วารสารวิชาการพระจอมเกล้าพระนครเหนือ ปีที่ 31, ฉบับที่ 3 ก.ค.–ก.ย. 2564
The Journal of KMUTNB., Vol. 31, No. 3, Jul.–Sep. 2021

บทความวิจัย

ซอฟต์แวร์ป้องกันการเกิดอาการคอมพิวเตอร์วิชันซินโดรม กรณีตรวจหาระยะห่างด้วยภาพใบหน้า

วันฉัตร พูขุนทด*
สาขาเทคโนโลยีมัลติมีเดีย คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเทพสตรี
จิติมนต์ อั่งสกุล และ ธรา อั่งสกุล
สาขาวิชาเทคโนโลยีสารสนเทศ สำนักวิชาเทคโนโลยีสังคม มหาวิทยาลัยเทคโนโลยีสุรนารี

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 06 2132 6768 อีเมล: exandwhy666@yahoo.com
DOI: 10.14416/j.kmutnb.2021.05.019
รับเมื่อ 16 มีนาคม 2563 แก้ไขเมื่อ 5 มิถุนายน 2563 ตอบรับเมื่อ 8 มิถุนายน 2563 เผยแพร่ออนไลน์ 24 พฤษภาคม 2564

บทคัดย่อ

คอมพิวเตอร์วิชันซินโดรมเป็นอาการเจ็บป่วยที่เกิดจากเพ่งมองจออุปกรณ์สารสนเทศ บทความนี้นำเสนอการพัฒนาซอฟต์แวร์ที่ป้องกันการเกิดอาการด้วยการตรวจหาระยะห่างของตาผู้ใช้กับหน้าจอ เมื่อผู้ใช้มีระยะการมองที่มีความเสี่ยงต่อการเกิดอาการ ระบบจะทำการแจ้งเตือนให้ผู้ใช้ปรับระยะการมองที่เหมาะเพื่อเลี่ยงความเสี่ยงของเกิดอาการ การวัดระยะห่างใช้ภาพที่รับจากกล้องที่เชื่อมกับอุปกรณ์นั้นๆ แล้วประยุกต์ใช้เทคโนโลยีการตรวจหาใบหน้าเพื่อค้นหาตำแหน่งของดวงตา วิธีการคำนวณเริ่มจากการสร้างเส้นสมมติจากตำแหน่งของกล้องและดวงตา เพื่อที่จะหาค่าตามหลักการตรีโกณมิติ การดำเนินการวิจัยใช้การตรวจหาใบหน้าสามวิธี ได้แก่ ฮาร์ ฮอก และโครงข่ายประสาทเทียม เพื่อเปรียบเทียบและเลือกวิธีการที่ดีที่สุด การประเมินแบ่งออกเป็นสามด้าน ได้แก่ ด้านความเร็ว ด้านความถูกต้องแม่นยำ และด้านความสามารถในการใช้งานได้ ผลการประเมินซอฟต์แวร์ใช้ความเร็วในการตรวจหาระยะจากภาพขนาด 640 × 480 เท่ากับ 36 มิลลิวินาทีต่อเฟรม ความถูกต้องและแม่นยำมีค่าร้อยละ 99 ขึ้นไป ค่าเฉลี่ยความคาดเคลื่อนสัมบูรณ์เท่ากับ 0.26 ค่าเฉลี่ยความคลาดเคลื่อนกำลังสองเท่ากับ 0.47 ค่ารากที่สองของความคาดเคลื่อนเฉลี่ยเท่ากับ 0.22 และซอฟต์แวร์มีความสามารถในการใช้งานได้เป็นอย่างดี

คำสำคัญ: คอมพิวเตอร์วิชันซินโดรม การตรวจหาใบหน้า ระยะห่าง

การอ้างอิงบทความ: วันฉัตร พูขุนทด, จิติมนต์ อั่งสกุล, และ ธรา อั่งสกุล, "ซอฟต์แวร์ป้องกันการเกิดอาการคอมพิวเตอร์วิชันซินโดรม กรณีตรวจหาระยะห่างด้วยภาพใบหน้า," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 31, ฉบับที่ 3, หน้า 574–586, ก.ค.–ก.ย. 2564.

