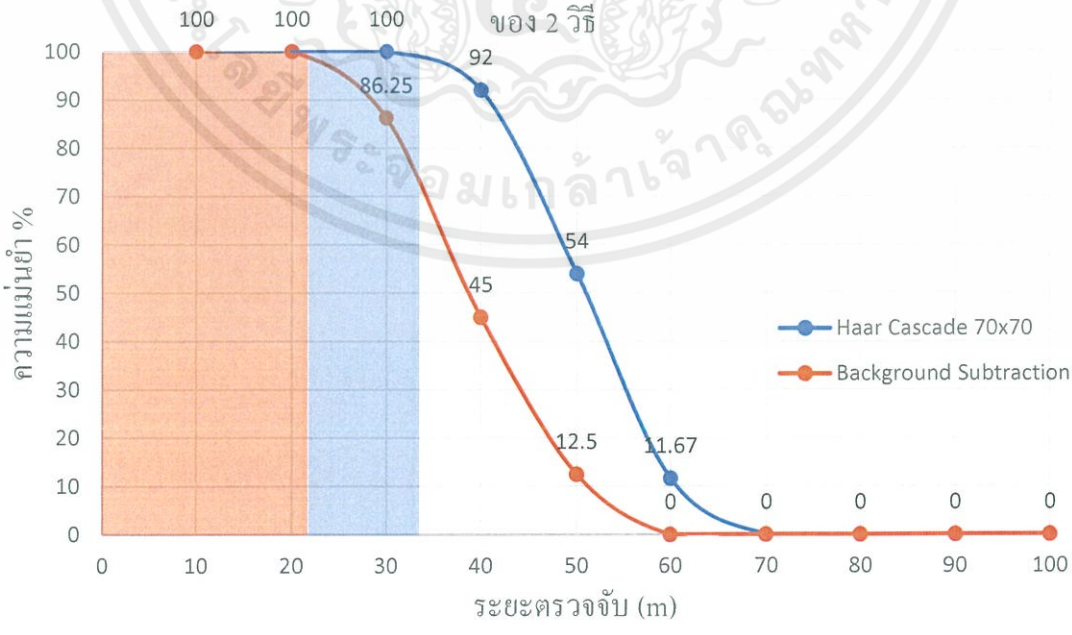
จากข้อมูลข้างต้น เมื่อนำระยะตรวจจับ กับ ความแม่นยำมาเฉลี่ย แล้วพล็อตกราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับกับความแม่นยำ (ดังรูปที่ 5.3) จะพบว่าความแม่นยำของอุปกรณ์โดยวิธี Background Subtraction นั้นมีค่าลดลงอย่างมาก เมื่อตรวจจับที่ระยะเกิน 20 เมตร ดังนั้น เพื่อให้ได้ความแม่นยำที่ 99% จึงกำหนดให้อุปกรณ์ มีระยะตรวจจับที่ 22 เมตร

รูปที่ 5.3 กราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับกับความแม่นยำ



เมื่อนำกราฟของทั้ง2วิธีข้างต้นมาเปรียบเทียบกัน จะพบความสัมพันธ์ กันดังรูปที่ 5.4

รูปที่ 5.4 กราฟแสดงความสัมพันธ์ระหว่างระยะตรวจจับ กับ ความแม่นยำ ของ 2 วิธี



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางบันทึกผล และ กราฟความสัมพันธ์ทั้ง 2 วิธีนั้น ดังรูปที่ 5.4 จะเห็นว่า การตรวจจับด้วยวิธี Haar Cascade ในช่วงของความแม่นยำสูงที่ 99% จะมีระยะตรวจจับได้ไกลกว่าประมาณ 10 เมตร แต่วิธี Background Subtraction สามารถตรวจจับในช่วงเวลาที่แสงน้อย ๆของวันได้ (9-9000 LUX)

ตารางบันทึกผล โดยกระบวนการ Haar Cascade ชุดที่2

เป็นการเก็บข้อมูลความผิดพลาดโดยวิธี Haar Cascade จากการตรวจจับวัตถุแปลกปลอมขนาด 1.0 x 1.0 , 0.5 x 0.5 , 1.7 x 0.5 และ 2.0 x 2.0 ในแต่ละระยะ แบ่งเป็นช่วงความเข้มแสงต่ำกว่า 2000 LUX และ ช่วงความเข้มแสงมากกว่า 10000 LUX เพื่อหาความผิดพลาดของอุปกรณ์ในสภาพแสงที่แตกต่างกัน

จากการทดสอบทั้งช่วง 2 ความเข้มแสง พบว่า อุปกรณ์ไม่สามารถตรวจจับสิ่งแปลกปลอมใด ๆ

ตารางบันทึกผลการทดสอบ ในช่วง ความเข้มแสงต่ำกว่า 2000 LUX

		ระยะต่าง ๆ				
		10	20	30	40	50
วัตถุขนาดต่างๆ (m x m)	0.5 x 0.5	✗	✗	✗	✗	✗
	1.0 x 1.0	✗	✗	✗	✗	✗
	1.7 x 0.5	✗	✗	✗	✗	✗
	2.0 x 2.0	✗	✗	✗	✗	✗

ตารางบันทึกผลการทดสอบ ในช่วง ความเข้มแสงมากกว่า 10000 LUX

		ระยะต่าง ๆ				
		10	20	30	40	50
วัตถุขนาดต่างๆ (m x m)	0.5 x 0.5	✗	✗	✗	✗	✗
	1.0 x 1.0	✗	✗	✗	✗	✗
	1.7 x 0.5	✗	✗	✗	✗	✗
	2.0 x 2.0	✗	✗	✗	✗	✗

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผล โดยกระบวนการ Background Subtraction ชุดที่2

เป็นการเก็บข้อมูลความผิดพลาดโดยวิธี Background Subtraction จากการตรวจจับวัตถุแปลกปลอมขนาด 1.0 x 1.0 , 0.5 x 0.5 , 1.7 x 0.5 และ 2.0 x 2.0 ในแต่ละระยะ แบ่งเป็นช่วงความเข้มแสงต่ำกว่า 2000 LUX และ ช่วงความเข้มแสงมากกว่า 10000 LUX เพื่อหาความผิดพลาดของอุปกรณ์ในสภาพแสงที่แตกต่างกัน

