

โปรแกรมตรวจสอบสัญญาณบ่งบอกอาการง่วงนอน จากลักษณะดวงตา

นริศา ชุตินารา,* ชยางกูร จันทวิชัย, ทิพย์ภา ชัยงาม, ภัททิรัญ ตนประเสริฐกุล,
รุ่งระพี กรานคำยี่, สาโรช เมลานนท์ และ ศิริสุข เทียนรุ่งโรจน์

บทคัดย่อ

อาการง่วงนอนเป็นสาเหตุหลักที่ก่อให้เกิดอุบัติเหตุทางรถยนต์ ผู้วิจัยตระหนักถึงปัญหาจึงได้พัฒนาโปรแกรมตรวจสอบอาการง่วงนอนซึ่งจะมีเสียงแจ้งเตือนผู้ขับขี่เป็นระยะเพื่อป้องกันผู้ขับขี่จากอุบัติเหตุ โดยติดตั้งกล้องไว้ในรถเพื่อบันทึกภาพวิดีโอของผู้ขับขี่ การตรวจจับใบหน้าใช้อัลกอริทึมวิเคราะห์สีผิว และการตรวจหาดวงตาใช้หลักการหาพื้นที่ที่พบพิกเซลสีขาวบนใบหน้า งานวิจัยนี้อธิบายถึงวิธีติดตามสถานะดวงตาเพื่อหาว่าตาเปิดหรือปิด ถ้าพบว่าตาปิดค่าความง่วงนอนจะเพิ่มขึ้น ในทางตรงกันข้ามค่าความง่วงนอนจะลดลงถ้าค่าความง่วงนอนมีค่ามากกว่าหรือเท่ากับค่าเกณฑ์ที่กำหนดระบบจะส่งสัญญาณเตือนอย่างต่อเนื่อง ผู้วิจัยได้ทดสอบระบบกับภาพวิดีโอ 9 คลิป และค่าพิสัยภาพสี YCbCr จำนวน 3 ช่วง พบว่าการวิเคราะห์สถานะดวงตาถูกต้อง 69.71% และวิเคราะห์สถานะความง่วงนอนได้ถูกต้อง 67.6% สุดท้ายระบบยังมีข้อผิดพลาดอยู่ ในกรณีที่ภาพวิดีโอมีแสงเข้ามาทำให้เกิดพื้นหลังเป็นสีดำและความเข้มของแสงไม่เท่ากัน และกรณีที่ระบบตรวจพบคิ้วส่งผลให้สถานะดวงตาเป็นเปิดซึ่งไม่ถูกต้อง

คำสำคัญ: อัลกอริทึมวิเคราะห์สีผิว ภาพสี YCbCr การวิเคราะห์สถานะดวงตา ค่าวัดความง่วงนอน

Driver Drowsiness Detection Program through Eye Motion Analysis

Narisa Chutinara*, Chayangkoon Chantaravitchai, Thipnapha Saingam,
Pattirun Thronpasertkul, Roongrapee Krancomyee,
Saroch Maolanont and Sirinoot TienRunroj

ABSTRACT

Drowsy driving is one of the main reasons causing traffic accidents. This paper aims to develop a program to detect driver drowsiness and alert drivers timely to prevent them from accidents. In order to analyze driver's eyes movement, a small camera is mounted inside the car to capture the driver's face. The program detects the face using pixel-based skin color algorithm and distinguishes the eyes by finding area of white pixels on face. This paper describes a method of tracking the eyes' status whether they are closed or open. If the eyes are found closed, the drowsiness index will be increased, on the other hand, the drowsiness index will be decreased, if the eyes become open. Whenever the drowsiness reaches the threshold, the system will issue a warning signal continuously. From the experiment with 9 video clips and 3 ranges of YCbCr colour, it shows that the system is able to analyze eye status with 69.71% accuracy and detect driver drowsiness with 67.67% accuracy across subjects. The system still has some errors when source of light is not consistency and when the system detects eyebrows, the system indicates the eye status is open which is not correct.

Keywords: Skin color Algorithm, YCbCr image, eye state analysis, drowsiness index (Peakdow)

บทนำ

ในปัจจุบันนี้เรามักได้ยินเรื่องราวที่เกี่ยวกับการเกิดอุบัติเหตุจากการขับขี่เป็นจำนวนมาก ทำให้มีผู้ได้รับอันตรายจากการเกิดอุบัติเหตุอันเนื่องมาจากสาเหตุความประมาทหรือการพักผ่อนไม่เพียงพอของผู้ขับขี่ แนวทางหนึ่งที่จะสามารถช่วยลดอุบัติเหตุที่เกิดจากอาการง่วงนอนนี้ได้คือการแจ้งเตือนผู้ขับขี่ขณะขับรถ โดยใช้เสียงแจ้งเตือนเป็นระยะๆ งานวิจัยนี้สามารถตรวจสอบอาการง่วงนอนของคนขับรถและมีสัญญาณแจ้งเตือนในเวลาที่เหมาะสม เพื่อช่วยป้องกันการเกิดอุบัติเหตุจำนวนมากและประหยัดงบประมาณในการรักษาผู้บาดเจ็บจากอุบัติเหตุโดยติดตั้งกล้องขนาดเล็กไว้ในรถเพื่อตรวจสอบใบหน้าผู้ขับขี่ โปรแกรมตรวจจับสัญญาณบ่งบอกอาการง่วงนอนจากลักษณะของดวงตาสามารถนำไปประยุกต์ใช้ในอุตสาหกรรมการผลิตรถยนต์โดยฝังโปรแกรมไว้ในอุปกรณ์หน้ารถ

องค์ความรู้ที่เกี่ยวข้องและอุปกรณ์เครื่องมือที่ใช้ในการพัฒนาระบบ

1. งานวิจัยที่เกี่ยวข้อง

งานวิจัยของ Devi และคณะ [1] ตรวจสอบอาการง่วงนอนของผู้ขับขี่และมีสัญญาณแจ้งเตือนในเวลาที่เหมาะสม โดยการติดตั้งกล้องขนาดเล็กไว้ในรถเพื่อตรวจสอบใบหน้าผู้ขับขี่ ขั้นตอนการทำงานคือรับภาพเคลื่อนไหวซึ่งเป็นภาพสีจากกล้องวิดีโอ ตำแหน่งใบหน้าคนขับอยู่ที่จุดศูนย์กลางของกล้อง โดยกล้องมีความละเอียด 352×288 พิกเซล การตรวจจับใบหน้า (Face Detection) กระทำโดยตัดภาพนิ่งที่เป็นภาพสีแบบ RGB (Red Green Blue) จากวิดีโอ จากนั้นทำการแยกใบหน้าออกจากสีพื้นหลัง โดยใช้อัลกอริทึมการวิเคราะห์สีผิว (Skin Color Algorithm) และนำภาพสีแบบ RGB ที่รับเข้ามาเปลี่ยนเป็นภาพ YCbCr

การเปลี่ยนภาพสีแบบ RGB ให้เป็น YCbCr เป็นไปดังสมการที่ (1)

$$\left. \begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.169R - 0.322G + 0.500B \\ Cr &= 0.500R - 0.419G - 0.081B \end{aligned} \right\} \quad (1)$$

