ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์

Driving Assistant and Warning System on Android

ปฐวี นาคบุตร
พีรภัทร พาปาน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์

Driving Assistant and Warning System on Android

ปฐวี นาคบุตร

พีรภัทร พาปาน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญานิพนธ์ ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์

DRIVING ASSISTANT AND WARNING SYSTEM ON ANDROID

ผู้จัดทำ

1. นายปฐวี นาคบุตร รหัสนักศึกษา 54010768

2. นายพีรภัทร พาปาน รหัสนักศึกษา 54010940

ชมพูนุท จินจาคาม อาจารย์ที่ปรึกษา

(ดร.ชมพูนุท จินจาคาม)

อาจารย์ที่ปรึกษา

(ผศ.เจริญ วงษ์ชุ่มเย็น)

อาจารย์ที่ปรึกษา

(ดร.ปกรณ์ วัฒนจตุรพร)

ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์

นายปฐวี	นาคบุตร	54010768
นายพีรภัทร	พาปาน	54010940
ดร.ชมพูนุท	จินจาคาม	อาจารย์ที่ปรึกษา
ผศ.เจริญ	วงษ์ชุ่มเย็น	อาจารย์ที่ปรึกษาร่วม
ดร.ปกรณ์	วัฒนจตุรพร	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2557

บทคัดย่อ

ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ เป็นแอพพลิเคชันที่นำเทคโนโลยีการประมวลผลภาพมาประยุกต์ใช้งาน ผู้ใช้งานสามารถติดตั้งสมาร์ทโฟนที่เปิดแอพพลิเคชันไว้ที่บริเวณส่วนกลางของคอนโซลหน้า ระบบทำการจับภาพจากกล้องของอุปกรณ์ที่ติดตั้งแอพพลิเคชัน จากนั้นนำภาพมาประมวลผลตรวจหาเส้นเลนของถนน และตรวจจับวัตถุที่อยู่ข้างหน้ารถ จากนั้นแสดงเส้นเลนของถนน และแสดงกรอบสี่เหลี่ยมล้อมรอบวัตถุที่ตรวจพบหน้ารถ หากรถออกนอกเลน หรือวัตถุที่อยู่ข้างหน้าเข้ามาใกล้ระยะที่กำหนด ระบบจะทำการส่งสัญญาณเตือนไปยังผู้ขับ ให้ผู้ขับทราบถึงอันตราย และได้สติกลับมา การประมวลผลของแอพพลิเคชันจะทำงานต่อเนื่องจนกว่าผู้ใช้จะปิดการใช้งาน

Driving Assistant and Warning System on Android

Mr. Patthawee	Narkbutr	54010768
Mr. Pirapat	Paparn	54010940
Dr. Chompoonuch	Jinjakam	Advisor
Asst.Prof. Charoen	Vongchumyen	Co-Advisor
Dr. Pakorn	Watanachaturaporn	Co-Advisor

Academic Year 2014

Abstract

Driving assistant and warning system on Android is an application that applies image processing technology. The user sets up smart phone that runs this application at car's console, the system takes video from smart phone's camera, then system processes and analyzes lane's lines and object that appear in front of the car, after that system show virtual lines for lane's lines and box around the object in front of the car. If car is departing from lane or object ahead is in define range, system will warning to make user aware of the dangers. The application processing continues until user close it.

กิตติกรรมประกาศ

รายงานปริญญานิพนธ์ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ ฉบับนี้จะสำเร็จได้ด้วยความอนุเคราะห์จาก ดร.ชมพูนุท จินจาคาม อาจารย์ที่ปรึกษาโครงงาน ที่ให้การปรึกษา และแนะนำทั้งการออกแบบ พัฒนาแอพพลิเคชั่น ตรวจสอบ ปรับปรุง แก้ไขรูปเล่มรายงานจนเสร็จสิ้นสมบูรณ์ ผศ.เจริญ วงษ์ชุ่มเย็น และดร.ปกรณ์ วัฒนจตุรพร อาจารย์ที่ปรึกษาร่วม ที่ให้คำปรึกษาด้านการออกแบบ และพัฒนาแอพพลิเคชัน รวมถึงการวางแผนการดำเนินการ ขอขอบคุณดร.ชัยวัฒน์ หนูทอง ที่ให้คำปรึกษาด้านการเลือกใช้เครื่องมือ ที่เหมาะสมในการพัฒนา

ท้ายนี้ ผู้จัดทำหวังว่าปริญญานิพนธ์นี้จะเป็นประโยชน์ต่อผู้ที่มีความสนใจต่อไป

ปฐวี นาคบุตร

พีรภัทร พาปาน

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง หน้า

สารบัญภาพ

ภาพ หน้า

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันการคมนาคมส่วนใหญ่ มักใช้รถยนต์เป็นยานพาหนะหลักสำหรับการเดินทาง และสิ่งที่มักจะเกิดขึ้นมาพร้อมกับการใช้รถใช้ถนนก็คืออุบัติเหตุ อุบัติเหตุสร้างความเสียหายทั้งชีวิตและทรัพย์สินมากมาย สาเหตุของการเกิดอุบัติเหตุส่วนใหญ่ เกิดจากความประมาทของคนขับเอง ทั้งรีบร้อน หรือพักผ่อนไม่เพียงพอ จนง่วงและหลับในระหว่างขับรถ ทำให้รถยนต์รุ่นใหม่ๆ ที่ผลิตออกมา ล้วนเพิ่มฟีเจอร์ที่จะช่วยเสริมความปลอดภัยให้คนขับ อีกทั้งในปัจจุบันอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์นั้นเป็นที่นิยมในการใช้งานอย่างกว้างขวาง ทางคณะผู้วิจัยจึงได้เลือกอุปกรณ์แอนดรอยด์เป็นพื้นฐานในการพัฒนา ระบบสนับสนุน และแจ้งเตือนคนขับบนระบบปฏิบัติการแอนดรอยด์ขึ้น

โดยระบบจะใช้กล้องของอุปกรณ์แอนดรอยด์ในการตรวจจับตำแหน่งของรถในเลนถนน และวัตถุที่อยู่ข้างหน้ารถ แสดงเส้นเลนเสมือนบนหน้าจออุปกรณ์แอนดรอยด์ หากรถส่ายหรือเคลื่อนที่ออกจากเลน จะทำการส่งสัญญาณ แจ้งเตือนคนขับ ที่อาจกำลังง่วงนอนว่ากำลังอยู่ในอันตราย รถอาจจะหลุดออกจากถนน หรือเกิดอุบัติเหตุได้ และหากวัตถุที่อยู่หน้ารถเข้าใกล้รถในระยะ 10 เมตร จะทำการส่งสัญญาณแจ้งเตือนคนขับให้ทราบถึงอันตราย เพื่อป้องกันการขับชนวัตถุข้างหน้า อีกทั้งระบบนี้เป็นระบบติดตั้งเพิ่มเติมสามารถใช้ได้กับรถยนต์ทั่วไปโดยไม่ขึ้นอยู่กับรุ่นหรือยี่ห้อของรถยนต์

โครงงานวิจัยนี้เหมาะสำหรับผู้ใช้งานรถยนต์ทั่วไป ที่ต้องการเพิ่มความปลอดภัยในการขับขี่รถยนต์ โดยการแจ้งเตือนจากระบบ ทั้งยังช่วยให้ลดการเกิดอุบัติเหตุบนท้องถนนอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อเพิ่มความปลอดภัยให้กับผู้ขับขี่รถยนต์
2) เพื่อพัฒนาระบบสนับสนุนผู้ขับขี่ ที่สามารถใช้ได้อย่างทั่วถึง โดยไม่ต้องคำนึงถึงรุ่นของรถยนต์
3) เพื่อเพิ่มอรรถประโยชน์ในการทำงานให้กับอุปกรณ์แอนดรอยด์
4) เพื่อศึกษาและพัฒนาแอพพลิเคชันบนอุปกรณ์แอนดรอยด์
5) ลดอุบัติเหตุบนท้องถนน

1.3 ขอบเขตของโครงงาน

ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์นี้ จะทำหน้าที่ตรวจจับเส้นเลนบนถนน ตำแหน่งของรถบนเลน รวมถึงวัตถุที่อยู่ด้านหน้าของรถ แสดงเส้นเลนที่ตรวจจับได้ และหากรถยนต์ส่ายหรือ เคลื่อนที่ออกนอกเลน ที่ไม่ใช่การขับแซง หรือเลี้ยว หรือวัตถุที่อยู่ด้านหน้ารถเข้ามาใกล้ในระยะที่กำหนด จะทำการส่งสัญญาณเพื่อแจ้งเตือนผู้ขับขี่ให้ได้สติ รู้สึกถึงอันตราย และกลับมาควบคุมรถได้

การใช้งานผู้ใช้ทำการติดตั้งอุปกรณ์แอนดรอยด์ที่ติดตั้งแอพพลิเคชันของระบบไว้บนฐานตั้งที่วางไว้บริเวณคอนโซลข้างผู้ขับ ระบบจะทำงานโดยการตรวจจับตำแหน่งของรถ วัตถุที่อยู่ด้านหน้ารถ และเลนของถนน โดยใช้วีดีโอที่ได้ถ่ายได้จากกล้องของอุปกรณ์แอนดรอยด์ ระบบทำงานโดยการรับภาพจากกล้องมาประมวลผลตามเวลาจริงเพื่อตรวจจับเส้นเลนของถนน และตรวจจับวัตถุที่อยู่ด้านหน้ารถ เมื่อพบเส้นเลนของถนน จะทำการแสดงเส้นเลนเสมือนขึ้นมาบนหน้าจอของอุปกรณ์แอนดรอยด์ และทำการตรวจสอบตำแหน่งของรถว่าอยู่ในเลนหรือไม่ หากรถเคลื่อนออกจากเลนแบบผิดปกติ ที่ไม่ได้เกิดจากการแซง หรือเลี้ยวรถ หรือพบวัตถุที่อยู่ด้านหน้ารถอยู่ในระยะ 10 เมตร ระบบจะส่งสัญญาณเตือนไปยังผู้ขับ ระบบจะสามารถตรวจจับเลนของถนนได้ดีในสภาพอากาศที่แจ่มใส สภาพแวดล้อมที่มีแสงมาก แต่ข้อจำกัดของระบบคือไม่สามารถประมวลผลได้หากรถขับด้วยความเร็วที่สูงเกิน 60 กิโลเมตรต่อชั่วโมงหรือสภาพอากาศไม่ดี บนหน้าจอของอุปกรณ์แอนดรอยด์จะแสดงวีดีโอที่ถ่ายได้จากกล้อง และเส้นเลนเสมือนจริงที่ได้จากการวิเคราะห์ หากมีการเคลื่อนออกจากเลน จะมีข้อความแสดงบนหน้าจอ พร้อมกับส่งสัญญาณเตือน

1.3 วิธีการดำเนินการ

1) ศึกษาวิธีการที่ใช้ในการประมวลผลภาพเพื่อตรวจจับเส้นเลน และทดสอบประสิทธิภาพ
2) ศึกษาการพัฒนาแอพพลิเคชันบนอุปกรณ์แอนดรอยด์
3) ออกแบบรายละเอียดการทำงานของระบบ รวมถึงยูสเซอร์อินเตอร์เฟส
4) วิเคราะห์และออกแบบโปรแกรมของระบบแต่ละส่วน
5) พัฒนาโปรแกรมของระบบ
6) ทดสอบการทำงานของระบบ และแก้ไขข้อผิดพลาด
7) จัดทำเอกสาร คู่มือการใช้งานระบบสนับสนุนและแจ้งเตือนบนระบบปฏิบัติการแอนดรอยด์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้แอพพลิเคชันที่ช่วยเหลือด้านความปลอดภัยในการขับขี่รถ
2) ได้ความรู้และประสบการณ์ในการพัฒนาแอพพลิเคชันบนอุปกรณ์แอนดรอยด์
3) ได้นำความรู้เรื่องวิธีการประมวลผลภาพแบบต่างๆ มาประยุกต์ใช้ให้ประโยชน์
4) สามารถนำแอพพลิเคชันที่พัฒนาไปใช้ประโยชน์ในเชิงธุรกิจได้
5) สามารถนำไปพัฒนาต่อยอดโดยการเพิ่มความสามารถในการทำงาน เช่น แจ้งหากทางข้างหน้าเป็นทางโค้ง ทางแยก หรือตรวจจับวัตถุที่อยู่ด้านหน้าของรถ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้ หลักการเกี่ยวกับการนอนหลับ และการหลับใน

2.1.1 การนอนหลับ [3]

การนอนหลับเป็นสภาวะหนึ่งของร่างกาย ที่การเคลื่อนไหว การรับรู้ต่อสิ่งต่างๆ จะลดลง เป็นเวลาหนึ่ง การนอนหลับอย่างเต็มที่จะทำให้ร่างกายสดชื่น มีเรี่ยวแรง การนอนหลับจะเกิดขึ้นเป็น 2 ช่วง ซึ่งจะเกิดขึ้นสลับกันในการนอนหลับแต่ละคืน

2.1.1.1 Non-rapid eye movement Sleep (NREM Sleep)

เป็นช่วงการหลับที่จะหลับลึกลงไปเรื่อยๆ โดยจะแบ่งออกเป็น 4 ระยะดังนี้

ระยะที่ 1 เริ่มมีความง่วง เป็นช่วงที่เริ่มเปลี่ยนจากการตื่นไปยังการนอน มีระยะเวลาประมาณ 30 วินาที ถึง 7 นาที หากถูกกระตุ้นเพียงเล็กน้อย ก็จะตื่นได้

ระยะที่ 2 หลับตื้น การหลับในช่วงต้น ไม่ได้ยินเสียงจากภายนอก เริ่มต้นการหลับสามารถปลุกให้ตื่นได้ง่าย