ตารางบันทึกผลการทดสอบ ในช่วง ความเข้มแสงต่ำกว่า 2000 LUX

วัตถุขนาดต่างๆ (m x m)	ระยะต่าง ๆ				
	10	20	30	40	50
0.5 x 0.5	✗	✗	✗	✗	✗
1.0 x 1.0	✓	✓	✓	✗	✗
1.7 x 0.5	✓	✓	✗	✗	✗
2.0 x 2.0	✓	✓	✓	✓	✗

ตารางบันทึกผลการทดสอบ ในช่วง ความเข้มแสงมากกว่า 10000 LUX

วัตถุขนาดต่างๆ (m x m)	ระยะต่าง ๆ				
	10	20	30	40	50
0.5 x 0.5	✓	✗	✗	✗	✗
1.0 x 1.0	✓	✓	✓	✓	✗
1.7 x 0.5	✓	✓	✓	✓	✓
2.0 x 2.0	✓	✓	✓	✓	✓

จากข้อมูลทั้งหมดข้างต้น พบว่า Haar Cascade จะดีกว่า Background subtraction อย่างเห็นได้ชัดในเรื่องของการไม่ตรวจจับสิ่งแปลกปลอมที่ผ่านกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 สรุปผลการทดลอง

จากการทำชุดต้นแบบการตรวจจับรถไฟด้วยกระบวนการ image Processing เพื่อตรวจสอบว่ามีขบวนรถไฟวิ่งเข้ามาในบริเวณกล้องหรือไม่แล้วทำการส่งสัญญาณแจ้งเตือนอัตโนมัติเมื่อระบบตรวจพบว่ามีขบวนรถไฟวิ่งเข้าในบริเวณที่สนใจพบว่า

สรุปผลการทดสอบด้วยกระบวนการ Haar Cascade

จากข้อมูลดังกล่าวข้างต้น ชุดทดลองจะสามารถเริ่มตรวจจับขบวนรถไฟได้ เมื่อบริเวณโดยรอบมีค่าความเข้มแสงอยู่ในช่วงประมาณ 20,000 LUX ขึ้นไป และจะไม่สามารถตรวจจับขบวนรถไฟได้ เมื่อบริเวณโดยรอบมีค่าความเข้มแสง อยู่ในช่วงประมาณต่ำกว่า 10,000 LUX โดยจะมีระยะการตรวจจับสูงสุดที่ 60 เมตร และ ระยะการตรวจจับแม่นยำที่ 33 เมตร

ผลการทดสอบจะพบว่าความผิดพลาดของระบบในกรณีที่มีวัตถุอื่นนอกจากขบวนรถไฟเคลื่อนที่ผ่าน เช่น คน, สุนัข หรือ เงาต้นไม้ จะไม่เกิดขึ้นในกระบวนการ Haar Cascade แต่ระบบนี้จะทำการส่งสัญญาณเมื่อตรวจจับบริเวณช่วงลำตัวของขบวน และหน้าต่างรถไฟด้วย ซึ่งมีลักษณะคล้ายกับภาพ Positive ของด้านหน้าขบวนรถไฟ ดังนั้นเมื่อทำการทดลองโดยการเพิ่มภาพ Positive เข้ามาในระบบจำนวน 80 รูป พบว่าค่าเปอร์เซ็นต์การจับภาพด้านข้างขบวนรถมีค่าลดลงเฉลี่ย 14% โดยจำนวนภาพทั้งหมดที่ใช้ในการทดสอบและป้อนเข้าสู่ระบบเพื่อให้ระบบทำการเรียนรู้สุดท้ายเป็นจำนวน 3,190 รูป ซึ่งแบ่งเป็นภาพ Positive 628 รูป และภาพ Negative 2,562 รูป

สรุปผลการทดสอบด้วยกระบวนการ Background subtraction

จากผลการทดสอบชุดทดลองจะสามารถเริ่มตรวจจับขบวนรถไฟได้ทั้งวัน ซึ่งที่ในช่วงเวลากลางวันระบบสามารถตรวจจับขบวนรถไฟได้สูงสุดที่ระยะ 50 เมตร และ ระยะการตรวจจับแม่นยำอยู่ที่ 22 เมตร ในกรณีที่มีวัตถุแปลกปลอมนอกจากขบวนรถไฟเคลื่อนที่ผ่าน เช่น คน, สุนัข หรือ เงาต้นไม้ จะอยู่ที่ขนาดเมื่อเทียบกับพื้นที่เส้นที่ลากตรวจจับ ถ้าวัตถุแปลกปลอมอยู่ใกล้มาก ๆ หรือมีขนาดใหญ่พอ (ขนาดใกล้เคียงขบวนรถไฟ) ก็จะทำให้เกิดการส่งสัญญาณผิดพลาดได้ ส่วนในช่วงเวลากลางคืนระบบสามารถตรวจจับขบวนรถไฟได้จากแสงหน้ารถขบวนรถไฟที่ระยะทางน้อยกว่า 30 เมตร แต่จะเกิดความผิดพลาดได้จากแสงอื่น ๆที่ผ่านในบริเวณรางเช่นกัน ส่งผลให้การตรวจจับด้วยกระบวนการนี้อาจจะมีความผิดพลาดได้ในเวลากลางคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบระหว่าง 2 กระบวนการจะพบว่ากระบวนการ Background subtraction สามารถทำงานในค่าความเข้มแสงต่ำๆได้ดีกว่ากระบวนการ Haar Cascade ก็จริง แต่ Haar Cascade ก็มีข้อดีตรงที่มีระยะตรวจจับที่ไกลกว่าถึง 10 เมตร และค่าเปอร์เซ็นต์ความผิดพลาดของกระบวนการ Haar Cascade จากการตรวจจับสิ่งแปลกปลอมน้อย และยังสามารถลดลงได้อีกด้วยการเพิ่มรูปภาพที่เราต้องการตรวจจับ (Positive) เข้ามาในระบบ เพื่อทำให้กระบวนการ Machine Learning มีประสิทธิภาพสูงขึ้น แต่ก็ต้องแลกมากับการที่อุปกรณ์ต้องมีการประมวลผลที่หนักขึ้น ทำให้อาจจะต้องเพิ่ม spec ของอุปกรณ์ ดังนั้น การนำไปใช้จึงขึ้นอยู่กับความถนัด ความเหมาะสมต่อพื้นที่งาน และ ลักษณะของ Software และ Hardware ของผู้ใช้

5.3 ข้อเสนอแนะ

5.2.1 ในกระบวนการ Haar Cascade การใส่ภาพ positive และภาพ negative ในปริมาณมาก จะทำให้ระบบมีความแม่นยำที่สูงขึ้น แต่จะแลกมาด้วยประสิทธิภาพในการประมวลผลที่จะใช้ CPU และ หน่วยความจำเยอะ ซึ่งอาจทำให้เกิดอาการหน่วงได้