Research Article

Software Prevent Occurrence of Computer Vision Syndrome Case of Detecting Distance from the Face Image

Wanchat Pookhuntod*

Technology Multimedia, Faculty of Information Technology, Thepsatri Rajabhat University, Lop Buri, Thailand

Jitimon Angskun and Thara Angskun

School of Information Technology, School of Social Technology Suranaree University of Technology, Nakhon Ratchasima, Thailand

* Corresponding Author, Tel. 06 2132 6768, E-mail: exandwhy666@yahoo.com
DOI: 10.14416/j.kmutnb.2021.05.019
Received 16 March 2020; Revised 5 June 2020; Accepted 8 June 2020; Published online: 24 May 2021

Abstract

Computer Vision Syndrome is vision-related problems associated with focusing the eyes on a digital screen of computer or other display device for protracted periods of time. This article presents the development of software that prevents the occurrence of such symptoms by detecting the distance of the user's eyes to screen. When users are in a view range that increase a risk of symptom development, the system will alert them so they can adjust the suitable viewing distance to avoid the risk of symptom occurrence. The distance measurement system uses image processing from the camera connected to that device while appliying face detection technology to detect eye pupils and measure pupillary distance. The calculation is performed by creating vectors from the position of the camera to eyes. In order to find the value according to trigonometric principles, this research employed three methods of face detection, called HAAR HOG and convolutional neural networks to compare and select the best method. The as sessment is divided into three areas: speed, accuracy, and the usability. The software evaluation results use the speed of distance measurement from 640 × 480 image size is 36 ms/f, the percentage accuracy and precision is described as 99 percent or above, MAE is 0.26, MSE is 0.47, RMSE is 0.22 . Overall, the software efficiency and effectiveness are substantiated.

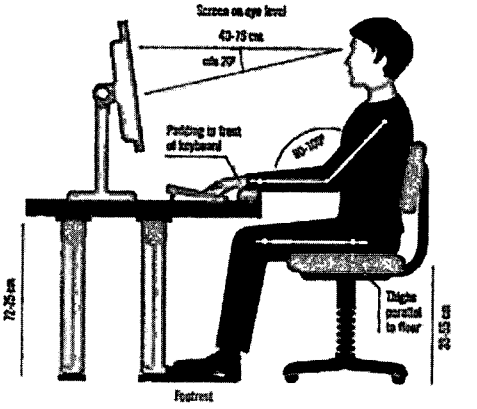
Keywords: Computer Vision Syndrome, Face Detection, Distance

Please cite this article as: W. Pookhuntod, J. Angskun, and T. Angskun, "Software prevent occurrence of computer vision syndrome case of detecting distance from the face image," *The Journal of KMUTNB*, vol. 31, no. 3, pp. 574–586, Jul.–Sep. 2021 (in Thai).

1. บทนำ

คอมพิวเตอร์วิชันซินโดรมเป็นกลุ่มโรคอาการเจ็บป่วยทางดวงตาและการมองเห็น อาทิ อาการปวดตา ตาล้า ปวดหัว ปวดคอ ปวดหลัง การเจ็บป่วยทางสายตา หรืออาการอื่นๆ อันมีสาเหตุมาจากการเพ่งมองจอคอมพิวเตอร์ สมาร์ตโฟน หรือแท็บเล็ต เป็นต้น [1] กลุ่มโรคดังกล่าวพบมากในผู้คนในปัจจุบัน โดยจะพบผู้มีอาการในกลุ่มผู้ใช้งานคอมพิวเตอร์และอุปกรณ์สารสนเทศถึงร้อยละ 90 ซึ่งมีทุกกลุ่มอายุ [2] ทั้งนี้ ผู้ที่มีอาการทั่วโลกทวีจำนวนเพิ่มขึ้นเรื่อยๆ ตามจำนวนการเติบโตของการใช้อุปกรณ์สารสนเทศ เนื่องด้วยเราต้องพึ่งพาสิ่งดังกล่าวอย่างมาก ในประเทศไทยคนใช้เวลากับอุปกรณ์เหล่านี้เฉลี่ย 9 ชั่วโมง 11 นาทีต่อวัน [3] ซึ่งเวลาส่วนนี้จะส่งผลต่อการเกิดอาการ [4] ซึ่งอาจกล่าวได้ว่าการใช้ชีวิตประจำวันในทุกวันนี้เป็นการยากที่จะไม่สุ่มเสี่ยงกับกลุ่มโรคดังกล่าว สาเหตุการเกิดอาการกลุ่มคอมพิวเตอร์วิชันซินโดรมมีอยู่หลายปัจจัย และแต่ละปัจจัยจะก่อให้เกิดอาการเจ็บป่วยที่แตกต่างกันไป ซึ่งการป้องกันการเกิดอาการเป็นเรื่องง่าย เพียงปรับพฤติกรรมการใช้งานอุปกรณ์สารสนเทศ เพราะอาการดังกล่าวเกิดจากพฤติกรรมการใช้งานที่ไม่เหมาะสม แต่อย่างไรก็ตาม ผู้ใช้อุปกรณ์สารสนเทศไม่ได้ใส่ใจการป้องกันหรือไม่รู้ตัวว่าพฤติกรรมของตนมีความเสี่ยงต่อการเกิดอาการ

