ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถ
SIMULATION OF DROWSY DRIVER DETECTION AND WARNING SYSTEM

กฤติณ วงษ์ขจิตรเพ็ชร

KITTIN WONGKHAJITPETCH

กฤษฎา บัวพลอย

KRITSADA BUAPLOY

กันตพงศ์ พุ่มน้อย

KANTAPONG POOMNOI

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถ

SIMULATION OF DROWSY DRIVER DETECTION AND WARNING SYSTEM

โดย

กฤติณ วงษ์ขจิตรเพ็ชร

กฤษฎา บัวพลอย

กันตพงศ์ พุ่มน้อย

อาจารย์ที่ปรึกษา

ดร.อิทธิภูมิ บุญพิคำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

หัวข้อปริญญานิพนธ์	ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถ		
นักศึกษา	นาย กฤติณ	วงษ์ขจิตรเพ็ชร	รหัสประจำตัว 54010047
	นาย กฤษฎา	บัวพลอย	รหัสประจำตัว 54010054
	นาย กันตพงศ์	พุ่มน้อย	รหัสประจำตัว 54010086
ปริญญา	วิศวกรรมศาสตร์บัณฑิต		
สาขาวิชา	อิเล็กทรอนิกส์		
ปีการศึกษา	2557		
อาจารย์ที่ปรึกษา	ดร.อิทธิภูมิ บุญพิคำ		

บทคัดย่อ

โครงงานนี้นำเสนอการออกแบบ แบบจำลองระบบตรวจจับและเตือนคนขับรถง่วงนอน โดยอาศัยกล้องวิดีโอ และรูปแบบของการประมวลผลภาพ เพื่อตรวจจับคนขับรถง่วงนอนโดยตรวจดูดวงตาของคนขับรถ โดยต่อกล้องวิดีโอเข้ากับคอมพิวเตอร์ ที่ทำหน้าที่เป็นตัวประมวลผลทางภาพ พร้อมต่อเข้ากับไมโครคอนโทรลเลอร์ที่ควบคุมระบบเตือนในส่วนระบบเตือนประกอบด้วยส่วนสร้างเสียงเตือนและหลอด LED เพื่อแจ้งเตือนในกรณีที่ระบบสามารถตรวจจับได้ว่ามีอาการง่วงนอนเกิดขึ้น เพื่อปลุกคนขับรถที่ง่วงนอน นอกจากนั้นยังมีหน้าจอ LCD เพื่อแสดงสถานะของระบบอีกด้วย แต่หากกล้องวิดีโอไม่สามารถจับใบหน้าของคนขับได้ ผู้ขับขี่สามารถเพิ่มระบบอินฟราเรดเพื่อจับดวงตาได้ดียิ่งขึ้น

Thesis Title	Simulation Drowsy Driver Warning System.		
Student	Mr.Kittin	Wongkhajitpetch	Student ID 54010047
	Mr.Kritsada	Buaploy	Student ID 54010054
	Mr.Kantapong	Poomnoi	Student ID 54010086
Degree	Bachelor of Engineering.		
Program	Electronics Engineering.		
Year	2014		
Thesis Advisor	Ittibhoom Boonpikum , Ph. D.		

ABSTRACT

This project presents a simulate design of drowsy driver detection and warning system using video camera and image processing algorithms to detect drowsy driver by monitoring the driver's eyes. The camera is connected to a computer that serve as the video processor which interfaces with the microcontroller that controls the warning system. The warning system consists of a piezo buzzer and LEDs to wake the driver in case the system can detect the driver's drowsiness. In addition, the LCD display is added to provide massage to warn the driver and passenger.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จได้อย่างดี ด้วยคำแนะนำและคำปรึกษาจาก ดร.อิทธิภูมิ บุญพิคำ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ข้าพเจ้ารู้สึกซาบซึ้งในความอรุเคราะห์จากท่านอาจารย์และขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ทุกท่านในภาควิชา วิศวกรรมอิเล็กทรอนิกส์ที่คอยช่วย ชี้แนะให้คำปรึกษารวมทั้งเอื้อเฟื้อสถานที่ในการทำงาน รวมถึงพี่ๆ เพื่อนๆ ในภาควิชาทุกคนที่ให้คำแนะนำในด้านต่างๆ คอยให้ความช่วยเหลือ และกำลังใจเสมอมา

ขอขอบคุณเอกสารค้างอิงต่างๆ ที่ข้าพเจ้าได้หยิบยกมา ก่อให้เกิดองค์ความรู้ที่นำมาประยุกต์ใช้ในโครงงานนี้ ซึ่งข้าพเจ้าคาดหวังเป็นอย่างยิ่งว่าโครงงานนี้จะเป็นประโยชน์และช่วยลดอุบัติเหตุที่เกิดขึ้นบนท้องถนนในอนาคตได้

กฤติณ วงษ์ขจิตรเพ็ชร

กฤษฎา บัวพลอย

กันตพงศ์ พุ่มน้อย

ปริญญานิพนธ์ปีการศึกษา 2557

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถ

SIMULATION OF DROWSY DRIVER DETECTION AND WARNING SYSTEM

ผู้จัดทำ	นายกฤติณ	วงษ์ขจิตรเพ็ชร	รหัสประจำตัว 54010047
	นายกฤษฎา	บัวพลอย	รหัสประจำตัว 54010054
	นายกันตพงศ์	พุ่มน้อย	รหัสประจำตัว 54010086

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ดร.อิทธิภูมิ บุญพิคำ)

อาจารย์ที่ปรึกษา

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญรูป.....VII

สารบัญตาราง.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมุติฐานของการศึกษา.....1

1.4 ขอบเขตการวิจัย.....2

1.5 ประโยชน์ที่คิดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎี.....3

2.1 MATLAB.....3

2.1.1 โครงสร้างของ MATLAB.....3

2.2 Image Processing.....5

2.3 ความรู้พื้นฐานด้านการประมวลผลภาพดิจิตอล.....5

2.4 ภาพเชิงตัวเลข (Digital Image).....6

2.4.1 ภาพไบนารี่ (Binary image / black and white image).....6

2.4.2 ภาพสีเทา (Intensity Image /Monochrome Image/Gray Image).....7

2.4.3 ภาพสี (Color Image/RGB Image).....8

2.5 การทำภาพให้เป็นระดับสีขาวดำ (Binary Image).....9

2.5.1 การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า (Pre-assiged Threshold Value).....10

2.5.2 การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value).....11

2.6 การสร้างข้อมูลเลขฐานสองจากรูปภาพ.....11

2.7 การหาการเปลี่ยนแปลงความเข้มของใบหน้า.....12

2.8 การหาตำแหน่งดวงตาในแนวดิ่ง.....13

2.9 การตรวจจับอาการง่วงนอน.....14

2.10 การบอกสภาพง่วงนอน.....15

2.11 ระบบ Real-Time.....15

2.12 การจับภาพ.....15

2.13 การหาค่า Binarization Threshold ที่ถูกต้อง.....16

2.14 การตรวจจับใบหน้า (Face Detection).....16

บทที่ 3 การออกแบบระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถ.....19

3.1 ส่วนประกอบของระบบ.....19

3.1.1 ฮาร์ดแวร์.....19

3.1.1.1 กล้อง.....20

3.1.1.2 คอมพิวเตอร์.....21

3.1.1.3 FT232RL.....21

3.1.1.4 ATmega328.....22

3.1.1.5 ระบบแจ้งเตือน.....23

3.1.2 ซอฟแวร์.....26

3.2 ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถที่พร้อมใช้งาน.....28

บทที่ 4 วิธีการและผลการทดสอบ.....29

4.1 การทดสอบภาวะแสงปกติ.....30

4.2 การทดสอบภาวะแสงน้อย.....33

4.3 การทดสอบภาวะแสงน้อยร่วมกับแสงอินฟราเรด.....36

4.4 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดด.....39

4.5 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดดร่วมกับแสงอินฟราเรด.....42

บทที่ 5 สรุปผลการทดสอบและข้อเสนอแนะ.....45

บรรณานุกรม.....46

ภาคผนวก.....47

ภาคผนวก ก.....48

ภาคผนวก ข.....53

สารบัญรูป

หน้า

รูปที่ 2.1 พิกัดของระบบภาพดิจิตอล ¼ ของแบบ double array.....5

รูปที่ 2.2 ภาพแบบสองสีแสดงให้เห็นแต่ละ Pixel ถูกกำหนดให้มีค่าเป็น 0 แสดงเป็นสีดำและ 1 แสดงเป็นสีขาว.....6

รูปที่ 2.3 ภาพ Binary และ Pixel value ของภาพ Binary.....7

รูปที่ 2.4 (a) แสดง Gray image และ Pixel value ของภาพ Binary.....7

รูปที่ 2.5 รูปแสดงภาพ RGB และ Pixel value ของภาพ RGB.....8

รูปที่ 2.6 ภาพแสดงการเปรียบเทียบระหว่าง 1)Binary Image 2)Gray Image 3)Color Image เรียงจากซ้ายไปขวา.....8

รูปที่ 2.7 แสดงผลของการเปลี่ยนค่า threshold (a) ค่า threshold 100 (b) ค่า threshold 150 (C) ค่า threshold 200.....12

รูปที่ 2.8 แสดงด้านบนของใบหน้าและการเปลี่ยนแปลงความเข้มแสง 2 ตำแหน่งแรก.....13

รูปที่ 2.9 กราฟค่าเฉลี่ยในแนวนอนสำหรับด้านซ้ายของใบหน้า.....13

รูปที่ 2.10 ตำแหน่งดวงตาด้านซ้ายจากการหา Valleyจากกราฟข้างต้น.....14

รูปที่ 2.11 เปรียบเทียบระหว่างดวงตาเปิดและปิด.....14

รูปที่ 2.12 Examples of the Saar features.....17

รูปที่ 2.13 The integral image trick.....17

รูปที่ 2.14 The classifier cascade is a chain.....18

รูปที่ 3.1 การออกแบบระบบตรวจจับคนง่วงขณะขับรถ.....19

รูปที่ 3.2 แผนผังแสดงการทำงานของฮาร์ดแวร์.....20

รูปที่ 3.3 กล้อง Anitect ความละเอียด 16 ล้านพิคเซล.....20

รูปที่ 3.4 โมดูล FTDI FT232RL.....21

รูปที่ 3.5 แผนผังของวงจรโมดูล FTDI FT232RL.....22

รูปที่ 3.6 ตำแหน่งขาของไมโครคอนโทรลเลอร์ ATmega328.....23

รูปที่ 3.7 แผนผังวงจรระบบแจ้งเตือน.....24

รูปที่ 3.8 ระบบแจ้งเตือนเมื่อประกอบเสร็จแล้ว.....26

รูปที่ 3.9 แผนผังแสดงการทำงานของซอฟแวร์.....26

รูปที่ 3.10 ผังงานแสดงการทำงานของระบบ.....27

รูปที่ 3.11 การเขียนโปรแกรมใน MATLAB.....28

รูปที่ 3.12 ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถที่พร้อมใช้งาน.....28

รูปที่ 4.1 แสดงส่วนที่ใช้กำหนดค่าเทรชโช.....29

รูปที่ 4.2 การทดสอบในภาวะแสงปกติ a) เมื่ออยู่ภาวะปกติ b) เมื่ออยู่ภาวะง่วงนอน.....31

รูปที่ 4.3 การทดสอบในภาวะแสงน้อย a) เมื่ออยู่ภาวะปกติ b) เมื่ออยู่ภาวะง่วงนอน.....34

รูปที่ 4.4 การทดสอบในภาวะแสงน้อยร่วมกับแสงอินฟราเรด a) เมื่ออยู่ภาวะปกติ b) เมื่ออยู่ภาวะง่วงนอน.37

รูปที่ 4.5 การทดสอบกรณีที่ผู้ขับสวมแว่นตากันแดด a) เมื่ออยู่ภาวะปกติ b) เมื่ออยู่ภาวะง่วงนอน.....40

รูปที่ 4.6 การทดสอบกรณีที่ผู้ขับสวมแว่นตากันแดดร่วมกับแสงอินฟราเรด a) เมื่ออยู่ภาวะปกติ
b) เมื่ออยู่ภาวะง่วงนอน0.....43

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงการเชื่อมต่อระหว่างโมดูล FTDI FT232RL กับ ไมโครคอนโทรลเลอร์.....22

ตารางที่ 3.2 แสดงการเชื่อต่อระหว่าง LCD 16*2 กับ ไมโครคอนโทรลเลอร์.....25

ตารางที่ 4.1 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงปกติโดยใช้ค่าเทรชโช 20.....30

ตารางที่ 4.2 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงน้อย โดยใช้ค่าเทรชโช 20.....33

ตารางที่ 4.3 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงน้อยร่วมกับอินฟราเรดโดยใช้
ค่าเทรชโช 5.....36

ตารางที่ 4.4 ผลการทดสอบการตรวจจับดวงตาในกรณีที่ผู้ขับรถสวมแว่นตากันแดด
โดยใช้ค่าเทรชโช 20.....39

ตารางที่ 4.5 ผลการทดสอบการตรวจจับดวงตาในกรณีที่ผู้ขับรถสวมแว่นตากันแดดร่วมกับอินฟราเรด
โดยใช้ค่าเทรชโช 1.....42

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอุบัติเหตุทางรถยนต์ส่วนใหญ่เกิดขึ้นจากความประมาทและการทำผิดกฎจรจารของผู้ขับขี่และสาเหตุหนึ่งของอุบัติเหตุเกิดจากการเมาแล้วขับและการหลับในของผู้ขับขี่เพราะฉะนั้นพวกเราจึงสนใจทำการระบบที่สามารถช่วยลดอุบัติเหตุลงได้ด้วยอุปกรณ์เตือนคนง่วงนอนขณะขับรถขึ้นมา

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของโครงงานนี้เพื่อ สร้างแบบจำลองการแจ้งเตือนเมื่อผู้ขับขี่รถยนต์เกิดการหลับในเพื่อลดอุบัติเหตุบนท้องถนน โดยใช้กล้องตรวจจับดวงตาแล้วนำมาประมวลผลผ่านโปรแกรมคอมพิวเตอร์

1.3 สมมุติฐานของการศึกษา

การตรวจจับอาการง่วงนอนของคนขับรถจากดวงตาของคนขับ โดยอาศัยกล้องของคอมพิวเตอร์

สำหรับการตรวจจับอาการง่วงนอนของคนขับรถจากดวงตาของคนขับนั้นใช้หลักการที่ว่าความเข้มของแสงบริเวณดวงตามนุษย์นั้นต่างๆไปจากความเข้มของแสงบริเวณใบหน้าอย่างมาก ทำให้สามารถตรวจจับดวงตาจากใบหน้าได้ ว่าอยู่ในสภาวะปิดหรือเปิด

1.4 ขอบเขตการวิจัย

โครงงานนี้มีขั้นตอนการศึกษาแบ่งออกเป็นสองส่วนคือส่วนของโปรแกรมและส่วนของวงจรควบคุมการทำงานและประมวลผลโดยเราต้องหาข้อมูลทางทฤษฎีที่จำเป็นจะต้องให้ในโครงงานนี้ เช่น โปรแกรมที่ใช้ในการประมวลผลหรือวงจรควบคุมการทำงานว่าเราจะต้องออกแบบวงจรแบบไหนที่สามารถตอบสนองการทำงานได้

โครงงานฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท คือ

บทที่ 1 กล่าวถึงความเป็นมาของโครงงาน ความมุ่งหมาย วัตถุประสงค์ สมมติฐาน แนวคิดและทฤษฎีที่ใช้ขอบเขตของโครงงานและขั้นตอนของการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีที่ใช้ในการทำโครงงานนี้

บทที่ 3 การออกแบบการทดลอง

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 เป็นบทสรุปการศึกษาและข้อเสนอแนะ