5.2.2 ในกระบวนการ Background subtraction การขีดเส้นไลน์ (Line) ควรขีดไว้ตามเส้นแนวรางโดยมีความกว้างประมาณขนาดของตู้รถไฟ เพื่อให้ระบบสามารถตรวจจับรถได้อย่างแม่นยำ

5.2.3 ในระหว่างติดตั้งอุปกรณ์ ณ สถานที่จริง ควรมีอุปกรณ์ยึดเหนี่ยวกล้องที่ใช้อย่างมั่นคงเพื่อป้องกันความผิดพลาดที่เกิดจากการสั่นโดยขบวนรถเมื่อเคลื่อนที่ผ่านบริเวณที่ติดตั้งกล้อง

5.2.4 หากต้องการตรวจจับขบวนรถไฟในเวลากลางคืนอาจเพิ่มชุดอุปกรณ์ Infrared ในการถ่ายหรือบันทึกภาพ

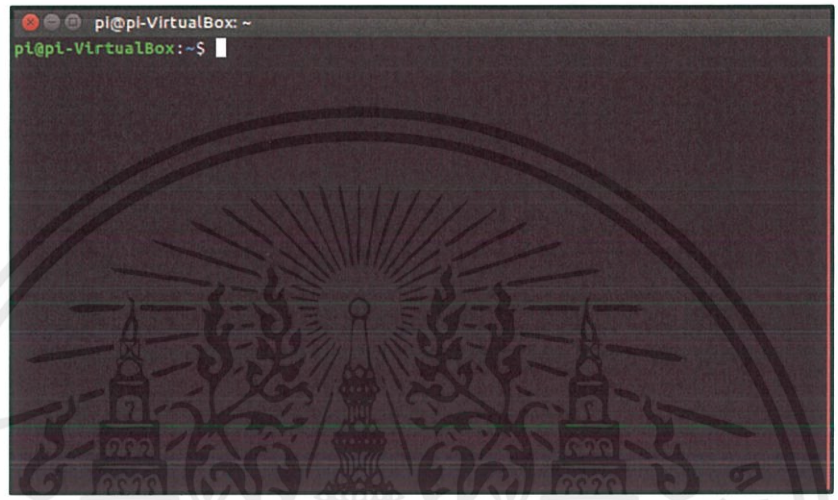
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

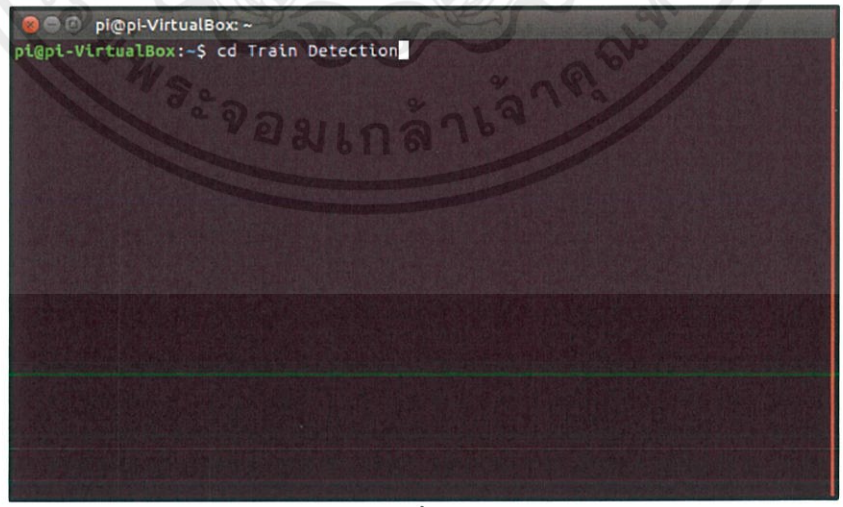
ขั้นตอนการเปิดโค้ด

เสียบสายแพรกล้องและทำการเปิด Raspberry Pi เมื่อสำเร็จ เปิด Terminal ขึ้นมา โดยทำการคลิกขวา เลือก Open Terminal จะขึ้นหน้าต่างดังรูป ก.1



รูป ก.1

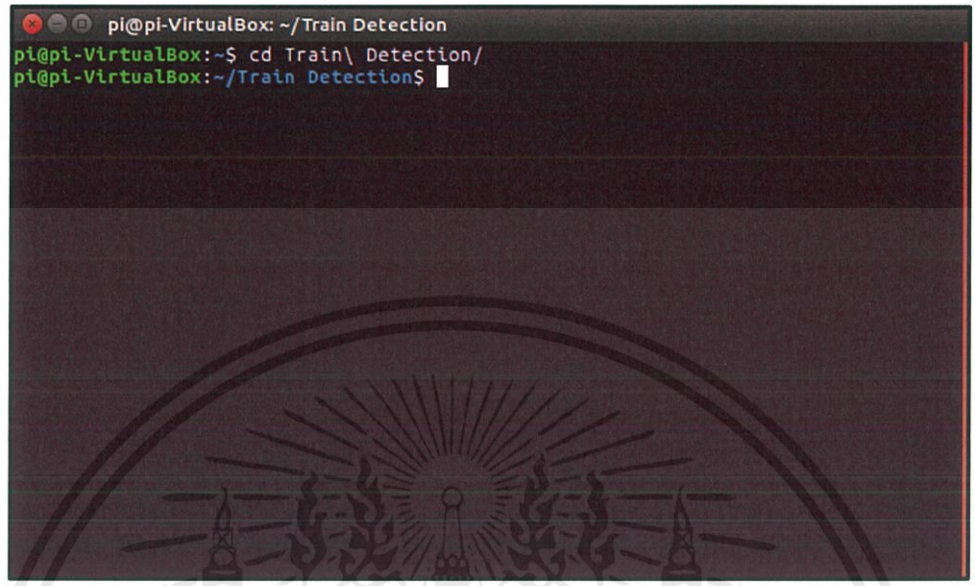
Browse ไฟล์ไปที่ที่อยู่ของโค้ด ตามตัวอย่าง จะอยู่ที่โฟลเดอร์ Train Detection จึงพิมพ์ cd ตามด้วยชื่อโฟลเดอร์ เป็น cd Train Detection แล้วกด Enter ดังรูปที่ ก.2