โดย Y คือ ค่าความเข้มของแสง และ Cb, Cr คือ ค่าความเข้มของแสงสีน้ำเงินและสีแดง

ถ้าค่า Cb, Cr ของแต่ละพิกเซลอยู่ในพิสัยที่ (2) และ (3) จึงจำแนกว่าเป็นสีผิว

$$76 < Cb < 126 \quad (2)$$

$$130 < Cr < 172 \quad (3)$$

เมื่อพบสีผิวแล้วจึงตรวจจับหาใบหน้าโดยใช้คุณสมบัติการหาพื้นที่สีเหลี่ยมผืนผ้าที่ครอบคลุมบริเวณใบหน้า จากนั้นตรวจหาดวงตา (Eyes Detection) โดยแบ่งรูปเป็น 2 ส่วนคือครึ่งบนและครึ่งล่าง แล้วเติมสีดำลงในภาพครึ่งล่าง หลังจากนั้นตีกรอบสีเหลี่ยมรอบบริเวณดวงตาในภาพครึ่งบน แล้วค้นหาดวงตาโดยใช้ Circular Hough Transform เพื่อค้นหาตำแหน่งดวงตา เมื่อเจอแล้วก็ทำวงกลมรอบดวงตาทั้งสอง

การประเมินสถานะดวงตาวิเคราะห์โดยวัดระยะห่างระหว่างดวงตาทั้ง 2 ดวง และระยะห่างของเฟรมปัจจุบันกับเฟรมก่อนหน้านั้น

โดย eyedist คือ ระยะห่างของดวงตาทั้งสองในเฟรมแรก

d คือ ระยะห่างระหว่างดวงตา 2 ดวงในเฟรมปัจจุบัน

dd คือ ความแตกต่างของค่าเฟรมปัจจุบันและค่าเฟรมก่อนหน้านั้น

กำหนดให้ค่าที่รับได้ (tolerance value) เท่ากับ 10 สำหรับใช้เป็นเงื่อนไขในการระบุสถานะดวงตา ถ้าเงื่อนไขนี้เป็นจริงแสดงว่าดวงตาเปิด

if (d > (eyedist - 10) && dd < 10)

แต่ถ้าดวงตาปิดจะไม่สามารถวงกลมได้จึงทำให้การคำนวณระยะห่างระหว่างดวงตาทั้งสองผิดพลาด จึงใช้ค่าเฟรมปัจจุบันและค่าเฟรมก่อนหน้ามาหาความต่างต่างกัน (dd) ซึ่งถ้า $dd \geq 10$ แสดงว่าดวงตาปิด

การตรวจสอบอาการง่วงนอน เมื่อมีภาพที่ตาปิดติดต่อกัน 8 ภาพ แสดงว่ามีอาการง่วงนอน เมื่อตรวจเจออาการง่วงนอนจะส่งสัญญาณแจ้งเตือนเป็นเสียงร้อง

ซึ่งงานวิจัยของ Devi [1] นั้นได้ตรวจหาอาการง่วงนอนของผู้ขับขี่โดยใช้อัลกอริทึมวิเคราะห์สีผิวเพื่อตรวจหาใบหน้าและใช้ Circular Hough Transform เพื่อตรวจหาดวงตา แต่ยังคงมีข้อจำกัดตรงที่พื้นหลังของภาพมีสีดำจึงทำให้การตรวจหาใบหน้าและดวงตาผิดพลาด ส่งผลให้สัญญาณเตือนไม่ถูกต้อง จากปัญหาดังกล่าวผู้วิจัยจึงได้กำหนดขอบเขตขนาดพิกเซลของภาพที่รับเข้ามาให้มากขึ้น แสงที่เข้ามามีระดับเท่ากันและการพัฒนาโปรแกรมก็กำหนดค่าพิสัยของสี YCbCr เป็น 3 ช่วงแล้วทำการทดสอบเพื่อเลือกช่วงที่ดีที่สุดที่เหมาะสมสำหรับคนสีผิวสองสี

2. ทฤษฎีที่ใช้ในงานวิจัย

แบบจำลองสี RGB [2]

แบบจำลองสีนี้เป็นแบบจำลองของแม่สีหลัก 3 สี (Primary Colors) ประกอบด้วยสี 3 สี ได้แก่ แดง (Red) เขียว (Green) และน้ำเงิน (Blue) เมื่อนำมาผสมผสานกันทำให้เกิดสีต่างๆ ซึ่งใกล้เคียงกับสีที่ตาเรามองเห็น ถ้าสีมีความเข้มมากเมื่อนำมาผสมกันจะทำให้เกิดเป็นสีขาว จึงเรียกระบบสีนี้ว่าแบบ Additive หรือการผสมสีแบบบวกปกติถูกนำมาใช้งานในด้านการแสดงผลข้อมูลบนจอภาพ รวมไปถึงการเก็บข้อมูลภาพในระบบคอมพิวเตอร์

แบบจำลองสี YCbCr

แบบจำลองสี YCbCr เป็นวิธีหนึ่งของการเก็บข้อมูลสีของระบบภาพและวิดีโอดิจิทัล เป็นการเอาภาพ RGB มาเข้ารหัส ส่วนการถอดรหัสก็จะทำโดยเอาข้อมูลจาก YCbCr มาเทียบกับตารางสี RGB อีกที ความแตกต่างของ YCbCr กับ RGB คือการแสดงผลของ YCbCr จะใช้ความสว่าง และ ค่าความต่างสี ส่วน RGB แสดงสีแดง สีเขียว และสีน้ำเงิน โดยที่ Y มาจากความสว่าง (Luminance) Cb คือสีน้ำเงินที่ตัดความสว่างออกไป (B - Y) และ Cr คือ สีแดงที่ตัดความสว่างออกไป (R - Y) ซึ่งเป็นโครงสร้างสีที่ใช้กันในการเข้ารหัสแบบ MPEG และ JPEG ระบบสี YCbCr ถูกพัฒนาขึ้นเพื่อใช้กับภาพวิดีโอดิจิทัล ซึ่ง Basilio และคณะ [3] ได้สรุปค่าความสว่าง Y มีค่าอยู่ระหว่าง 16-235 ส่วน Cb และ Cr จะมีค่าอยู่ระหว่าง 16-240

การเปลี่ยนภาพ RGB เป็น YCbCr แสดงดังสมการ (4) [3]

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (4)$$

จากการทดลองระบบสี YCbCr นั้น [3] เมื่อเก็บค่า Y จะพบว่าถ้า Y มีค่าตั้งแต่ 50 ลงไป ภาพจะมีดเกินไป แต่ถ้า Y มีค่ามากกว่า 240 ภาพจะสว่าง ดังนั้นในการนำภาพมาวิเคราะห์จึงเลือกใช้ภาพที่ Y มีค่ามากกว่า 80

Kukharev และคณะ [4] พบว่าค่าพิสัย CbCr ในช่วง $85 < Cb < 135$ และ $135 < Cr < 180$ จะเหมาะกับคนผิวขาว ส่วน Chai และ Ngai [5] ได้พัฒนาอัลกอริทึมให้ดีขึ้นเพื่อวิเคราะห์สีผิวมนุษย์พบว่าช่วง Cb และ Cr ที่ดีที่สุดคือ $77 \leq Cb \leq 127$ และ $133 \leq Cr \leq 173$ ผู้วิจัยได้นำค่าพิสัยตามงานของ Chai [5] มาทดลองแล้วพบว่าค่าพิสัยดังกล่าวไม่สามารถใช้ร่วมกับคนสีผิวสองสีได้อย่างถูกต้อง ส่วนค่าพิสัย ใน [1] $76 < Cb < 126$ และ $130 < Cr < 172$ เหมาะกับคนสีผิวคล้ำกลางๆ และค่าพิสัยใน [3] $80 \leq Cb \leq 120$ และ $133 \leq Cr \leq 173$ เหมาะกับคนผิวสองสี