จากการศึกษาปัญหาข้างต้น ผู้วิจัยจึงได้พัฒนาซอฟต์แวร์เพื่อใช้ป้องกันอาการคอมพิวเตอร์วิชันซินโดรม โดยพิจารณาตามปัจจัยที่เป็นสาเหตุของอาการที่สามารถเชื่อมโยงกับเทคโนโลยีที่นำมาประยุกต์ใช้ในการพัฒนาซอฟต์แวร์ จึงได้แนวคิดในการใช้เทคโนโลยีการตรวจหาใบหน้ามาประยุกต์ใช้ในการตรวจหาระยะการเพ่งมองจอ ซึ่งเป็นปัจจัยหลักที่ทำให้เกิดอาการตาเมื่อยล้า ปวดหัว ปวดคอ ปวดไหล่ และปวดหลัง เป็นต้น [5] โดยระบบการทำงานของซอฟต์แวร์จะตรวจหาระยะเพ่งจอของผู้ใช้ เมื่อมีระยะที่สุ่มเสี่ยงต่อการเกิดอาการ จะแจ้งเตือนให้ผู้ใช้ทราบ เพื่อให้ปรับระยะการมองของตนและหลีกเลี่ยงความเสี่ยงนั้น ระยะการเพ่งมองจอที่ความเหมาะสมควรมีระยะห่างจากจอประมาณ 40–75 ซม. [6] ดังแสดงดังรูปที่ 1 การเกิดอาการจากระยะการเพ่งมองที่ไม่เหมาะสม มีสาเหตุมาจากระยะเปลี่ยนแปลงโดยไม่รู้ตัว เนื่องจากสายตาต้องการความคมชัดที่เพิ่มขึ้น เพราะการเพ่งมองวัตถุนาน ดวงตาจะเสียความคมชัด จึงจำเป็นต้องปรับระยะการมองของตนแบบไม่รู้ตัว ทำให้เกิดการโน้มตัวบ้างเอียงตัวบ้าง และบางครั้งฝืนร่างกายตนเอง ทำให้เกิดอาการเจ็บป่วยตามมา โดยการป้องกันทำได้ง่ายเพียงรักษาระยะการมองที่เหมาะสมไว้เท่านั้น



รูปที่ 1 ระยะการเพ่งมองจอและท่านั่งที่เหมาะสม

งานวิจัยที่ใช้เทคโนโลยีคอมพิวเตอร์เพื่อป้องกันอาการคอมพิวเตอร์วิชันซินโดรมมีหลากหลายด้าน หลายสาเหตุปัจจัย เช่น Jennifer และ Shavmila [7] ใช้การแปลงภาพดวงตาเป็นไบนารีหาความจุในนั้น Vieiva และคณะ [8] หาอัตราส่วนของพิกัดดวงตา หาสถานะการลืมตาและหลับตาเพื่อตรวจหาอัตราการกระพริบตา Julius และคณะ [9] ลดการเพ่งมองจอคอมพิวเตอร์เป็นเวลานาน ด้วยการเตือนจากการนับเวลาการเปิดเครื่อง

อย่างไรก็ตาม บทความนี้มุ่งเน้นที่การป้องกันอาการคอมพิวเตอร์วิชันซินโดรมด้วยการตรวจหาระยะห่างของดวงตากับหน้าจอและมีงานวิจัยที่มีแนวทางคล้ายกัน เช่น Ho และคณะ [10] นำเสนอการแจ้งเตือนเมื่อผู้ใช้มีระยะการมองสมาร์ตโฟนที่เสี่ยงต่ออาการดังกล่าว ระบบจะวัดระยะห่างด้วยการเปรียบเทียบขนาดของใบหน้าที่อยู่ในระยะที่เหมาะสม ส่วนการแจ้งเตือนจะเป็นรูปแบบไม่คุกคามผู้ใช้ เช่น ขอบจอกระพริบ หรือข้อความแจ้งเตือน Toda และคณะ