รูปที่ ก.2

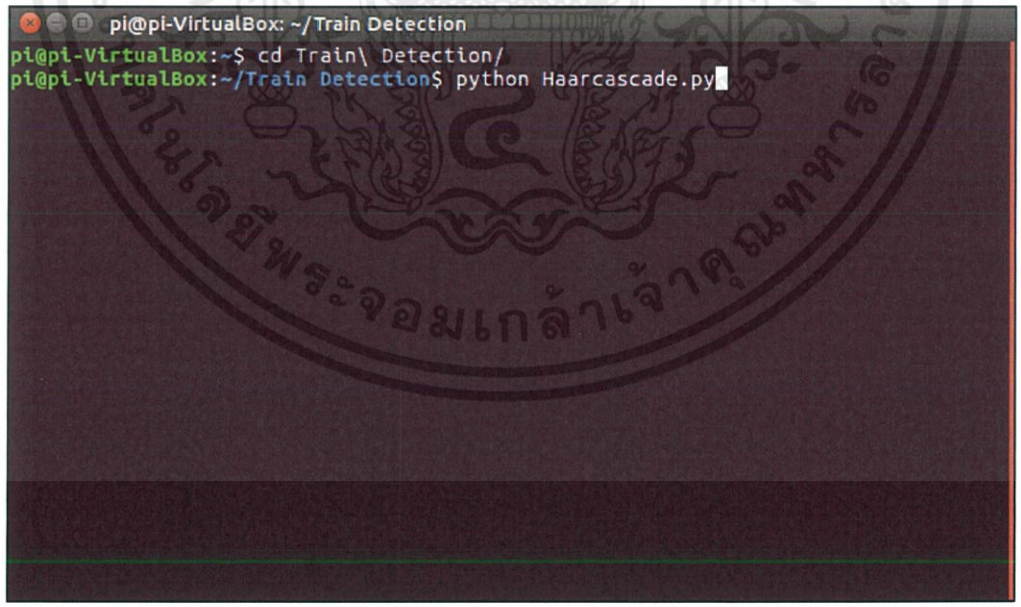
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้ามาถูกต้อง จะขึ้นบรรทัดใหม่ แล้วมีชื่อโฟลเดอร์ Train Detection สีฟ้า ขึ้นมาดังรูปที่ ก.3



รูปที่ ก.3

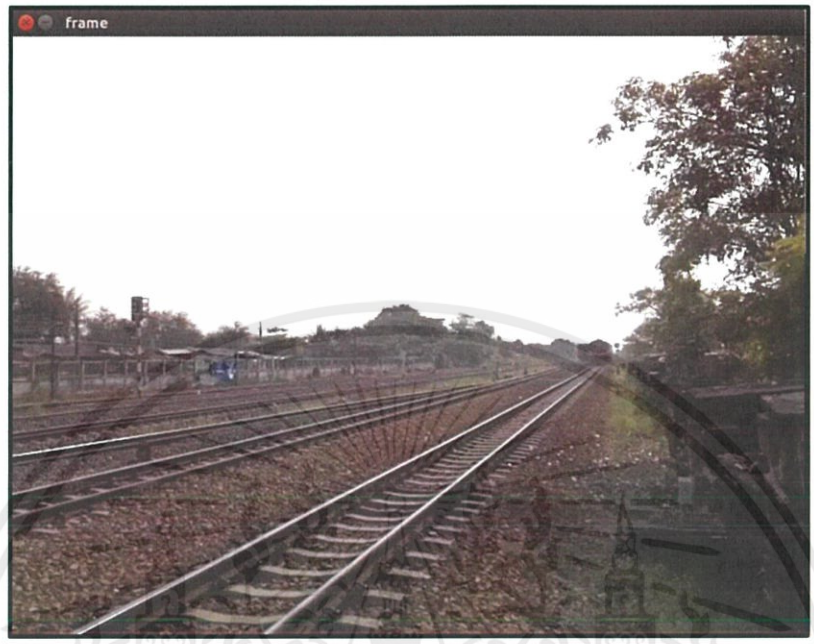
ทำการรันโค้ดโดย พิมพ์ python Haarcascade.py ดังรูปที่ ก.4



รูปที่ ก.4

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกด Enter จะปรากฏภาพจากกล้องขึ้นมา พร้อมใช้งาน ดังรูปที่ ก.5



รูปที่ ก.5

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โค้ดภาษา Python ในการพัฒนาชุดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.) Haarcascade Code

```
import cv2

import time

import datetime

cap = cv2.VideoCapture("Haar.mp4")

TrainCascade = cv2.CascadeClassifier("cascade.xml")

while(True):

        timestamp = datetime.datetime.now()

        ret, frame = cap.read()

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        trains =
TrainCascade.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(70, 70))

        print("Found {0}".format(len(trains)))

        for (x, y, w, h) in trains:

                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

                x1 = w/2

            y1 = h/2

            cx = x+x1

            cy = y+y1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                centroid = (cx,cy)

                    cv2.circle(frame,(int(cx),int(cy)),2,(0,0,255),-1)

                    imageFilename = "car_at_" +
datetime.datetime.now().strftime("%Y%m%d_%H%M%S") + ".jpg"

                    cv2.imwrite(imageFilename,frame)

        cv2.imshow('frame', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):

                    break

cap.release()

cv2.destroyAllWindows()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) Background Subtraction Code

```
from picamera.array import PiRGBArray

from picamera import PiCamera

import cv2

import time

import numpy as np

import RPi.GPIO as GPIO

 ********************Camera Setup **********************

camera = PiCamera()

camera.resolution = (768, 576)

camera.framerate = 30

camera.vflip = True

camera.hflip = True

rawCapture = PiRGBArray(camera, size=camera.resolution)

time.sleep(0.9)

faceCascade = cv2.CascadeClassifier("cascade.xml")