ดังนั้นในงานวิจัยนี้จึงได้ทดสอบค่า Cb และ Cr เป็น 3 ช่วงดังนี้

ช่วงที่ 1 Cb มีค่าอยู่ระหว่าง 76 ถึง 126

Cr มีค่าอยู่ระหว่าง 130 ถึง 172

ช่วงที่ 2 Cb มีค่าอยู่ระหว่าง 77 ถึง 127

Cr มีค่าอยู่ระหว่าง 133 ถึง 177

ช่วงที่ 3 Cb มีค่าอยู่ระหว่าง 80 ถึง 120

Cr มีค่าอยู่ระหว่าง 133 ถึง 173

และได้เกณฑ์ช่วงที่ดีที่สุดคือช่วงที่ 3 คือ Cb มีค่าอยู่ระหว่าง 80 ถึง 120 และ Cr มีค่าอยู่ระหว่าง 133 ถึง 173

การแปลงภาพ YCbCr เป็นภาพขาวดำโดยใช้ค่าเกณฑ์ (threshold)

การแปลงภาพที่มีค่าพิกเซลอยู่ระหว่าง 0-255 ให้เป็นภาพที่มีค่าเพียงสองระดับ (Binary Image) [6] โดยมีเงื่อนไขว่า ถ้าค่าพิกเซลนั้นมีค่าต่ำกว่าค่า threshold ให้จุดภาพนั้นมีค่าเป็น 0 หรือเป็นสีดำ แต่ถ้าค่าพิกเซลนั้นมีค่าสูงกว่าหรือเท่ากับค่า threshold ให้จุดภาพนั้นมีค่าเป็น 255 หรือสีขาว

$$g(x,y) = 0 \quad \text{if } f(x,y) < \text{threshold value}$$

$$g(x,y) = 255 \quad \text{if } f(x,y) \geq \text{threshold value}$$

โดย $f(x,y)$ คือ ค่าพิกเซลของภาพต้นฉบับในตำแหน่ง (x,y)

$g(x,y)$ คือ ค่าพิกเซลของภาพผลลัพธ์ในตำแหน่ง (x,y)

การขยายขนาดข้อมูลภาพ (Dilation)

การขยายขนาดข้อมูลภาพใช้สัญลักษณ์ \oplus เป็นการขยายพิกเซลที่สว่างให้ใหญ่ขึ้นสำหรับรูปภาพแบบขาวดำหรือสีเทาเพื่อจุดประสงค์บางประการเช่นเพื่อปิดรูเล็กๆ ในวัตถุหรือใช้เพื่อช่วยให้วัตถุ 2 วัตถุที่ไม่มีสมาชิกร่วมกันแต่อยู่ใกล้กันสามารถเชื่อมต่อกันได้เป็นต้น การขยายขนาดทำได้โดยวางสมาชิโครงสร้าง (Structuring Element) ลงบนภาพแล้วเลื่อนสมาชิโครงสร้างมีขั้นตอนดังนี้

1. ถ้าจุดศูนย์กลางของสมาชิโครงสร้างตรงกับค่า '0' ในภาพไม่ต้องดำเนินการใดๆ แล้วให้เลื่อนสมาชิโครงสร้างไปยังจุดภาพถัดไป
2. ถ้าจุดศูนย์กลางของสมาชิโครงสร้างตรงกับค่า '1' ในภาพให้ดำเนินการด้วยตัวดำเนินการ OR (หรือ) ระหว่างภาพกับสมาชิโครงสร้าง

การกร่อนข้อมูลภาพ (Erosion)

การกร่อนข้อมูลภาพใช้สัญลักษณ์ \otimes เป็นการดำเนินการเช่นเดียวกับการขยายขนาดแต่ให้ลักษณะตรงกันข้ามคือลดขนาดของพิกเซลที่สว่างลง การกร่อนจะลดขนาดของวัตถุให้เล็กลงมีการดำเนินการดังนี้

1. ถ้าจุดศูนย์กลางของสมาชิโครงสร้างตรงกับค่า '0' ในภาพ ไม่ต้องดำเนินการใดๆ และให้เลื่อนสมาชิโครงสร้างไปยังจุดภาพถัดไป
2. ถ้าจุดศูนย์กลางของสมาชิโครงสร้างตรงกับค่า '1' ในภาพ ให้พิจารณาว่าถ้ามีจุดภาพค่า '1' ในสมาชิโครงสร้างเลยออกมาจากวัตถุซึ่งมีค่า '1' ในภาพให้เปลี่ยนค่า '1' ณ ตำแหน่งที่เป็นจุดศูนย์กลางในภาพเป็นค่า '0'

การเปิดและปิดภาพ (Opening and Closing)

การเปิดภาพ (Opening) ใช้สัญลักษณ์ \circ เป็นการกำจัดรายละเอียดขนาดเล็กของภาพคือกระบวนการกร่อนข้อมูลภาพ (Erosion) จากนั้นจึงทำการขยายขนาดข้อมูลภาพ (Dilation) ซึ่งผลลัพธ์จากการทำ Opening จะทำให้พิกเซลของภาพถูกเปิดออกมากขึ้น ดังสมการต่อไปนี้

$$A \circ B = (A \otimes B) \oplus B$$

โดย A คือ ภาพต้นฉบับ และ B เป็นสมาชิโครงสร้าง (Structuring element)

การปิดภาพ (Closing) ใช้สัญลักษณ์ \bullet ทำในวิธีตรงข้ามกับ Opening จะเป็นการทำให้ภาพมีการเชื่อมต่อกันมากขึ้นและการทำ Closing จะทำให้พิกเซลของภาพจะถูกปิดเชื่อมต่อกันมากขึ้นวิธีการทำ Closing คือการทำ Dilation ก่อนจึงทำ Erosion ดังสมการต่อไปนี้

$$A \bullet B = (A \oplus B) \otimes B$$

อุปกรณ์และวิธีการทดลอง

ซอฟต์แวร์ (Software) ดังต่อไปนี้

1. โปรแกรม Visual Studio 2015 พัฒนาขึ้นโดยไมโครซอฟท์เป็นเครื่องมือที่ช่วยทำเว็บไซต์ เว็บแอปพลิเคชัน และ เว็บเซอร์วิส ในปัจจุบัน Visual Studio นั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษา ดอตเน็ตในโปรแกรมเดียวกัน
2. OpenCV- 3.1.0 เป็น Library ในภาษา C++ และ Python สำหรับการพัฒนาโปรแกรมที่เกี่ยวข้องกับ Image Processing และ Computer Vision

ฮาร์ดแวร์ (Hardware) ใช้เครื่องมือดังต่อไปนี้

1. คอมพิวเตอร์ระบบปฏิบัติการวินโดวส์
2. โทรศัพท์มือถือ Samsung galaxy grand II กล้องดิจิทัล 8 ล้านพิกเซล