ระยะที่ 3 หลับปานกลาง หัวใจจะเต้นช้าลง สมองจะทำงานช้าลง สติจะหายไป ตื่นได้ยากแม้จะถูกกระตุ้นจากภายนอก

ระยะที่ 4 หลับลึก เป็นช่วงหลับสนิทที่สุดของการนอน ใช้เวลาประมาณ 30-50 นาที อุณหภูมิร่างกาย และความดันโลหิต จะลดลง

2.1.1.2 Rapid eye movement Sleep (REM Sleep)

เป็นช่วงการหลับที่กล้ามเนื้อต่างๆ ของร่างกาย หยุดการทำงานเกือบทั้งหมด ยกเว้น หัวใจ กระบังลม กล้ามเนื้อตา หลอดเลือด และลำไส้ การหลับในช่วงนี้จะเป็นช่วงที่เราจะฝันเนื่องจากสมองยังคงทำงานอยู่ การหลับช่วงนี้จะใช้เวลาประมาณ 30 นาที จากนั้นจะกลับไปเริ่มระยะที่ 1 ของ NREM

2.1.2 การหลับใน

การหลับใน (Microsleep) คือการที่ร่ายกายและสมองหลับเป็นเวลาสั้นๆ ประมาณ 2 – 3 วินาที โดยที่ยังลืมตาอยู่ การหลับนี้เป็นการหลับในช่วง NREM คนที่หลับใน ภายนอกจะเหมือนคนปกติ แต่สมองไม่สั่งงาน สาเหตุของการหลับในเกิดจาการที่ร่างกายพักผ่อน นอนหลับไม่เพียงพอ ร่ายกายจึงต้องทำการนอนชดเชย เป็นระยะเวลาสั้นๆ

2.1.3 พฤติกรรมของผู้ขับรถหลับใน

ผู้ขับที่กำลังจะหลับใน จะเริ่มต้นจากมีอาการหาวบ่อยครั้ง ไม่มีสมาธิจดจ่ออยู่กับการขับรถ อัตราการเต้นของหัวใจจะลดลง ส่วนพฤติกรรมการขับรถ จะขับส่ายไปมา ขับออกนอกเส้นทาง ขับออกนอกเลนถนนที่ขับอยู่

2.2 ขั้นตอนวิธีการที่ใช้ในการประมวลผลภาพ

2.2.1 การตรวจจับเลนของถนน (Lane Detection) [4]

การตรวจจับเลนของถนนจะใช้หลักการหาเส้นตรงในภาพ ซึ่งเส้นตรงนั้นหาได้จากการหาเส้นขอบของภาพอีกที การตรวจจับเส้นเลนจึงแบ่งออกเป็นสองส่วน คือ เตรียมภาพโดยหาเส้นขอบของภาพ (Edge Detection) และนำภาพที่ได้ไปตรวจจับหาเส้นตรงที่อยู่ในทิศทางที่ต้องการ (Hough Transform) ซึ่งแต่ละส่วนมีวิธีการดังนี้

2.2.1.1 เตรียมภาพโดยการหาเส้นขอบ (Edge Detection)

การหาเส้นขอบในที่นี้จะใช้วิธี Canny Edge Detection ในการหาเส้นขอบของภาพ ซึ่งภาพที่จะนำไปใช้นั้น จะต้องถูกแปลงเป็น Grayscale ก่อน จากนั้นจะนำไปดำเนินการอีก 5 ขั้นตอน ดังนี้

2.2.1.1.1 เตรียมภาพโดยการทำภาพให้เรียบ (Smoothing)

ภาพที่นำมาใช้ส่วนใหญ่จะมี Noise การกำจัด Noise เพื่อทำให้ภาพดูเรียบขึ้น จะทำโดยใช้ Gaussian Filter ทำการ Convolution กับรูปภาพ

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \qquad (2.1)$$

จากสมการ (2.1)

A คือ ภาพที่ต้องการหาเส้นขอบ

B คือ ผลลัพธ์

ภาพ 2.1 ภาพตั้งต้น

ภาพ 2.2 ภาพหลังทำให้เรียบขึ้น

2.2.1.1.2 คำนวณหา Gradients ของรูปภาพ

เนื่องจากบริเวณที่เป็นเส้นขอบจะมีขนาดของ Gradient ที่สูงกว่าบริเวณรอบๆ ซึ่งการหาค่า Gradient ของรูปที่เรียบโดยการใช้ตัวดำเนินการ Sobel กับรูปภาพ เพื่อหา Gradient ในแนวแกน x และ y ตัวดำเนินการ Sobel ที่ใช้เป็นเมตริกซ์ดังนี้

$$K_{GX}=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \tag{2.2}$$

$$K_{GY}=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \tag{2.3}$$

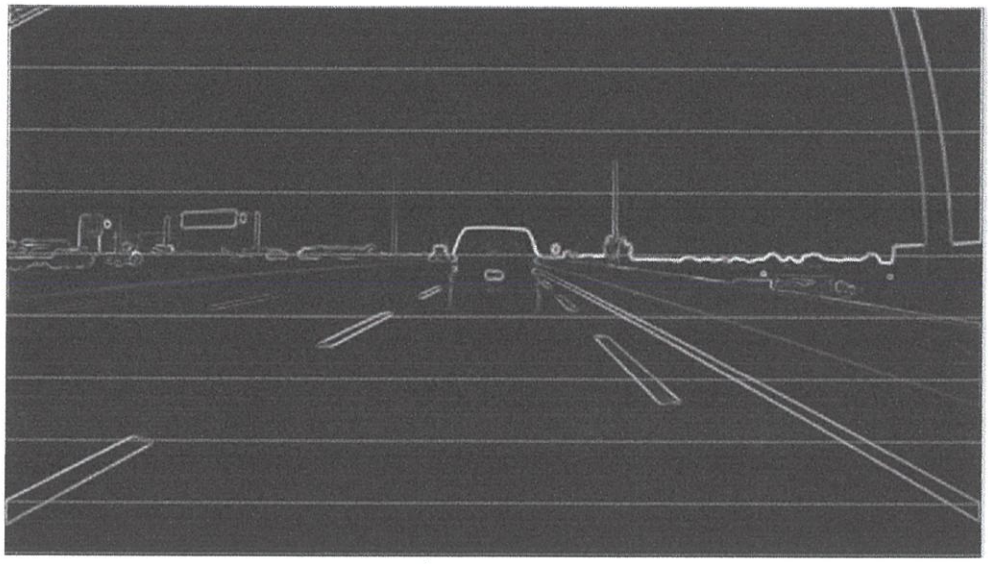
จากนั้นทำการคำนวณหาขนาดของ Gradient หรือ Edge strengths ซึ่งจะสามารถหาค่าได้โดยใช้กฎของพีทาโกรัสดังสมการ (2.4)

$$|G|=\sqrt{G_x^2+G_y^2} \tag{2.4}$$

เมื่อ G_x คือGradient ในแนวแกน x และ G_y คือ Gradient ในแนวแกน y สำหรับทิศทางของ Gradicnt หาโดยใช้สมการ (2.5)

$$\theta=\sin^{-1}\left(\frac{|G_y|}{|G_x|}\right) \tag{2.5}$$

ภาพที่ได้จากการหาขนาดของ Gradient จะได้เป็นเส้นขอบ และองค์ประกอบอื่นๆ ที่มีคุณสมบัติใกล้เคียงกันดังภาพ

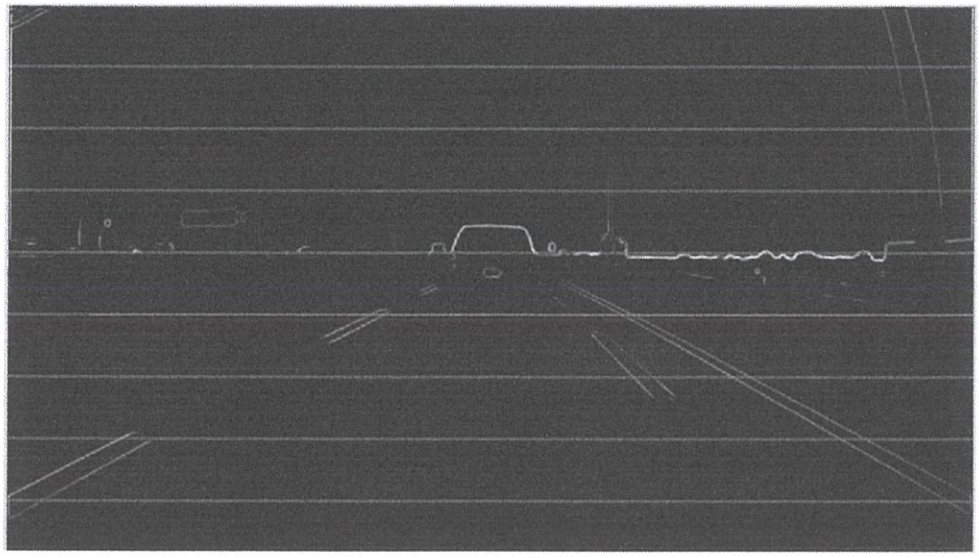


ภาพ 2.3 ภาพหลังที่ได้จากการคำนวณหา Gradient

2.2.1.1.3 กำจัดส่วนที่ค่าขนาดของ Gradient มีค่าน้อย เพื่อทำให้เส้นขอบชัดเจนขึ้น

เป็นการแบ่งรูปภาพที่ได้จากข้อ 2 ออกเป็นส่วนย่อยๆ โดยพิจารณาจากทิศทาง และเก็บค่าขนาดของ Gradient สูงสุดของแต่ละส่วนย่อยๆ ไว้ ส่วนค่าที่น้อยกว่าจะตัดทิ้ง โดยมีขั้นตอนดังนี้

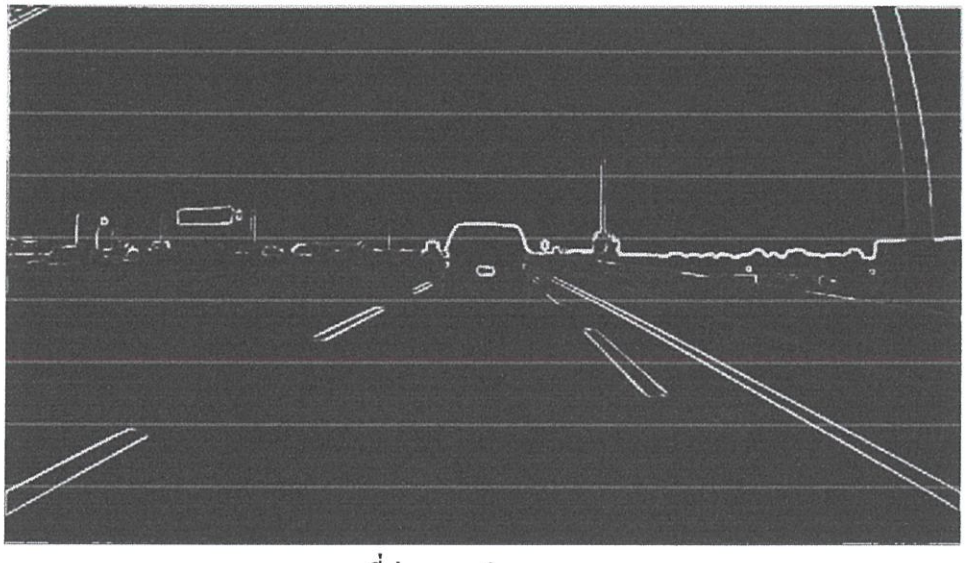
1) ปัดค่าทิศทางของ Gradient (θ) ให้ใกล้เคียงกับทิศทาง 8 ทิศ (0, 45, 90, 135, 180, 225, 270, 315 องศา)
2) เปรียบเทียบค่าขนาดของ Gradient ของแต่ละ pixel กับ pixel อื่นๆ ที่อยู่รอบ ๆ โดยพิจารณาจากทิศทางของ Gradient โดยจะเปรียบเทียบกับ pixel อยู่ในทิศทางเดียวกัน และตรงข้ามกับทิศของ Gradient เช่น หากทิศทางเป็น 90 องศา ก็จะเปรียบเทียบขนาดของ Gradient กับ pixel ที่อยู่ด้านบน และ ด้านล่างของ pixel นั้นๆ
3) หากค่าขนาดของ Gradient ของ pixel นั้นมากที่สุด ให้ทำการเก็บค่านั้นไว้ใน pixel เดิม แต่ถ้าค่านั้นไม่ใช่ค่าที่มากที่สุดให้ตัดค่านั้นทิ้งไป



ภาพ 2.4 ภาพเส้นขอบหลังจากกำจัดส่วนที่ขนาดของ Gradient มีค่าน้อยแล้ว

2.2.1.1.4 Double thresholding

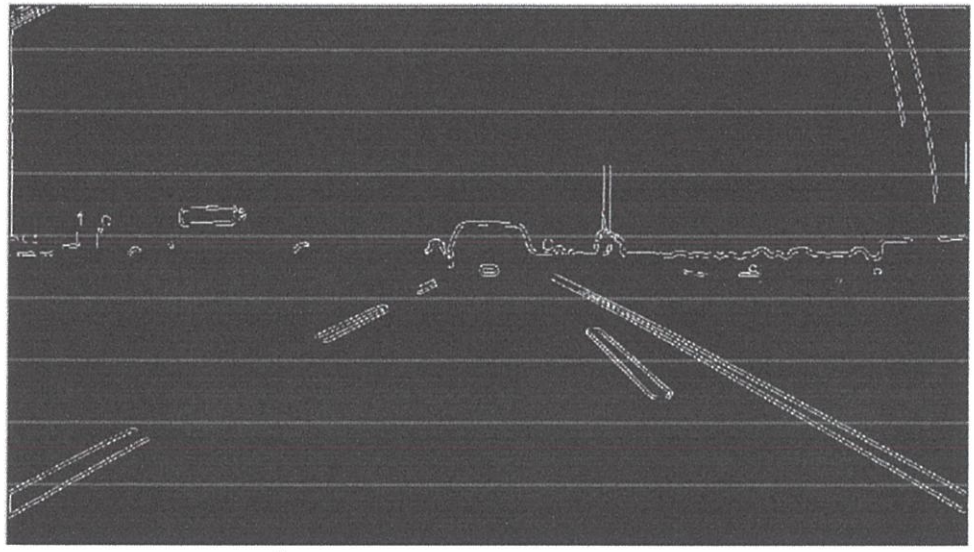
เป็นการทำให้เส้นขอบชัดเจนขึ้น โดยการใช้ค่ากลางสองค่าในการแยกแยะระหว่างเส้นขอบ กับส่วนประกอบอื่นของรูป ที่ยังคงหลงเหลืออยู่ ค่ากลางนี้จะแบ่งเป็นสองค่า เรียกว่า High threshold และ Low threshold โดยค่าขนาดของ Gradient ของ pixel ที่มากกว่า High threshold จะถูกกำหนดว่า เป็นเส้นหนา pixel ที่ขนาดของ Gradient น้อยกว่า Low threshold จะถูกตัดทิ้ง และ pixel ที่ขนาดของ Gradient อยู่ในช่วงระหว่าง High threshold กับ Low threshold จะถูกกำหนดว่าเป็นเส้นบาง