1.5 ประโยชน์ที่คิดว่าจะได้รับ

สามารถช่วยลดอุบัติเหตุลงได้ด้วยระบบจำลองการเตือนคนง่วงนอนขณะขับรถ

บทที่ 2

ทฤษฎี

2.1 MATLAB

MATLAB เป็นภาษาคอมพิวเตอร์ชั้นสูง (High-level Language) สำหรับการคำนวณทางเทคนิคที่ประกอบด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อน และการจำลองแบบเพื่อให้มองเห็นภาพพจน์ได้ง่ายและชัดเจน ชื่อของ MATLAB ย่อมาจาก matrix laboratory เดิมโปรแกรม MATLAB ได้เขียนขึ้นเพื่อใช้ในการคำนวณทาง matrix หรือเป็น matrix software ที่พัฒนาจากโพกเจ็กที่ชื่อ LINKPACK และ EISPACK MATLAB ได้พัฒนามาด้วยการแก้ปัญหาที่ส่งมาจากหลายๆอย่างผู้ใช้เป็นระยะเวลาหลายปีจึงทำให้โปรแกรม MATLAB มีฟังก์ชั่นต่างๆให้เลือกใช้มากมาย ในบางมหาวิทยาลัยได้ใช้โปรแกรม MATLAB เป็นหลักสูตรพื้นฐานในการศึกษาทางด้านคณิตศาสตร์ วิศวกรรมศาสตร์ และวิทยาศาสตร์แขนงต่างๆ ตลอดจนในด้านอุตสาหกรรมได้ใช้โปรแกรม MATLAB เป็นเครื่องมือสำหรับใช้ในงานวิจัย พัฒนาและวิเคราะห์โปรแกรม MATLAB จะมีกล่องเครื่องมือที่ใช้ในการหาคำตอบเรียกว่า Toolbox โดยโปรแกรม MATLAB จะมี toolbox ในแต่ละสาขา เช่น การประมวลผลสัญญาณ (Signal processing toolbox) การประมวลผลภาพ (image processing toolbox) ระบบควบคุม (control system toolbox) โครงข่ายประสาท (neural networks toolbox) ฟัซซี่ลอจิก (fuzzy logic toolbox) เวฟเลท (wavelet toolbox) การติดต่อสื่อสาร (communication toolbox) สถิติ (Statistics toolbox) และสาขาอื่นๆ มากมาย ภายใน toolbox แต่ละสาขาก็จะมีฟังก์ชั่นต่างๆ ที่เกี่ยวข้องการแก้ปัญหาในสาขานั้นๆ

2.1.1 โครงสร้างของMATLAB

โครงสร้างของโปรแกรม MATHLAB ประกอบด้วย 5 ส่วนใหญ่ๆ คือ

2.1.1.1 ภาษาโปรแกรม MATHLAB (MATLAB LANGUAGE)

MATLAB เป็นโปรแกรมภาษาชั้นสูงที่ใช้ควบคุม Flow statement ฟังก์ชั่น โครงสร้างข้อมูล อินพุท/เอาท์พุท และลักษณะโปรแกรม Object-Oriented Programming ทำให้การเขียนโปรแกรมไม่ยุ่งยาก เมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาอื่นๆ เช่น C , Fortran , Basic เป็นต้น

2.1.1.2 สถาปัตยกรรมในการทำงานของ MATLAB

MATLAB จะมีกลุ่มของเครื่องมือที่เป็นประโยชน์สำหรับการทำงานของผู้ใช้โปรแกรมหรือโปรแกรมเมอร์ ประโยชน์ที่กล่าวนี้ก็คือ การจัดตัวแปรใน Workspace การนำข้อมูลหรือการผ่านค่าตัวแปร เข้า/ออก และกลุ่ม ของเครื่องมือต่างๆ นี้ก็จะใช้สำหรับพัฒนา จัดการตรวจสอบความผิดพลาดของโปรแกรม (Debugging) ที่ได้เขียนขึ้น

2.1.1.3 ฟังก์ชั่นในการคำนวณทางคณิตศาสตร์ (Mathematical Function Library)

MATLAB จะมีไลบรารีทั่วไปที่ใช้ในการคำนวณอย่างกว้าง เช่น Sine , Cosine และพีชคณิตเชิงซ้อนโดนสามารถนำไปประยุกต์ใช้เป็นฟังก์ชั่นหรือไลบรารีเพิ่มเติมจากไลบรารีที่ใช้กันโดยทั่วไป เช่น ฟังก์ชั่นในการหา Eigenvalues และ Eigenvectors การแยกตัวประกอบ และ ส่วนประกอบของเมตริกซ์ด้วยวิธีต่างๆ การวิเคราะห์ข้อมูล การหาความน่าจะเป็น และ การแก้ปัญหาระบบของสมการเชิงเส้นที่เป็นพื้นฐานของสาขาวิชาต่างๆ เป็นต้น ทำให้โปรแกรมMATLAB มีฟังก์ชั่นสำหรับใช้งานค่อนข้างมากและครอบคลุมในรายละเอียดของการคำนวณสาขาต่างๆได้มากขึ้น

2.1.1.4 Handle Graphics

ระบบกราฟิกของ MATLAB จะประกอบด้วยคำสั่งชั้นสูงสำหรับการพล็อตกราฟโดยมีพื้นฐานอยู่บนแนวความคิดที่ว่า ทุกๆสิ่งบนหน้าต่างรูปภาพของโปรแกรม MATLAB จะเป็นวัตถุ (Object) ซึ่งมีเอกลักษณ์เฉพาะตัว Handle Graphics

ประกอบด้วยคำสั่งชั้นสูงให้คุณได้เลือกใช้ในการสร้าง Graphic User lnterface บนพื้นฐานการประยุกต์ใช้งาน นอกจากนี้โปรแกรม MATLAB ยังมีฟังก์ชั่นที่ใช้สำหรับการแสดงภาพสองมิติ ภาพสามมิติ และการสร้างภาพเคลื่อนไหว

2.1.1.5 The MATLAB Application Program lnterface (API)

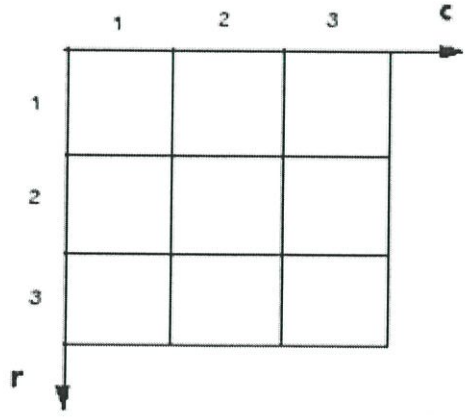
API จะใช้เพื่อสนับสนุนการติดต่อจากภายนอกโดยใช้โปรแกรมที่เป็น mex ไฟล์ซึ่งเป็นไฟล์ที่เขียนขึ้นโดยใช้ mex ฟังก์ชั่นใน MATLAB ซึ่งจะเรียกใช้รูทีนจากโปรแกรมภาษา C และ Fortran หรืออาจกล่าวได้ว่า API เป็น ไลบรารีที่เขียนด้วยโปรแกรมภาษา C และ Fortran ที่มีการเชื่อมต่อกับโปรแกรม MATLAB ด้วยไฟล์ที่เป็น mex ฟังก์ชั่นอีกทั้ง MATLAP API นี้ยังสามารถเรียก Routine จาก MATLAB (Dynamic linking) ก็ได้

2.2 Image processing

Image Processing คือการนำเสนอรายละเอียดของวัตถุ (Object) โดยการจับเอาหรือถ่ายเท(Maniltipulation) รายละเอียดของภาพที่ได้จากวัตถุเพื่อนำรายละเอียดของภาพเหล่านั้นไปทำประโยชน์ให้มากยิ่งขึ้น วิธีการทาง Image Processing จะต่างกับวิธีการทาง Computer graphic กล่าวคือ วิธีการทาง Computer graphic ตัวคอมพิวเตอร์เองที่จะเป็นตัวที่สร้างภาพ แต่เทคนิคทาง Image Processing นั้นใช้คอมพิวเตอร์สำหรับจัดเก็บข้อมูล

2.3 ความรู้พื้นฐานด้านการประมวลผลภาพดิจิตอล

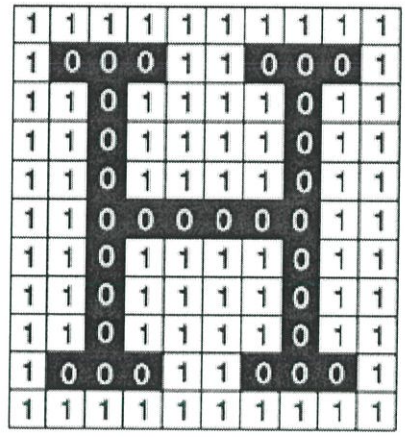
การประมวลผลภาพดิจิตอล จะเกี่ยวข้องกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอลเพื่อที่จะสามารถนำเอาข้อมูลนี้ไปผ่านกระบวนการต่างๆด้วยคอมพิวเตอร์ ได้ ซึ่งการทำงานของคอมพิวเตอร์ระบบการรับข้อมูลเข้าหรือส่งข้อมูลออกจะอยู่ในรูปแบบดิจิตอลเท่านั้น ในการประมวลผลภาพดิจิตอล เมื่อระบบได้รับข้อมูลภาพเข้าไปแล้วจะทำการคำนวณและส่งออกมาเป็นข้อมูลที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปร array ดังรูปที่ 2.1 โดยค่าในแต่ละช่องจะแสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่อง array เป็นตัวกำหนดตำแหน่งของจุดภาพ



รูปที่ 2.1 พิกัดของระบบภาพดิจิตอล 1/4 ของแบบ double array

2.4 ภาพเชิงตัวเลข (Digital Image)

Digital Image เป็นการจับภาพจากสิ่งแวดล้อมหรือทำสำเนาภาพจากเอกสารให้อยู่ในรูปแบบของอิเล็กทรอนิกส์ เช่น รูปถ่าย เอกสารที่เขียนด้วยมือ และพิมพ์เขียว เป็นต้น โดย Digital image จะอยู่ในรูปของแผ่นตารางโดยแต่ละช่องจะเป็นส่วนหนึ่งของภาพหรืออักษร เรียกแต่ละจุดหรือช่องนั้นว่า " Pixel " แต่ละ Pixel จะถูกกำหนดให้มีระดับของความเข้ม (สีดำ สีขาว สีเทาหรือสีอื่นๆ แต่ละ Pixel ก็จะแทนด้วย Binary Digital (Bit) จะถูกเก็บเป็นลำดับใน Computer และโดยทั่วไปจะถูกลดขนาดลงด้วยวิธีทางคณิตศาสตร์ (บีบอัดให้เล็กลง) แต่ละ Bit จะถูกแปลและอ่านโดย Computer ให้เป็นแบบ Analog ซึ่งเป็นรูปภาพ หรือ แผ่นพิมพ์

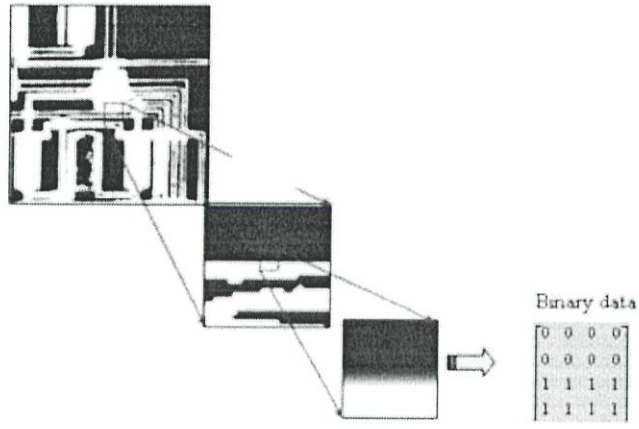


รูปที่ 2.2 ภาพแบบ 2 สี แสดงให้เห็นแต่ละ Pixel ถูกกำหนดให้มีค่าเป็น 0 แสดงเป็นสีดำและ 1 แสดงเป็นสีขาว

ชนิดของภาพเชิงตัวเลข (Digital Image Type) แบ่งออกเป็นหลายประเภท ได้แก่

2.4.1 ภาพไบนารี่ (Binary Image/Black and White Image)

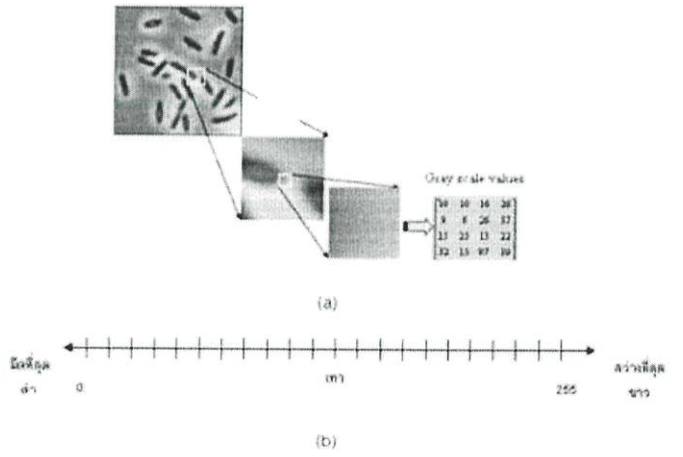
ค่าในแต่ละ pixel ของ binary image จะใช้แค่ 1 บิต ซึ่งจะมีค่าที่เป็น ไปได้คือ 0 (สีดำ) และ 1 (สีขาว) เท่านั้น ดังรูปที่ 2.3



รูปที่ 2.3 ภาพ binary และ pixel value ของภาพ binary

2.4.2 ภาพสีเทา (Intensity Image /Monochrome Image/Gray Image)

ค่าในแต่ละ pixel ของ gray image คือค่าความเข้มของแสง ณ แต่ละตำแหน่งของ pixel ซึ่ง จะอยู่ในรูปของ gray scale (gray level) ดังรูปที่ 2.4 (a) ค่าที่เป็นไปได้ของ gray scale จะขึ้นอยู่กับจำนวน bit ที่ใช้ตัวอย่างเช่น 8-bit monochrome จะมี gray scale ทั้งหมด 256 ระดับ ดังรูปที่ 2.4 (b)

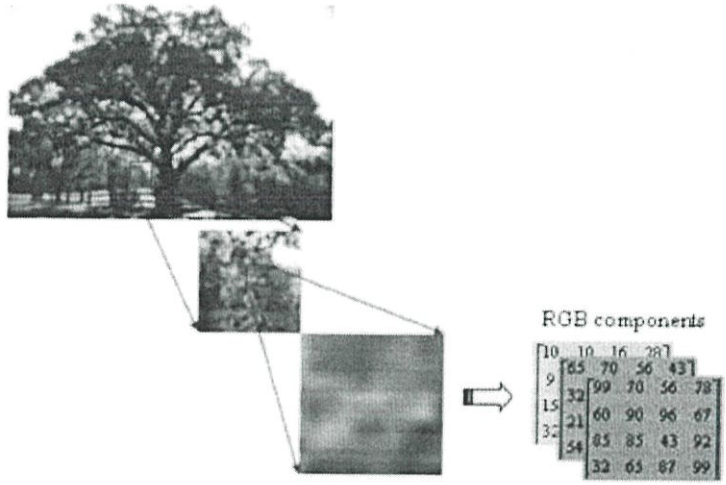


รูปที่ 2.4 (a) แสดง gray image และ pixel value ของ gray image (b) แสดง 8 bit gray scale

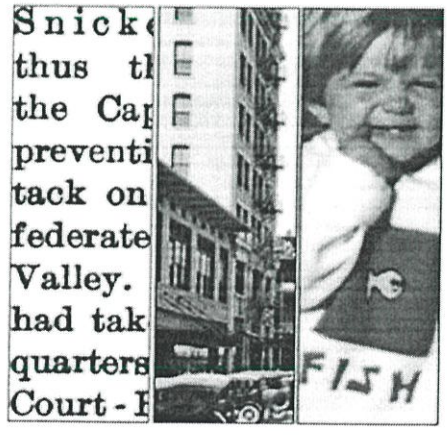
2.4.3 ภาพสี (Color Image/RGB Image)

ค่าในแต่ละ pixel ของ color image จะประกอบไปด้วย vector ที่แสดงค่าของสีแดง สีเขียว และสีน้ำเงิน อย่างละ 8 บิต ดังนั้น RGB image 1 pixel จะประกอบไปด้วยจำนวนบิตทั้งหมด 24 บิต ทำให้ RGB

image มีจำนวนสีที่เป็นไปได้ทั้งหมด 2^24 สี ประมาณ 16.7 ล้านสี ตัวอย่างดังรูปที่ 2.5



รูปที่ 2.5 รูปแสดงภาพ RGB และ pixel value ของภาพ RGB



รูปที่ 2.6 ภาพแสดงการเปรียบเทียบระหว่าง 1)Binary Image 2)Gray Image 3)Color Image เรียงจากซ้ายไปขวา

2.5 การทำภาพให้เป็นระดับสีขาวดำ (Binary image)

อุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือ สีขาวกับสีดำยังมีการใช้กันอย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer) เครื่องโทรสาร (Fax) จอแสดงผลแบบโมโนโครม (Monochrome) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ที่มีราคาถูกดังนั้นการแสดงผลหรือการพิมพ์รูปภาพที่มีความเข้มของภาพหลายระดับซึ่งมีมากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงสองระดับ เท่านั้น