วิธีดำเนินงาน

อธิบายการทำงานตามผังงานในภาพที่ 1 ดังนี้

1. นำเข้าข้อมูลเป็นภาพเคลื่อนไหว

ทดสอบโดยนำภาพวิดีโอจากโทรศัพท์มือถือ Samsung galaxy grand II มีความละเอียด 8 ล้านพิกเซล แล้วตัดภาพนิ่งออกจากภาพวิดีโอ

2. แปลงภาพ RGB ให้อยู่รูปแบบ YCbCr

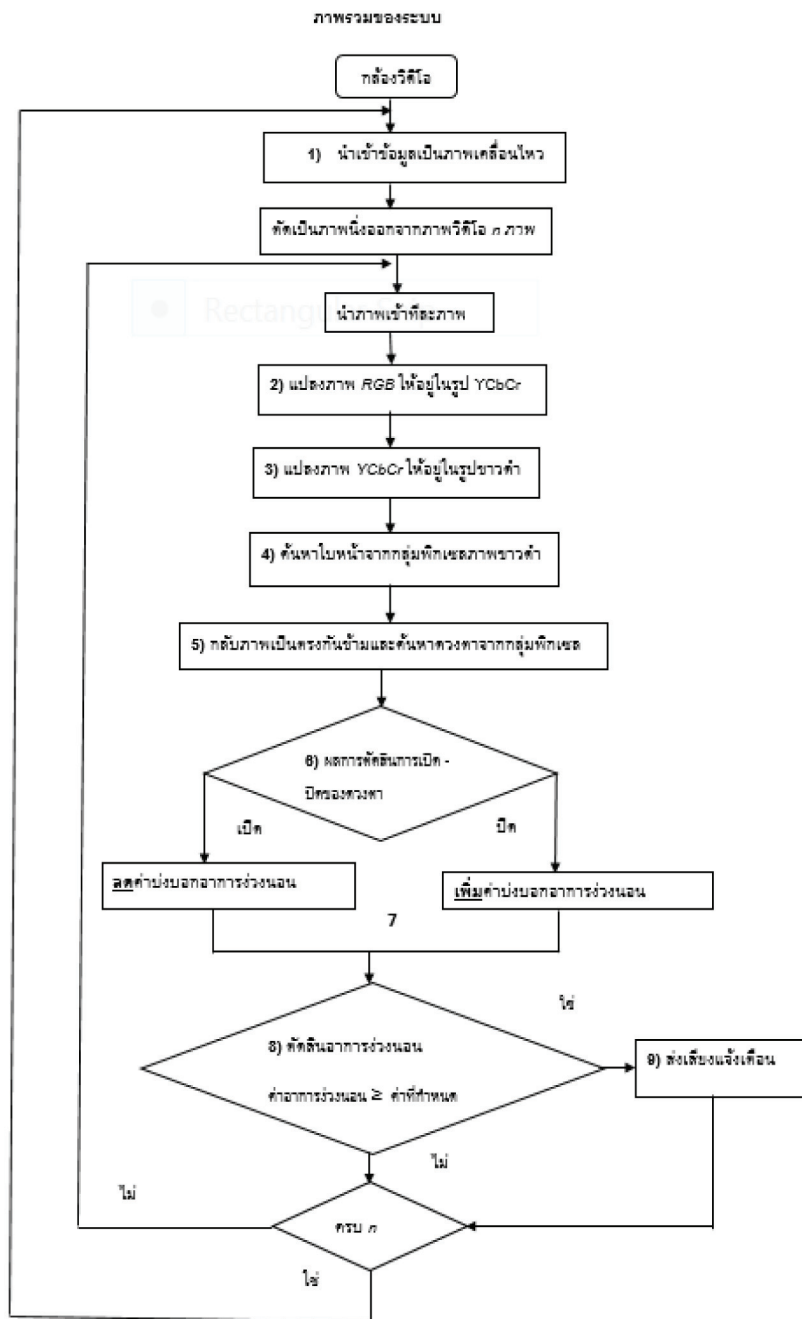
เพื่อลดผลกระทบจากแสงโดยแปลงภาพ RGB เป็น YCbCr ดังสมการ (4) ได้ผลลัพธ์ดังภาพ 2

3. แปลงภาพ YcbCr ให้อยู่รูปขาวดำโดยใช้ค่าเกณฑ์ (threshold)

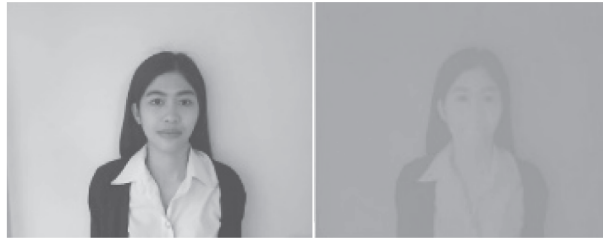
ทำการเปลี่ยนภาพ YCbCr ให้เป็นภาพขาวดำโดยใช้ค่าเกณฑ์ (threshold) ดังภาพที่ 3 การระบุส่วนสีผิวมนุษย์นั้นจะทำการตรวจสอบค่าพิกเซล Cb และ Cr ว่าอยู่ในค่าพิสัยที่กำหนดหรือไม่ โดยในระบบได้กำหนดพิสัย Cb และ Cr เป็น 3 ช่วงดังระบุในทฤษฎีที่ใช้ในงานวิจัย แล้วทำการทดสอบเพื่อหาช่วงที่เหมาะสมที่สุด

4. ค้นหาใบหน้าจากภาพขาวดำ

โดยทำการ opening ภาพจะได้ค่าพิกเซลสว่างเป็นสีผิว กำหนดให้เป็นใบหน้า ดังภาพที่ 4 และแบ่งกลุ่มพิกเซลออกจากกันเพื่อป้องกันไม่ให้เกิดกลุ่มพิกเซลที่เป็นใบหน้าเชื่อมต่อกับสิ่งแวดล้อมอื่นที่อยู่ติดกัน



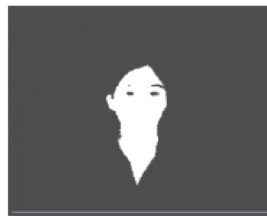
ภาพที่ 1 ผังงานของระบบ



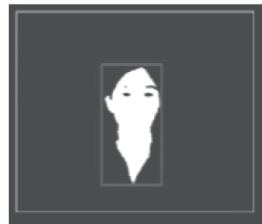
ก) รูป RGB

ข) รูป YCbCr

ภาพที่ 2 แปลงภาพ RGB ให้อยู่ในรูป YCbCr



ภาพที่ 3 ภาพขาวดำที่ได้หลังจากการแปลงจาก YCbCr



ภาพที่ 4 กำหนดกลุ่มพิกเซลสว่างที่มีขนาดใหญ่ที่สุดเป็นใบหน้า

5. ค้นหาดวงตาจากกลุ่มพิกเซลใบหน้า

เมื่อได้ใบหน้าในภาพที่ 8 แล้วให้ reverse ภาพขาวดำอีกครั้งดังภาพที่ 5 ในบริเวณใบหน้าจะได้ค่าพิกเซลสว่างเป็นดวงตา เพื่อให้กลุ่มพิกเซลดวงตาที่มีความสมบูรณ์มากขึ้นค่าความสว่างต้องอยู่บนใบหน้าเท่านั้น