[11] เสนอการตรวจหาระยะห่างของผู้ใช้และการงอคอในการมองจอ การหาระยะห่างใช้การคำนวณค่าขนาดพื้นที่ของกรอบตรวจหาใบหน้า โดยมีความคาดเคลื่อนเท่ากับร้อยละ 2.40 ถึง 9.42 Wasnik และ Jeyakumar [12] นำเสนอการตรวจหาสาเหตุของการเกิดคอมพิวเตอร์วิชันซินโดรมด้วยเซนเซอร์ โดยใช้คลื่นอัลตราโซนิก (Ultrasonic) ตรวจหาระยะห่าง มาตรวัดความเร่ง (Accelerometer) ตรวจหาองศาการมอง ตัวรับรู้ชีพจร (Pulse Sensors) ตรวจหาการเต้นของชีพจรอุปกรณ์ดังกล่าวทั้งหมดจะติดบนตัวของผู้ใช้ ระบบทำงานผ่านไมโครคอนโทรลเลอร์ (Microcontroller) เชื่อมโยงกับระบบคลาวด์และแจ้งเตือนผ่านข้อความบนโทรศัพท์มือถือ

อย่างไรก็ตาม จะเห็นได้ว่าแต่ละงานวิจัยที่ได้กล่าวมาข้างต้น เป็นป้องกันปัจจัยอย่างใดอย่างหนึ่ง งานวิจัยนี้จึงได้พัฒนาซอฟต์แวร์อายการ์ด (EyeGuard) [13] ที่รวบรวมวิธีการป้องกันต่างๆ ไว้ในชิ้นงานเดียว โดยซอฟต์แวร์จะทำการตรวจปัจจัยที่ก่อให้เกิดโรคด้วยภาพจากกล้องและแจ้งเตือนพฤติกรรมของผู้ใช้ที่มีความเสี่ยง ได้แก่ การกระพริบตา ระยะห่าง องศาการมอง และช่วงเวลาการเพ่งมองจอ การเปรียบเทียบคุณสมบัติของอายการ์ดและงานที่เกี่ยวข้องแสดงในตารางที่ 1

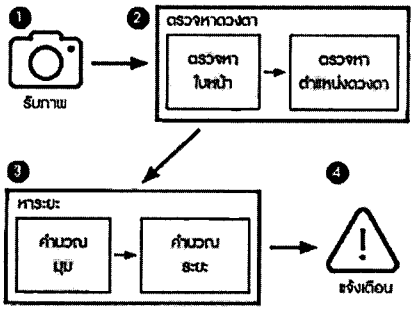
งานวิจัยนี้นำเสนอการตรวจหาระยะห่างเพื่อยกระดับคุณภาพของอายการ์ด โดยมีงานที่พัฒนาการวัดระยะห่างของดวงตาด้วยการตรวจหาใบหน้าหลากหลายวิธี เช่น Lönig และคณะ [14] นำเสนอการคำนวณระยะใบหน้ากับจอในสมาร์ตโฟนโดยใช้หลักการของกล้องรูเข็ม (Pinhole) ด้วยการเปรียบเทียบระยะระหว่างตากับรูปภาพที่ได้รับจากกล้องใช้เวลาคำนวณในแต่ละครั้ง 300 ms. ค่าเบี่ยงเบนเท่ากับ 2.1 เช่นเดียวกับ Li และคณะ [15] ใช้วิธีเดียวกันในการวัดระยะเพื่อปรับเปรียบการแสดงผลบนหน้าจอสมาร์ตโฟนซึ่งความเร็วสะสมที่ใช้เท่ากับ 2 วินาที และค่าคาดเคลื่อนกำลังสองเท่ากับ 0.68 Hossain และ Mukit [16] ก็ใช้ระยะห่างของดวงตาที่ได้จากการตรวจหาด้วยอัลกอรึทึมเอดาบูสท์ (AdaBoost) วัดระยะห่างของหน้าจอเช่นกัน Dong และคณะ [17] ได้ศึกษาการตรวจวัดระยะใบหน้ากับหน้าจอด้วยตรีโกณมิติโดยสร้างสามเหลี่ยมด้วยการตรวจหาดวงตาทั้งสองข้างและจมูก นำสิ่งที่ได้ไปเปรียบระหว่างภาพที่ได้รับและขนาดของวัตถุจริง ซึ่งมีค่าความถูกต้องร้อยละ 95 ความเร็วเท่ากับ 230 มิลลิวินาที และยังมีวิธีการคำนวณระยะจากพิกเซลของรูปภาพ งานของ Godard และคณะ [18] นำเสนองานวิจัยการวัดระยะและความลึกของวัตถุด้วยพิเซลที่อยู่ในภาพด้วยโครงข่ายประสาทเทียม ซึ่งได้ผลดีในระดับหนึ่งแต่เหมาะกับการวัดในพื้นที่กว้าง เช่นเดียวกับ Bianco และคณะ [19] เนื่องจากการวัดระยะข้างต้นยังมีความคาดเคลื่อนที่สูงและใช้เวลาที่มากอยู่และอายการ์ดจะทำให้สิ่งดังกล่าวดีขึ้น