จะเห็นได้ว่าการแสดงผลของภาพที่มีความเข้มหลายระดับบนอุปกรณ์ที่มีการแสดงผลได้แค่2 ระดับนั้น จะต้องแปลงข้อมูลภาพให้เป็นภาพแบบไบนารี่(Binary Image) ซึ่งการสร้างภาพแบบไบนารี่การแปลง ภาพที่มีความเข้มหลายระดับ(Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต(bit) นั่นเองโดย 0 แทน ด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสีดำเมื่อเสร็จขั้นตอนในการทำไบนารี่จึง นำภาพที่ได้ไปแสดงผลที่อุปกรณ์เหล่านั้นจะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารี่จึงมี ความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีระดับความเข้มของภาพหลายระดับบนอุปกรณ์ที่มี ความสามารถในการแสดงผลได้ 2 ระดับสำหรับประโยชน์อีกประการหนึ่งในการแปลงข้อมูลภาพนั้นเป็น ภาพไบนารี่คือการลดเนื้อที่การเก็บข้อมูลภาพจะใช้เนื้อที่การเก็บ 8 บิต หรือ 256 ระดับ เมื่อสร้างเป็นภาพไบนารี่เมื่อสร้างเป็นภาพไบนารี่แล้วสามารถลดลงจากเดิมได้ถึง 8 เท่า นั่นคือ 1 จุดจะใช้เนื้อที่ในการเก็บ ทั้งหมดคือ 1 บิต อีกทั้งยังสามารถนำไปใช้งานได้อีกแพร่หลาย เช่น การประมวลผลขั้นต้น (Preprocessing)

ในการสร้างภาพไบนารี่สามารถทำได้โดยการใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) เป็นการพิจารณาจุดพิกเซลในภาพว่าจุดใดควรจะเป็นจุดขาวที่มีค่าเท่ากับ 0 หรือจุดใดควรจะเป็นจุดที่มีค่าเท่ากับ 1 โดยจะทำการเปรียบเทียบค่าของแต่ละพิกเซล(f(x,y)) กับค่าคงที่ที่เรียกว่าค่า เทรชโฮล (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่มีความแตกต่างระหว่างวัตถุ (Object) และพื้นหลัง (Background) ซึ่งเป็นลักษณะเดียวกับสีขาวของป้ายทะเบียนและสีดำที่เป็นตัวอักษรของป้ายทะเบียน โดย ค่าของพิกเซลในภาพ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดเป็น 1 (จุดดำ) และถ้าค่าของพิกเซลใดๆใน ภาพที่มีค่ามากกว่าหรือเท่ากับค่าเทรชโฮลจะถูก0 (จุดขาว)

กำหนดให้เป็น

0 ; g(x,y) < Thr

B(x,y) =

1 ; g(x,y) >= Thr

b(x,y) ข้อมูลภาพผลลัพธ์เป็นภาพแบบไบนารี่

g(x,y) ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

Thr ค่าเทรชโฮลเป็นค่าคงที่ที่อยู่ระหว่าง 0 ถึง L ระดับ

0 คือจุดขาว , 1 คือจุดดำ โดยที่ L เป็นระดับความเข้มของจุดภาพสูงสุด

ในการสร้างภาพไบนารี่โดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่ดีและคมชัด จะต้องเกิดจากการเลือกค่าเทรชโฮลที่ถูกต้องและเหมาะสม ถ้าเลือกค่าเทรชโฮลไม่เหมาะสม เช่น ค่าเทรชโฮลที่มากเกินไปหรือน้อยเกินไป ภาพที่ได้จะขาดความคมชัด หรือออาจจะทำให้รายระเอียดของภาพขาดหายไป หรือภาพที่ได้ อาจจะมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน(Noise) เกิดขึ้นทำให้ภาพผลลัพธ์ที่ได้ไม่ ชัดเจนเท่าที่ควร ดังนั้นปัญหาในการสร้างภาพแบบไบนารี่คือ ทำอย่างไรจึงจะคำนวณค่าเทรชโฮลที่ เหมาะสมกับภาพแต่ละภาพที่จะมาสร้างเป็นภาพแบบไบนารี่ ซึ่งค่าเทรชโฮลสามารถคำนวณได้หลายวิธี แต่ ละวิธีจะเหมาะการทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า (Pre-assigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Midrange Threshold Value) แต่ละวิธีจะอธิบายได้ดังนี้

2.5.1 การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า (Pre-assigned Threshold Value)

การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้าเป็นวิธีที่ง่ายที่สุดเป็นการคำนวณค่าโดยการกำหนดเอง

ของผู้ใช้ ซึ่งการกำหนดนี้จะขึ้นอยู่กับประสบการณ์ของผู้ใช้ โดยการเลือกค่าคงที่ที่เรียกว่าค่าเทรชโฮลนั้น ค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าสูงสุดและค่าต่ำสุดของข้อมูลอินพุทแต่ละพิกเซลของภาพ เช่น ภาพข้อมูล อินพุทมีเกรย์เลเวล (Gray Level) 256 ระดับจะมีค่าทั้งหมด 0-255 เมื่อเลือกค่าเทรชโฮลแล้วก็สามารถนำค่าเทรชโฮลเป็นตัวกำหนดในการสร้างภาพไบนารี่ได้

2.5.2 การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value)

การหาค่าเทรชโฮลโดยพิจารณาจากค่ากลางเป็นการหาค่าเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า สำหรับวิธีนี้จะเป็นการคำนวณโดยอัตโนมัติโดยไม่ต้องให้ผู้ใช้เป็นผู้กำหนด โดย การหาค่าเทรชโฮล วิธีนี้อาศัยหลักการคำนวณพื้นฐานทางสถิติในเรื่องของการหาค่ากลางหรือค่าเฉลี่ย (Mean) มาประยุกต์ใช้ค่าเทรชโฮลที่คำนวณได้จากค่ากึ่งกลางที่อยู่ระหว่างค่าที่มีความเข้มสูงสุด (Maximum Level) และระดับความเข้มต่ำสุด (Minimum Level) ของข้อมูลอินพุท เมื่อทำการคำนวณค่าเทรชโฮลได้แล้ว ก็จะสามารถนำค่าเทรชโฮลเป็นตัวกำหนดในการสร้างภาพไบนารี่ได้

2.6 การสร้างข้อมูลเลขฐานสองจากรูปภาพ

ขั้นแรกของการบอกตำแหน่งของตา คือสร้างข้อมูลเลขฐานสอง(Binarization) จากภาพที่ได้จากกล้อง Binarization คือการแปลงภาพให้เป็นข้อมูลเลขฐานสอง

ภาพเลขฐานสองคือภาพที่ตำแหน่งภาพแต่ละตำแหน่งแสดงด้วยตัวเลข 0 หรือ 1 โดย 0 แทนบริเวณสีดำ และ 1 แสดงบริเวณสีขาว ส่วนบริเวณสีเทานั้นจะถูกแปลงเป็นเลขฐานสองผ่านค่า thresholding ส่วนเอาท์พุทภาพเลขฐานสองนั้นจะมีค่าเป็น 0(สีดำ) จากภาพดั้งเดิมที่มีความสว่างน้อยกว่าค่า threshold และมีค่าเป็น 1(สีขาว) สำหรับบริเวณอื่นๆ กำหนดค่า threshold นี้ตามเงื่อนไขแสงที่อยู่รอบๆ และความซับซ้อนของคนขับ หลังจากตรวจดูภาพใบหน้าหลายๆแบบภายใต้เงื่อนไขความสว่างแสง ได้ค่า 150 ที่ถือว่าเป็นค่าที่ดีที่สุด วิธีที่ใช้ในการเลือกค่า threshold ที่ถูกต้องนี้ อาศัยแนวคิดที่ว่าภาพเลขฐานสองของใบหน้าคนขับรถนี้ ส่วนใหญ่แล้วจะมีสีขาว และมีส่วนที่เป็นสีดำเพียงเล็กน้อย อันได้แก่ ตา จมูก และปาก

รูปที่ 2.7 แสดงผลของการเปลี่ยนค่า threshold ใช้ค่า threshold เป็น 100, 150, และ 200 ตามลำดับ คือตัวอย่างของภาพในรูปเลขฐานสองสำหรับรูปแบบการตรวจจับดวงตา ที่มีพื้นหลังเป็นสีดำสม่ำเสมอ และใบหน้าเป็นสีขาว ทำให้สามารถหาขอบใบหน้าดังที่จะได้อธิบายในหัวข้อต่อไป

(a)

(b)

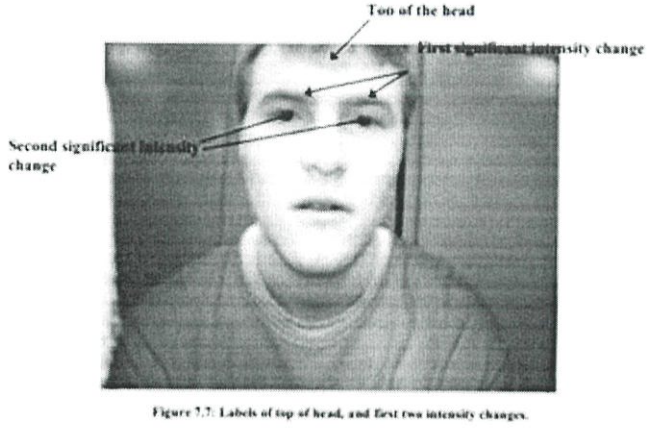
(c)

รูปที่ 2.7 แสดงผลของการเปลี่ยนค่า threshold (a) ค่า threshold 100 (b) ค่า threshold 150 (2) ค่า threshold 200

2.7 การหาการเปลี่ยนแปลงความเข้มของใบหน้า

ขั้นตอนต่อไปในการหาตำแหน่งดวงตา คือหาการเปลี่ยนแปลงความเข้มแสงบนใบหน้า โดยใช้ภาพเริ่มแรก ไม่ใช่ภาพเลขฐานสอง ขั้นแรกคือคำนวณค่าเฉลี่ยความเข้มแสงสำหรับตำแหน่ง y แต่ละตำแหน่ง เรียกว่าค่าเฉลี่ยในแนวนอน(horizontal average) เนื่องจากค่าเฉลี่ยนี้ได้จากค่าในแนวนอน(ค่า x) ค่าที่เปลี่ยนแปลงอย่างมากตามแนวแกนจะแสดงการเปลี่ยนแปลงค่าความเข้ม(ดำขาว) เมื่อลงจุดค่าในแนวนอน จะพบว่ามีค่าที่เปลี่ยนแปลงเล็กๆนี้จำนวนมาก ซึ่งไม่ได้แทนการเปลี่ยนแปลงค่าความเข้มแสง แต่เป็นผลจากค่าความแตกต่างค่าเฉลี่ยที่ไม่สูงมากนัก แก้ปัญหาโดยใช้วิธี “smoothing” ซึ่งกำจัดค่าการเปลี่ยนแปลงที่ไม่สูงเหล่านี้ ทำให้ค่าเรียบขึ้น และได้กราฟที่ชัดขึ้น

หลังจากได้ค่าเฉลี่ยในแนวแกนนอน ขั้นต่อไปคือหากค่าที่เปลี่ยนแปลงที่สูง นั่นคือจะบอกตำแหน่งของดวงตา สมมติว่าใบหน้านั้นโล่ง(มีผมปกคลุมหน้าผากเล็กน้อย) this is based on the notion จากด้านบนขอบใบหน้า ลงมา และการเปลี่ยนแปลงความเข้มแสงแรกขนตา การเปลี่ยนแปลงต่อไป

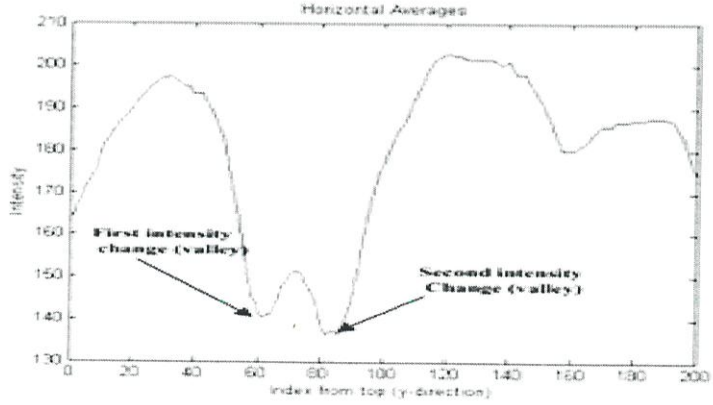


รูปที่ 2.8 แสดงด้านบนของใบหน้า และการเปลี่ยนแปลงความเข้มแสง 2 ตำแหน่งแรก

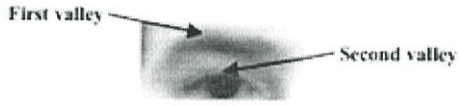
ค่าเปลี่ยนแปลงที่พบโดยหาการเปลี่ยนแปลงความชันของเส้นกราฟจากลบเป็นบวกและยอกของเส้นกราฟที่พบจากการเปลี่ยนแปลงความชันจากบวกไปเป็นลบ ขนาดของการเปลี่ยนแปลงนี้หาได้จากระยะระหว่างยอดและ valley และเมื่อพบ “valleys” ก็จะนำมาจัดเรียงตามความความแตกต่าง

2.8 การหาตำแหน่งดวงตาในแนวดิ่ง

“Valley” ค่ามากสุดแรกที่มีค่าพิกัด y ต่ำสุดคือจนตา รองลงมาคือดวงตา แสดงดังรูปที่ 2.9 และ 2.10



รูปที่ 2.9 กราฟค่าเฉลี่ยในแนวนอนสำหรับด้านซ้ายของใบหน้า



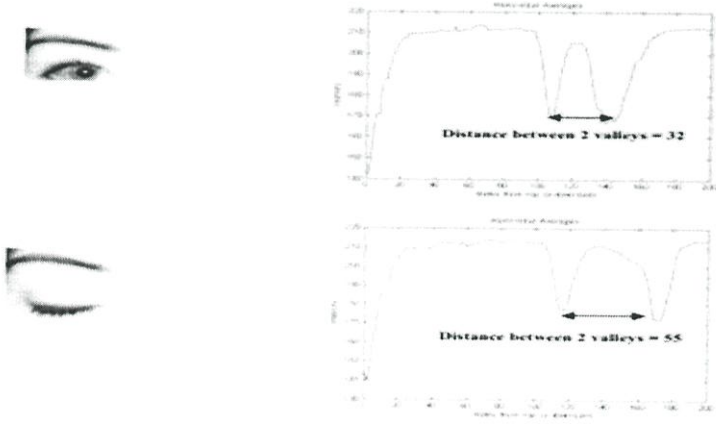
รูปที่ 2.10 ตำแหน่งดวงตาด้านซ้าย จากการหา “valleys” จากกราฟข้างต้น

ขั้นตอนนี้สำหรับการหาตำแหน่งด้านซ้ายและด้านขวาของใบหน้า และบริเวณด้านซ้ายและด้านขวาของดวงตาที่พบ จะนำมาเปรียบเทียบกัน เพื่อตรวจสอบบริเวณดวงตาที่พบนั้นถูกต้องหรือไม่ การคำนวณค่าเฉลี่ยบริเวณทางด้านซ้าย นั้นใช้ค่าเฉลี่ยจากขอบด้านซ้ายไปจนถึงจุดศูนย์กลางของใบหน้า เช่นเดียวกับวิธีการทางด้านขวาของใบหน้า เหตุผลในการคิดแยกด้านสองด้านก็เนื่องจาก เมื่อศีรษะของคนขับรถมีการเอียง ค่าเฉลี่ยในแนวนอนจะไม่ถูกต้อง ตัวอย่างเช่น หากศีรษะเอียงไปทางขวา ค่าเฉลี่ยในแนวนอนของบริเวณขนตาจะเป็นขนตาทางด้านซ้ายแทน และเป็นไปได้ที่จะเป็นด้านขวาของหน้าผาก

2.9 การตรวจจับอาการง่วงนอน

การหาสถานะของดวงตา

สถานะ(สภาพ)ของดวงตา(ไม่ว่าเปิดหรือปิด) หาได้จากระยะระหว่างการเปลี่ยนแปลงความเข้มแสงชุดแรกที่พบในขั้นตอนข้างต้น เมื่อดวงตาปิด ระยะระหว่างพิกัด y ของการเปลี่ยนแปลงความเข้มแสงจะมาก เมื่อเปรียบเทียบกับเมื่อตาเปิด แสดงดังรูปที่ 2.11



รูปที่ 2.11 เปรียบเทียบระหว่างดวงตาเปิดและปิด

ข้อจำกัดวิธีการนี้คือ หากคนขับรถเคลื่อนใบหน้าเข้าหรือออกจากตัวกล้อง ระยะทางจะเปลี่ยนแปลง เนื่องจากจำนวน pixel ของใบหน้าจะเปลี่ยนไปด้วย ดังแสดงด้านล่าง เนื่องจากข้อจำกัดนี้ ระบบจะถือว่าใบหน้าของคนขับรถจะอยู่ห่างจากตัวกล้องเป็นระยะเท่ากันตลอดเวลา