ภาพ 2.5 ภาพที่ผ่านการทำ Double thresholding

2.2.1.1.5 ตรวจสอบความต่อเนื่องของเส้นขอบ

เส้นขอบหนา ส่วนใหญ่จะคือเส้นขอบขอบรูปจริงๆ ส่วนเส้นขอบบาง คือ ส่วนที่เชื่อมต่อเส้นขอบหนาส่วนต่างๆ เข้าด้วยกัน การตรวจสอบความต่อเนื่องของเส้นขอบนั้นจะใช้วิธี BLOB (Binary Large Object) ซึ่งจะใช้วิธีคล้ายกับการดำเนินการในข้อ 3 แต่จะตรวจสอบ pixel ที่สนใจ กับ pixel รอบๆ 8 pixel หากค่าขนาดของ Gradient ของ pixel นั้นมากที่สุดก็จะเก็บค่าไว้ แต่ถ้าไม่ ก็จะตัดทิ้งเช่นกัน



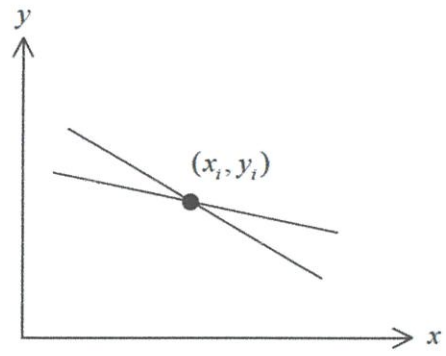
ภาพ 2.6 ภาพที่ได้หลังจากดำเนินการ Canny Edge Detection เสร็จ

2.2.1.1.2 Hough Transform

เป็นวิธีการที่ใช้ตรวจจับเส้นตรงหรือเส้นโค้งที่ยาวต่อเนื่องกัน โดยการแปลงข้อมูลจาก Cartesian Space เป็น Parameter Space ซึ่งจะสามารถอธิบายความสัมพันธ์ระหว่างจุดที่อยู่บนเส้นตรงเดียวกันได้ โดยมีหลักการดังนี้

พิจารณารูปภาพที่มีเส้นตรงในภาพ กำหนดให้จุดใดๆ ในรูปภาพคือ (x_i,y_i) และเส้นตรงทุกเส้นที่ลากผ่านจุดนี้ มีสมการเป็นสมการ (2.6) โดยที่ความชัน (m) และค่าคงที่ (c) สามารถเปลี่ยนแปลงได้ โดยจะเขียนในรูป (m,c)

$$y_i=mx_i+c \tag{2.6}$$

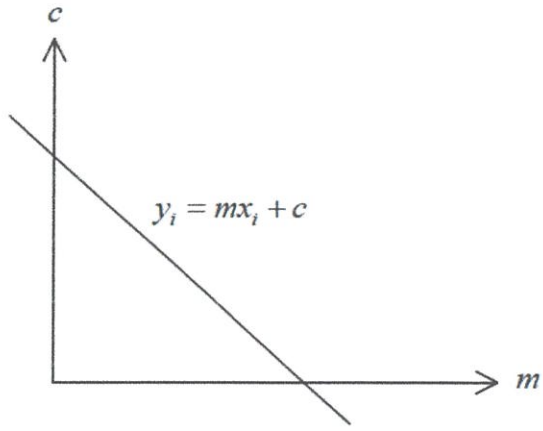


ภาพ 2.7 เส้นตรงที่ตัดผ่านจุดเดียวกัน แต่จะมีค่า (m,c) ต่างกัน

หากทำการจัดรูปสมการ (2.6) ใหม่ โดยใช้ตัวแปรเป็น (m,c) แทนพิกัด (x_i,y_i) จะได้สมการ (2.7) ดังนี้

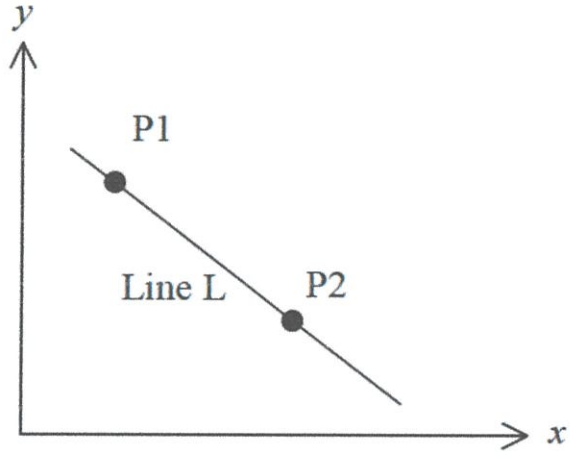
$$c=y_i- mx_i \tag{2.7}$$

ซึ่งสมการ (2.7) นี้จะเขียนด้วยกราฟดังภาพ ในระนาบ(m,c) หรือกล่าวได้ว่าเส้นตรงในระนาบ (x,y) จะแสดงเป็นจุดในระนาบ(m,c)

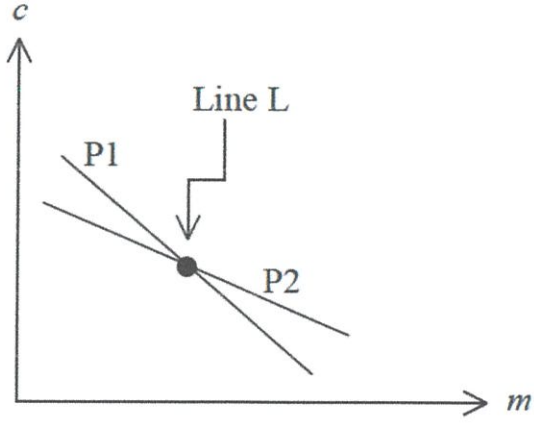


ภาพ 2.8 สมการ (6) ในระนาบ (m,c)

พิจารณาจุดสองจุด P1 และ P2 ซึ่งอยู่บนเส้นตรงเดียวกัน ในระนาบ (x,y) สำหรับแต่ละจุด สำหรับแต่ละจุด เราสามารถแสดงเส้นทุกเส้นที่สามารถลากผ่านจุดนั้นได้ด้วยเส้นตรงเส้นเดียวในระนาบ (m,c)ฉะนั้นเส้นตรงที่ผ่านทั้งจุด P1 และ P2 ในระนาบ (x,y) จะต้องอยู่ที่จุดตัดของเส้น P1 และ P2 ในระนาบ (m,c) ซึ่งหมายความว่า จุดทุกจุดที่อยู่ในเส้นตรงเดียวกันในระนาบ (x,y) จะแสดงเป็นเส้นตรงที่ผ่านจุดๆเดียวในระนาบ (m,c) ซึ่งสามารถดูได้จากภาพ



ภาพ 2.9 จุด P1 และ P2 ที่อยู่บนเส้นตรงเดียวกัน



ภาพ 2.10 จุด P1 และ P2 เขียนในระนาบ (m,c)

จากหลักการข้างต้นจะสามารถนำไปประยุกต์ใช้ในการตรวจจับเส้นตรงในรูปภาพ โดยใช้ขั้นตอนดังนี้

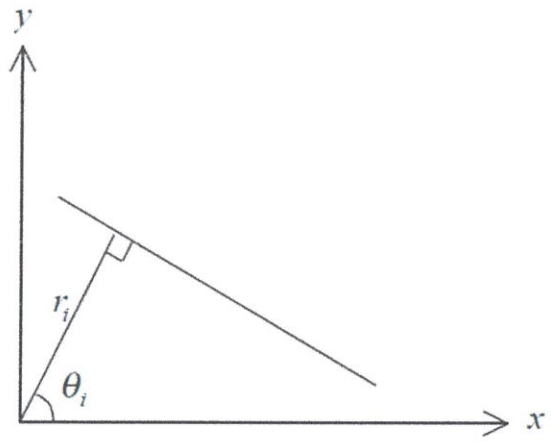
1) ตรวจจับเส้นขอบของรูปด้วยวิธีใดก็ได้ ซึ่งในที่นี้ใช้ Canny Edge Detection
2) Quantize ระนาบ (m,c) ไปเป็น เมตริกซ์ 2 มิติ M ซึ่งขนาดขึ้นอยู่กับ Quantization Level
3) กำหนดค่าเริ่มต้นของเมตริกซ์ M เป็น 0
4) สำหรับแต่ละช่องของเมตริกซ์ $H(m,c)$ ที่ตรงกับเส้นขอบ จะถูกเพิ่มค่าขึ้นอีก 1 ผลที่ได้นี้จะเป็นเมตริกซ์ซึ่งแสดงค่าฮิสโตแกรมของจุดที่อยู่ในเส้นตรงเดียวกัน
5) ช่องของเมตริกซ์ที่มีค่ามาก คือช่องที่แทนเส้นตรงที่อยู่ในภาพนั่นเอง โดยที่ยิ่งค่ามาก แสดงถึงความยาวของเส้นที่มากตาม

ในทางปฏิบัติการใช้ระนาบ (m,c) นั้นมีไม่เป็นที่ เนื่องจากเส้นตรงส่วนมีความยาวมาก ทำให้จำนวนเส้นที่ลากผ่านจุดในระนาบ (m,c) มีจำนวนมาก ทำให้กาตรวจจับเส้นตรงนั้นทำได้ยาก เนื่องจากค่า m ที่ได้เป็นค่าอนันต์ จึงทำให้ในทางปฏิบัติ จะใช้พิกัดเชิงขั้วในการคำนวณแทน

2.2.1.1.2.1 การใช้พิกัดเชิงขั้ว

โดยการใช้สมการ (2.8) แทนสมการ (2.6) ดังนี้

$$x \cos\theta + y \sin\theta = r \tag{2.8}$$



ภาพ 2.11 เส้นตรงจากระนาบ (x,y) ที่แสดงความสัมพันธ์ในรูปแบบพิกัดเชิงขั้ว

จากนั้นใช้หลักการเดิมคือการใช้เมตริกซ์ H แต่ เปลี่ยนเป็นเมตริกซ์ 3 มิติ คือค่าของ รัศมี และ พิกัด x y ตำแหน่งจุดศูนย์กลาง

2.2.2 การตรวจจับวัตถุ (Object Detection) [5]

การตรวจจับวัตถุจะใช้อัลกอริทึม cascade classifier จะแบ่งการทำงานหลักๆเป็นสองขั้นตอนคือ การเตรียมข้อมูลหรือการ training และการตรวจจับค้นหา

2.2.2.1 การเตรียมข้อมูล

การเตรียมข้อมูลของวัตถุที่เราต้องการจะตรวจจับค้นหา โดยมีวิธีดังนี้

1) การเตรียมชุดของตัวอย่างโดยจะแบ่งเป็นตัวอย่างที่ดีกับตัวอย่างที่ไม่ดี
2) สร้าง training set จากภาพและภาพพื้นหลัง
3) สร้าง test set ในรูปแบบของไฟล์ภาพ jpg
4) ทำการแปลง test set ให้อยู่ในรูปแบบของ vec
5) นำไปเข้าในฟังก์ชัน Cascade Training
6) ผลลัพธ์ที่ได้จะอยู่ในไฟล์ xml เช่น cascade.xml ซึ่งเราจะนำไฟล์ cascade.xml ไปใช้ในการทำงานส่วนที่สองการตรวจจับ

2.2.2.2 การตรวจจับค้นหาวัตถุ

หลังจากที่ได้ไฟล์ cascade.xml มาแล้วเราก็จะนำไฟล์มาใช้กับตัว Classifier เพื่อที่จะทำการตรวจจับค้นหาวัตถุที่เราสนใจ โดยการทำงานของตัว Classifier จะทำการสร้างพื้นที่ๆสนใจ (โดยมีขนาดเท่ากับตอนที่เราใช้ทำ training) บนภาพที่อินพุทเข้ามา ตัว classifier จะให้ผลลัพธ์เป็น 1 หากภาพในพื้นที่ๆสนใจคล้ายกับภาพวัตถุที่เราใช้ตอนทำ training หรือให้ผลลัพธ์เป็น 0 ในกรณีที่ไม่เหมือน

หลังจากนั้นก็จะเลื่อนตัว หน้าต่างการค้นหา ไปเพื่อตรวจสอบตำแหน่งถัดไปในภาพเรื่อยๆ จนถึงจุดสิ้นสุดของภาพ

classifier ถูกออกแบบมาเพื่อให้ง่ายต่อการ resize เพื่อสามารถหาวัตถุที่สนใจในหลายๆขนาดที่ต่างกัน ซึ่งวิธีนี้มีประสิทธิภาพกว่าการ resize ภาพอินพุท