ตารางที่ 1 การเปรียบเทียบคุณสมบัติของอายการ์ดและงานที่เกี่ยวข้อง

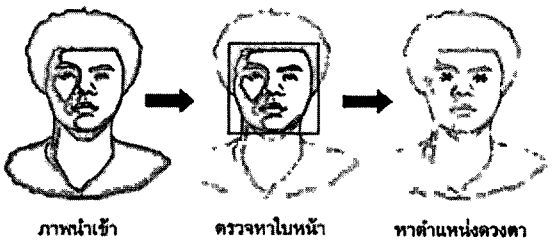
Research Work	Approach				Platform	
	Blink	Distance	Angle	Duration	Computer	Smartphone
Jennifer	✓				✓	
Vieira	✓				✓	✓
Julius				✓	✓	
Ho		✓				✓
Toda		✓	✓		✓	
Wasnik		✓	✓		✓	
EyeGuard (This work)	✓	✓	✓	✓	✓	✓

2. วัสดุ อุปกรณ์และวิธีการวิจัย

โครงสร้างการทำงานของระบบสามารถแสดงได้ดังรูปที่ 2 การทำงานของระบบประกอบด้วยสี่ขั้นตอนหลัก ได้แก่ ขั้นตอนการนำเข้าภาพหรือรับภาพจากกล้องที่เชื่อมกับอุปกรณ์ของผู้ใช้ ขั้นตอนการตรวจหาตำแหน่งดวงตา ขั้นตอนคำนวณระยะห่างของผู้ใช้กับอุปกรณ์ และเมื่อมีความเสี่ยงต่อการเกิดอาการคอมพิวเตอร์วิชันซินโดรม ซอฟต์แวร์จะเข้าสู่ขั้นตอนการแจ้งเตือน



รูปที่ 2 โครงสร้างการทำงานของระบบ



รูปที่ 3 การตรวจหาตำแหน่งดวงตา

2.1 ขั้นตอนการรับภาพ

ซอฟต์แวร์จะรับภาพจากกล้องโดยใช้ไลบรารีโอเพนซีวี (OpenCV) [20] และรับภาพขนาด 640 x 480 x 3 โดยช่องสีของภาพอยู่ในรูปแบบบีจีอาร์ (BGR; Blue Green Red) และส่งภาพไปในขั้นตอนต่อไป

2.2 ขั้นตอนการตรวจหาตำแหน่งดวงตา

ขั้นตอนการตรวจหาตำแหน่งของดวงตา สามารถแสดงได้ดังรูปที่ 3 โดยเริ่มต้นจากการหาจุดพื้นที่ภายในภาพเพื่อหาตำแหน่งที่สนใจหรือเรียกว่า อาร์โอไอ (ROI; Region of Interest) โดยส่วนมากการหาตำแหน่งของดวงตาจะใช้ตำแหน่งของใบหน้าเป็นจุดสนใจ การตรวจหาใบหน้ามีวิธีการหลากหลายวิธี ซึ่งบทความนี้มุ่งเน้นกระบวนการประมวลผลบนซีพียู ซึ่งได้รวบรวมการตรวจหาใบหน้าจากโมเดลที่ได้มีการสร้างไว้แล้ว (Pre-model) ประกอบด้วย

ฮาร์ (Haar Cascasde) [21] เป็นการตรวจหาใบหน้าโดยใช้คุณลักษณะในการจำแนกเป็นรูปสี่เหลี่ยม ซึ่งใช้เวฟเล็ท (Wavelet) สร้าง สอนการเรียนรู้ให้เครื่องโดยใช้อัลกอริทึมเอดาบูสท์ ในการจำแนกคุณลักษณะ

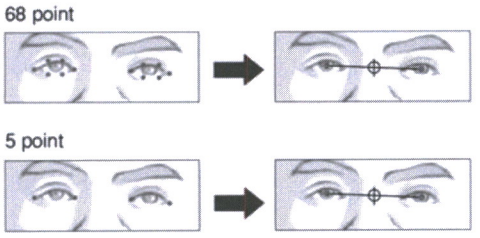
ฮอก (HOG; Histogram of Oriented Gradients) [22] เป็นการตรวจหาวัตถุด้วยการประเมินค่าฮิสโทแกรมโดยการกระจายไล่ระดับความเข้มและควบคุมทิศทางของขอบวัตถุในภาพ แบ่งภาพเป็นบล็อกเรียงต่อกันและแบ่งเป็นความเข้มและทิศทางของขอบวัตถุ ซึ่งเซลล์ทั้งหมดจะเป็นตัวบอกคุณลักษณะของภาพ เมื่อได้ภาพจากขั้นตอนดังกล่าวแล้วนำไปผ่านกระบวนการเรียนรู้ของเครื่องแบบเอสวีเอ็ม (SVM; Support Vector Machine) เพื่อสร้างการจำแนกคุณลักษณะของวัตถุ [23]