*********************************************************

counter = 0

ix,iy = -1,-1

fx,fy = -1,-1

drawing = False

setup_complete = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tracking = False
prompt = ''
def prompt_on_image(txt):
    global image
    cv2.putText(image, txt, (10, 35),
    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
# mouse callback function for drawing capture area
def draw_rectangle(event,x,y,flags,param):
    global ix,iy,fx,fy,drawing,setup_complete,image, org_image, prompt
     if event == cv2.EVENT_LBUTTONDOWN:
       drawing = True
       ix,iy = x,y
    elif event == cv2.EVENT_MOUSEMOVE:
       if drawing == True:
           image = org_image.copy()
           prompt_on_image(prompt)
           cv2.rectangle(image,(ix,iy),(x,y),(0,255,0),2)
      elif event == cv2.EVENT_LBUTTONUP:
       drawing = False
       fx,fy = x,y
       image = org_image.copy()
       prompt_on_image(prompt)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cv2.rectangle(image,(ix,iy),(fx,fy),(0,255,0),2)
        #print ('ix,iy =') ,ix,iy
        #print ('fx,fy =') ,fx,fy
    cv2.namedWindow("Camera")
    cv2.moveWindow("Camera", 10, 40)
    # call the draw_rectangle routines when the mouse is used
    cv2.setMouseCallback('Camera',draw_rectangle)
*********************************************************************************************
    camera.capture(rawCapture, format="bgr", use_video_port=True)
    image = rawCapture.array
    rawCapture.truncate(0)
    org_image = image.copy()
    prompt = "Draw detection area - press c to continue"
    prompt_on_image(prompt)
    # wait while the user draws the monitored area's boundry
    while not setup_complete:
        cv2.imshow("Camera",image)
        #wait for for c to be pressed
        key = cv2.waitKey(1) & 0xFF
        # if the `c` key is pressed, break from the loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if key == ord("c"):
        break
cv2.destroyAllWindows()
for fff in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
    frame = fff.array
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=2.5,
            minNeighbors=5,
            minSize=(20, 20)
    )
    print("Found {0} ".format(len(faces)))
    for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
            x1 = w/2
            y1 = h/2
            cx = x+x1
            cy = y+y1
            centroid = (cx,cy)
            cv2.circle(frame,(int(cx),int(cy)),2,(0,0,255),-1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                GPIO.setmode(GPIO.BOARD)
                GPIO.setup(7,GPIO.OUT)
                if (cx >= ix)and(cx <= fx):
                        if (cy > iy) and (cy < fy):
                                cv2.putText(frame, 'Found', (10, 35),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
                                GPIO.output(7,GPIO.HIGH)
                                time.sleep(2)
                                GPIO.output(7,GPIO.LOW)
                                time.sleep(2)
                                GPIO.output(7,GPIO.HIGH)
                                time.sleep(2)
                        GPIO.cleanup()
        cv2.imshow('Output', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        rawCapture.truncate(0)
cv2.destroyAllWindows()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

การติดตั้งซอฟท์แวร์สำหรับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งระบบ Raspbian OS

Win32DiskImager	https://sourceforge.net/projects/win32diskimager/
RPI.img	

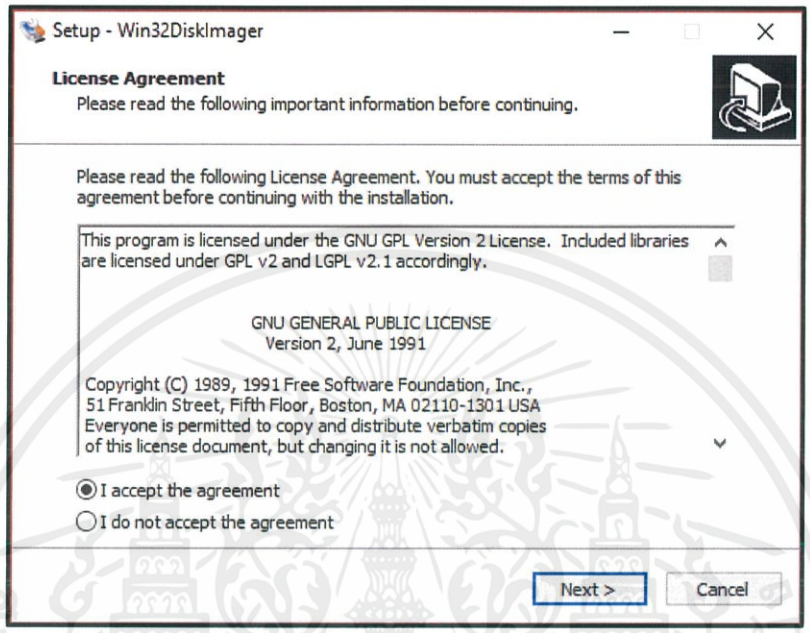
ตารางที่ ค.1 ไฟล์ที่จำเป็นในการติดตั้ง

เข้าที่อยู่ลิ้งค์ตามในตาราง จะขึ้นดังรูปที่ ค.1 คลิกที่ปุ่ม Download สีเขียว เพื่อทำการดาวน์โหลดโปรแกรม

รูปที่ ค.1

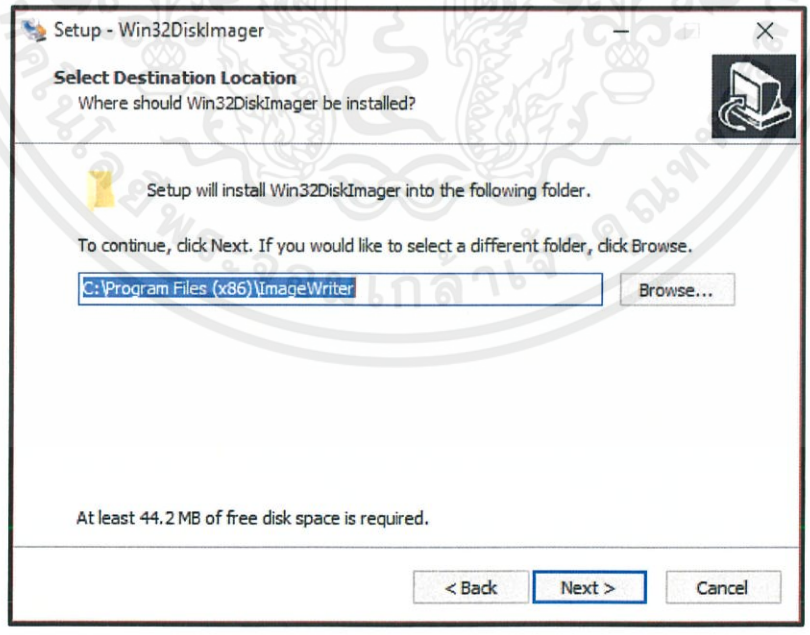
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการดาวน์โหลดเสร็จเรียบร้อย ให้ทำการติดตั้งโปรแกรม โดยดับเบิ้ลคลิกตัวติดตั้งที่โหลดมา จะขึ้นดังรูปที่ ค.2 แล้วทำการติ๊กที่ช่อง I accept the agreement แล้วกด Next