2.3 OpenCV [1]

เป็น Library ที่พัฒนาเพื่อการประมวลผลที่เกี่ยวกับ Computer Vision สามารถใช้งานได้หลากหลาย Platform ถูกพัฒนาโดยใช้ภาษา C++ รองรับการประยุกต์ใช้งานได้หลากหลาย เช่น Facial recognition, Gesture recognition, Object Identification, Segmentation, Motion Tracking เป็นต้น ซึ่งในโครงงานได้ใช้ OpenCV ช่วยในการประมวลผลเกี่ยวกับการแยกส่วนประกอบของภาพ ตรวจจับวัตถุ และประมวลผลที่ได้ในระบบปฏิบัติการแอนดรอยด์

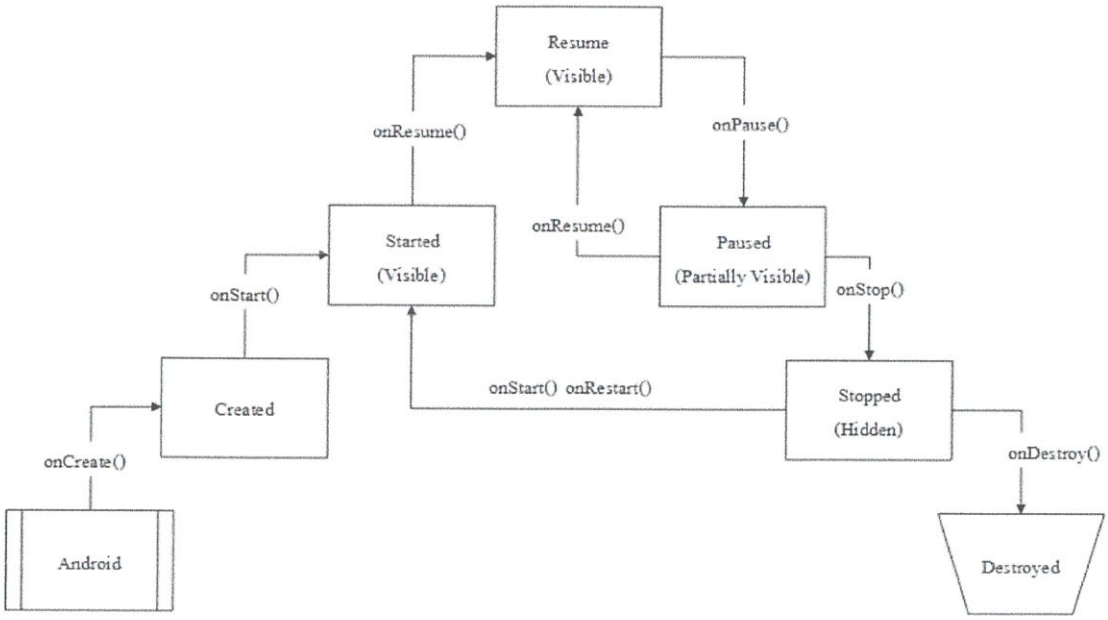
2.4 การทำงานของแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ [2]

แอพพลิเคชันของระบบปฏิบัติการแอนดรอยด์ จะเรียกการทำงานของ Method ต่างๆ ใน Activity ซึ่งแตกต่างจากแอพพลิเคชันทั่วไปที่จะเรียกในฟังก์ชัน Main ทำให้การพัฒนาแอพพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ จะต้องเข้าใจวงจรชีวิตของ Activity (Lifecycle of Activity) และ Method ต่างๆ ที่ทำงานร่วมกัน

Activity ที่กำลังทำงานอยู่ จะแบ่งออกเป็น 3 สถานะ ดังนี้

1) Running หรือเรียกอีกอย่างว่า Resume เป็นสถานะที่แอพพลิเคชันกำลังทำงานอยู่บนหน้าจอ
2) Paused เป็นสถานะที่แอพพลิเคชันบนหน้าจอ มีบางส่วนมาบัง และไม่รับ input จากผู้ใช้ แต่ตัวแอพพลิเคชันยังคงอยู่ในหน่วยความจำหลัก
3) Stopped เป็นสถานะที่แอพพลิเคชันจะทำงานอยู่เบื้องหลัง แต่ยังก็ยังถูกเก็บอยู่ในหน่วยความจำหลัก หากระบบปฏิบัติการต้องการหน่วยความจำไปใช้งานเมื่อไหร่ ระบบปฏิบัติการจะปิดแอพพลิเคชันทิ้งทันที

แอพพลิเคชันที่อยู่ในสถานะ Paused หรือ Stopped จะถูกปิดการทำงานได้ หากระบบปฏิบัติการต้องการหน่วยความจำหลักไปใช้งาน



ภาพ 2.12 วงจรชีวิตของ Activity

จากรูปที่ 2.12 จะเห็นว่า มี Method อยู่ 6 Method ที่จัดการเกี่ยวกับสถานะต่างๆ ดังนี้

- onCreate() จะถูกเรียกเมื่อ Activity ถูกสร้างครั้งแรก เป็นการสร้าง Layout ของ Activity นั้นๆ
- onStart() จะถูกเรียกเมื่อ Activity เริ่มทำงาน
- onResume() จะถูกเรียกเมื่อแอพพลิเคชันที่ถูก Paused กลับมาทำงานบนหน้าจออีกครั้ง จะทำการสร้าง field ต่างๆ หรือเรียก Service ขึ้นมาอีกครั้ง
- onPause() จะถูกเรียกเมื่อมี Activity อื่นมาแสดงผลทับ ใช้ในการปิด Service ที่ใช้งานต่างๆ ได้
- onStop() ถูกเรียกเมื่อแอพพลิเคชันถูกย้ายไปทำงานด้านหลัง ใช้ในการปิดการเชื่อมต่อกับฐานข้อมูล หรือการเชื่อมต่ออื่นๆ
- onDestroy() ถูกเรียกเมื่อแอพพลิเคชันถูกปิด โดยเอาออกจากหน่วยความจำ

ในการควบคุมแต่ละ Activity จะใช้ Method เหล่านี้ในการควบคุมส่วนต่างๆ ตั้งแต่การกำหนดค่าเริ่มต้น การเชื่อมต่อกับส่วนอื่นๆ

2.4 ปัจจัยที่มีผลต่อระบบ

2.4.1 แสง

ระบบจะทำงานได้ดีในสภาพแวดล้อมที่มีแสงเหมาะสม เนื่องจากสภาพแวดล้อมที่มีแสงน้อย จะทำให้เกิดสัญญาณรบกวนในภาพ ทำให้ขั้นตอนวิธีการที่ใช้ในการตรวจจับเส้นและวัตถุนั้นทำงานได้ไม่ดี และเกิดข้อผิดพลาดในการตรวจจับ

2.4.2 ทัศนวิสัย

ระบบจะทำงานได้ดีในสภาพอากาศที่ท้องฟ้าแจ่มใส ไม่มีเมฆฝน หมอก รบกวน

2.4.3 ความเร็วในการขับรถ

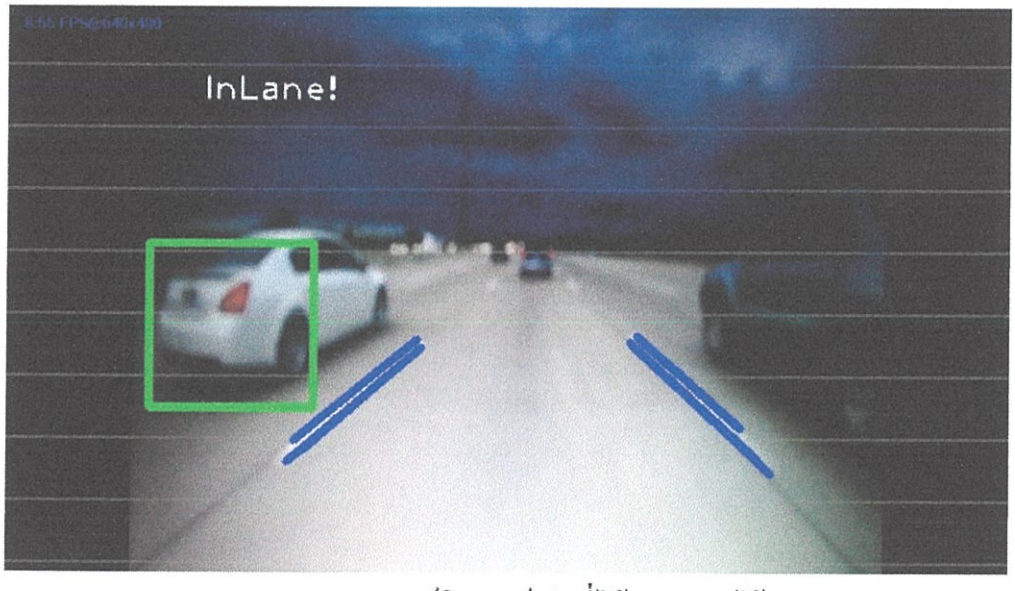
คาดว่าระบบจะสามารถทำงานได้ที่ที่ความเร็วรถประมาณไม่เกิน 60 กิโลเมตรต่อชั่วโมง

บทที่ 3
การออกแบบและพัฒนา

3.1 ภาพรวมของระบบ

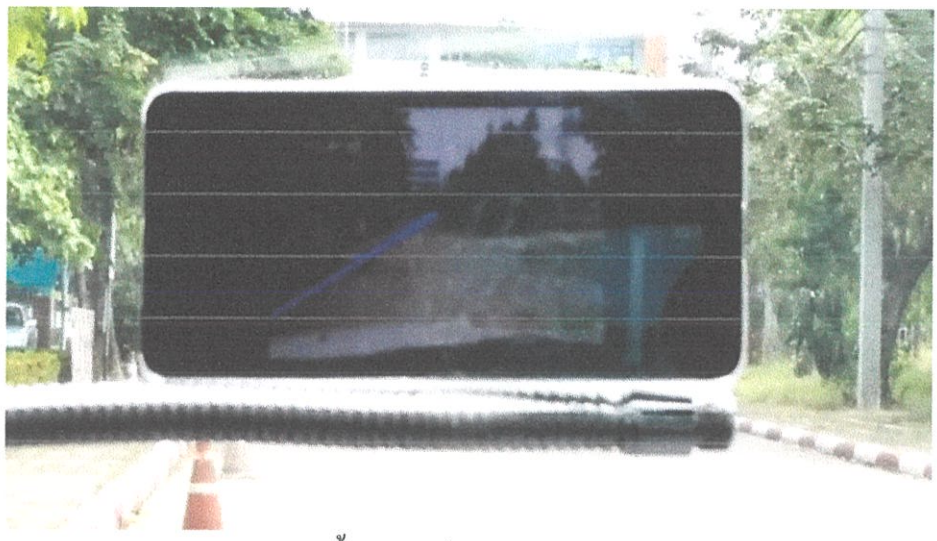
ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ เป็นแอพพลิเคชันที่ทำงานบนอุปกรณ์แอนดรอยด์ โดยจะรับภาพจากกล้องของอุปกรณ์แอนดรอยด์มาประมวลผล การออกแบบทั้งหมดจะอยู่ที่การทำงานของโปรแกรมเป็นหลัก ซึ่งโปรแกรมจะสามารถทำงานได้ดังนี้

1) ตรวจจับตำแหน่งของเลนและรถบนถนน
2) ตรวจจับวัตถุที่อยู่ด้านหน้ารถได้
3) แสดงเส้นจำลองของเลนถนน
4) ส่งสัญญาณแจ้งเตือนผู้ขับ หากรถส่าย หรือออกนอกเลน และหากวัตถุที่อยู่ด้านหน้ารถเข้าใกล้ในระยะ 10 เมตร

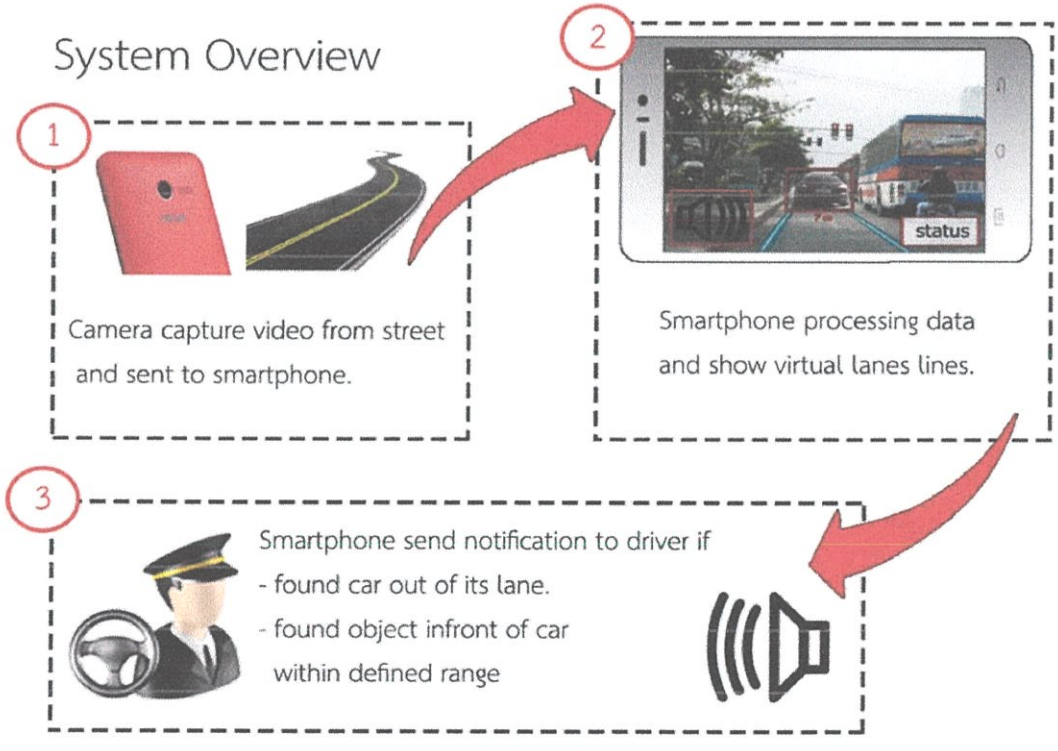


ภาพ 3.1 ยูสเซอร์อินเตอร์เฟสที่ได้ออกแบบไว้

ยูสเซอร์อินเตอร์เฟส ซึ่งในส่วนของยูสเซอร์อินเตอร์เฟสจะเป็นส่วนที่ติดต่อสื่อสารกับผู้ใช้ รวมถึงแสดงเส้นเลนจำลองที่ได้จากการประมวลผล กรอบสี่เหลี่ยมล้อมรอบวัตถุหน้ารถที่ตรวจจับได้ และหากรถส่าย หรือออกนอกเลน จะมีสัญลักษณ์แจ้งเตือนพร้อมกับเสียงเพื่อเตือนให้ไปยังผู้ขับ