ดีเอ็นเอ็น (DNN: Deep Neural Network) เป็นการอนุมานโครงข่ายเชิงลึก รองรับการทำงานกับโครงข่ายเชิงลึกหลายเฟรมเวิร์ก เช่น แคฟฟี (Caffe) เทนเซอโฟลว์ (Tensor-Flow) ทอช (Torch) ดาร์กเน็ต (Darknet) โอเอ็นเอ็นเอ็กซ์ (ONNX) ในงานวิจัยนี้ได้ใช้โมเดลโครงข่ายเชิงลึกที่ถูกสอนไว้ล่วงหน้าด้วยกรอบการทำงานเทนเซอร์โฟลว์ (Tensorflow Framework) ซึ่งงานวิจัยนี้ได้ใช้โมเดลที่มีการเทรนไว้ล่วงหน้าซึ่งอยู่ในไลบรารีโอเพนซีวี [20] และฟังชันก์ในไลบรารีดังกล่าวจะสร้างการอนุมานโครงข่ายขึ้นมา

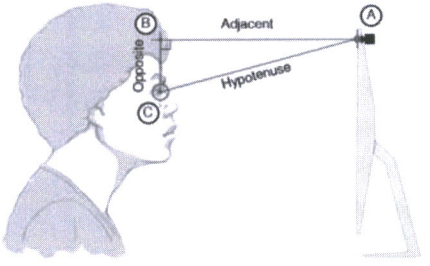
เมื่อได้ตำแหน่งของใบหน้าแล้วซอฟต์แวร์จะหาตำแหน่งของด้วยตาด้วยวิธีการสร้างเฟซแลนด์มาร์ก (Facial Landmark) ในบทความนี้ใช้โมเดลที่ได้สร้างไว้ก่อนแล้วคือการตรวจจับ 68 จุด และ 5 จุด [24] พร้อมทั้งเปรียบเทียบการใช้ทรัพยากรและความเร็ว เมื่อการสร้างเฟซแลนด์มาร์กเสร็จสิ้น จะใช้เฉพาะจุดตำแหน่งดวงตา ทั้งนี้ ตำแหน่งของดวงตาประกอบด้วยตาซ้ายและขวาซึ่งมีตำแหน่งที่แตกต่างกัน งานวิจัยนี้จึงสร้างจุดกึ่งกลางเสมือนของการมองเห็น ดังแสดงในรูปที่ 4 ซึ่งจุดกึ่งกลางของตาซ้ายขวาดังกล่าวจะใช้ในการวัดระยะการมองของผู้ใช้

2.3 ขั้นตอนการคำนวณหาระยะห่าง

เมื่อซอฟต์แวร์สามารถตรวจหากึ่งกลางของการมองเห็นได้แล้ว จุดกึ่งกลางดังกล่าวจะถูกนำมาใช้ในการหาระยะห่างระหว่างดวงตาผู้ใช้กับหน้าจออุปกรณ์ ในงานวิจัยนี้



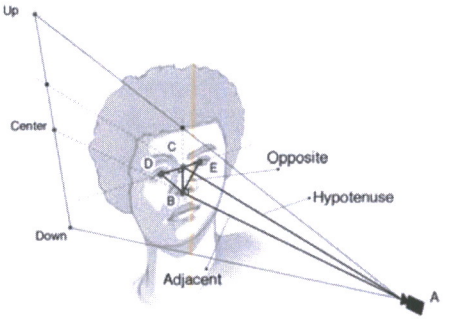
รูปที่ 4 จุดกึ่งกลางของการมองเห็น



รูปที่ 5 ตัวอย่างการคำนวณระยะห่างของตาผู้ใช้กับหน้าจอ

ได้ใช้วิธีการคำนวณตรีโกณมิติ (Trigonometry) เพื่อคำนวณหาระยะ ดังแสดงตัวอย่างในรูปที่ 5 แสดงตัวอย่างแนวคิดการใช้ตรีโกณมิติคำนวณระยะห่างของงานวิจัยนี้ โดยเส้นประชิดมุมหรือคือระยะห่างของดวงตากับจอ เส้นตรงข้ามมุมหรือคือเส้นตั้งฉากของมุมกล้องกับตำแหน่งดวงตา เส้นตรงข้ามมุมฉาก หรือคือเส้นที่สร้างการทำมุมของดวงตากับกล้อง