2.10 การบอกสภาพง่วงนอน

เมื่อเฟรมจับภาพดวงตาปิดได้ติดต่อกัน 5 เฟรม เสียงเตือนจะทำงานเพื่อกระตุ้นให้คนขับตื่น อย่างไรก็ตามต้องมี closed frames ที่ตามมาด้วยเพื่อหลีกเลี่ยงการทำงานผิดพลาดในกรณีที่ตาปิดเนื่องจากกระพริบ

2.11 ระบบ Real-Time

ระบบ real-time นี้รวมการทำงานอีกส่วนหนึ่งเมื่อตรวจจับคนขับรถ เพื่อให้ระบบมีความเสถียร โดยมีสถานะเริ่มต้น ที่ซึ่งใน 4 เฟรมแรก จะถือว่าดวงตามคนขับรถนั้นเปิดอยู่ และระยะระหว่างพิกัด y ที่ซึ่งความเข้มแสงเปลี่ยน จะถือว่าเป็นค่าอ้างอิง หลังสถานะเริ่มต้นผ่านไป ระย่ห่างที่คำนวณได้จะถูกนำมาเปรียบเทียบกับระยะที่พบในสถานะเริ่มต้น หากพบระยะที่น้อยกว่า (ความแตกต่างระหว่าง 5-80 pixels) จะถือว่าดวงตานี้ปิด

ส่วนเพิ่มเติมสำหรับระบบ real-time คือการเปรียบเทียบด้านซ้ายและด้านขวาของดวงตาที่พบ โดยแยกพิจารณา แล้วนำตำแหน่งทั้งสองมาเปรียบเทียบกัน โดยถือว่าศรีษะของคนขับนั้นไม่ทำมุม(เอียง ส่วนพิกัด y ทางด้านซ้าย และด้านขวาควรมีระยะเท่ากัน หากแตกต่างกัน ระบบจะถือว่าไม่พบตำแหน่งของดวงตา และเอาท์พุทจะส่งข้อความเพื่อบอกว่าระบบไม่สามารถตรวจจับดวงตาได้ ทำให้ระบบจะพยายามหาตำแหน่งดวงตาต่อไปอีก ส่วนนี้ยังมีประโยชน์ในกรณีที่คนขับรถนั้นอยู่นอกมุมกล้อง ในกระณีนี้ระบบควรจะรายงานว่าไม่ทำการตรวจจับสภาวะง่วงนอน สำหรับกรณีทีไม่สามารถตรวจจับดวงตาได้ภายใน 5 เฟรมติดต่อกันหรือมากกว่า ต้องส่งเสียงเตือนออกมา ซึ่งป้องกันกรณีที่ศรีษะของคนขับรถนั้นตกลงมา ซึ่งเป็นกรณีที่ต้องส่งเสียงเตือน

2.12 การจับภาพ

สิ่งที่สำคัญที่สุดคือการส่งรูปภาพที่ได้จากกล้องไปยังคอมพิวเตอร์ ซึ่งมีส่วนประกอบด้วยกันสองส่วน ดังนี้ 1) จับภาพและส่งภาพอย่างต่อเนื่อง 2) สามารถเข้าถึงข้อมูลภาพที่เก็บไว้ได้ เริ่มจากการสร้าง “Dirt Cheap Frame Grabber” และทดสอบ

2.13 หาค่า binarization threshold ที่ถูกต้อง

เนื่องจากใบหน้าและแสงสว่างรอบๆ นั้นแตกต่างกัน จึงเป็นการยากที่จะหาค่า threshold สำหรับการ binarization แก้ปัญหาโดยเลือกค่าใดค่าหนึ่งที่ทำให้มีจุดดำบนใบหน้าน้อยที่สุด ตามด้วยสังเกตภาพเลขฐานสองของคนที่แตกต่างกัน สรุปได้ว่าค่า threshold ที่ต้องการนี้ที่ให้ผลกับคนทุกคนนั้นเป็นไปไม่ได้ สุดท้ายจึงต้องตัดสินใจว่าจุดดำบนใบหน้านั้นยอมรับได้ และใช้วิธีการกำจัดจุดดำนี้ในภายหลังแทน

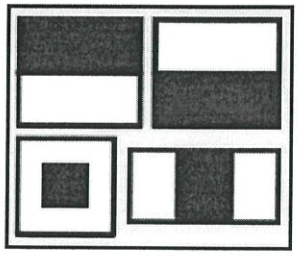
2.14 การตรวจจับใบหน้า (Face Detection)

คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอหลังจากนั้นก็จะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และ อัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธีซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมาก

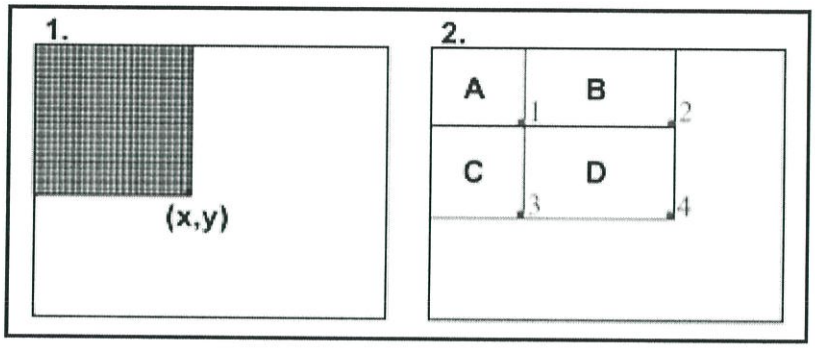
วิธีการหนึ่งที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตราความถูกต้องในการตรวจหาสูง ซึ่งPaul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์

ในปี ค.ศ. 2001 โดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอวิธีการแทนรูปภาพที่ เรียกว่า"Integral Image" ซึ่งช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้น และได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้น ฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features (features ที่ให้ classifiers ที่มีประสิทธิภาพสูงสุด) นอกจากนี้ยังได้อธิบายถึงการรวม classifiers แบบ cascade ซึ่งช่วยให้ส่วนพื้น หลังของภาพถูกปฏิเสธได้เร็วและเน้นการคำนวณไปที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการสแกน sub-window เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต การประมวลผลภาพแบบทั่วไปจะใช้การปรับขนาดภาพขาเข้าแตกต่างกันหลายๆขนาด และใช้ตัวตรวจหา (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ ซึ่งวิธีนี้กินเวลาในการคำนวณมากเนื่องมาจากการคำนวณบนรูปภาพที่มีขนาดแตกต่างกัน Viola-Jones ได้เสนอวิธีใหม่โดยการปรับขนาดตัวตรวจหาแทนที่จะปรับขนาดภาพขาเข้า และใช้ตัวตรวจหาค้นหาวัตถุหลายๆรอบ (แต่ละรอบใช้ขนาดแตกต่างกัน) ซึ่งทั้ง สองวิธีน่าจะใช้เวลาในการคำนวณไม่ต่างกันมากนัก แต่ Viola-Jones ได้คิดค้นตัวตรวจหาที่ใช้จำนวนครั้งในการคำนวณคงที่แม้จะมีขนาดของภาพแตกต่างกัน โดยตัวตรวจหาดังกล่าวนี้สร้างขึ้นโดยใช้ features ของHaar wavelets (รูปที่ 2.12) และ Integral Image (รูปที่ 2.13)



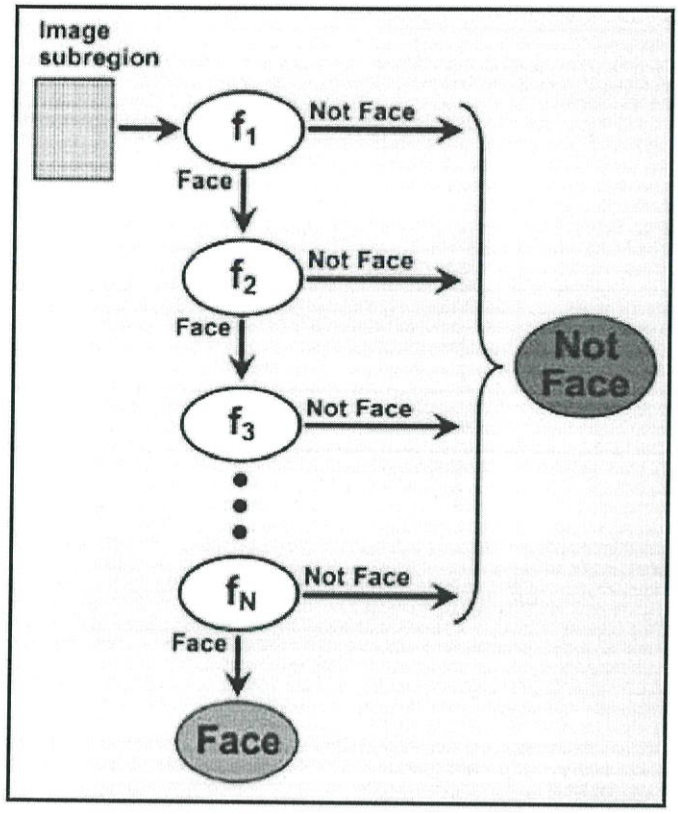
รูปที่ 2.12 Examples of the Haar features



รูปที่ 2.13 he Integral Image trick

1. หลังจากการอินทิเกรตพิกเซล (x, y) ประกอบด้วยผลรวมของค่าพิกเซลภายในสี่เหลี่ยมที่แรงา
2. ผลรวมของพิกเซลในสี่เหลี่ยม D คือ (x4, y4) - (x2, y2) - (x3, y3) + (x1, y1)

หลักการของอัลกอริทึมค้นหาหน้าของ Viola-Jones คือการใช้ตัวตรวจหา สแกนหลายๆ ครั้งบนภาพเดิม แต่ด้วยขนาดที่แตกต่างกัน ถึงแม้ว่าจะมีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ได้โดยใช้หลักการ "ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า" เพราะการตัดสินใจว่าบริเวณใดๆ ไม่ใช่ใบหน้านั้น ทำได้เร็วกว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.14 ซึ่งเมื่อ sub-window ถูกจัดประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้นถูกจำแนกเป็น มีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายิ่งมีจำนวนชั้น ของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยิ่งมีมากขึ้น

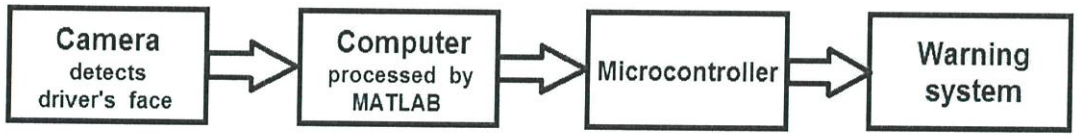


รูปที่ 2.14 ตัวจำแนกเกิดจากการนำฟิลเตอร์หลายตัวมาต่อกัน.

บทที่ 3

หลักการออกแบบระบบจำลองการตรวจจับ และแจ้งเตือนคนง่วงขณะขับรถ

ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถจะทำการตรวจจับอาการง่วงนอนจากดวงตาของผู้ขับรถโดยใช้กล้องเว็บแคมที่เชื่อมต่อกับคอมพิวเตอร์เพื่อทำการประมวลผลภาพบริเวณใบหน้าของผู้ขับขี่ที่ได้รับมาจากกล้องผ่านโปรแกรม MATLAB จากนั้นจึงส่งข้อมูลเข้าสู่ไมโครคอนโทรลเลอร์เพื่อเปิดการแจ้งเตือนในกรณีที่ระบบสามารถตรวจจับได้ว่าผู้ขับรถมีอาการง่วงนอน



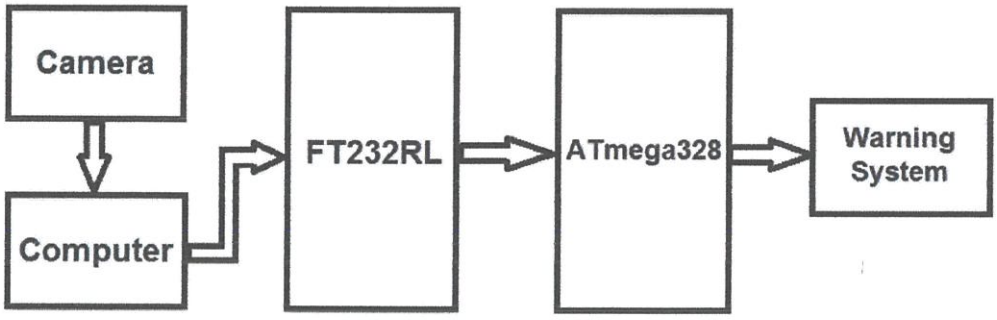
รูปที่ 3.1 การออกแบบระบบตรวจจับคนง่วงขณะขับรถ

3.1 ส่วนประกอบของระบบ

ระบบตรวจจับและแจ้งเตือนคนง่วงขณะขับรถแบ่งออกเป็น 2 ส่วนหลัก ได้แก่ ส่วนของฮาร์ดแวร์ซึ่งเป็นส่วนที่ใช้เป็นอุปกรณ์ในการตรวจจับ ประมวลผล ส่งผ่านข้อมูลรวมถึงอุปกรณ์และวงจรต่างๆที่เกี่ยวข้องในโครงงาน และส่วนของซอฟแวร์ซึ่งเป็นโปรแกรมที่ทำหน้าที่ตรวจรับภาพจากกล้องมาทำการประมวลผลรวมถึงโปรแกรมที่ใช้ในการสั่งเปิดระบบแจ้งเตือน

3.1.1 ฮาร์ดแวร์

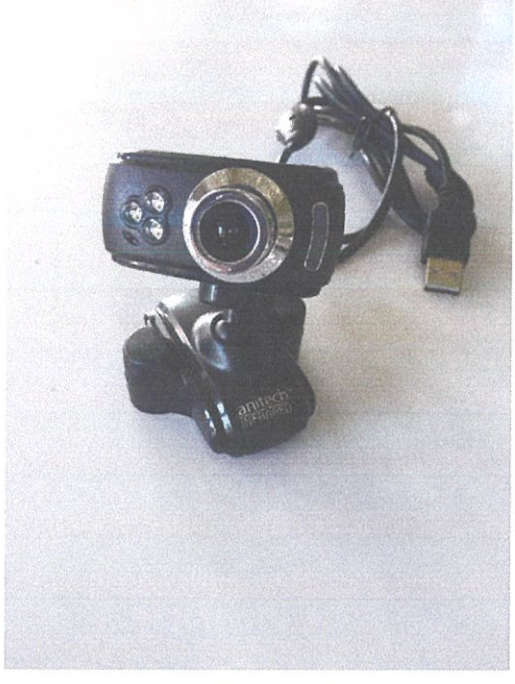
ฮาร์ดแวร์ของระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถแบ่งออกเป็น 5 ส่วน คือ กล้อง คอมพิวเตอร์ FT232RL Atmega328 และระบบแจ้งเตือน ซึ่งแผนผังการทำงานของฮาร์ดแวร์แสดงดังรูปที่ 3.2



รูปที่ 3.2 แผนผังแสดงการทำงานของฮาร์ดแวร์

3.1.1.1 **กล้อง**

ทำหน้าที่เป็นอุปกรณ์ตรวจจับอาการง่วงนอนโดยจะเล็งไปที่บริเวณใบหน้าของผู้ขับรถเพื่อทำการจับภาพและนำภาพที่ได้มาทำการประมวลผลผ่านทางคอมพิวเตอร์ต่อไป ในโครงงานนี้ใช้กล้อง Webcam ของ Anitech ซึ่งมีความละเอียด 16 ล้านพิกเซลซึ่งระบบปรับความสว่างอัตโนมัติและแหล่งกำเนิดแสงอินฟราเรดในตัว



รูปที่ 3.3 กล้อง Anitech ความละเอียด 16 ล้านพิกเซล

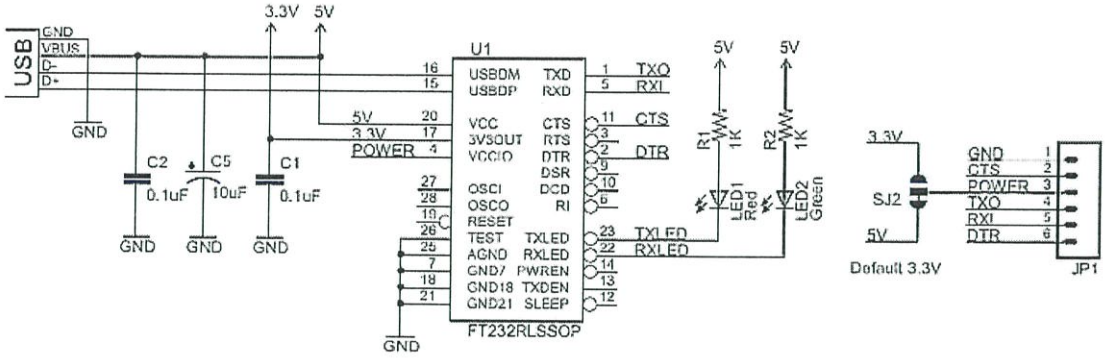
3.1.1.2 **คอมพิวเตอร์**

โครงงานนี้ใช้คอมพิวเตอร์โน้ตบุ๊ก Toshiba Satellite M840 Core i5 ความเร็วในการประมวลผล 2.50 GHz ถึง 3.10 GHz บนระบบปฏิบัติการ Window 8 เป็นหน่วยประมวลผลหลักของระบบจำลองการตรวจจับคนง่วงโดยรับภาพจากกล้องมาทำการประมวลผลในโปรแกรม MATLAB (Version R2013a) ก่อนที่จะส่งสัญญาณไปควบคุมระบบแจ้งเตือนเมื่อโปรแกรมประเมินว่าผู้ขับรถมีอาการง่วงนอน