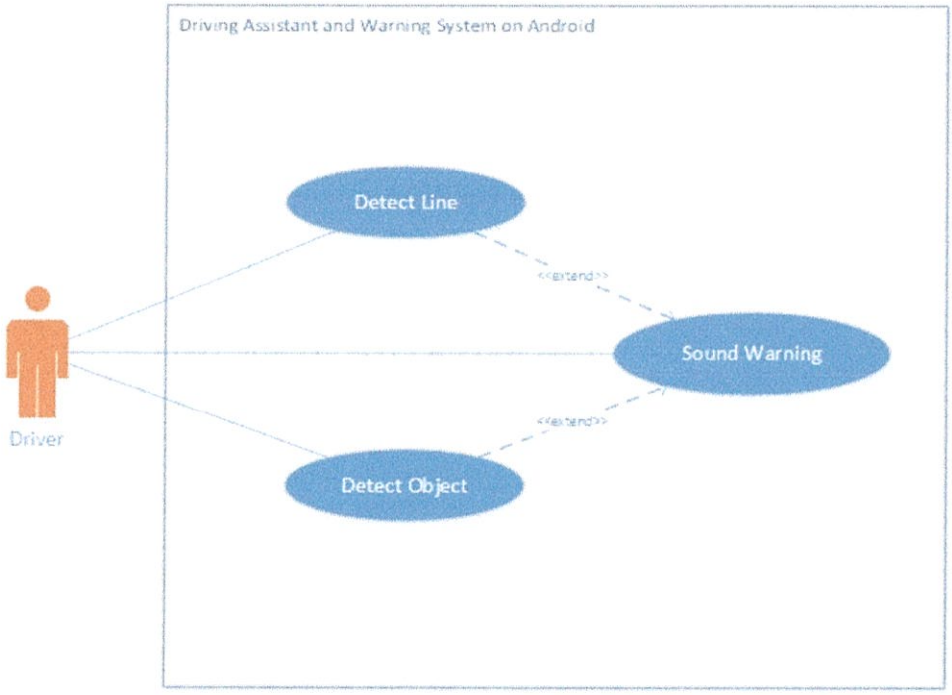
ภาพ 3.2 ติดตั้งอุปกรณ์พร้อมใช้งานแอพพลิเคชัน



ภาพ 3.3 ภาพรวมของระบบ

3.2 Use Case Diagram ของระบบ

ระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ ทำงานโดยกล้องของอุปกรณ์แอนดรอยด์ ถ่ายวีดีโอแล้วนำวีดีโอที่ได้นั้นไปประมวลผล และแสดงเส้นจำลองของเลนถนน และกรอบรอบวัตถุที่อยู่หน้ารถขึ้นมา จากนั้นจะตรวจสอบตำแหน่งของรถบนเลนถนน และตรวจสอบระยะห่างระหว่างรถกับวัตถุด้านหน้า หากรถล่ายหรือออกจากเลนถนน หรือวัตถุที่อยู่หน้ารถเข้าใกล้ในระยะ 10 เมตร จะทำการส่งสัญญาณเตือนเพื่อแจ้งเตือนไปยังคนขับ ให้ทราบถึงอันตราย



ภาพ 3.4 Use Case Diagram ของระบบ

จากภาพ 3.2 Use Case Diagram จะเห็นว่าระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ จะแบ่งการทำงานออกเป็น 3 ส่วนหลักๆ คือ ตรวจจับเลน ตรวจจับวัตถุ และส่งเสียงเตือน โดยผู้ใช้ซึ่งก็คือ ผู้ขับขี่ มีหน้าที่รับข้อมูลตำแหน่งของรถในเลน และวัตถุที่อยู่ด้านหน้ารถ รวมถึงรับสัญญาณเตือนเมื่ออยู่ในอันตราย

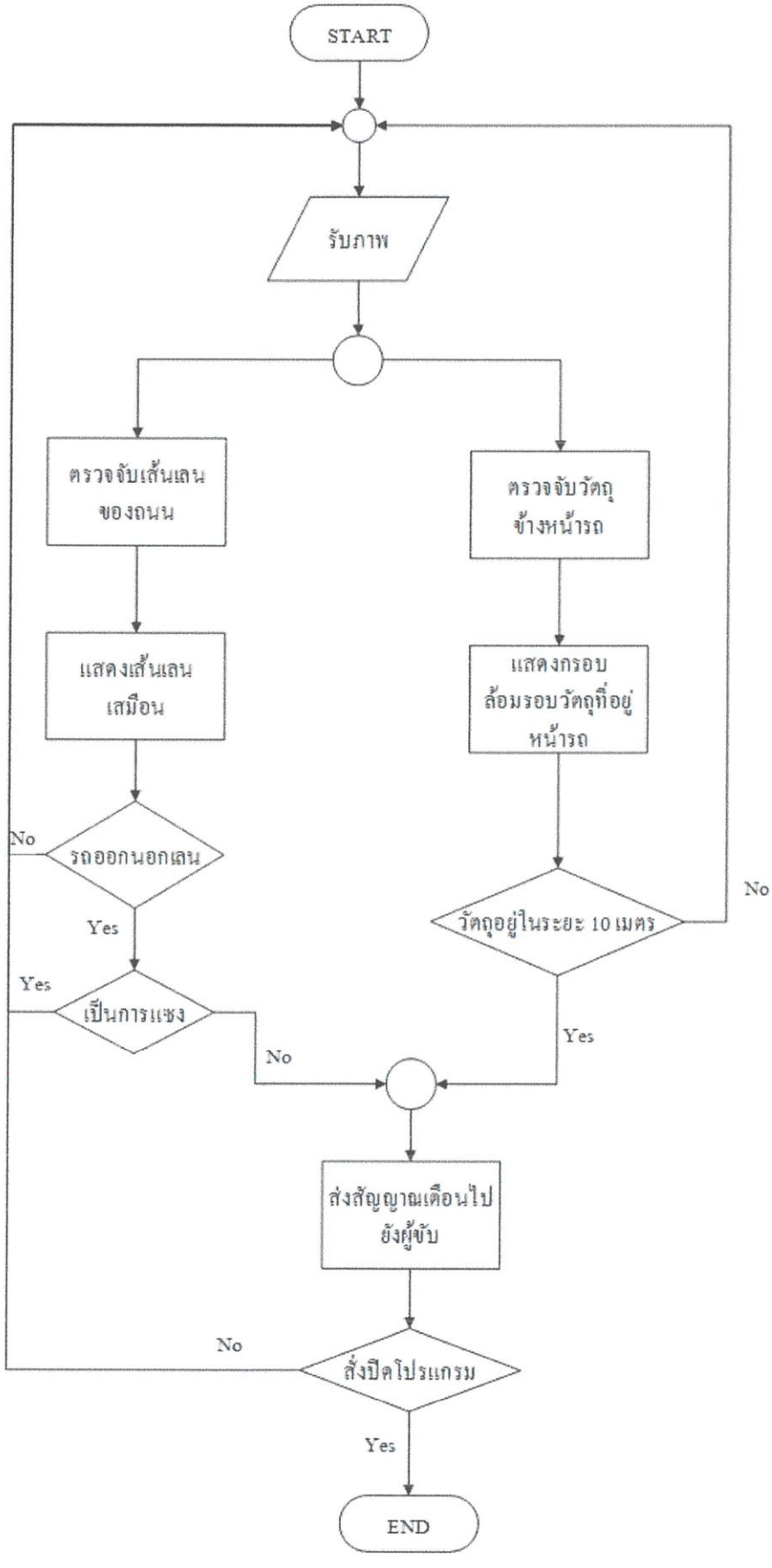
3.2.1 คำอธิบาย Use Case ของระบบ

จากรายละเอียดของระบบสนับสนุนและแจ้งเตือนผู้ขับขี่บนระบบปฏิบัติการแอนดรอยด์ และภาพ 3.2 จะได้คำอธิบายของ Use Case ดังนี้

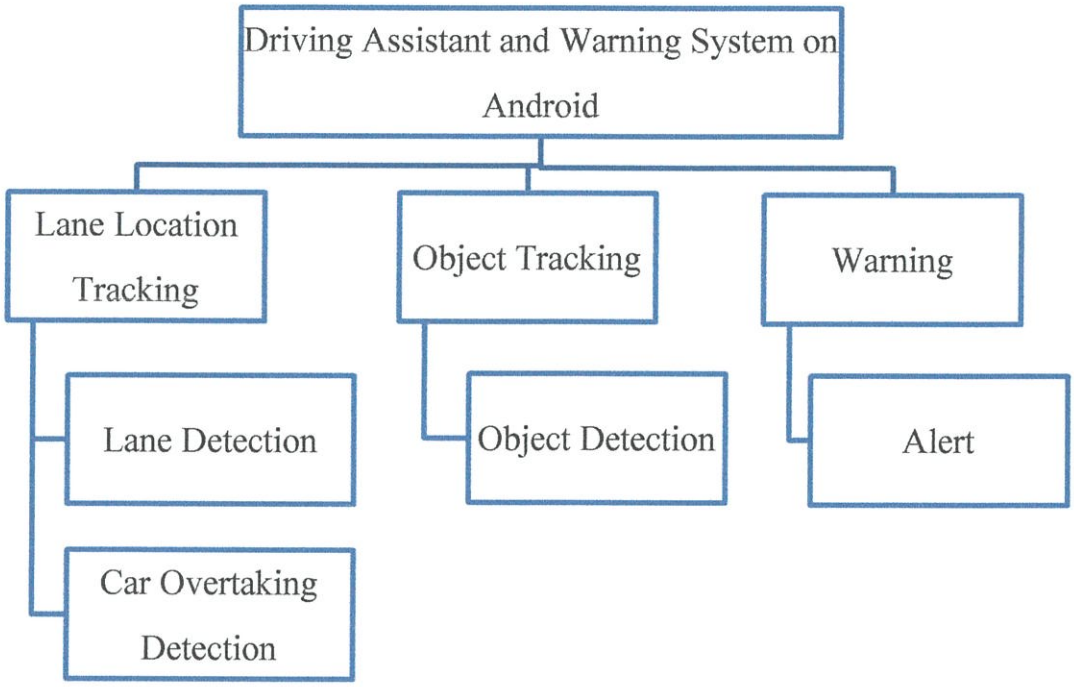
Use Case Title : Detect Lane	Use Case Id : 1
Primary Actor : Driver	
Stakeholder Actor : -	
Main Flow : ระบบจะแสดงเส้นเลนจำลองจากการประมวลผลภาพที่ถ่ายจากกล้องของอุปกรณ์ให้ผู้ขับได้ทราบถึงตำแหน่งของรถบนเลน	
Exceptional Flow : รถส่ายออกนอกเลน จะสั่งการให้ส่วนแจ้งเตือนทำงาน	

Use Case Title : Detect Object	Use Case Id : 2
Primary Actor : Driver	
Stakeholder Actor : -	
Main Flow : ระบบจะแสดงตำแหน่งของวัตถุที่อยู่ด้านหน้ารถจากการประมวลผลภาพที่ถ่ายจากกล้องของอุปกรณ์ให้ผู้ขับได้ทราบถึงตำแหน่งของวัตถุนั้น	
Exceptional Flow : วัตถุที่อยู่หน้ารถเข้าใกล้ในระยะ 10 เมตร จะสั่งการให้ส่วนแจ้งเตือนทำงาน	

Use Case Title : Sound Warning	Use Case Id : 3
Primary Actor : Driver	
Stakeholder Actor : -	
Main Flow : เมื่อส่วนตรวจจับทราบถึงความผิดปกติ เช่นรถส่ายออกนอกเลน หรือวัตถุที่อยู่หน้ารถเข้ามาใกล้ในระยะ 10 เมตร จะทำการส่งสัญญาณเตือนไปยังผู้ขับ	



ภาพ 3.5 ภาพรวมการทำงานของระบบ



ภาพ 3.6 แผนภาพแสดงการออกแบบระบบย่อย

จากภาพ 3.5 ระบบสนับสนุนและแจ้งเตือนผู้ขับบนระบบปฏิบัติการแอนดรอยด์ถูกแบ่งออกเป็นสองส่วนย่อยดังนี้

1. ระบบตรวจจับตำแหน่งของรถยนต์บนเลนถนน
   1.1 ระบบตรวจจับตำแหน่งของเลน ส่วนนี้จะทำการเตรียมภาพที่ได้จากกล้อง และนำไปประมวลผลเพื่อตรวจจับตำแหน่งของเลนบนถนน
   1.2 ระบบตรวจจับการออกจากเลนของรถ
   ส่วนนี้จะทำการตรวจจับตำแหน่งของรถในเลน โดยจะตรวจจับหากรถเคลื่อนที่ล่ายหรือออกนอกเลน ซึ่งจะสามารถแยกความแตกต่างระหว่างการแซงหรือเลี้ยวรถ กับการออกจากเลนด้วยสาเหตุอื่นได้
2. ระบบตรวจจับวัตถุหน้ารถ
   ตรวจจับวัตถุที่อยู่หน้ารถ แสดงเส้นกรอบล้อมรอบวัตถุ และตรวจสอบระยะของวัตถุ
3. ระบบแจ้งเตือนผู้ขับขี่
   ทำการแจ้งเตือนผู้ขับขี่เมื่อรถล่าย เคลื่อนออกนอกเลน หรือวัตถุหน้ารถเข้ามาใกล้ในระยะ 10 เมตร โดยการส่งเสียงเตือน