เพื่อหาระยะห่างตามการคำนวณตรีมิติแสดงในสมการที่ (1) งานวิจัยนี้ได้สร้างเส้นสมมติจากตำแหน่งดวงตาที่ตรวจหาได้กับการกำหนดระยะการมองเห็นวัตถุของกล้อง ดังแสดงตัวอย่างในรูปที่ 6 โดย A คือ ตำแหน่งของกล้อง B คือ จุดตัดฉากของตำแหน่งกึ่งกลางการมองเห็นและกึ่งกลางกล้อง C คือ ตำแหน่งกึ่งกลางการมองเห็นของผู้ใช้ที่ตรวจหาได้ D และ E คือ ตำแหน่งตาขวาซ้ายตามลำดับ (โดยทั้งสองมีค่าพิกัดแนวตั้งเท่ากับพิกัดแนวตั้งของ C) Center คือ จุดกึ่งกลางการกล้อง Up และ Down คือ ขอบเขตที่กล้องสามารถมองเห็นวัตถุในมุมเงยและก้มได้จากระยะห่างที่กำหนดตามลำดับ เมื่อเปรียบเทียบสมการกับเส้นสมมติที่สร้างขึ้น $\overline{AB}$คือระยะห่างที่เราตั้งการหา ดังนั้นจะต้องหาค่า $\overline{BC}$และ BAC



รูปที่ 6 ตัวอย่างเส้นสมมติที่ใช้หาระยะห่าง

$$\tan\theta = \frac{o}{a} \tag{1}$$

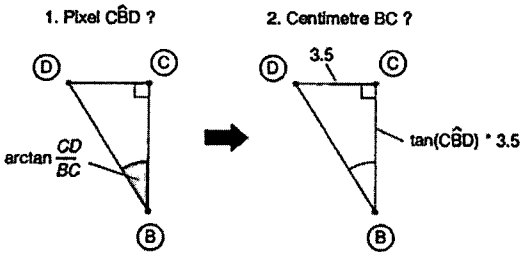
จากรูปที่ 6 การหาค่า $\overline{BC}$ หรือเส้นตั้งฉากของมุมกล้องกับตำแหน่งดวงตา สามารถหาได้จากการตรวจหาในภาพเป็นหน่วยพิกเซล ซึ่งในการคำนวณหาระยะห่างของผู้ใช้กับจอเป็นการคำนวณระยะที่มีหน่วยเป็นเซนติเมตร และการกำหนดระยะการมองเห็นวัตถุจะใช้หน่วยเซนติเมตรเช่นเดียวกัน เพื่อให้สอดคล้องกับหน่วยการคำนวณการหาค่าดังกล่าว ขั้นตอนการคำนวณจะประกอบด้วยสองขั้นตอนย่อย ได้แก่ การคำนวณมุม CBD และการแปลง $\overline{BC}$เป็นเซนติเมตร

$$CBD = \arctan\left(\frac{\overline{CD}}{\overline{BC}}\right) \tag{2}$$

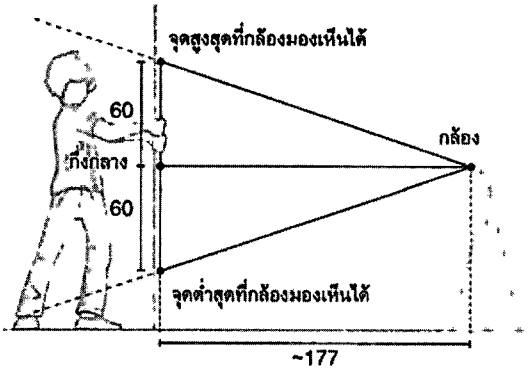
$$\tan(CBD) = \frac{3.5}{\alpha} \tag{3}$$

โดยค่า CBD คำนวณด้วยสมการที่ (2) เป็นการคำนวณหามุมตามหลักตรีโกณมิติ โดย $\overline{CD}$ คือ เส้นตรงข้ามมุมและ $\overline{BC}$ คือ เส้นประชิดมุม การแปลง $\overline{BC}$ เป็นเซนติเมตร คำนวณด้วยสมการที่ (3) ให้ $\overline{BC}$ หรือเส้นตั้งฉากของมุมกล้องกับดวงตาในหน่วยเซนติเมตรเป็น α แล้วกำหนดระยะกึ่งกลางของดวงตากับตาขวาหรือ $\overline{CD}$ เท่ากับ 3.5 (ค่าปริยายที่ได้จากการวัดจากดวงตาเฉลี่ยของผู้ทดลอง โดยสามารถปรับได้ตามความเหมาะสม) ตัวอย่างการหาค่าเส้นตั้งฉากของมุมกล้องกับดวงตาในหน่วยเซนติเมตรแสดงในรูปที่ 7