3.1.1.3 **FT232RL**

ในการโปรแกรมข้อมูลลงไมโครคอนโทรเลอร์และการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรเลอร์จะถูกทำผ่านทางโมดูล FTDI FT232RL ซึ่งต่ออยู่กับพอร์ต COM4 ของคอมพิวเตอร์ โดยโมดูล FT232RL จะทำหน้าที่แปลงสัญญาณจากพอร์ต USB จากคอมพิวเตอร์ไปเป็นสัญญาณอนุกรม UART เพื่อสื่อสารกับไมโครคอนโทรลเลอร์และเป็นช่องทางจ่ายไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์อีกด้วย การเชื่อมต่อระหว่างโมดูล FT232RL กับไมโครคอนโทรเลอร์แสดงดังตารางที่ 3.1

รูปที่ 3.4 โมดูล FTDI FT232RL



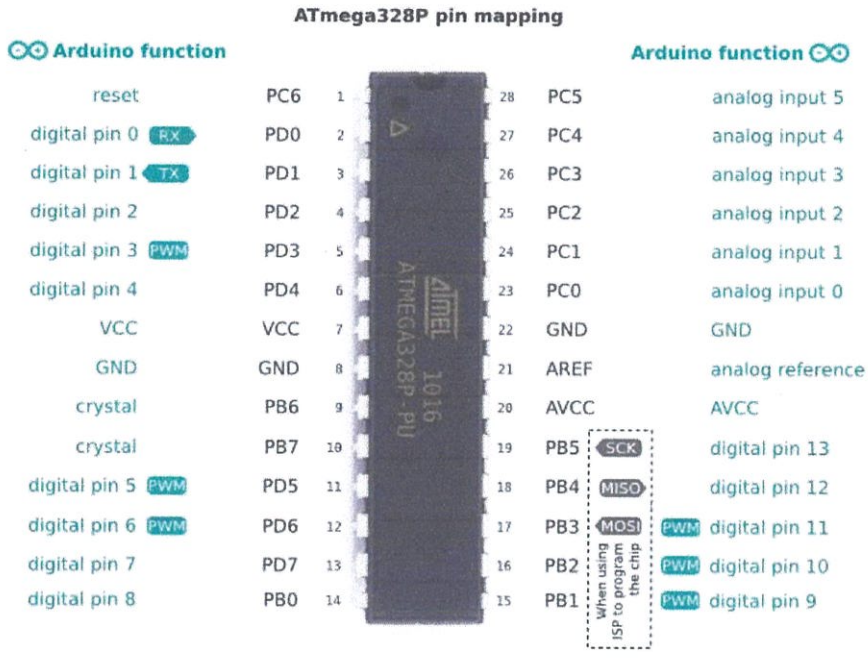
รูปที่ 3.5 แผนผังวงจรของโมดูล FTDI FT232RL

ตารงที่ 3.1 แสดงการเชื่อมต่อระหว่างโมดูล FTDI FT232RL กับไมโครคอนโทรเลอร์

FTDI 232RL	Microcontroller (ATmega328)
GND	GND
CTS	-
VCC	VCC
TX	RX
RX	TX
DTR	GRN (Reset)

3.1.1.4 **ATmega328**

ทำหน้าที่เป็นไมโครคอนโทรลเลอร์ที่ใช้สั่งการระบบแจ้งเตือนโดยรับข้อมูลจากคอมพิวเตอร์ ซึ่งผ่านการแปลงจากโมดูล FT232RL ในโครงงานนี้ใช้ไมโครคอนโทรลเลอร์ ATmega328 ซึ่งเป็นตัวประมวลผลเดียวกับบอร์ด Arduino Uno R3

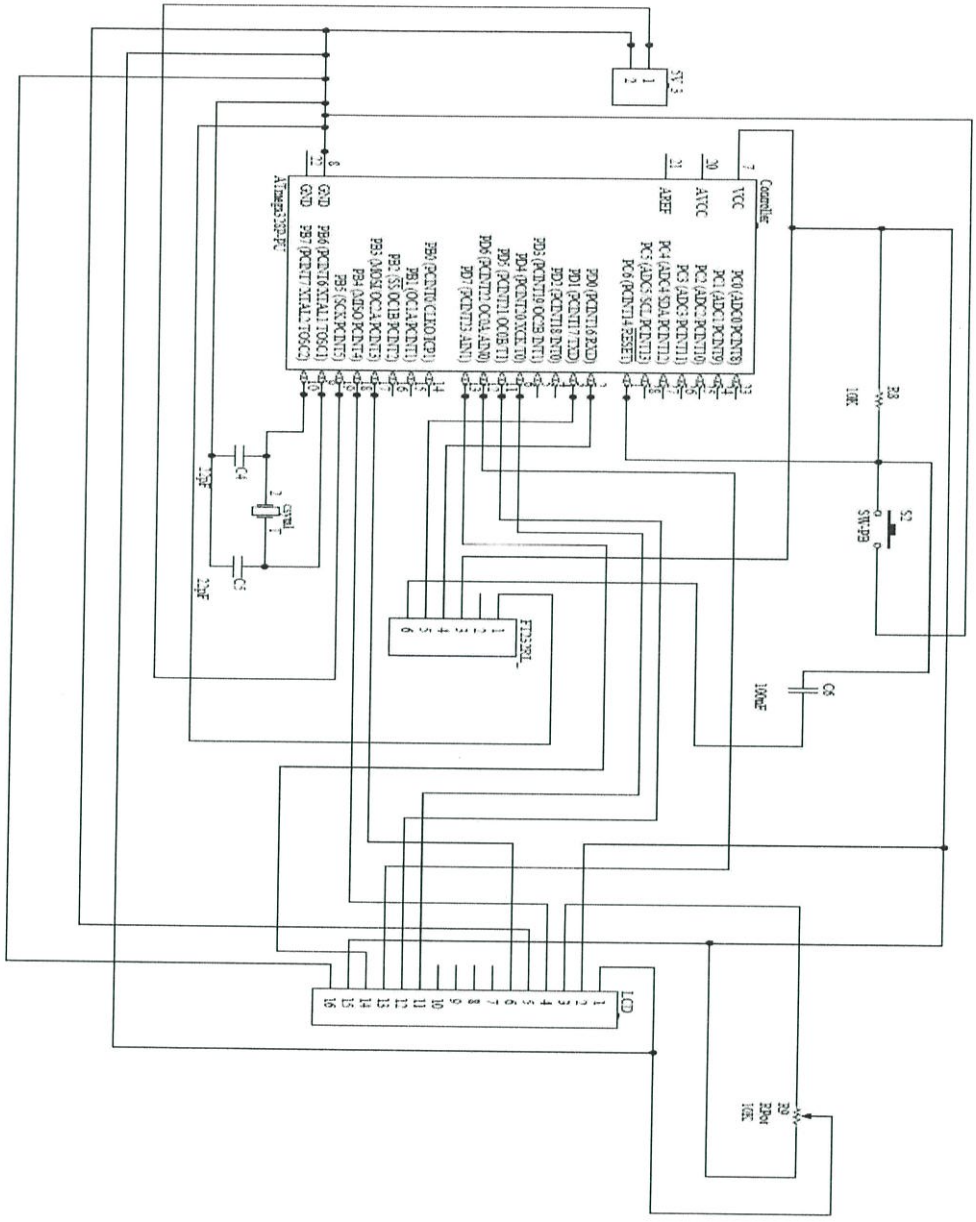


รูปที่ 3.6 ตำแหน่งขาของไมโครคอนโทรลเลอร์ ATmega328

3.1.1.5 ระบบแจ้งเตือน

ระบบแจ้งเตือนประกอบด้วยวงจรแจ้งเตือนด้วยเสียงพร้อมไฟกระพริบและการแสดงข้อความแจ้งเตือนบนจอ LCD โดยระบบแจ้งเตือนด้วยเสียงพร้อมไฟกระพริบประกอบด้วยลำโพง 12V 2ตัว และ LED 3 ตัวต่อขนานกันและถูกต่อเข้ากับขาเอาต์พุตของไอซี NE555 ซึ่งเป็นตัวสร้างสัญญาณพัลส์

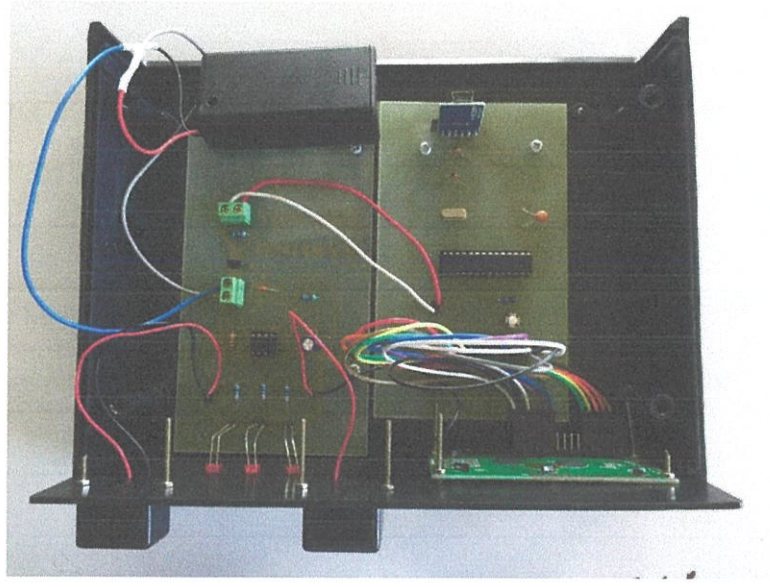
เมื่อมีสัญญาณจากไมโครคอนโทรลเลอร์ผ่านเข้ามาที่ขา Digital pin 13 หรือขาลำดับที่ 19 ของไมโครคอนโทรเลอร์ซึ่งต่อกับระบบแจ้งเตือน ไอซี NE555 จะส่งสัญญาณพัลส์ทำให้เกิดเสียงและไฟกระพริบเป็นจังหวะเพื่อทำการแจ้งเตือน นอกจากนั้นยังมี LCD ที่กับไมโครคอนโทรลเลอร์ซึ่งในภาวะปกติจะแสดงข้อความ “Normal Condition” และจะเปลี่ยนเป็น “Drowsiness Detect” เมื่อระบบแจ้งเตือนถูกเปิดขึ้นซึ่งการเชื่อต่อระหว่าง LCD กับไมโครคอนโทรเลอร์แสดงดังตารางที่ 3.2



รูปที่ 3.7 แผนผังวงจรระบบแจ้งเตือน

ตารางที่ 3.2 แสดงการเชื่อต่อระหว่าง LCD 16*2 กับไมโครคอนโทรเลอร์

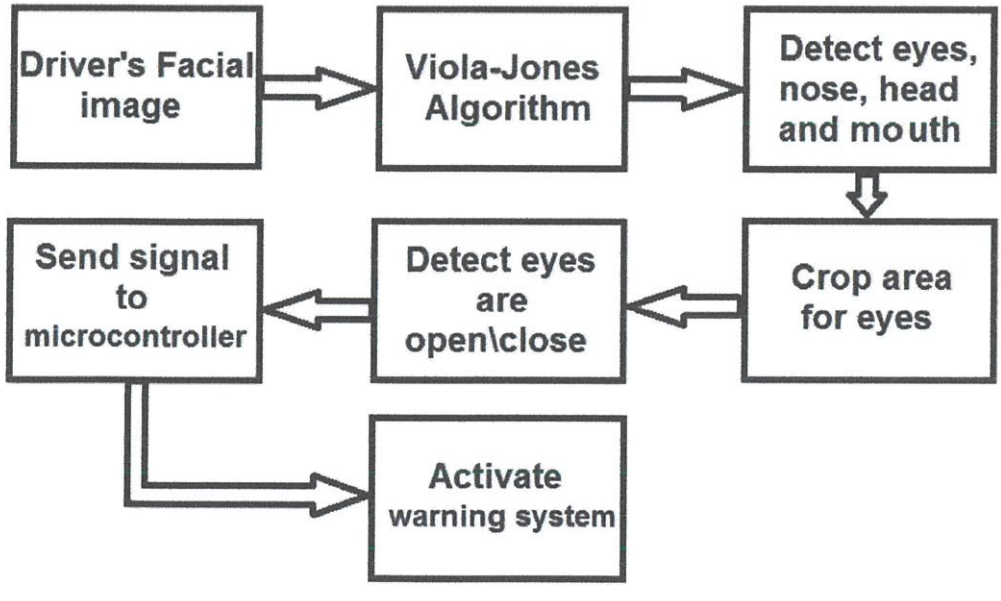
VR 10 kΩ	LCD	Microcontroller (ATmega328)
GND	VSS/GND	GND
VCC	VDD	+5 VDC
Signal	VO/VEE	-
-	RS	Digital pin 12
-	RW	GND
-	E/EN	Digital pin 11
-	DB4	Digital pin 4
-	DB5	Digital pin 5
-	DB6	Digital pin 6
-	DB7	Digital pin 7
-	A	+5 VDC
-	K	GND



รูปที่ 3.8 ระบบแจ้งเตือนเมื่อประกอบเสร็จแล้ว

3.1.2 ซอฟแวร์

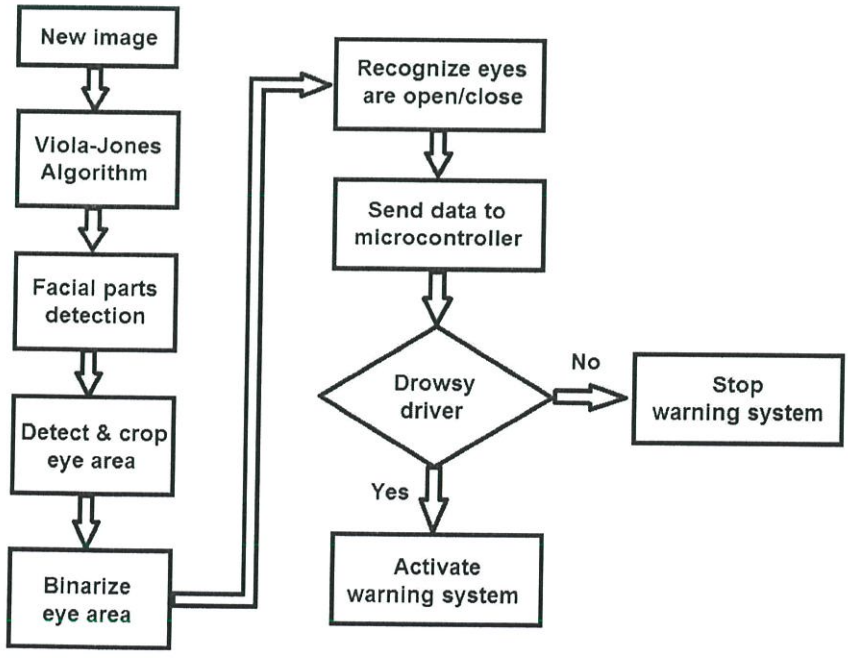
ส่วนของซอฟแวร์จะเป็นอัลกอลิทึมที่ใช้ในการตรวจจับส่วนต่างๆของใบหน้าแล้วนำมาทำการประมวลผลว่าผู้ขับรถมีการการง่วงนอนหรือไม่รวมถึงชุดคำสั่งที่ใช้ในการควบคุมระบบแจ้งเตือนระบบแจ้งเตือน