6. การตัดสินผลการเปิด-ปิดของดวงตา

โดยแบ่งใบหน้าเป็นครึ่งซ้ายและครึ่งขวา และหาว่ามีกลุ่มพิกเซลที่อยู่ซ้ายบนและขวาบนของใบหน้าหรือไม่ ถ้ามีทั้งสองข้างจะคืนค่าเป็น true (ดวงตาเปิด) และถ้าไม่จะคืนค่าเป็น false (ดวงตาปิด)

7. การเพิ่ม-ลดค่าวัตถุการง่วงนอน (Peakdow)

peakdow คือ ค่าวัตถุการง่วงนอน เมื่อได้ผลการเปิด-ปิดดวงตาแล้วก็จะนำผลที่ได้มาเพิ่ม-ลดค่าวัตถุการง่วงนอน โดยจะเพิ่มค่าเมื่อสถานะดวงตาปิด และลดค่าเมื่อสถานะดวงตาเปิด แต่จะไม่ลดลงมาต่ำกว่า 0 ดังโค้ดในภาพที่ 7 และจะถูกรีเซ็ตกลับเป็น 0 ใหม่ เมื่อดวงตาเปิดติดต่อกันโดยไม่มีดวงตาปิดเลยเป็นระยะเวลาหนึ่งดังภาพที่ 6

8. ตัดสินอาการรบกวน

จะตัดสินใจให้ผู้ขับชื้ออยู่ในอาการรบกวนเมื่อค่า peakdow มีมากกว่าหรือเท่ากับค่า threshold ก็จะส่งเสียงร้องเตือนโดยค่า threshold นี้จะแปรผันตามจำนวนเฟรมต่อวินาทีของภาพวิดีโอที่นำเข้า ดังนั้น $threshold = fps \times second$ (fps : frame per second) ในที่นี้ใช้ 1 วินาที เพราะ 1 วินาทีเป็นเวลาที่เหมาะสมอยู่ในขั้นที่ต้องแจ้งเตือนแล้ว แต่ถ้ามากกว่า 1 วินาที ค่า threshold จะสูงเกินไปทำให้ส่งสัญญาณเตือนซ้ำ

9. ส่งเสียงแจ้งเตือน

ออกแบบรูปแบบของสัญญาณเตือนในระยะเวลา 1000 มิลลิวินาทีด้วยความถี่ 300 เฮิรตซ์และค่าระยะเวลาจะลดลงเมื่อค่าวัดอาการรบกวน (peakdow) มากขึ้นทำให้มีการเตือนถี่ขึ้นเรื่อยๆ ดังโค้ดภาพที่ 8



ก) หากกลุ่มพิกเซลสว่างที่อยู่บนใบหน้า ข) เมื่อเจอระบุว่า เป็นดวงตา

ภาพที่ 5 การระบุหากกลุ่มพิกเซลสว่างที่อยู่บนใบหน้า

```

Frame per seconds : 28
Limit of dowsiness : 28
frame 0 true 0
frame 1 true 0
frame 2 true 0
frame 3 true 0
frame 4 true 0
frame 5 true 0
frame 6 true 0
frame 7 true 0
frame 8 true 0
frame 9 false 2
  
```

ภาพ 6 เมื่อดวงตาเปิดติดต่อกันโดยไม่มีดวงตาปิดเลยเป็นระยะเวลาหนึ่งจะรีเซ็ตค่า peakdow เป็น 0

```

if (peakdow>0)
{
    peakdow--;
}
presta = true;
}
else {
    cout << "false ";
    save im(3, photocount, compression_params);
    if (!presta) {
        peakdow++;
    }
    else {
        peakdow += 2;
    }
}
  
```

ภาพที่ 7 โค้ดในการตัดสินใจเพิ่มหรือลดค่า peakdow

```

void sound_beepp()
{
    while (program_run)
    {
        if(soundop) {
            Beep(1000, 300);
            Sleep(1000/((peakdow/ (limitofdow/2) )+1));
        }
    }
}

```

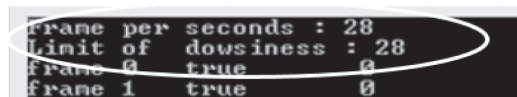
ภาพที่ 8 Method sound_beepp แสดงเสียงสัญญาณเตือน

ผลการดำเนินงานของการทดสอบเปิด-ปิดดวงตา

เราได้ทำการตรวจสอบความถูกต้องของการแยกแยะสีผิว โดยใช้วิดีโอในการทดสอบทั้งหมด 9 คลิป โดยในแต่ละคลิปจะถูกแยกแยะโดยช่วงของสีผิวทั้ง 3 ช่วง เป็นดังนี้

| ช่วงที่ 1 | ช่วงที่ 2 | ช่วงที่ 3 |
|----------------|----------------|----------------|
| Cb = 76 - 126 | Cb = 77 - 127 | Cb = 80 - 120 |
| Cr = 130 - 172 | Cr = 133 - 177 | Cr = 133 - 173 |

โดยอัตราการตัดภาพวิดีโอเป็นภาพนิ่งของแต่ละวิดีโอไม่เท่ากันแสดงตัวอย่างของวิดีโอที่ 5 ดังในภาพที่ 9










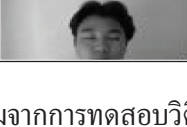
```

frame per seconds : 28
limit of dowsiness : 28
frame 0 true 0
frame 1 true 0

```

ภาพที่ 9 แสดงจำนวนเฟรมต่อวินาทีของวิดีโอที่ 5

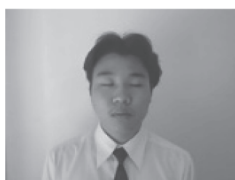
ผลการดำเนินงานของโปรแกรมจากการทดสอบวิดีโอที่ 5 ในช่วงที่ 1 ได้ผลลัพธ์บางส่วนดังภาพที่ 10

| ผลลัพธ์การทำงานของโปรแกรมจากการทดสอบวิดีโอที่ 5 | | | | |
|---|--|----------------|--|---------------------------|
| เฟรม ที่ | ผลที่โปรแกรมตัดสิน (True = ตาเปิด, False = ตาปิด) | ค่า peakdow | รูปภาพแสดงผลลัพธ์ | ความถูกต้อง ของโปรแกรม |
| 0 | True | 0 |  | ถูก |
| 6 | True | 0 |  | ถูก |
| 7 | True | 0 |  | ถูก |
| 8 | True | 0 |  | ถูก |
| 9 | False | 2 |  | ถูก |
| 10 | True | 1 |  | ผิด |
| 11 | True | 0 |  | ผิด |
| 12 | False | 2 |  | ถูก |

ภาพที่ 10 ผลลัพธ์การทำงานของโปรแกรมจากการทดสอบวิดีโอที่ 5

• ผลลัพธ์ที่โปรแกรมดำเนินการผิดพลาด

เฟรมที่ 10 โปรแกรมประมวลได้ค่า true แต่ค่าความถูกต้องคือ false เพราะตาปิด ดังภาพที่ 11