ตาราง 3.1 สรุปความสามารถของระบบ

ลำดับที่	หัวข้อ	ผลการทดสอบที่ควรจะเป็น (ได้/ไม่ได้)
1.	**ระบบตรวจจับตำแหน่งของรถยนต์บนเลนถนน**	
1.1	ตรวจจับตำแหน่งของเลนถนนได้ หากรถขับด้วยความเร็วไม่เกิน 60 กิโลเมตร/ชั่วโมง	✓
1.2	ตรวจจับตำแหน่งของรถบนเลนถนนได้	✓
1.3	แสดงเส้นจำลองของเลนถนนได้	✓
1.4	สามารถแยกแยะการส่ายของรถ กับการขับแซงรถคันข้างหน้าได้	✓
1.5	สามารถตรวจจับวัตถุที่ข้างหน้ารถในระยะ 10 เมตรได้	✓
1.6	สามารถตรวจจับเลนถนนได้ในสภาพแวดล้อมที่มีแสงน้อย	✗
1.7	สามารถตรวจจับเลนถนนได้ในขณะที่ทัศนวิสัยไม่ดี	✗
2.	**ระบบแจ้งเตือนผู้ขับขี่**	
2.1	สามารถส่งสัญญาณเตือนเมื่อรถมีการส่ายหรือออกนอกเลน	✓
2.2	สามารถส่งสัญญาณเตือนเมื่อวัตถุข้างหน้ารถ เข้าใกล้รถในระยะ 10 เมตร	✓
2.3	ไม่ส่งสัญญาณเตือน หากรถออกนอกเลนด้วยการแซงรถคันข้างหน้า	✓

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองที่ 1 การหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับเส้นเลนถนน

การทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับเส้นเลนถนน จะทำการหาเปอร์เซ็นต์ของความถูกต้องในการตรวจหาเส้นเลนถนน โดยจะนำวิดีโอที่บันทึกได้จากแอพพลิเคชั่นมาคำนวณโดยจะใช้วิธีคิดค่าเฉลี่ยจาก 3 วิดีโอ โดยแต่ละวิดีโอที่ใช้จะมีความยาวประมาณ 1 นาที เฟรมเรทประมาณ 7 เฟรมต่อวินาที การสุ่มเลือกเฟรมคือ 1 เฟรมทุกๆ 3 วินาที และวิเคราะห์ผลลัพธ์

วิดีโอที่ 1 และ 2 เป็นการรันแอพพลิเคชั่นผ่านหน้าจอคอมพิวเตอร์ แต่วิดีโอที่ 3 เป็นการรันแอพพลิเคชั่นบนรถของจริง

4.1.1 จุดประสงค์ในการทดลอง

เพื่อหาเปอร์เซ็นต์ของความถูกต้องของการตรวจจับเส้นเลนถนน

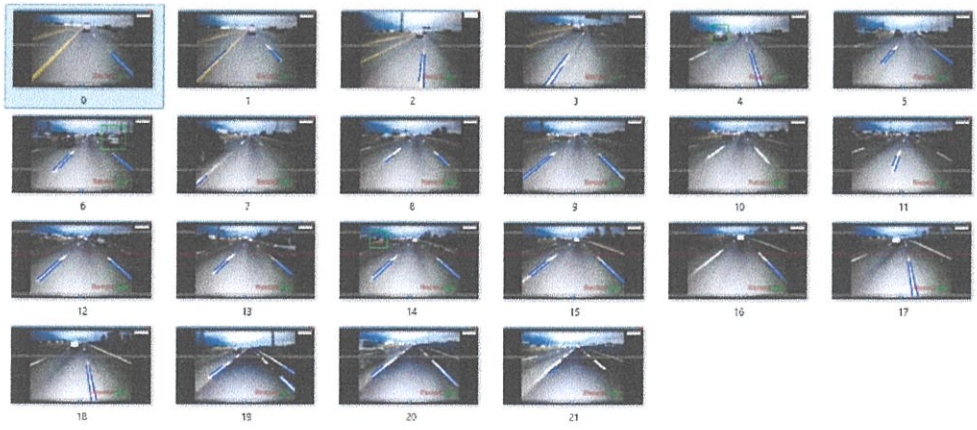
4.1.2 เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1) Asus Zenfone 5 LTE
2) คอมพิวเตอร์ระบบปฏิบัติการ Windows 8 ,CPU core 2 duo 2.26 GHz, Ram 4 GB
3) Windows Media Player

4.1.3 วิธีการทดลอง

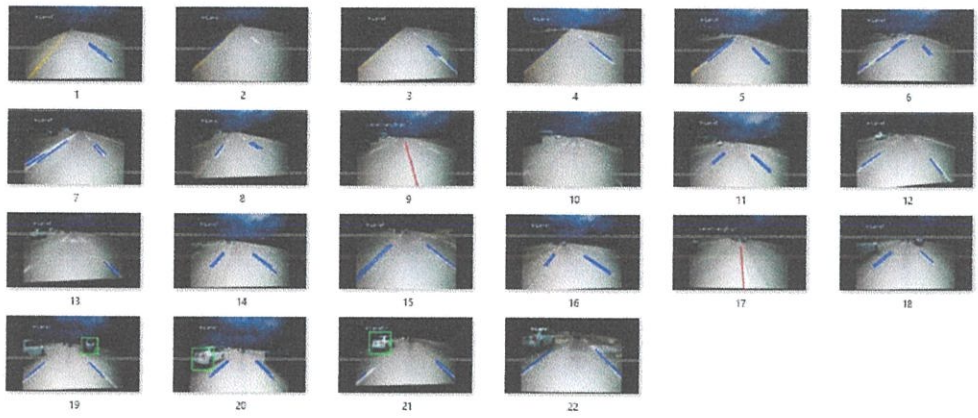
1) เฟรมที่ได้จากการสุ่มเลือกจากทั้ง 3 วิดีโอ

วิดีโอที่ 1 ได้ 22 เฟรม



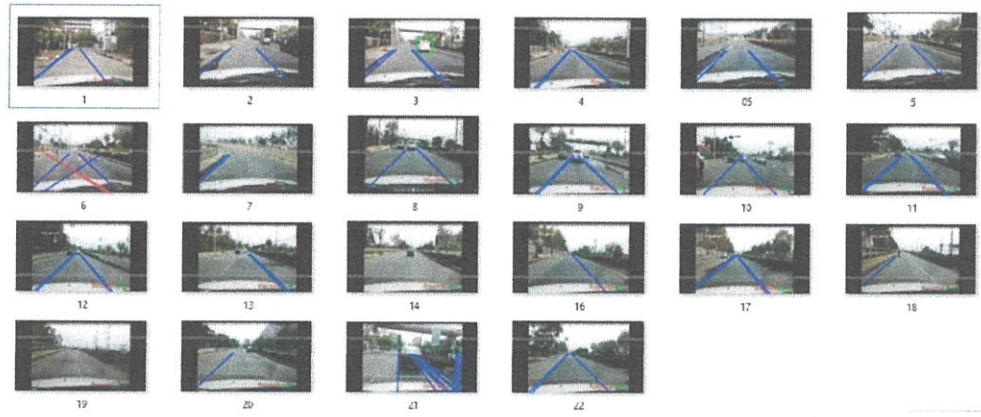
ภาพ 4.1 ตัวอย่างจากเฟรมของวีดีโอ 1 สำหรับการทดลองที่ 1

วิดีโอที่ 2 ได้ 22 เฟรม



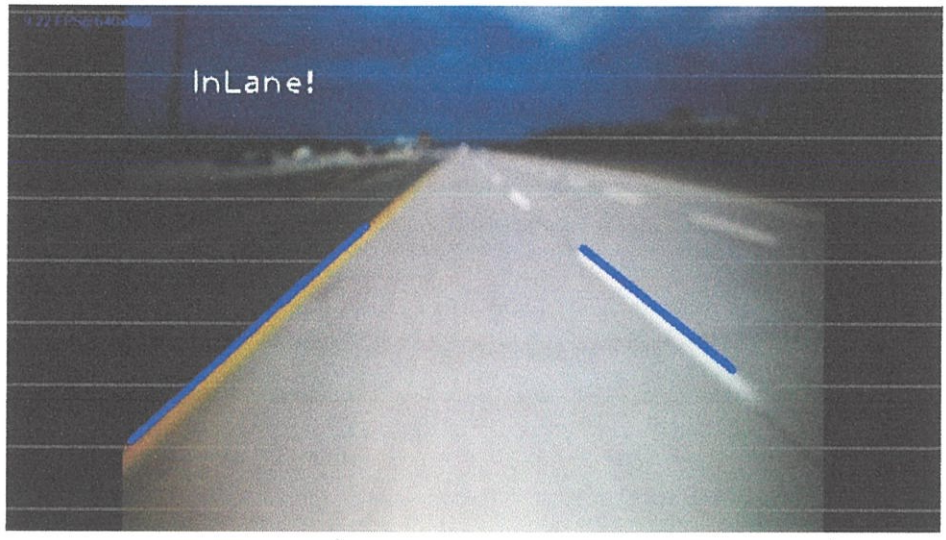
ภาพ 4.2 ตัวอย่างจากเฟรมของวีดีโอที่ 2 สำหรับการทดลองที่ 1

วิดีโอที่ 3 ได้ 22 เฟรม



ภาพ 4.3 ตัวอย่างจากเฟรมของวีดีโอที่ 3 สำหรับการทดลองที่ 1

2) ทำการวิเคราะห์ความถูกต้องของภาพที่ถูกสุ่มขึ้นมา



ภาพ 4.4 ตัวอย่างภาพที่ตรวจจับเส้นได้ถูกต้อง สำหรับการทดลองที่ 1

ภาพ 4.5 ตัวอย่างภาพที่ตรวจจับเส้นได้ไม่ถูกต้อง สำหรับการทดลองที่ 1

3) บันทึกผลที่ได้ลงในตาราง 4.1

ตาราง 4.1 ผลการทดลองที่ 1

	วิดีโอที่ 1		วิดีโอที่ 2		วิดีโอที่ 3	
ภาพที่	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง
1		✓		✓	✓	
2	✓		✓		✓	
3	✓		✓		✓	
4	✓		✓			✓
5	✓		✓		✓	
6	✓		✓		✓	
7	✓		✓		✓	
8		✓	✓			✓
9	✓		✓		✓	
10	✓			✓	✓	
11		✓	✓		✓	
12	✓		✓		✓	
13	✓			✓	✓	
14	✓		✓			✓
15	✓		✓			✓
16	✓		✓			✓
17	✓	✓	✓		✓	
18	✓		✓			✓
19	✓		✓			✓
20	✓		✓			✓
21	✓		✓			✓
22	✓		✓		✓	
รวม	18	4	19	3	13	9

4) คำนวณเปอร์เซ็นต์ความถูกต้อง

หาได้จาก (ภาพผลลัพธ์ที่ถูกต้อง * 100) / จำนวนภาพผลลัพธ์ทั้งหมด

วิดีโอที่ 1 = (18 * 100) / 22

= 81.82 %

วิดีโอที่ 2 = (16 * 100) / 22

= 86.36 %

วิดีโอที่ 3 = (18 * 100) / 22

= 59.09 %

เฉลี่ย = (81.82+86.36+59.09) / 3

= 76.09 %

4.1.4 ปัญหาและอุปสรรค

1) ภาพผลลัพธ์บางภาพทำการแยกแยะยากระหว่างภาพที่ดีกับภาพที่ไม่ดี
2) เมื่อใช้วิดีโอจากบนรถจริงพบว่าได้ภาพเบลอเป็นบางครั้งเนื่องจากการสั่นสะเทือนของรถ
3) เส้นเลนถนนบางแห่ง มีสีที่จางหรือขาดหาย ทำให้การตรวจจับเส้นมีความยาก

4.1.5 การแก้ไข

1) ทำการวิเคราะห์ให้ละเอียดขึ้นหรือ ทำการสุ่มภาพมาใหม่
2) ใช้ซอร์ฟแวร์ช่วยให้ภาพชัดขึ้นเช่น stabilizer
3) ใช้อุปกรณ์ที่มีคุณภาพสูงขึ้นเพื่อจะได้ภาพที่ชัดขึ้น และปรับค่า threshold ให้เหมาะสม

4.1.6 สรุปผลการทดลอง

จากการทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับเส้นเลนถนนพบว่าความความถูกต้องของแอพพลิเคชั่นในการตรวจจับเส้นเลนถนนที่ได้จากการสุ่มภาพมาวิเคราะห์และคำนวณเฉลี่ยอยู่ที่ 76.09 % และเมื่อเปรียบเทียบผลลัพธ์ระหว่างวิดีโอที่รันผ่านหน้าจอกับรันจริงพบว่า วิดีโอที่ได้จากการรันจริงจะมีความแม่นยำน้อยกว่าซึ่งเป็นผลมาจาก การสั่นของรถ

4.2 การทดลองที่ 2 การหาความแม่นยำในการตรวจจับวัตถุที่อยู่ข้างหน้า

การทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับวัตถุที่อยู่ข้างหน้า จะทำการหาเปอร์เซ็นต์ของความถูกต้องในการตรวจจับวัตถุที่อยู่ข้างหน้า แต่ละเฟรมที่นำมาใช้ทำการทดลองจะเลือกเฟรมที่มีภาพวัตถุปรากฏอยู่ โดยจะนำวิดีโอที่บันทึกได้จากแอพพลิเคชั่นมาทดสอบและคำนวณโดยใช้วิธี นับจำนวนวัตถุที่พบทั้งหมดแล้วนำมาวิเคราะห์ โดยจะทำการทดลองจาก 3 วิดีโอและนำมาคิดหาค่าเฉลี่ย