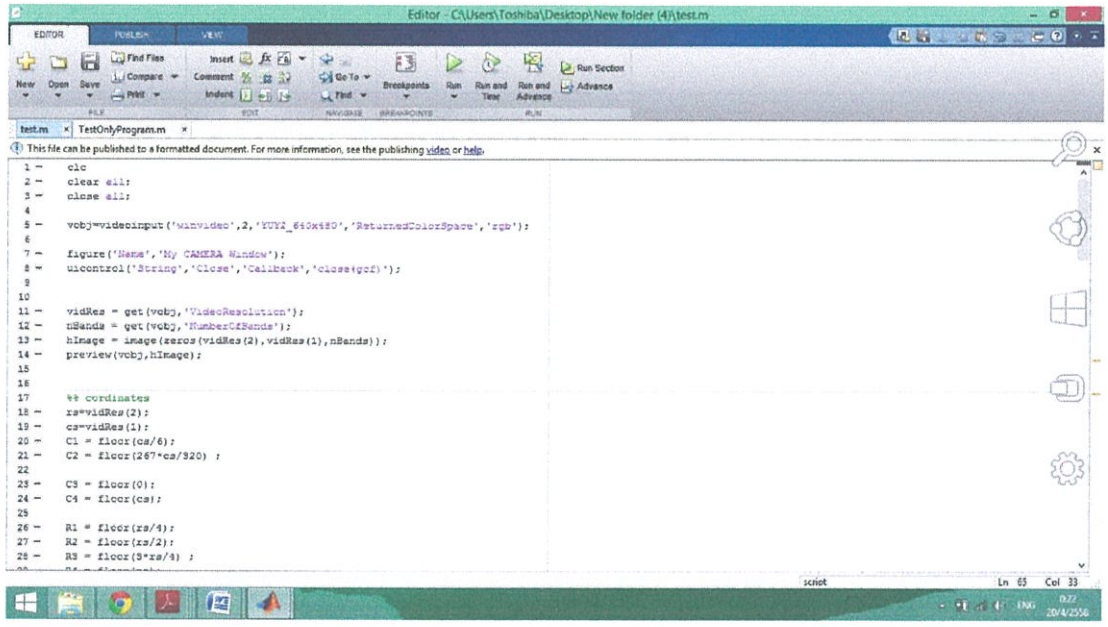
รูปที่ 3.9 แผนผังแสดงการทำงานของซอฟแวร์

การทำงานของซอฟแวร์สามารถอธิบายเป็นลำดับต่างๆดังนี้

- รับภาพจากกล้อง
- ใช้อัลกอลิทึม Viola-Jones
- ตรวจจับส่วนต่างๆของใบหน้าซึ่งประกอบด้วยตา จมูกและปาก
- แปลงภาพบริเวณดวงตาเป็นภาพไบนารี่
- ทำการจำแนกว่าดวงตาปิดหรือเปิด
- ส่งข้อมูลไปยังไมโครคอนโทรลเลอร์
- ประเมินว่าผู้ขับรถมีอาการง่วงนอนหรือไม่
- เปิดระบบแจ้งเตือนเมื่อประเมินว่าผู้ขับรถมีอาการง่วงนอน
- ปิดระบบแจ้งเตือนเมื่ออยู่ในสภาวะปกติ



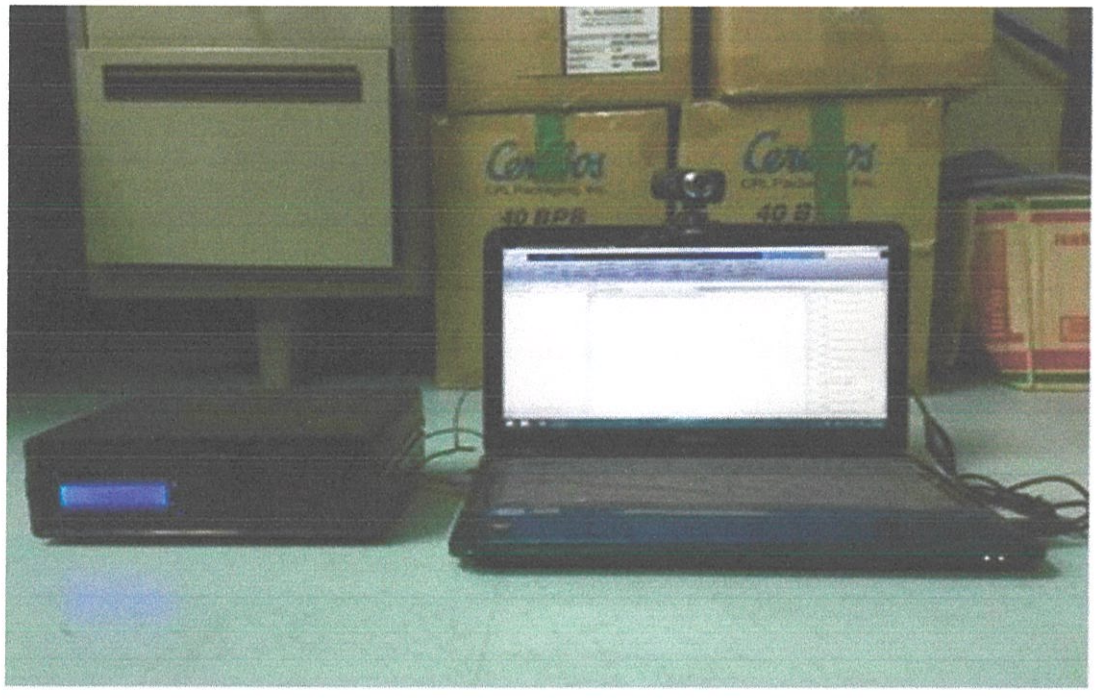
รูปที่ 3.10 ผังงานแสดงการทำงานของระบบ



รูปที่ 3.11 การเขียนโปรแกรมใน MATLAB

3.2 ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถที่พร้อมใช้งาน

เมื่อนำส่วนของฮาร์ดแวร์และซอฟต์แวร์แวร์ที่ทำการทดสอบในส่วนต่างๆเรียบร้อยแล้วก็จะได้ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถที่พร้อมใช้งานซึ่งแสดงดังรูปที่ 3.12



รูปที่ 3.12 ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถที่พร้อมใช้งาน

บทที่ 4

วิธีการและผลการทดสอบ

ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถถูกทดสอบโดยการนำระบบไปใช้งานในสภาวะแวดล้อมและกรณีที่ต่างกันโดยในขั้นตอนการตรวจจับบริเวณดวงตาในภาวะแสงที่แตกต่างกันนั้น โปรแกรมอาจมีความผิดพลาดเกิดขึ้นได้จึงอาจต้องมีการกำหนดค่าเทรชโชด้วยตนเอง โดยในโครงงานนี้จะใช้ค่าเทรชโชในช่วงตั้งแต่ 0 ถึง 30 ในการทำการทดสอบและ

การทดสอบในทุกภาวะที่แสดงในโครงงานนี้ถูกกำหนดให้มีสภาพแวดล้อมที่เหมือนกันโดยทำการทดสอบภายในห้องปิดขนาด 20 ตารางเมตร ใช้หลอดไฟนีออน 36 W 1 หลอดเป็นแหล่งกำเนิดแสง ยกเว้นในภาวะแสงน้อยที่ทำการปิดไฟ นอกจากนั้นยังกำหนดนิยามของคำต่างๆดังนี้

ภาวะปกติ คือ ภาวะที่ผู้ขับรถลืมตาหรือมีการกระพริบตา 15 - 20 ครั้งต่อนาทีซึ่งเป็นอัตราการกระพริบตาของคนทั่วไปในภาวะปกติโดยดวงตาของผู้ขับรถต้องอยู่ในบริเวณที่กำหนดไว้

ภาวะง่วงนอน คือ ภาวะที่ผู้ขับรถหลับตาเป็นระยะเวลานานเกินกว่า 3 วินาทีหรือมีการเคลื่อนที่ออกนอกบริเวณที่กำหนดไว้

ในการทดสอบแต่ละภาวะได้ทำการทดสอบ 10 ครั้งทั้งตอนเปิดตาและหลับตาและนำมาแปลงเป็นเปอเซนต์เพื่อหาประสิทธิภาพของระบบ

```
58 -    cnt=0;
59
60 -    shapeInserter = vision.ShapeInserter('BorderColor','Custom','CustomBorderColor',[255 255 0]);
61
62 -    EyeDetector1 = vision.CascadeObjectDetector('EyePairBig');
63 -    NoseDetector = vision.CascadeObjectDetector('Nose');
64 -    MouthDetector1 = vision.CascadeObjectDetector('Mouth');
65 -    EyeDetector1.MergeThreshold = 7;
66 -    [illegible]eady
67
68 -    s=serial('COM4','BAUD', 9600)
69 -    fopen(s);
70
```

ส่วนที่ใช้กำหนดค่าเทรชโช

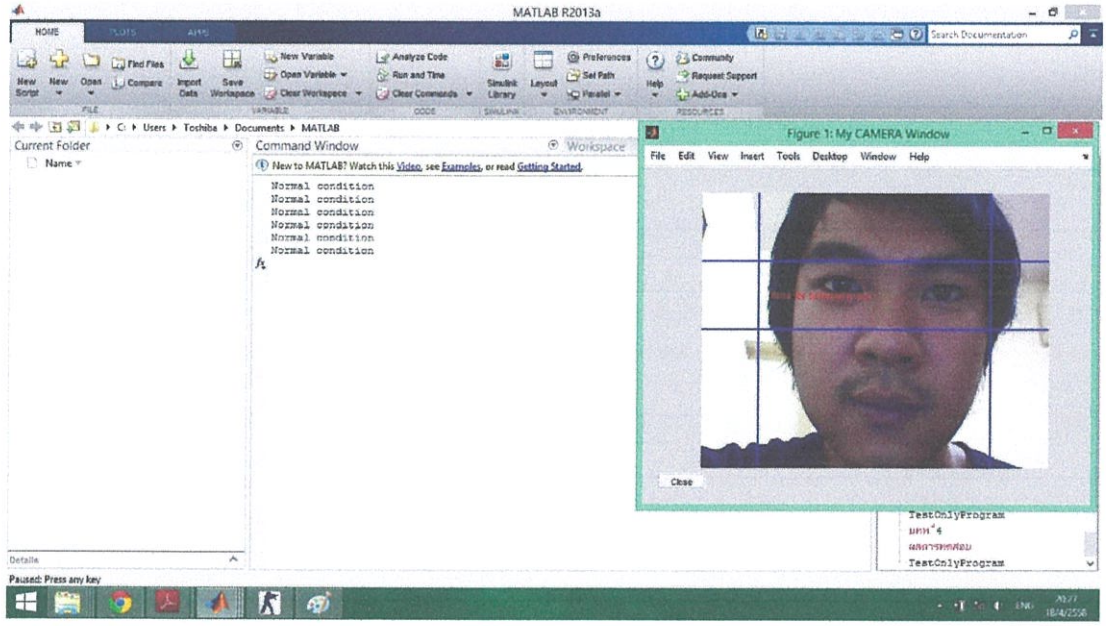
รูปที่ 4.1 แสดงส่วนที่ใช้กำหนดค่าเทรชโช

4.1 การทดสอบภาวะแสงปกติ

ทำการทดสอบโดยการนำระบบมาใช้งานในเวลากลางวันซึ่งเป็นช่วงเวลาที่คาดว่าระบบจะถูกใช้งานมากที่สุด พบว่าระบบมีความสามารถในการตรวจจับและแจ้งเตือนได้ดีเมื่อผู้ขับรถมีอาการง่วงนอนที่ค่าเทรชโชในช่วง 15 – 25 โดยมีประสิทธิภาพถึง 80% ผลการทดลองแสดงดังตารางที่ 4.1

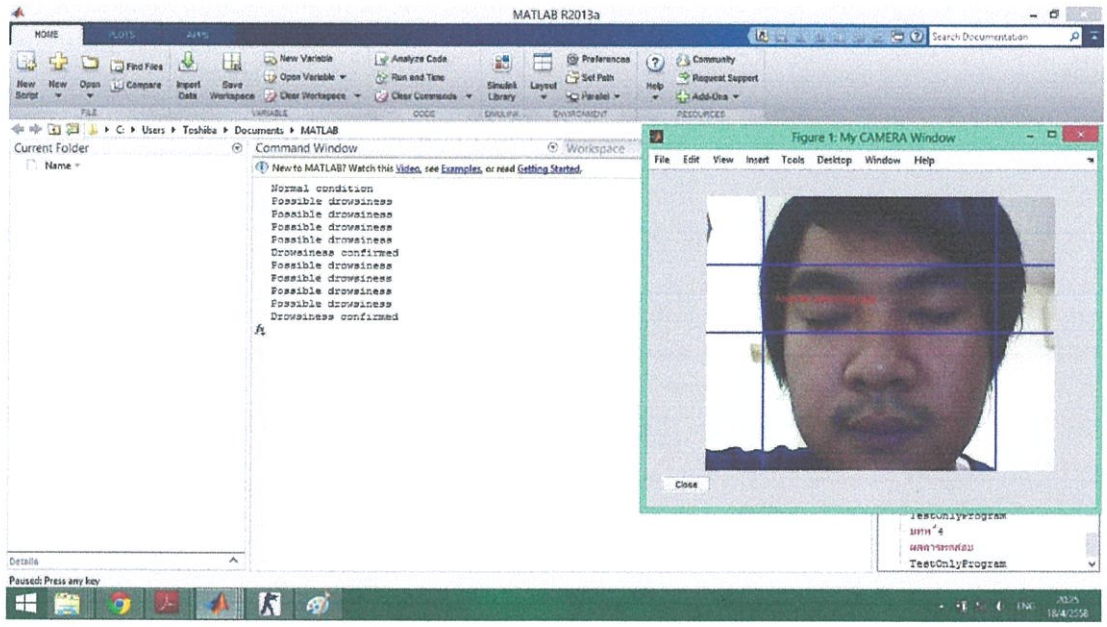
ตารางที่ 4.1 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงปกติโดยใช้ค่าเทรชโช 20

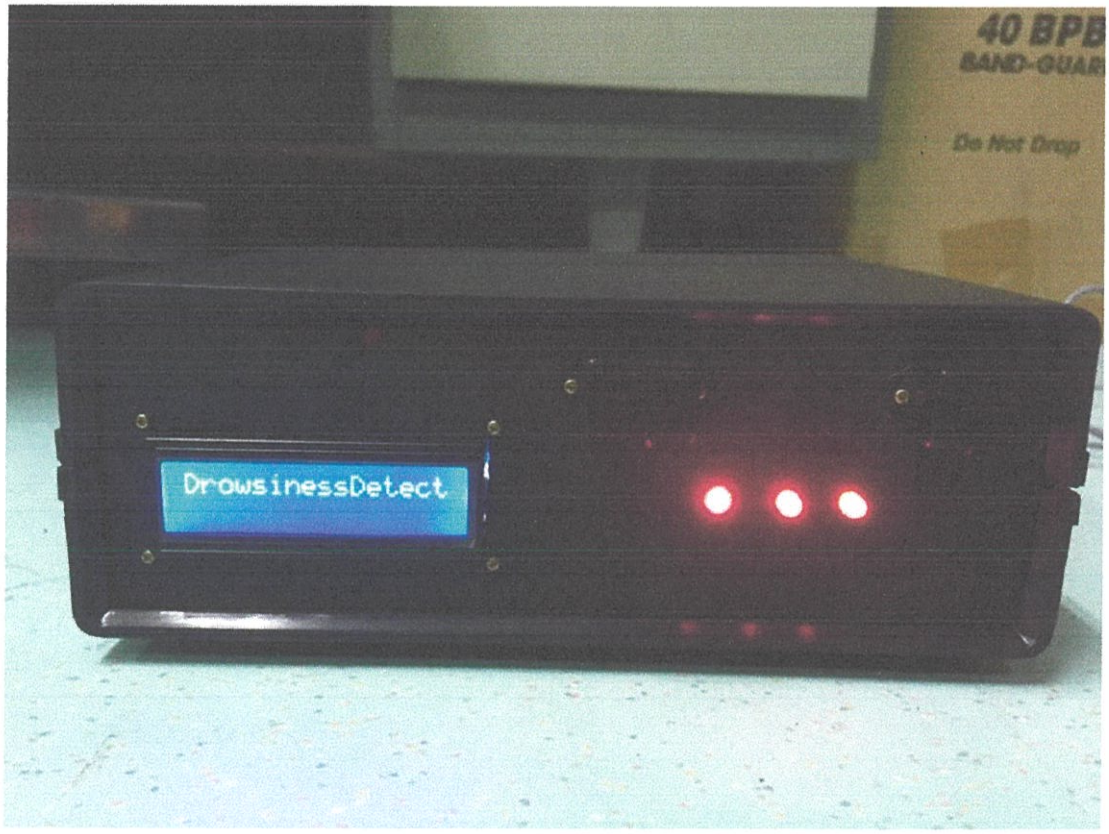
ครั้งที่	เปิดตา	หลับตา
1	✓	✓
2	✓	✓
3	✓	✓
4	✓	✓
5	✓	✗
6	✓	✓
7	✓	✓
8	✓	✓
9	✓	✗
10	✓	✓


MATLAB R2013a
Figure 1: My CAMERA Window
Command Window
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Close
TestOnlyProgram
TestOnlyProgram

Normal Condition
40 BPB
Do Not Drop

a)





b)