ภาพที่ 11 ผลลัพธ์ที่โปรแกรมดำเนินการผิดพลาดของวิดีโอที่ 5 เฟรมที่ 10

ผลการดำเนินงานของการวัดอาการง่วงนอน

การส่งเสียงร้องเตือนเมื่อผู้ขับขี่อยู่ในอาการง่วงนอนนั้น โดยนำค่าตัววัดอาการง่วงนอน (peakdow) มาเทียบกับค่า threshold ถ้า $\text{peakdow} \geq \text{threshold}$ ก็จะส่งเสียงร้องเตือน

ตัวอย่างเช่น เมื่อนำวิดีโอ 5 เข้ามาทดสอบซึ่งมีทั้งหมด 46 เฟรม (เฟรม 0 \Rightarrow เฟรม 45) frame per second(fps) = 28 ดังนั้น $\text{threshold} = \text{fps} \times \text{second} = 28 \times 1 = 28$

เมื่อแสดงผลโดยใช้ Microsoft visual basic 2015 ได้หน้าจอเป็นดังภาพที่ 12

```

Frame per seconds : 28
Limit of drowsiness : 28
frame 0 true 0
frame 1 true 0
frame 2 true 0
frame 3 true 0
frame 4 true 0
frame 5 true 0
frame 6 true 0
frame 7 true 0
frame 8 true 0
frame 9 false 2
frame 10 true 1
frame 11 true 0
frame 12 false 2
frame 13 false 3
frame 14 false 4
frame 15 false 5
frame 16 true 4
frame 17 false 6
frame 18 false 7
frame 19 false 8
frame 20 false 9
frame 21 false 10
frame 22 false 11
frame 23 true 10
frame 24 false 12
frame 25 false 13
frame 26 true 12
frame 27 true 11
frame 28 true 10
frame 29 true 9
frame 30 true 8
frame 31 false 10
frame 32 true 9
frame 33 false 11
frame 34 true 10
frame 35 false 12
frame 36 false 13
frame 37 false 14
frame 38 true 13
frame 39 false 15
frame 40 false 16
frame 41 true 15
frame 42 false 17
frame 43 false 18
frame 44 false 19
frame 45 false 20
Cannot read the frame from video file

```

ภาพที่ 12 ผลลัพธ์ที่ได้จากวิดีโอ 5 คอลัมน์ที่ 1 คือเฟรมที่ คอลัมน์ที่ 2 คือ สถานะของดวงตา true = เปิด false = ปิด คอลัมน์ที่ 3 คือ ค่าวัดอาการง่วงนอน (peakdow)

กำหนดค่าวัดอาการง่วงนอน (peakdow) เริ่มต้นเท่ากับ 0 ถ้าตาเปิดค่าสถานะดวงตาเป็น true และค่า peakdow จะลดลงแต่ไม่ต่ำกว่า 0 แต่ถ้าตาปิดค่าสถานะดวงตาจะเป็น false ค่า peakdow จะเพิ่มขึ้น ถ้าค่า peakdow ≥ 28 ก็จะกำหนดค่า soundop เป็น true ดังภาพที่ 13 (โดยตัวแปร limitofdow เป็นค่า threshold) แล้วไปเรียก method soundbeep จะมีการส่งเสียงสัญญาณเตือนในระยะเวลา 1000 milliseconds และทำการหยุดรอเป็นเวลาหนึ่ง โดยเวลารอจะลดลงแปรผันตามค่า peakdow

```

if (peakdow >= limitofdow) {
    soundop = true;
}
else {
    soundop = false;
}

```

ภาพที่ 13 โค้ดการหาค่าสถานะดวงตา

สรุปและวิจารณ์ผลการทดลอง

ผลความแม่นยำของการแยกแยะสถานะดวงตาว่าเปิด-ปิด

การตรวจสอบการแยกแยะสถานะดวงตานั้นอยู่ที่ความถูกต้องของการแยกแยะสีผิว โดยผู้วิจัยทดสอบช่วงของสีผิวเป็น 3 ช่วงดังได้กล่าวไปแล้ว ผลที่ได้จากโปรแกรมมาตรวจสอบกับค่าจริงจะได้ค่าความแม่นยำดังตารางที่ 1 โดย

Accuracy คือ ค่าความถูกต้องที่โปรแกรมประมวลออกมาได้ถูกต้อง

Precision = Accuracy * 100/จำนวนเฟรม

ตารางที่ 1 ผลความแม่นยำของการตัดสินใจการเปิด-ปิดดวงตา

| รายงานการตรวจผลการตัดสินใจการเปิด-ปิดของดวงตา | | | | | | | | | | | |
|---|---------------|-------------------------|----------------------------|-----|---------------------------------|----------------------------|-----|---------------------------------|----------------------------|-----|-----------------------------|
| วิดีโอ ที่ | จำนวน เฟรม | ความ ยาว (วินาที) | ช่วงที่ 1 | | | ช่วงที่ 2 | | | ช่วงที่ 3 | | |
| | | | ผลการประเมิน จากโปรแกรม | | ความ แม่นยำ(% (Precision) | ผลการประเมิน จากโปรแกรม | | ความ แม่นยำ(% (Precision) | ผลการประเมิน จากโปรแกรม | | ความแม่นยำ(% (Precision) |
| | | | ถูก (accur acy) | ผิด | | ถูก (accur acy) | ผิด | | ถูก (accur acy) | ผิด | |
| 1 | 48 | 16 | 26 | 22 | 54.17 | 32 | 16 | 66.67 | 45 | 3 | 93.75 |
| 2 | 52 | 17 | 49 | 3 | 94.23 | 50 | 2 | 96.15 | 46 | 6 | 88.46 |
| 3 | 42 | 14 | 17 | 25 | 40.48 | 18 | 24 | 42.86 | 33 | 9 | 78.57 |
| 4 | 77 | 24 | 33 | 44 | 42.86 | 33 | 44 | 42.86 | 56 | 21 | 72.73 |
| 5 | 46 | 14 | 41 | 5 | 89.13 | 40 | 6 | 86.95 | 41 | 5 | 89.13 |
| 6 | 83 | 25 | 50 | 33 | 60.24 | 51 | 32 | 61.45 | 80 | 3 | 96.39 |
| 7 | 70 | 23 | 56 | 14 | 80.00 | 58 | 12 | 82.86 | 54 | 16 | 77.14 |
| 8 | 63 | 19 | 30 | 33 | 47.62 | 27 | 36 | 42.86 | 52 | 11 | 82.54 |
| 9 | 47 | 14 | 32 | 15 | 68.09 | 25 | 22 | 53.19 | 24 | 23 | 51.06 |
| ค่าเฉลี่ย | | | | | 64.09 | | | 63.98 | | | 81.08 |
| ค่าเฉลี่ยรวม | | | | | 69.71 | | | | | | |

ผลความแม่นยำของการตัดสินใจการร่วงนอน

จากการทดสอบโปรแกรมจากจำนวนวิดีโอทั้งหมด 9 คลิป สามารถหาความแม่นยำของการตัดสินใจการร่วงนอนได้ โดยใช้การประเมินผลลัพธ์จากโปรแกรมเปรียบเทียบกับผลลัพธ์จริงที่เก็บจากคน [7] ซึ่งนิยามตัวแปรไว้ดังนี้

TP: True Positive คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่าจริง

TN: True Negative คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่าไม่จริง

FP: False Positive คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่าไม่จริง