วิดีโอที่ 1 และ 2 เป็นการรันแอพพลิเคชั่นผ่านหน้าจอคอมพิวเตอร์ แต่วิดีโอที่ 3 เป็นการรันแอพพลิเคชั่นบนรถของจริง

4.2.1 จุดประสงค์ในการทดลอง

เพื่อหาเปอร์เซ็นต์ของความถูกต้องของการตรวจจับวัตถุที่อยู่ข้างหน้า

4.2.2 เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1) Asus Zenfone 5 LTE

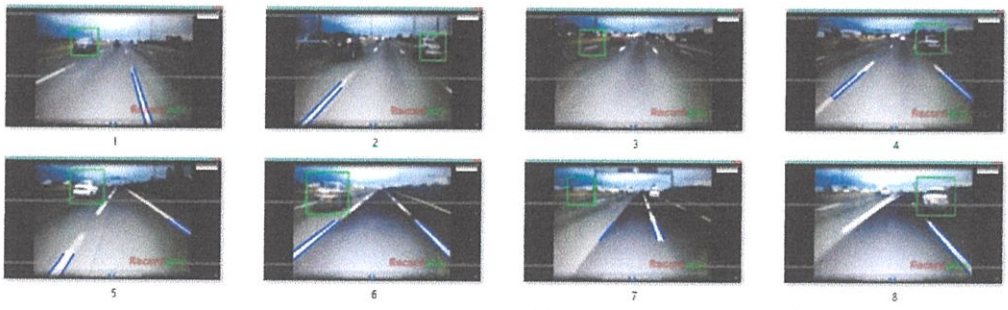
2) คอมพิวเตอร์ระบบปฏิบัติการ Windows 8 ,CPU core 2 duo 2.26 GHz, Ram 4 GB

3) Windows Media Player

4.2.3 วิธีการทดลอง

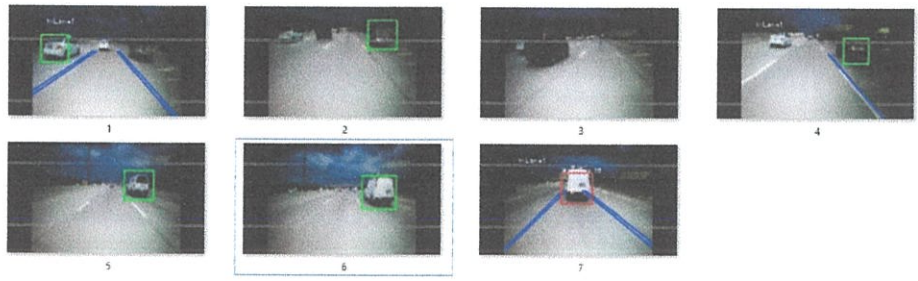
1) ทำการหาภาพที่มีวัตถุข้างหน้า หรือภาพที่มีการตรวจจับผิดพลาดในวิดีโอ

วิดีโอที่ 1 ได้ 8 เฟรม



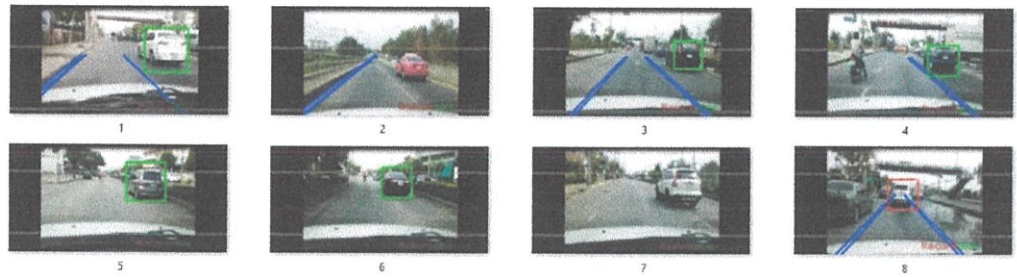
ภาพ 4.6 ตัวอย่างจากเฟรมวีดีโอที่ 1 สำหรับการทดลองที่ 2

- วิดีโอที่ 2 ได้ 7 เฟรม



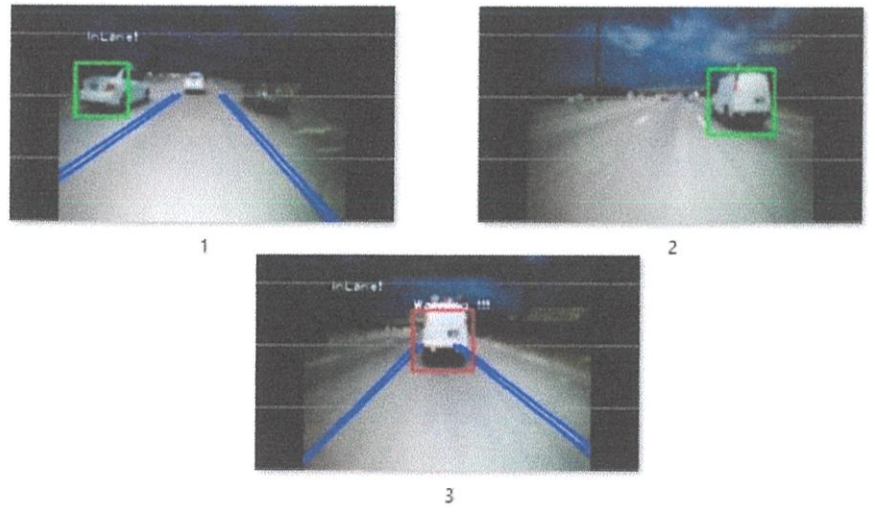
ภาพ 4.7 ตัวอย่างจากเฟรมวีดีโอที่ 2 สำหรับการทดลองที่ 2

- วิดีโอที่ 3 ได้ 8 เฟรม



ภาพ 4.8 ตัวอย่างจากเฟรมวีดีโอ 3 สำหรับการทดลองที่ 2

2) ทำการวิเคราะห์ความถูกต้องของภาพ



ภาพ 4.9 ตัวอย่างภาพที่ตรวจจับวัตถุได้ถูกต้อง สำหรับการทดลองที่ 2

n1

n2

n3

ภาพ 4.10 ตัวอย่างภาพที่ตรวจจับวัตถุได้ไม่ถูกต้อง สำหรับการทดลองที่ 2

3) บันทึกผลที่ได้ลงในตาราง 4.2

ตาราง 4.2 ผลการทดลองที่ 2

	วิดีโอที่ 1		วิดีโอที่ 2		วิดีโอที่ 3	
ภาพที่	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง
1	✓		✓		✓	
2	✓		✓			✓
3		✓		✓	✓	
4	✓		✓		✓	
5	✓		✓		✓	
6	✓		✓		✓	
7	✓		✓			✓
8		✓	-		✓	
รวม	6	2	6	1	6	2

3) คำนวณหา เปอร์เซ็นต์ความถูกต้อง

หาได้จาก (ภาพผลลัพธ์ที่ถูกต้อง * 100) / จำนวนภาพผลลัพธ์ทั้งหมด

วิดีโอที่ 1 = (6 * 100) / 8

= 75.00 %

วิดีโอที่ 2 = (6 * 100) / 7

= 85.71 %

วิดีโอที่ 3 = (6 * 100) / 8

= 75.00 %

ค่าเฉลี่ย = (75+85.71+75) / 3

= 78.57 %

4.2.4 ปัญหาและอุปสรรค

1) ภาพผลลัพธ์บางภาพทำการแยกแยะยากระหว่างภาพที่ดีกับภาพที่ไม่ดีความผิดพลาดบางส่วนเกิดจากเงาที่สะท้อนตอนทดสอบแอพพลิเคชั่นผ่านจอคอมพิวเตอร์

2) การตรวจจับรถยนต์บางครั้งเกิดความผิดพลาดเนื่องจาก ภาพรถยนต์ที่เข้ามาในกล้องนั้นเห็นรถยนต์ไม่เต็มคัน

4.2.5 การแก้ไข

1) เปลี่ยนไปใช้จอคอมพิวเตอร์แบบด้าน ที่ไม่สะท้อน

2) พัฒนาอัลกอริธึมให้สามารถตรวจจับรถยนต์จากด้านข้างได้

4.2.6 สรุปผลการทดลอง

จากการทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับวัตถุที่อยู่ข้างหน้า พบว่าความความถูกต้องของแอพพลิเคชั่นในการตรวจจับวัตถุอยู่ที่ 78.57 % พบว่าทั้งวิดีโอที่ได้จากการรันแอพพลิเคชั่นผ่านจอกับแบบรันของจริงมีความแม่นยำที่ใกล้เคียงกัน

4.3 การทดลองที่ 3 การหาความแม่นยำของระบบการแจ้งเตือน

การทดลองการหาความแม่นยำของระบบแจ้งเตือนของแอพพลิเคชั่นโดยการแจ้งเตือนจะเกิดขึ้นได้มี 2 กรณีคือ 1.รถยนต์ออกนอกเลนหรือเปลี่ยนเลน 2.พบวัตถุที่อยู่ข้างหน้าภายในระยะที่กำหนด โดยการทดลองจะทำการหาเปอร์เซ็นต์ของความถูกต้องในแจ้งเตือน แต่ละเฟรมที่นำมาใช้ทำการทดลองจะเลือกเฟรมที่มีการแจ้งเตือนปรากฏอยู่ โดยจะนำวิดีโอที่บันทึกได้จากแอพพลิเคชั่นมาทดสอบและคำนวณโดยใช้วิธีนับจำนวนการแจ้งเตือนที่พบทั้งหมดแล้วนำมาวิเคราะห์ โดยจะทำการทดลองจาก 3 วิดีโอและนำมาคิดหาค่าเฉลี่ย

วิดีโอที่ 1 และ 2 เป็นการรันแอพพลิเคชั่นผ่านหน้าจอคอมพิวเตอร์ แต่วิดีโอที่ 3 เป็นการรันแอพพลิเคชั่นบนรถของจริง

4.3.1 จุดประสงค์ในการทดลอง

เพื่อหาเปอร์เซ็นต์ของความถูกต้องในการแจ้งเตือนของแอพพลิเคชั่น

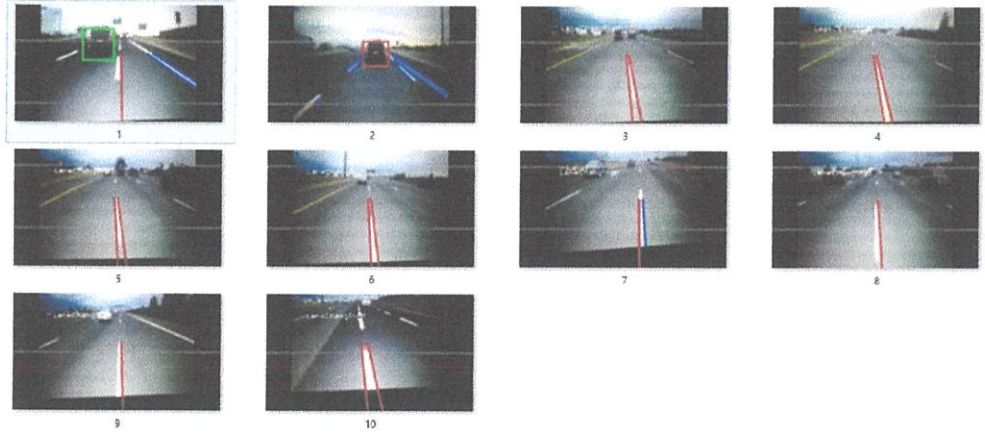
4.3.2 เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1) Asus Zenfone 5 LTE
2) คอมพิวเตอร์ระบบปฏิบัติการ Windows 8 ,CPU core 2 duo 2.26 GHz, Ram 4 GB
3) Windows Media Player

4.3.3 วิธีการทดลอง

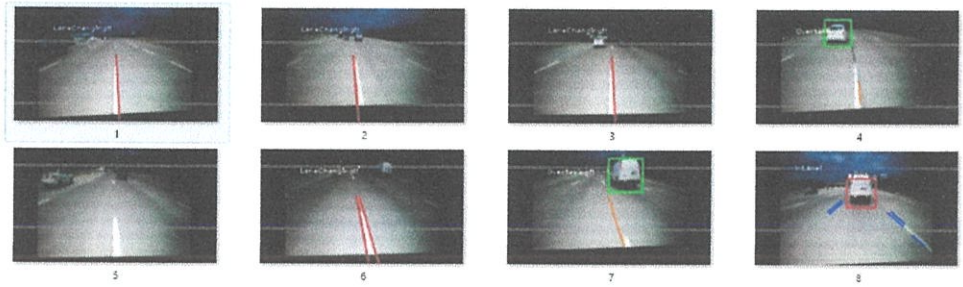
1) ทำการหาภาพที่มีการการแจ้งเตือน

วิดีโอที่ 1 ได้ 10 เฟรม



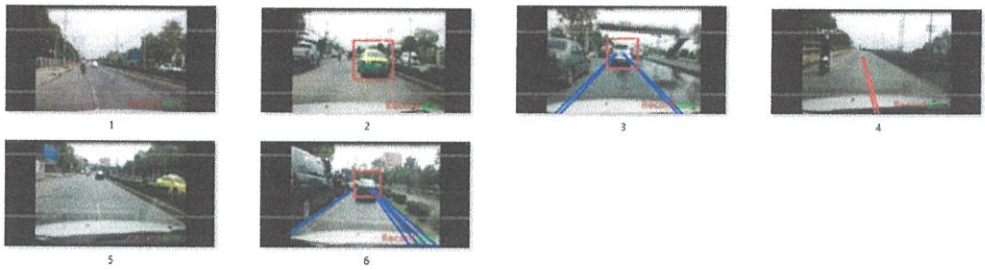
ภาพ 4.11 ตัวอย่างจากเฟรมวีดีโอที่ 1 สำหรับการทดลองที่ 3