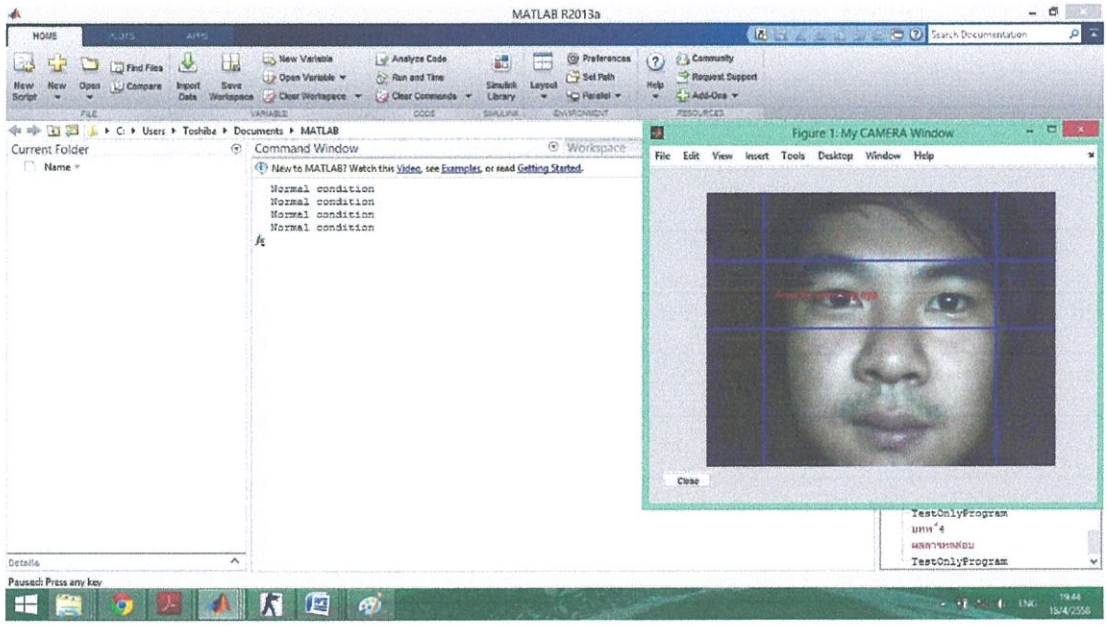
รูปที่ 4.2 การทดสอบในภาวะแสงปกติ a) เมื่ออยู่ในภาวะปกติ b) เมื่ออยู่ในภาวะง่วงนอน

4.2 การทดสอบภาวะแสงน้อย

เมื่อนำระบบมาใช้งานในภาวะแสงน้อยหรือเวลากลางคืนพบว่าระบบมีความคลาดเคลื่อนในการตรวจจับและแจ้งเตือนค่อนข้างมากในทุกค่าเทรชโช โดยมีความสามารถในการตรวจจับและแจ้งเตือนลดลงเหลือเพียง 40% ผลการทดลองแสดงดังตารางที่ 4.2

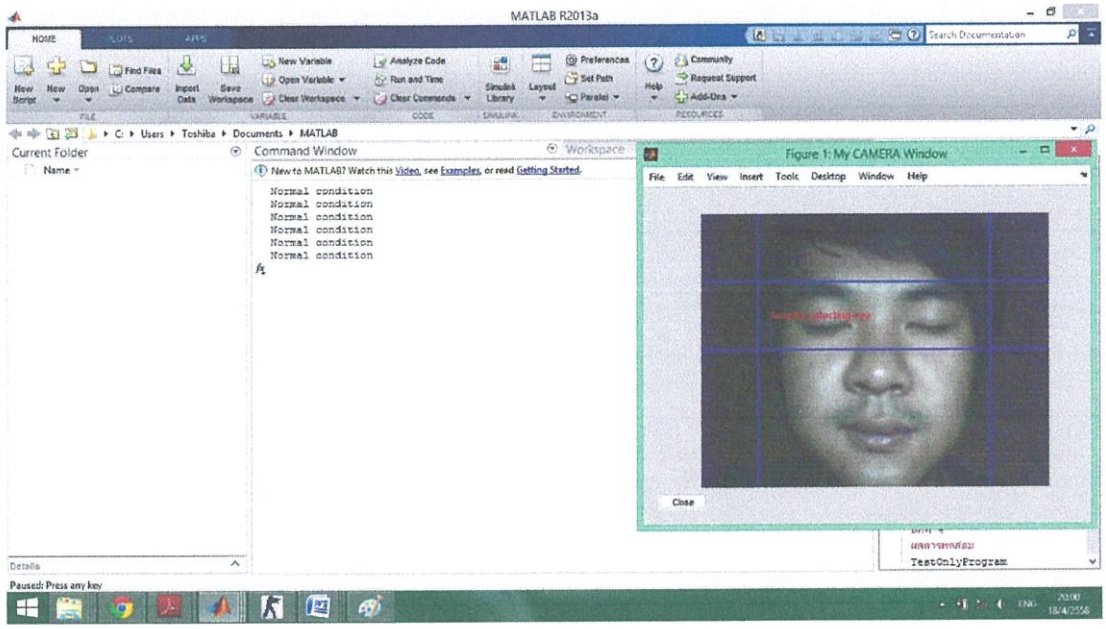
ตารางที่ 4.2 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงน้อย โดยใช้ค่าเทรชโช 20

ครั้งที่	เปิดตา	หลับตา
1	✓	✓
2	✓	✗
3	✓	✗
4	✓	✗
5	✓	✓
6	✓	✗
7	✓	✗
8	✓	✓
9	✓	✗
10	✓	✓


MATLAB R2013a
Command Window
Normal condition
Normal condition
Normal condition
Normal condition
Figure 1: My CAMERA Window
Close
Paused: Press any key

Normal Condition
40 BPB
Do Not Drop

a)



b)

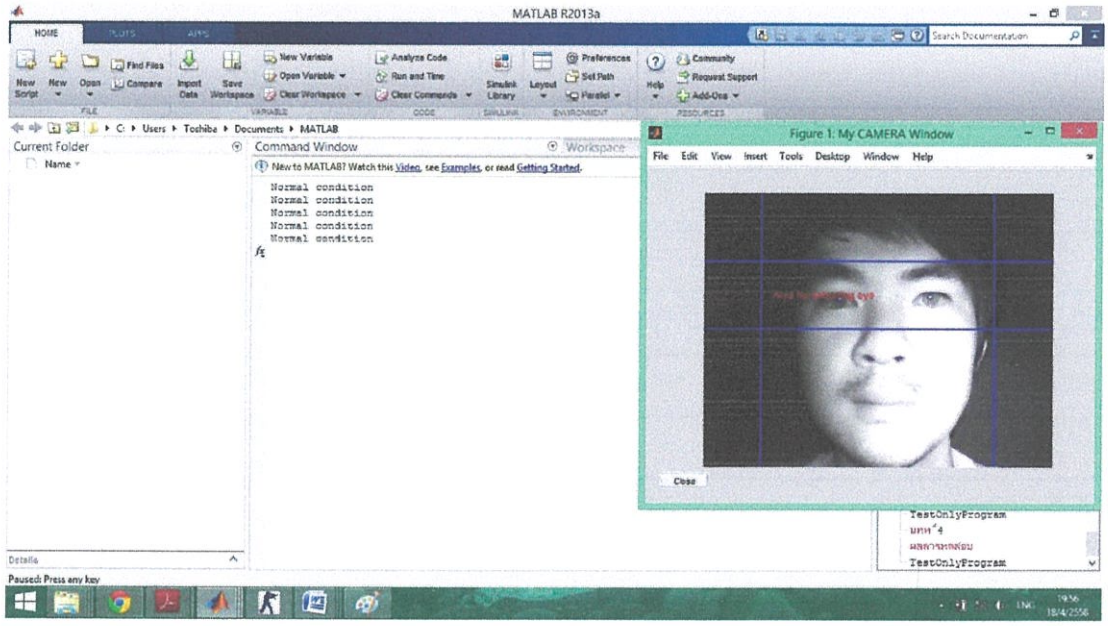
รูปที่ 4.3 การทดสอบในภาวะน้อย a) เมื่ออยู่ในภาวะปกติ b) เมื่ออยู่ในภาวะง่วงนอน

4.3 การทดสอบภาวะแสงน้อยร่วมกับแสงอินฟราเรด

ทำการทดสอบในเวลากลางคืนเหมือนในกรณีที่ 4.2 แต่ใช้แสงอินฟราเรดที่ติดอยู่กับกล้องส่องไปบริเวณดวงตาของผู้ขับรถและกำหนดค่าเทรชโชทประมาณ 5 พบว่าช่วยให้ระบบมีความสามารถในการตรวจจับและแจ้งเตือนเพิ่มขึ้นเป็น 80% ผลการทดลองแสดงดังตารางที่ 4.3

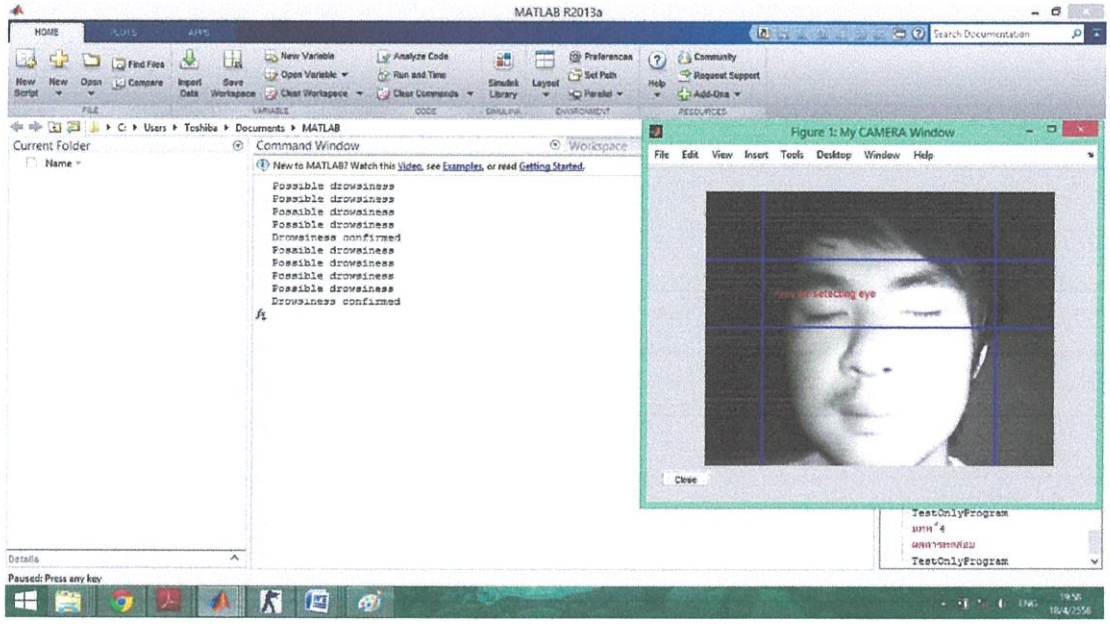
ตารางที่ 4.3 ผลการทดสอบการตรวจจับดวงตาในภาวะแสงน้อยร่วมกับอินฟราเรด โดยใช้ค่าเทรชโช 5

ครั้งที่	เปิดตา	หลับตา
1	✓	✓
2	✓	✓
3	✓	✓
4	✓	x
5	✓	✓
6	✓	✓
7	✓	✓
8	✓	✓
9	✓	✓
10	✓	x


MATLAB R2013a
Command Window
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Figure 1: My CAMERA Window
File Edit View Insert Tools Desktop Window Help
Close
TestOnlyProgram
TestOnlyProgram

Normal Condition
40 BPB
Do Not Drop

a)



b)

รูปที่ 4.4 การทดสอบในภาวะน้อยร่วมกับแสงอินฟราเรด

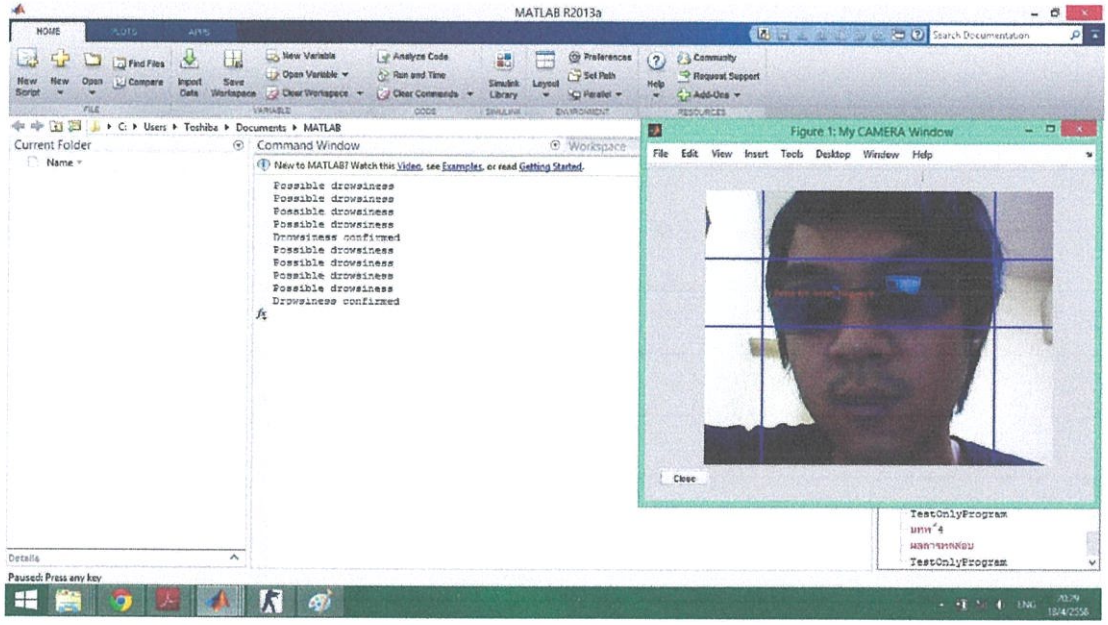
a) เมื่ออยู่ในภาวะปกติ b) เมื่ออยู่ในภาวะง่วงนอน

4.4 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดด

ใช้การทดสอบในเวลากลางวันเมื่อผู้ขับรถสวมแว่นกันแดด พบว่าระบบทำงานถูกต้องเพียง 10% เท่านั้นในทุกค่าเทรชโชเนื่องจากไม่สามารถระบุตำแหน่งของดวงตาได้ ผลการทดลองแสดงดังตารางที่ 4.4

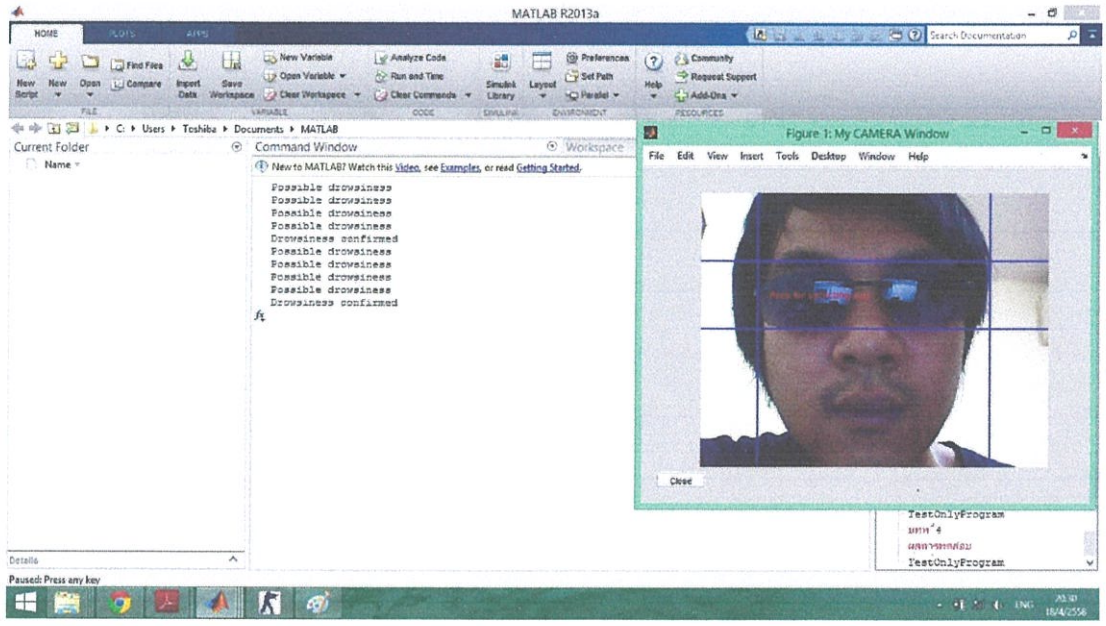
ตารางที่ 4.4 ผลการทดสอบการตรวจจับดวงตาในกรณีที่ผู้ขับรถสวมแว่นตากันแดด โดยใช้ค่าเทรชโช 20