FN: False Negative คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่าจริง

- Accuracy คือค่าที่บอกว่าโปรแกรมสามารถทำนายได้แม่นยำเพียงใด

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FN} + \text{TN} + \text{FP})$$

- Recall หรือ True Positive Rate (TPR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าจริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN})$$

- True Negative Rate (TNR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าไม่จริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด

$$\text{TNR} = \text{TN} / (\text{TN} + \text{FP})$$

- False Positive Rate (FPR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าจริง เป็นอัตราส่วนเท่าไรของไม่จริงทั้งหมด

$$\text{FPR} = \text{FP} / (\text{TN} + \text{FP})$$

- False Negative Rate (FNR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าไม่จริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด

$$\text{FNR} = \text{FN} / (\text{TP} + \text{FN})$$

- Precision คือ ค่าที่บอกว่าโปรแกรมทำนายว่าจริง ถูกต้องเท่าไร

หมายเหตุ : $\text{TPR} + \text{FNR} = 1$ และ $\text{TNR} + \text{FPR} = 1$

ตารางที่ 2 ผลความแม่นยำของการตัดสินใจการร่วงนอน

| ช่วงที่ 1 | | | | | | | | | | | | ช่วงที่ 2 | | | | | | | | | | | | ช่วงที่ 3 | | | | | | | | | | | |
|--------------------|-------|----|----|----|-----------|----------|-------|-------|-------|-------|--|--------------------|-------|----|----|----|-----------|----------|-------|-------|-------|-------|--|--------------------|-------|----|----|----|-----------|----------|-------|-------|-------|------|-------|
| วิดีโอ | TP | FN | TN | FP | precision | Accuracy | TPR | TNR | FPR | FNR | | วิดีโอ | TP | FN | TN | FP | precision | Accuracy | TPR | TNR | FPR | FNR | | วิดีโอ | TP | FN | TN | FP | precision | Accuracy | TPR | TNR | FPR | FNR | |
| วิดีโอ1 | 14 | 12 | 16 | 6 | 07 | 0.62 | 0.53 | 0.72 | 0.27 | 0.46 | | วิดีโอ1 | 13 | 19 | 9 | 7 | 0.65 | 0.45 | 0.40 | 0.56 | 0.43 | 0.59 | | วิดีโอ1 | 29 | 16 | 9 | 3 | 0.90 | 0.66 | 0.64 | 0.75 | 0.25 | 0.36 | |
| วิดีโอ2 | 27 | 22 | 0 | 3 | 09 | 0.51 | 0.56 | 0 | 1 | 0.44 | | วิดีโอ2 | 22 | 28 | 1 | 1 | 0.96 | 0.44 | 0.44 | 0.5 | 0.5 | 0.56 | | วิดีโอ2 | 34 | 12 | 0 | 6 | 0.85 | 0.65 | 0.73 | 0 | 1 | 0.26 | |
| วิดีโอ3 | 0 | 17 | 24 | 1 | 0 | 0.57 | 0 | 0.96 | 0.04 | 1 | | วิดีโอ3 | 0 | 18 | 24 | 0 | 0 | 0.57 | 0 | 1 | 0 | 1 | | วิดีโอ3 | 14 | 19 | 9 | 0 | 1 | 0.54 | 0.42 | 1 | 0 | 0.57 | |
| วิดีโอ4 | 0 | 33 | 44 | 0 | 0 | 0.57 | 0 | 1 | 0 | 1 | | วิดีโอ4 | 0 | 33 | 44 | 0 | 0 | 0.57 | 0 | 1 | 0 | 1 | | วิดีโอ4 | 47 | 9 | 0 | 21 | 0.69 | 0.61 | 0.83 | 0 | 1 | 0.16 | |
| วิดีโอ5 | 17 | 23 | 1 | 5 | 0.77 | 0.39 | 0.42 | 0.16 | 0.83 | 0.57 | | วิดีโอ5 | 16 | 28 | 2 | 0 | 1 | 0.39 | 0.364 | 1 | 0 | 0.63 | | วิดีโอ5 | 26 | 21 | 0 | 0 | 1 | 0.54 | 0.54 | 0 | 0 | 0.45 | |
| วิดีโอ6 | 13 | 37 | 33 | 0 | 1 | 0.56 | 0.26 | 1 | 0 | 0.74 | | วิดีโอ6 | 14 | 37 | 32 | 0 | 1 | 0.55 | 0.27 | 1 | 0 | 0.72 | | วิดีโอ6 | 42 | 38 | 3 | 0 | 1 | 0.54 | 0.62 | 1 | 0 | 0.47 | |
| วิดีโอ7 | 26 | 30 | 14 | 0 | 1 | 0.57 | 0.46 | 1 | 0 | 0.53 | | วิดีโอ7 | 18 | 40 | 12 | 0 | 1 | 0.42 | 0.314 | 1 | 0 | 0.68 | | วิดีโอ7 | 44 | 5 | 0 | 16 | 0.73 | 0.67 | 0.89 | 0 | 1 | 0.10 | |
| วิดีโอ8 | 19 | 11 | 15 | 16 | 0.54 | 0.55 | 0.63 | 0.48 | 0.51 | 0.36 | | วิดีโอ8 | 0 | 27 | 36 | 0 | 0 | 0.57 | 0 | 1 | 0 | 1 | | วิดีโอ8 | 26 | 27 | 8 | 3 | 0.89 | 0.52 | 0.48 | 0.72 | 0.27 | 0.51 | |
| วิดีโอ9 | 22 | 8 | 2 | 15 | 0.58 | 0.51 | 0.73 | 0.11 | 0.88 | 0.26 | | วิดีโอ9 | 24 | 1 | 0 | 22 | 0.52 | 0.51 | 0.96 | 0 | 1 | 0.04 | | วิดีโอ9 | 3 | 21 | 21 | 2 | 0.6 | 0.51 | 0.125 | 0.91 | 0.08 | 0.87 | |
| ค่าเฉลี่ย | | | | | 0.611 | 0.537 | 0.397 | 0.603 | 0.392 | 0.596 | | ค่าเฉลี่ย | | | | | 0.568 | 0.496 | 0.306 | 0.784 | 0.214 | 0.591 | | ค่าเฉลี่ย | | | | | | 0.861 | 0.582 | 0.575 | 0.486 | 0.40 | 0.415 |
| คิดเป็นร้อยละ | | | | | 61.1 | 53.7 | 39.7 | 60.3 | 39.2 | 59.6 | | คิดเป็นร้อยละ | | | | | 56.8 | 49.6 | 30.5 | 78.4 | 21.4 | 69.1 | | คิดเป็นร้อยละ | | | | | | 86.1 | 58.2 | 57.5 | 48.6 | 40 | 41.5 |
| Accuracy เท่านั้น | 53.7% | | | | | | | | | | | Accuracy เท่านั้น | 49.6% | | | | | | | | | | | Accuracy เท่านั้น | 58.2% | | | | | | | | | | |
| TPR เท่านั้น | 39.7% | | | | | | | | | | | TPR เท่านั้น | 30.5% | | | | | | | | | | | TPR เท่านั้น | 57.5% | | | | | | | | | | |
| TNR เท่านั้น | 60.3% | | | | | | | | | | | TNR เท่านั้น | 78.4% | | | | | | | | | | | TNR เท่านั้น | 48.6% | | | | | | | | | | |
| FPR เท่านั้น | 39.2% | | | | | | | | | | | FPR เท่านั้น | 21.4% | | | | | | | | | | | FPR เท่านั้น | 40% | | | | | | | | | | |
| FNR เท่านั้น | 59.5% | | | | | | | | | | | FNR เท่านั้น | 69.1% | | | | | | | | | | | FNR เท่านั้น | 41.5% | | | | | | | | | | |
| Precision เท่านั้น | 61.1% | | | | | | | | | | | Precision เท่านั้น | 56.8% | | | | | | | | | | | Precision เท่านั้น | 65.1% | | | | | | | | | | |