- วิดีโอที่ 2 ได้ 8 เฟรม



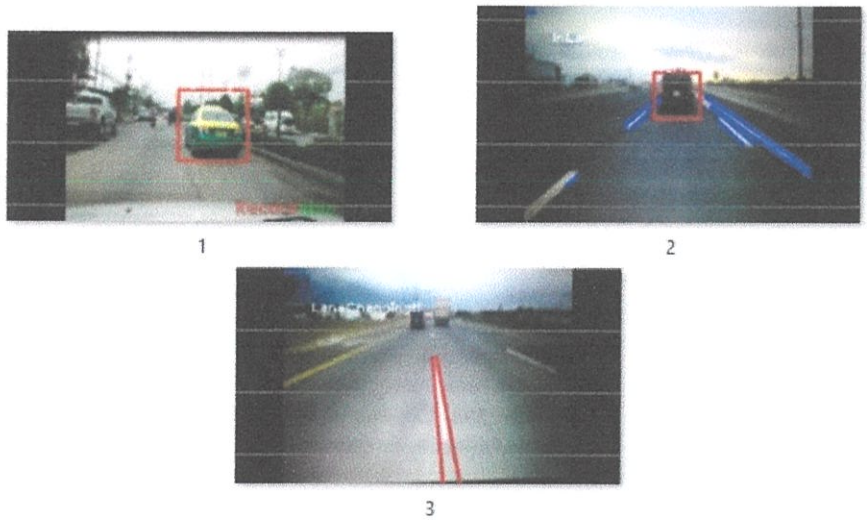
ภาพ 4.12 ตัวอย่างจากเฟรมวีดีโอที่ 2 สำหรับการทดลองที่ 3

- วิดีโอที่ 3 ได้ 6 เฟรม

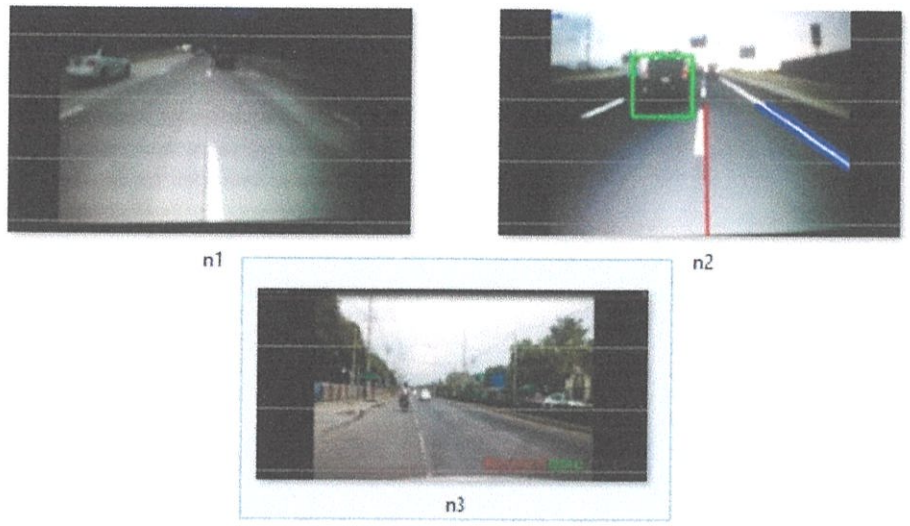


ภาพ 4.13 ตัวอย่างจากเฟรมวีดีโอ 3 สำหรับการทดลองที่ 3

2) ทำการวิเคราะห์ความถูกต้องของภาพ



ภาพ 4.14 ตัวอย่างภาพที่ทำการแจ้งเตือนได้ถูกต้อง สำหรับการทดลองที่ 3



ภาพ 4.15 ตัวอย่างภาพที่ทำการแจ้งเตือนได้ไม่ถูกต้อง สำหรับการทดลองที่ 3

3) บันทึกผลที่ได้ลงในตาราง 4.3

ตาราง 4.3 ผลการทดลองที่ 3

	วิดีโอที่ 1		วิดีโอที่ 2		วิดีโอที่ 3	
ภาพที่	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง	ผลลัพธ์ถูกต้อง	ผลลัพธ์ไม่ถูกต้อง
1		✓	✓			✓
2	✓		✓		✓	
3	✓		✓		✓	
4	✓		✓		✓	
5	✓			✓		✓
6	✓		✓		✓	
7		✓	✓		-	
8	✓		✓		-	
9	✓		-		-	
10	✓		-		-	
รวม	8	2	7	1	4	2

3) คำนวณหา เปอร์เซ็นต์ความถูกต้อง

หาได้จาก (ภาพผลลัพธ์ที่ถูกต้อง * 100) / จำนวนภาพผลลัพธ์ทั้งหมด

วิดีโอที่ 1 = (8 * 100) / 10

= 80.00 %

วิดีโอที่ 2 = (7 * 100) / 8

= 87.5 %

วิดีโอที่ 3 = (4 * 100) / 6

= 66.67 %

ค่าเฉลี่ย = (80+87.5+66.67) / 3

= 78.05 %

4.3.4 ปัญหาและอุปสรรค

1) การแจ้งเตือนบางครั้งเกิดความผิดพลาด เป็นผลมาจากประสิทธิภาพของสมาร์ทโฟนต่ำ ทำให้ประมวลผลไม่ทัน

2) การแจ้งเตือนบางครั้งเกิดความผิดพลาด แสงรบกวนที่เข้ามาในภาพ

4.3.5 การแก้ไข

1) ใช้สมาร์ทโฟนที่มีประสิทธิภาพสูงขึ้น หรือพัฒนาอัลกอริธึมให้ทำงานได้เร็วขึ้น

2) พัฒนาอัลกอริธึมให้ที่ช่วยลดแสงรบกวน

4.3.6 สรุปผลการทดลอง

จากการทดลองการหาความแม่นยำในการแจ้งเตือนของแอพพลิเคชั่น พบว่าความความถูกต้องของแอพพลิเคชั่นในการตรวจจับวัตถุอยู่ที่ 78.05 % พบว่าวิดีโอที่ได้จากการรันแอพพลิเคชั่นผ่านจอมีความแม่นยำมากกว่าแบบรันของจริง

4.4 สรุปผลการทดลองที่ได้จากผลการทดลองทั้งหมด

จากการทดลองทั้งหมดในโครงงาน สามารถสรุปผลการทดลองทั้งหมดได้ดังนี้

4.4.1 การทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับเส้นเลนถนน

สามารถตรวจจับเส้นเลนถนนได้ 50 ภาพจากทั้งหมด 66 ภาพ คิดเป็น 76.09 %

4.4.2 การทดลองการหาความแม่นยำของแอพพลิเคชั่นในการตรวจจับวัตถุที่อยู่ข้างหน้า

สามารถตรวจจับวัตถุที่อยู่ข้างหน้าได้ 18 ภาพจากทั้งหมด 23 ภาพ คิดเป็น 78.57%

4.4.3 การทดลองการหาความแม่นยำของระบบการแจ้งเตือน

สามารถทำการแจ้งเตือนได้ 19 ภาพจากทั้งหมด 24 ภาพ คิดเป็น 78.05%

บทที่ 5
วิเคราะห์และสรุปผล

5.1 สรุปผล

โครงงานนี้ได้พัฒนาจนสามารถใช้งานได้ตามฟังก์ชันการทำงานเบื้องต้น และขอบเขตที่กำหนดไว้ได้ดีในระดับหนึ่ง คือ ตรวจจับเส้นเลนของถนน ตรวจจับวัตถุที่อยู่ด้านหน้ารถ และสามารถแจ้งเตือนเมื่อรถออกนอกเลนหรือเข้าใกล้วัตถุที่อยู่ด้านหน้ารถในระยะทางที่กำหนดได้ แต่ยังสามารถทำงานได้ดีในสภาพแวดล้อมที่จำกัด คือ ตรวจจับได้ดีในตอนกลางวัน ผู้ขับควรขับรถด้วยความเร็วไม่เกิน 60 กิโลเมตรต่อชั่วโมง ส่วนระยะเวลาในการแจ้งเตือนเมื่อรถด้านหน้าเข้าใกล้ในระยะที่กำหนด และเมื่อเกิดการเปลี่ยนเลนนั้น ควรจะสัมพันธ์กับความเร็วของรถ ระยะในการตรวจจับวัตถุควรจะคำนวณจากมุมก้มของกล้อง และความสูงของรถที่ติดตั้งอุปกรณ์แอนดรอยด์เพื่อให้การแจ้งเตือนทำได้ถูกต้องและแม่นยำยิ่งขึ้น ด้านการพัฒนาต่อยอด ควรพัฒนาให้สามารถตั้งค่ากำหนดความเร็วที่ใช้ในการขับขี่ ช่วงเวลาในการขับรถ ความกว้าง และความสูงของรถ เพื่อใช้ในการคำนวณให้ระบบทำงานได้ดีในปัจจัยที่อยู่นอกเหนือจากขอบเขตที่กำหนด นอกจากนี้สามารถนำโครงการนี้ไปต่อยอดในการพัฒนาเทคโนโลยีที่เกี่ยวข้องเช่น การพัฒนารถยนต์ไร้คนขับ

5.2 ปัญหาและอุปสรรค

5.2.1 ความเร็วในการประมวลผลวีดีโอ

เนื่องจากภาพของกล้องจากอุปกรณ์ที่ใช้ที่ความละเอียด 1280 x 720 พิกเซล ซึ่งทำให้ใช้เวลาในการประมวลนาน จนผลที่ได้ไม่เป็นเรียลไทม์ แนวทางการแก้ปัญหาคือ ปรับปรุงขั้นตอนวิธีการที่ใช้การประมวลผล รวมถึงปรับคุณภาพของภาพที่นำมาประมวลผลให้เหมาะสม ทำให้สามารถประมวลผลข้อมูลได้อย่างถูกต้อง ภายในเวลาที่ต้องการได้

5.2.2 ความซับซ้อนของไลบรารี่ที่ใช้

การประมวลผลภาพและวิเคราะห์ข้อมูลต่างๆ ได้ใช้ ไลบรารี่ OpenCV ในการพัฒนา รวมถึงตัวโปรแกรมซึ่งเป็นแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ ทำให้ต้องศึกษาการพัฒนาแอพพลิเคชันบนแอนดรอยด์ และการพัฒนาโดยใช้ไลบรารี่ OpenCV การพัฒนาไลบรารี่ OpenCV บนระบบปฏิบัติการแอนดรอยด์ แตกต่างจากการใช้ OpenCV ในแพลทฟอร์มอื่น

พอสมควร ทำให้จำเป็นต้องศึกษาโครงสร้างการทำงานของฟังก์ชันต่างๆให้เข้าใจ แล้วจึงเริ่มพัฒนาระบบ

5.2.3 สัญญาณรบกวนในรูปภาพ

การประมวลผลภาพจำเป็นต้องจัดการกับสัญญาณรบกวนที่เกิดจากการถ่ายภาพเพื่อให้การประมวลผลข้อมูลได้ผลลัพธ์ถูกต้องตามที่ต้องการมากที่สุด ซึ่งจำเป็นต้องศึกษาและเลือกใช้วิธีการกำจัดสัญญาณรบกวนในภาพวิธีต่างๆ และทดสอบผลลัพธ์ที่ได้จนกระทั่งได้ผลลัพธ์ที่ต้องการ

5.3 แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นต่อไป

1) ปรับปรุงให้สามารถตรวจจับเลนถนนและ รถยนต์ที่อยู่ด้านหน้าได้ดีขึ้น รวมถึงพัฒนาให้สามารถตรวจจับวัตถุที่อยู่ด้านหน้าได้ทุกชนิด วิเคราะห์ชนิดของวัตถุที่อยู่ด้านหน้า และสามารถแจ้งเตือนโดยที่ระบุชนิดของวัตถุได้ ส่วนของตรวจจับเลนถนนควรปรับปรุงโดยการใช้ Spline Model ซึ่งเป็นอัลกอริทึมที่ตรวจสอบความต่อเนื่องของเส้น จะทำให้แอพพลิเคชันสามารถตรวจจับเส้นเลนที่เป็นเส้นโค้งได้ดีกว่าการใช้ Hough Transform
2) ปรับปรุงให้ระบบสามารถวัดระยะห่างระหว่างรถกับวัตถุที่อยู่ด้านหน้าเป็นตัวเลขได้
3) สามารถนำไปประยุกต์ใช้กับการพัฒนารถยนต์แบบไร้คนขับได้โดยนำแอพพลิเคชันไปใช้ในการตรวจจับตำแหน่งของรถในเลน และตรวจจับวัตถุที่อยู่ด้านหน้ารถเพื่อป้องกันอุบัติเหตุได้

บรรณานุกรม

[1] OpenCV dev team. **"OpenCV 2.4.9.0 Documentation"**

[Online] Available : http://docs.opencv.org/index.html

[2] Google. **"Android SDK"** [Online] Available : https://developer.android.com/sdk

[3] Hang-Bong Kang. **"Various Approaches for Driver and Driving Begavior Monitoring"**

[Online] Available : http://www.cs.dartmouth.edu/~lorenzo/Papers/mobisys13.pdf

[4] Jay Ramphia. **"Hough Transform"**

[Online] Available http://www.jayrambhia.com/blog/hough-transform/

[5] Abhishek Kumar Annamraju **"Train cascade and car using OpenCV"**

[Online] Available : http://abhishek4273.com/2014/03/16/traincascade-and-car-detection-using-opencv/