ครั้งที่	เปิดตา	หลับตา
1	✗	✓
2	✗	✓
3	✗	✓
4	✗	✓
5	✗	✓
6	✓	✓
7	✗	✓
8	✗	✓
9	✗	✓
10	✗	✓


MATLAB R2013a
Command Window
Possible drowsiness
Possible drowsiness
Possible drowsiness
Possible drowsiness
Drowsiness confirmed
Possible drowsiness
Possible drowsiness
Possible drowsiness
Possible drowsiness
Drowsiness confirmed
Figure 1: My CAMERA Window
File Edit View Insert Tools Desktop Window Help
Close
TestOnlyProgram
TestOnlyProgram
Paused: Press any key

DrowsinessDetect

a)



b)

รูปที่ 4.5 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดด

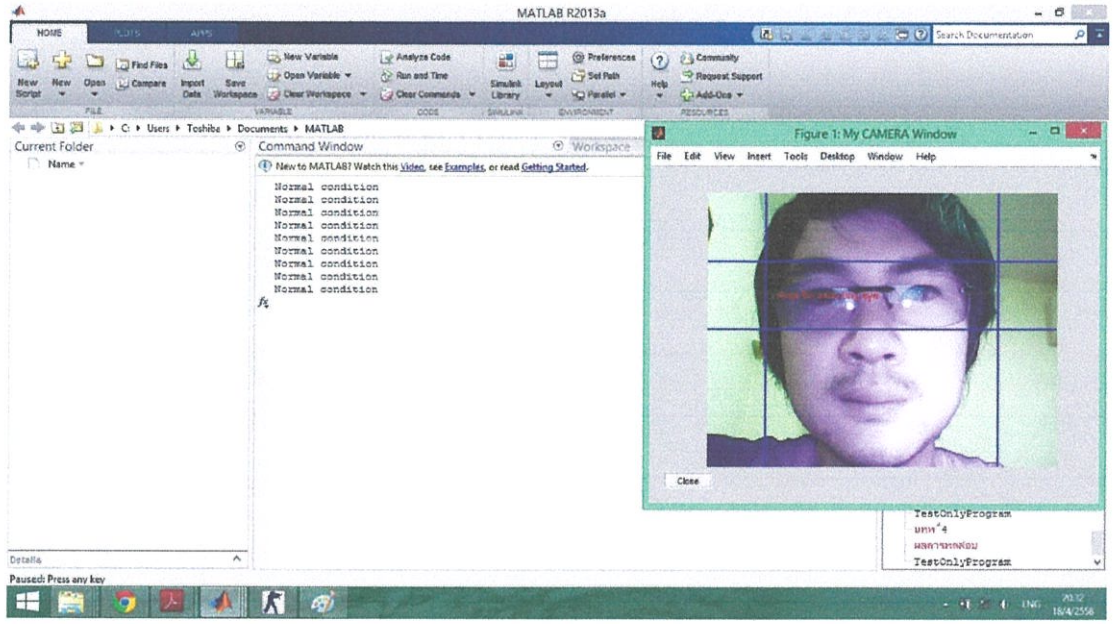
a) เมื่ออยู่ในภาวะปกติ b) เมื่ออยู่ในภาวะง่วงนอน

4.5 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดดร่วมกับแสงอินฟราเรด

ทำการทดสอบโดยผู้ขับรถสวมแว่นตากันแดดแต่ใช้แสงอินฟราเรดที่ติดอยู่กับกล้องส่องไปบริเวณดวงตาของผู้ขับรถและใช้ค่าเทรชโชประมาณ 1 พบว่าระบบสามารถกลับมาทำงานได้ดีโดยมีความสามารถในการตรวจจับและแจ้งเตือนได้ถึง 80% ผลการทดลองแสดงดังตารางที่ 4.5

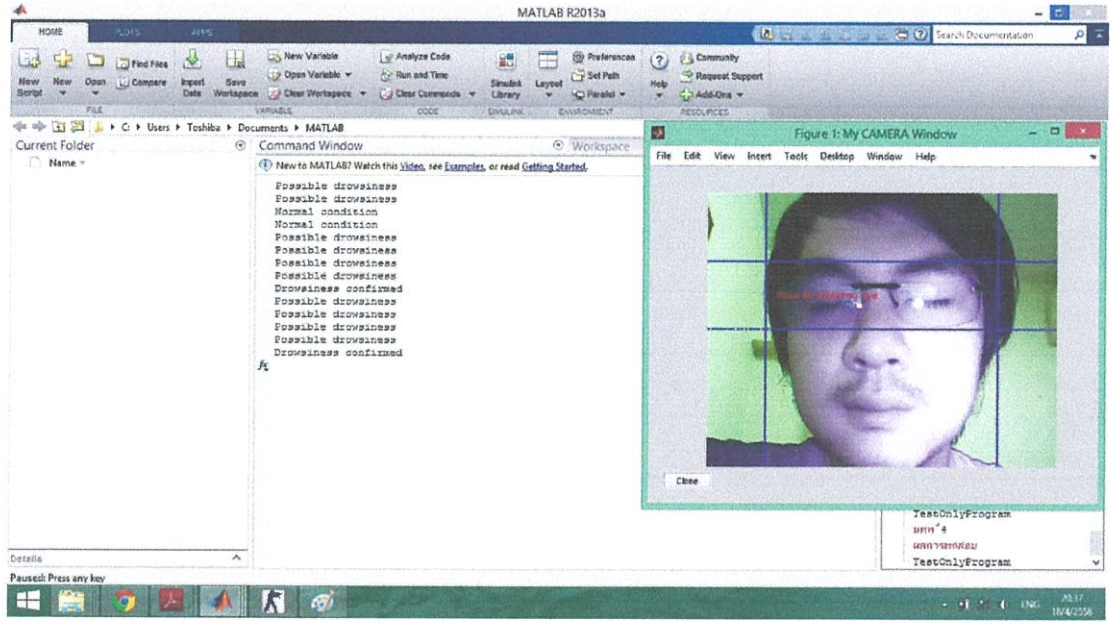
ตารางที่ 4.5 ผลการทดสอบการตรวจจับดวงตาในกรณีที่ผู้ขับรถสวมแว่นตากันแดดร่วมกับอินฟราเรด โดยใช้ค่าเทรชโช 1

ครั้งที่	เปิดตา	หลับตา
1	✓	✓
2	✓	✓
3	✓	✓
4	✓	✗
5	✓	✓
6	✓	✓
7	✓	✓
8	✓	✓
9	✗	✓
10	✓	✓


MATLAB R2013a
Figure 1: My CAMERA Window
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
Normal condition
TestOnlyProgram
TestOnlyProgram

Normal Condition

a)



รูปที่ 4.6 การทดสอบกรณีที่ผู้ขับรถสวมแว่นตากันแดดร่วมกับแสงอินฟราเรด

a) เมื่ออยู่ในภาวะปกติ b) เมื่ออยู่ในภาวะง่วงนอน

บทที่ 5

สรุปผลการทดสอบและข้อเสนอแนะ

ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถถูกพัฒนาขึ้นโดยใช้การประมวลผลภาพที่ได้รับมาจากกล้องโดยใช้โปรแกรม MATLAB ซึ่งมีฟักก์ชั่นในการตรวจจับส่วนต่างๆของใบหน้าแล้วนำมาประเมินว่าผู้ขับรถมีอาการง่วงนอนซึ่งในที่นี้คือการหลับตาเป็นระยะเวลานานหรือเคลื่อนตัวออกจากบริเวณที่กำหนด ซึ่งหากระบบประเมินว่าผู้ขับรถมีอาการดังกล่าวเกิดขึ้นจะทำการส่งข้อมูลผ่านโมดูล FTDI 232 ให้ไมโครคอนโทรลเลอร์ทำการเปิดระบบแจ้งเตือนซึ่งประกอบด้วยแสงกระพริบจากหลอด LED และเสียงจากลำโพงเพื่อปลุกคนขับให้ตื่นขึ้น นอกจากนั้นยังแสดงข้อความแจ้งเตือนบนจอ LCD อีกด้วย

การทำงานของระบบขึ้นอยู่กับสภาวะแวดล้อมและแสงในขณะใช้งาน เพื่อความแม่นยำของระบบในการตรวจจับดวงตาจึงอาจต้องทำการปรับค่าเทรชโชให้เหมาะสมโดยมีข้อสังเกตว่าในภาวะแสงน้อยควรปรับให้มีค่าต่ำและปรับให้มีค่าสูงเมื่อใช้งานในภาวะที่มีแสงมากแต่ทั้งนี้ควรใช้ค่าเทรชโชในช่วง 0 ถึง 30 นอกจากนี้ยังพบว่าในกรณีที่มีแสงน้อย การใช้การแสงอินฟราเรดเข้ามาช่วยจะทำให้ประสิทธิภาพในการตรวจจับและแจ้งเตือนเป็นได้ดีขึ้นเช่นเดียวกับกรณีที่ผู้ขับรถสวมแว่นตากันแดดซึ่งในการทำงานของระบบโดยปราศจากแสงอินฟราเรดจะไม่สามารถตรวจจับบริเวณดวงตาได้จึงต้องใช้แสงอินฟราเรดเข้าช่วยในส่องผ่านแว่นกันแดด ซึ่งโดยรวมแล้วภายใต้การกำหนดค่าเทรชโชที่เหมาะสม ระบบจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถสามารถทำงานได้อย่างมีประสิทธิภาพและมีความแม่นยำถึง 80%

เนื่องจากโครงงานนี้เป็นเพียงการจำลองการตรวจจับและแจ้งเตือนคนง่วงขณะขับรถเท่านั้น ในการนำไปใช้งานจริงยังประสบปัญหาหลายประการ เช่น สภาพแวดล้อมและแสงที่เปลี่ยนไปตามช่วงเวลาและบริเวณของรถที่แล่นไปทำให้การปรับค่าเทรชโชไม่สามารถทำได้ นอกจากนั้นการตรวจจับและแจ้งเตือนโดยใช้บริเวณดวงตาอาจไม่เพียงพอในการใช้งานจริงในชีวิตในการหลีกเลี่ยงอุบัติเหตุที่จะเกิดขึ้น ซึ่งในอนาคตหากสามารถสร้างระบบที่มีการเปลี่ยนแปลงค่าเทรชโชอัตโนมัติตามปริมาณของแสงและมีการพัฒนาระบบที่มีการตรวจว่ารถยังคงแล่นอยู่ในเลนถนนมาทำงานร่วมกันจะช่วยเพิ่มประสิทธิภาพของระบบตรวจจับและแจ้งเตือนคนง่วงขณะขับรถให้ดียิ่งขึ้นจนสามารถนำไปใช้ในชีวิตจริงซึ่งจะสามารถลดปริมาณอุบัติเหตุที่เกิดขึ้นบนท้องถนนให้น้อยลงได้

บรรณานุกรม

1. 2554. **การรู้จำใบหน้า (Face Recognition)**. [Online]. Available : http://www.bantronix.com/2011/10/face-recognition.html (04/03/2558)

2. 2558. Viola-Jones object detection framework. [Online]. Available : http://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

3. ชูชาติ ปิณฑวิรุจน์. 2552. **การประมวลผลภาพเชิงดิจิตอลขั้นสูงโดยใช้** MATLAB. พิมพ์ครั้งที่2. กรุงเทพ : งานบริการการเรียนการสอน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

4. ชูชาติ ปิณฑวิรุจน์. 2555. **การประมวลผลภาพดิจิตอลด้วย**MATLAB. พิมพ์ครั้งที่2. กรุงเทพ : งานบริการการเรียนการสอน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

5. Dhaval Pimplaskar, Dr. M.S. Nagmode, Atul Borkar. 2013. Real Time Eye Blinking Detection and Tracking Using Opencv. Journal of Engineering Research and Applications. Vol3.(Issue 5) : page 1780-1787

6. Mark Cataldi, Jesse Harvey, Dieter Laskowski. 2008. Drowsy Driver Warning System. RIT Department of Computer Engineering.

ภาคผนวก

ภาคผนวก ก

โปรแกรมตรวจจับคนง่วงใน MATLAB

```
clc
clear all;
close all;

vobj=videoinput('winvideo',2,'YUY2_640x480','ReturnedColorSpace','rgb');

figure('Name','My CAMERA Window');
uicontrol('String','Close','Callback','close(gcf)');


vidRes = get(vobj,'VideoResolution');
nBands = get(vobj,'NumberOfBands');
hImage = image(zeros(vidRes(2),vidRes(1),nBands));
preview(vobj,hImage);


%% cordinates
rs=vidRes(2);
cs=vidRes(1);
C1 = floor(cs/6);
C2 = floor(267*cs/320) ;

C3 = floor(0);
C4 = floor(cs);

R1 = floor(rs/4);
R2 = floor(rs/2);
R3 = floor(3*rs/4) ;
```

```
R4 = floor(rs);

x1=[C1 C1];
x2=[C2 C2];
x3=[C3 C3];
x4=[C4 C4];

x5=[C3 C4];

y1=[0 rs];
y2=[R1 R1];
y3=[R2 R2];
y4=[R3 R3];
y5=[R4 R4];

line(x1,y1,'Color','g','LineWidth',2)
line(x2,y1,'Color','g','LineWidth',2)

line(x5,y2,'Color','g','LineWidth',2)
line(x5,y3,'Color','g','LineWidth',2)

text2=text(19*cs/96,3*rs/8,'Area for setecting eye','Color','R');

%% initialize flags
FlagForeHead=0;
FlagEyes=0;
FlagNose=0;
FlagMouth=0;

cnt=0;
```

```
shapeInserter = vision.ShapeInserter('BorderColor','Custom','CustomBorderColor',[255
255 0]);

EyeDetector1 = vision.CascadeObjectDetector('EyePairBig');
NoseDetector = vision.CascadeObjectDetector('Nose');
MouthDetector1 = vision.CascadeObjectDetector('Mouth');
EyeDetector1.MergeThreshold = 5;
pause %% press any key while you are ready

s=serial('COM4','BAUD', 9600)
fopen(s);

for i=1:200
    I2=getsnapshot(vobj);
    pause(0.5)
    FirstSeg=imcrop(I2,[C1 0 C2-C1 R1]);

    BlackDetect=(FirstSeg(:,:,1)<70)&(FirstSeg(:,:,2)<70)&(FirstSeg(:,:,3)<70);
    BW1 = imfill(BlackDetect,'holes');
    BW2 = bwareaopen(BW1,2000);

    [Mat1 Nr]=bwlabel(BW2);
    if Nr ~= 0
        FlagForeHead=1;
    else
        FlagForeHead=0;
    end


    SecondSegment=imcrop(I2,[C1 R1 C2-C1 R2-R1]);
    bbox_eye1 = step(EyeDetector1,SecondSegment);
    I_Eye = step(shapeInserter,SecondSegment,int32(bbox_eye1));
```

```
 if isempty(bbox_eye1)~=1
     FlagEyes=1;
 else
     FlagEyes=0;
 end

 ThirdSegment=imcrop(I2,[C1 R2 C2-C1 R3-R2]);
 bbox_Nose1 = step(NoseDetector,ThirdSegment);
 I_Nose = step(shapeInserter,ThirdSegment,int32(bbox_Nose1));

 if isempty(bbox_Nose1)~=1
     FlagNose=1;
 else
     FlagNose=0;
 end

 FourthSegment=imcrop(I2,[C1 R3 C2-C1 R4-R3]);
 bbox_Mouth1 = step(MouthDetector1,FourthSegment);
 I_Mouth = step(shapeInserter,FourthSegment,int32(bbox_Mouth1));
 if isempty(bbox_Mouth1)~=1
     FlagMouth=1;
 else
     FlagMouth=0;
 end

 if((FlagEyes==1))
     disp('Normal condition')
     cnt=0;
     fprintf(s,101);
 else
     disp('Possible drowsiness')
```

```
        cnt=cnt+1;

    end
    if cnt>2
        cnt=0;
        disp('Drowsiness confirmed')
        pause(1);
        fprintf(s,100);
    end
end
```

ภาคผนวก ข

โปรแกรมควบคุมระบบแจ้งเตือนใน ATmega328P-PU

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 4, 5, 6, 7);

const int ledpin=13;

int recValue;


void setup()

{

lcd.begin(16, 2);

Serial.begin(9600);

pinMode(13, OUTPUT);

}


void loop()

{

 if(Serial.available()>0)

 {

   recValue=Serial.read();
```

```
  if  (recValue == 100)
   {
     digitalWrite(13, HIGH);
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("DrowsinessDetect");
     }
      if(recValue == 101)
       {
       digitalWrite(13, LOW);
       lcd.clear();
       lcd.setCursor(0,0);
       lcd.print("Normal Condition");
       }
  }
}
```