หมายเหตุ: ค่า TP สิ่งที่โปรแกรมทำนายว่าจริงและคนบอกว่าจริงในวิดีโอ 3 และ 4 เป็น 0 ในช่วงที่ 1 และ 2 เพราะโปรแกรมทำนายว่าตบและคนบอกว่าตบ แสดงดังภาพที่ 14 และ 15



ภาพที่ 14 ภาพไบนารีที่มีค่ามืดเป็นจุดที่เป็นสีผิวและค่าสว่างไม่ใช่ผิวตาปิด (วิดีโอที่ 3)



ภาพที่ 15 ภาพผลลัพธ์ที่ได้จากการประมวลผลของโปรแกรมตัดสินว่าตาปิด (วิดีโอที่ 4)

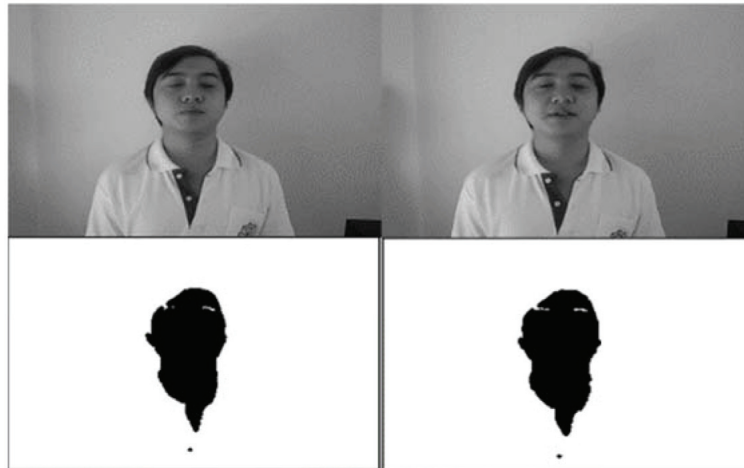
จากผลการทดสอบหาผลความแม่นยำของการแยกแยะสถานะดวงตาว่าเปิด-ปิดและผลความแม่นยำของการตัดสินอาการง่วงนอนจะพบว่าค่าพิสัย $CbCr$ ช่วงที่ 3 ที่มีพิสัยคือ $80 < Cb < 120$ และ $133 < Cr < 173$ จะมีความถูกต้องที่สุด

ปัญหาและแนวทางในการพัฒนาต่อไป

ปัญหาที่ทำให้โปรแกรมมีการวิเคราะห์ดวงตาผิดพลาดมีดังต่อไปนี้

1. เนื่องจากมีปัญหาเรื่องการควบคุมความสว่างของแสง ผู้วิจัยจึงกำหนดขอบเขตให้แสงที่เข้ามามีระดับใกล้เคียง ไม่เลือกภาพที่มีทั้งมมมืดและสว่างในเวลาเดียวกัน

2. วัตถุบนใบหน้ามีผลต่อความถูกต้องในการตัดสินใจ เช่น คิ้ว ริมฝีปาก ยกตัวอย่างเช่น เมื่อตรวจจับใบหน้าแล้วกำหนดจุดที่ไม่ใช่สีผิว ถ้ามีอยู่ 2 ตำแหน่งเราจะกำหนดว่าเป็นดวงตา แต่เมื่อเจอคิ้วมันจะได้สถานะเป็นเปิด ซึ่งจริงๆ แล้วดวงตาปิด ดังภาพที่ 16



ภาพที่ 16 โปรแกรมตัดสีสถานะดวงตาได้ไม่ถูกต้อง ในกรณีนี้ตาปิดแต่โปรแกรมตรวจจับเจอคือ
จึงแสดงสถานะเป็นตาเปิด

เนื่องจากการตรวจสอบความถูกต้องของการแยกแยะสีผิวนั้น ขึ้นอยู่กับช่วงของ YCbCr และคุณภาพของภาพขาวดำ ดังนั้นผู้วิจัยจึงเสนอแนะแนวทางในการพัฒนาต่อไปดังนี้

1. พัฒนารูปแบบการแยกสีผิวมนุษย์โดยหาช่วง YCbCr ที่เหมาะสมกว่านี้
2. เลือกความสมบูรณ์ของภาพขาวดำโดยควรพิจารณาจาก ความเร็วแห่งของใบหน้าแล้วเติมภาพจุดดำและขาวในส่วนที่ขาดหายไป
3. พัฒนาเป็นระบบ real time บนโทรศัพท์มือถือเพื่อให้ผู้ขับขี่สามารถดาวน์โหลดโปรแกรมไปใช้เพื่อเตือนผู้ขับขี่ได้ทันทั่วทั้ง

กิตติกรรมประกาศ

ขอขอบคุณ นายภัทริชญ์ ตนประเสริฐกุล อาจารย์และเจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ที่ให้คำแนะนำและช่วยตรวจทานเป็นอย่างดี ทำให้บทความนี้สำเร็จลุล่วงไปด้วยดี

เอกสารอ้างอิง

1. Devi, M.S., Choudhari, M.V. and Bajaj, P. 2011. Driver Drowsiness Detection Using Skin Color Algorithm and Circular Hough Transform. India: N.S.I.T, Ahmadabad. Available from URL: <http://ieeexplore.ieee.org/xpls/abs-all.jsp?arnumber=6120568&tag=1>. 15 September 2015.
2. Elgammal, A., Muang, C. and Hu, D. 2011. Skin Detection. Available from URL: <http://www.cs.rutgers.edu/~elgammal/pub/skin.pdf>. 16 September 2015.

3. Basilio, J.A.M., Torres, G.A., Pérez, G.S.P., Medina, L.K.T. and Meana, H.M.P. 2011. Explicit image detection using YCbCr space color model as skin detection. In: American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications. 29-31 January 2011. Puerto Morelos, Mexico. p. 123-128.
4. Kukharev, G., and Nowosielski, A. 2004. Visitor Identification-Elaborating Real Time Face Recognition System. In: 12th Winter School on computer Graphics (WSCG), Plzen, Czech Republic, p. 157-164
5. Chai, D. and Ngai, K.N. 1999. Face Segmentation Using Skin Color Map in Videophone Applications. In: IEEE Trans. On Circuits and Systems for video Technology, 9(4). p. 551-564
6. Sonka, M., Hlavac, V., and Boyle, R. 2008. Image Processing, Analysis, and Machine Vision. 3rd Ed. Thomson. p. 40, 176 and 661-666.
7. Szeliski, R. 2011. Computer Vision Algorithms and Applications. Springer. p. 79,201.

ได้รับบทความวันที่ 19 สิงหาคม 2559

ยอมรับตีพิมพ์วันที่ 23 พฤศจิกายน 2559