รูปที่ ค.2

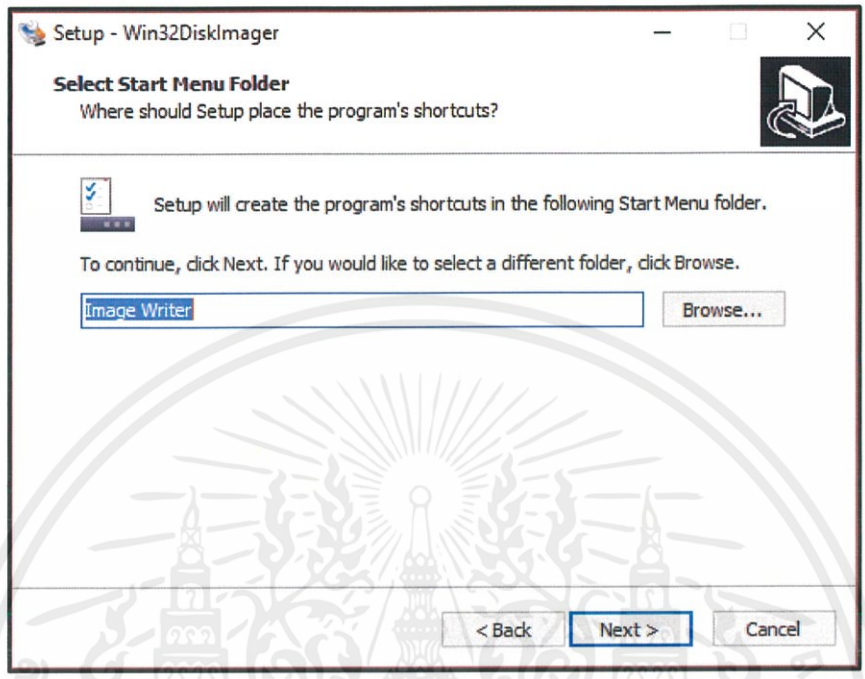
กด Next



รูปที่ ค.3

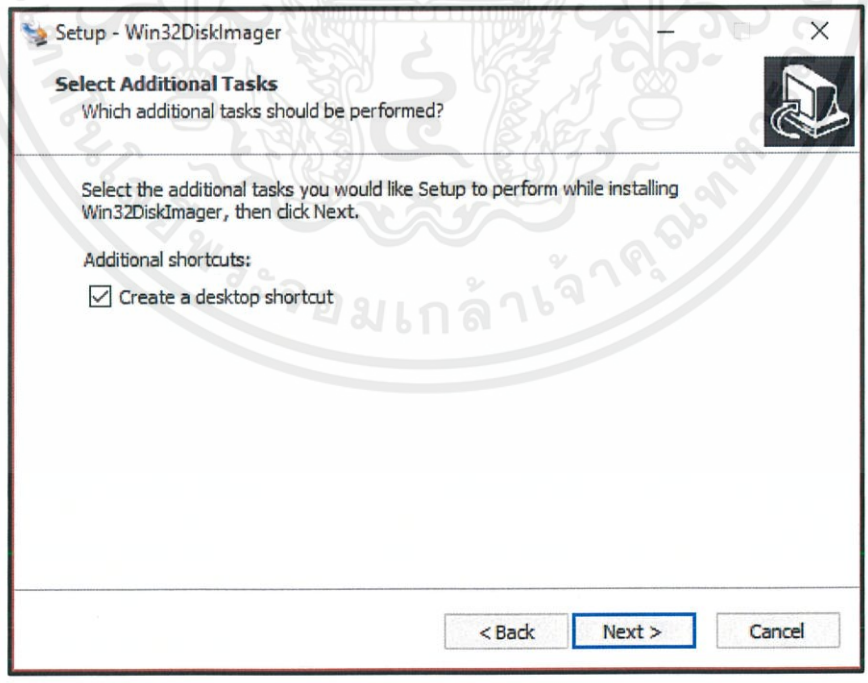
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด Next



รูปที่ ค.4

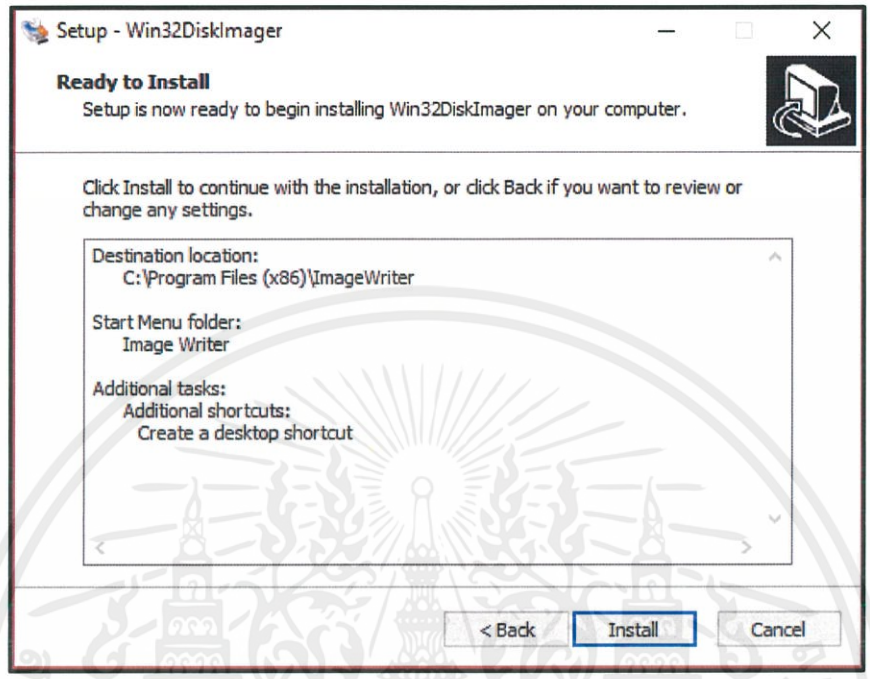
กดทำเครื่องหมาย ที่ช่อง Create a desktop shortcut แล้วกด Next



รูปที่ ค.5

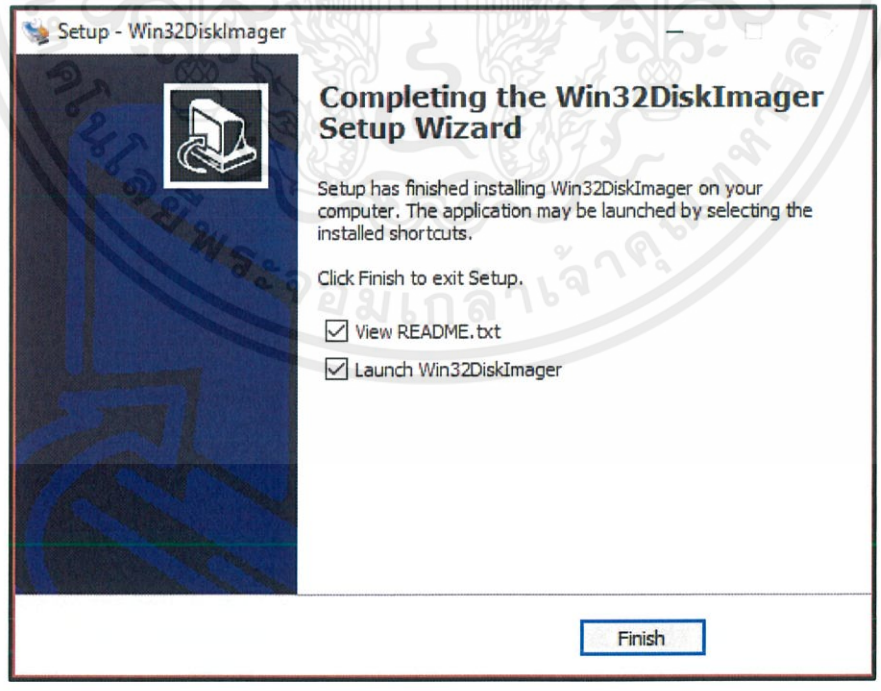
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด Install



รูปที่ ค.6

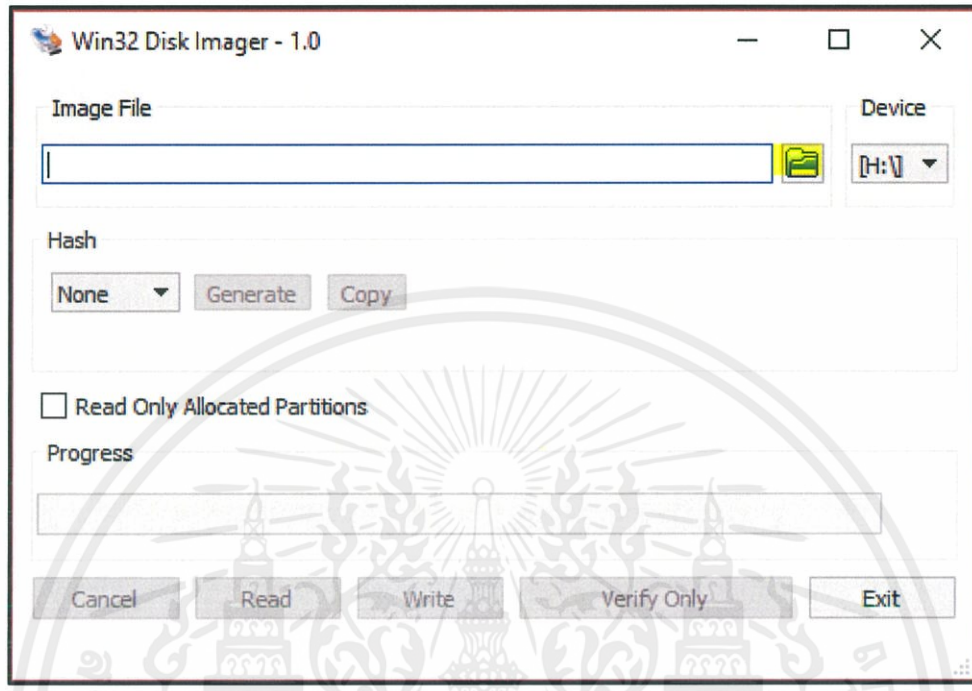
กด Finish เพื่อเสร็จสิ้นการติดตั้ง



รูปที่ ค.7

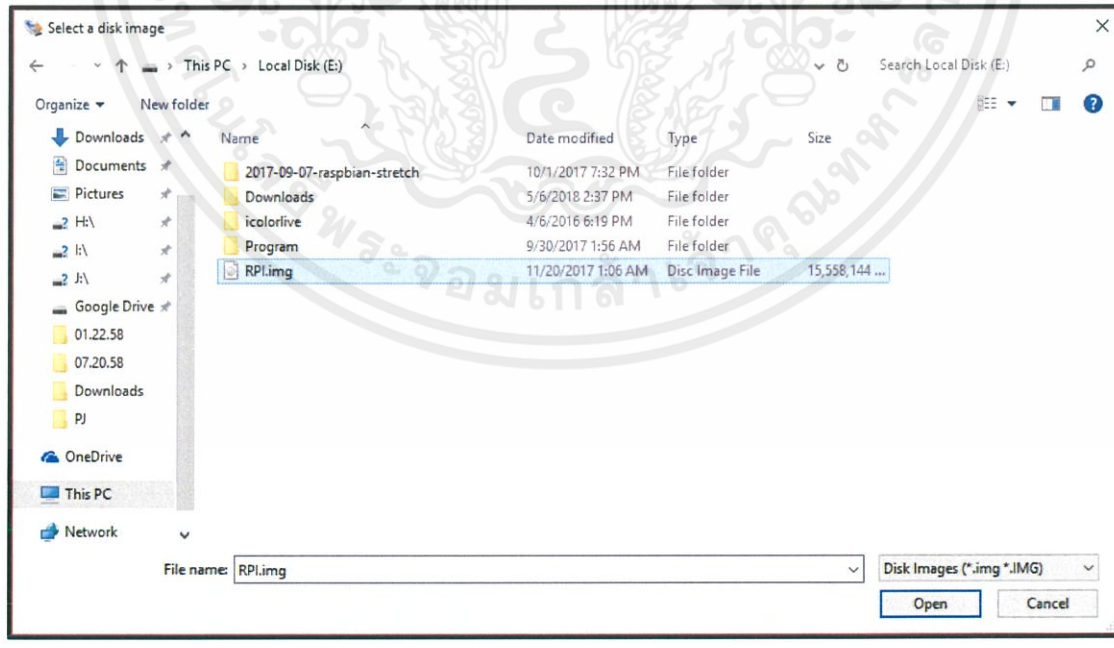
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าตาโปรแกรมจะเป็นดังรูป กดที่กล่องรูปภาพสีเหลือง



รูปที่ ค.8

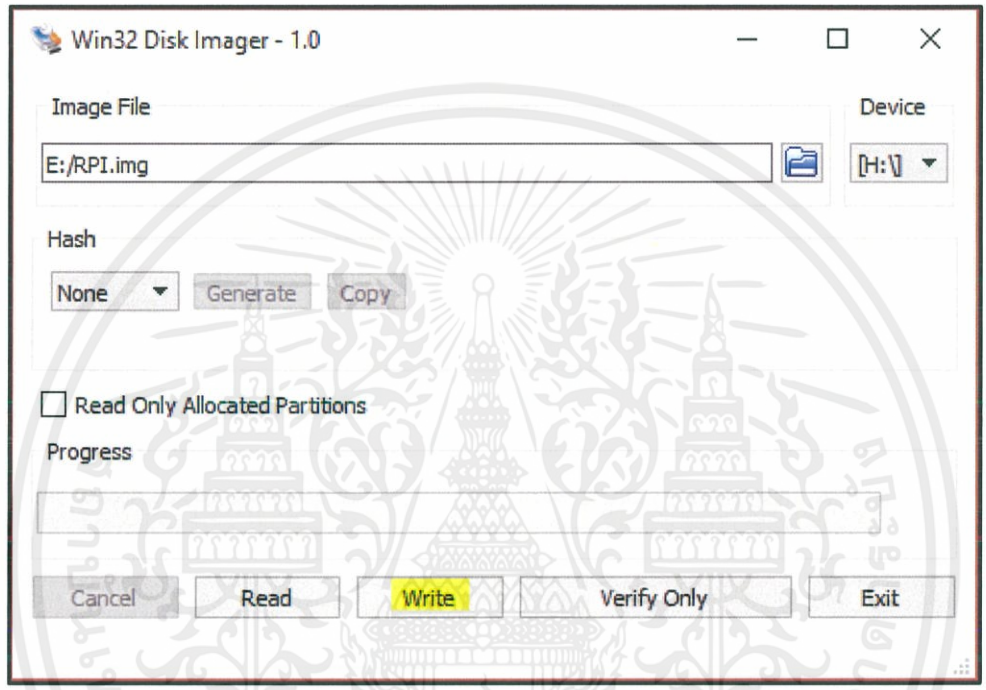
เลือกไปที่ไฟล์ RPI.img ที่เราทำการดาวน์โหลดมา แล้วกด Open



รูปที่ ค.9

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเลือกไปที่ไดร์วของ Memory Card ที่ช่อง Device ด้านขวา แล้วกด Write เพื่อทำการเขียนไฟล์ลง Memory Card เมื่อเสร็จสิ้น ถอด Memory Card และนำไปเสียบกับ Raspberry Pi เพื่อเริ่มต้นระบบปฏิบัติการ



รูปที่ ค.10

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] คมกฤช วงษ์แก้ว และ ชัยวัฒน์ แสนบุญศิริ(2550).รายงานการวิจัยเรื่อง เครื่องเตือนรถไฟบริเวณทางข้ามรถไฟ

[2] บริพัตร แซมทอง,ดิศรณ์ มุทธเมธา,วิทยา เสาโมก และพลวัฒน์วรคำ.รายงานการวิจัยเรื่อง ชุดระบบเตือนภัยทางข้ามทางรถไฟ

[3] ปุณณรัตน์ เมืองราช และวรเชษฐ์ ขันโยธา(2558). รายงานการวิจัยเรื่อง ระบบแจ้งเตือนสิ่งกีดขวางสำหรับคนพิการ

[4] Rakesh Kumar, Tepesh Parashar, Gopal Verma. Background Modeling and Substraction Based People Counting for Real Time Video Sueveillance.

[5] M. Oliveira, V.Santos. Automatic Detection of Cars in Real Roads using Haar-like Features.

[6] ศรราม หงษ์พรหม,ธีรัตน์ ชวนวัน และปาริฉัตร เสริมวุฒิสาร. รายงานการวิจัยเรื่อง ระบบตวรจจับการเคลื่อนไหวของมนุษย์ภายในแบบอัตโนมัติ.

[7] Facebook Master Tong Apollo ,
URL: https://www.facebook.com/mastertong.apollo, access on 24/10/2560

[8] ตรวจสอบเวลาการเดินรถ การรถไฟแห่งประเทศไทย, แหล่งที่มา
http://www.railway.co.th/checktime/checktime.asp, เข้าดูเมื่อวันที่ 16/09/2560

[9] สถานีรถไฟหัวตะเข้,แหล่งที่มา https://th.wikipedia.org/wiki/สถานีรถไฟหัวตะเข้,
เข้าดูเมื่อวันที่ 16/09/2560

[10] เทคโนโลยีการประมวลผลภาพ(2013), แหล่งที่มา https://silllovely.wordpress.com
เข้าดูเมื่อวันที่ 17/09/2560

[11] MarcusCode,Introduction Python(2017),
URL: http://marcuscode.com/lang/python/introduction ,access on 17/09/2560

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[12] Haar-Like Feature,From Wikipedia,
URL: https://en.wikipedia.org/wiki/Haar-like_feature access on 01/10/2560

[13] Playelek,การสร้างไฟล์ Cascade ไว้ตรวจจับวัตถุ ,
แหล่งที่มา https://playelek.com/haar-cascade-create/ เข้าดูเมื่อวันที่ 05/10/2560

[14] Haar-Like for face detection,
URL: https://www.quora.com/How-can-I-understand-Haar-like-feature-for-face-detection ,acess on 05/10/2560

[15] Background subtraction,
URL: https://en.wikipedia.org/wiki/Background_subtraction, access on 05/10/2560

[16] Open Source Computer (OpenCV),
URL: https://docs.opencv.org/3.2.0/d1/dc5/tutorial_background_subtraction.html ,access on 07/10/2560

[17] ทฤษฎีสีของระบบสี RGB ,SARA DEE,
URL: https://xn--12cf0dj0aaufkr9l0ai2m6ab4p.blogspot.com/2014/06/rgb.html ,access on 15/10/2560

[18] แนะนำการประมวลผลภาพดิจิตอล,ระบบสี HSV,
แหล่งที่มา http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm เข้าดูเมื่อวันที่ 15/10/2560

[19] บทความการพัฒนาโปรแกรมบน Raspberry Pi,
แหล่งที่มา http://thaieasyelec.com/article-wiki/embedded-electronics-application/บทความการพัฒนาโปรแกรมบน-raspberry-pi เข้าดูเมื่อวันที่ 16/10/2560

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้