การหาค่า BÂC หรือมุมองศาดวงตากับกล้องเพื่อ

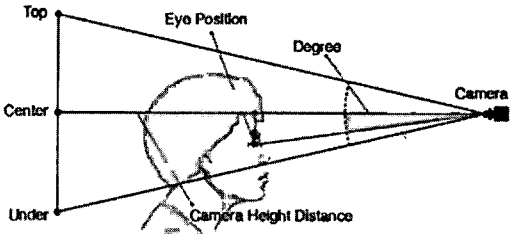


รูปที่ 7 การคำนวณเส้นตั้งฉากมุมกล้องกับดวงตา



รูปที่ 8 วิธีการสร้างระยะเพื่อใช้ในการคำนวณมุมกล้อง

นำไปใช้หาระยะห่างของผู้ใช้สามารถคำนวณได้ ด้วยการสร้างรูปสามเหลี่ยมเพื่อประยุกต์ใช้หลักการตรีโกณมิติ โดยผู้วิจัยกำหนดให้กล้องมองเห็นวัตถุที่มีขนาดความสูง 120 เซนติเมตร ซึ่งในมุมกล้องที่รับภาพขนาด 640*480 สามารถมองเห็นความสูงดังกล่าวได้ในระยะประมาณ 177 เซนติเมตร ดังตัวอย่างในรูปที่ 8 จากระยะและมุมมองข้างต้นสามารถนำมาคำนวณด้วยสมการที่ (4) การหามุมสามเหลี่ยมและตัวอย่างรูปที่ 9 การคำนวณมุมของดวงตากับกล้อง การทำมุมองศาของดวงตากับจอ (Degree) เกิดจากจุดกึ่งกลางการมองเห็น (Eye Position) ได้เบนออกจากจุดกึ่งกลางของกล้อง (Center) Up คือ ขอบเขตที่กล้องสามารถมองเห็นวัตถุในมุมเงยได้จากระยะห่างที่กำหนด Down คือ ขอบเขตที่กล้องสามารถมองเห็นวัตถุในมุมก้มได้จากระยะห่างที่กำหนด จากสมการดังกล่าวกำหนดให้ θ เป็นมุมองศา ค่า $\boldsymbol{p}$ คือ จุดกึ่งกลางการมองเห็นในแนวตั้ง (ตำแหน่งกึ่งกลางของดวงตาทั้งสองข้าง) ค่า $\boldsymbol{m}$ คือ ระยะจากจุดกึ่งกลางกล้องถึงขอบเขต



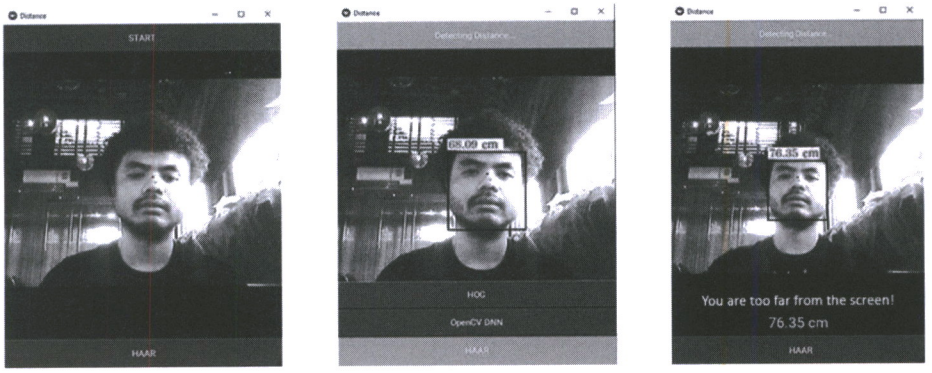
รูปที่ 9 แนวคิดการคำนวณมุมของดวงตากับกล้อง

รูปที่ 10 ตัวอย่างการตรวจหาระยะห่างด้วยภาพใบหน้า

ที่กล้องสามารถมองเห็นวัตถุในมุมเงยได้ ระยะห่างดังกล่าวมีค่า 60 เซนติเมตร ค่า $\boldsymbol{l}$ คือ ความสูงจากกึ่งกลางภาพถึงขอบบนสุดของภาพที่นำเข้าจากกล้องมีค่าเท่ากับ 240 พิกเซล $\boldsymbol{t}$ คือ ระยะห่างที่กำหนดให้กล้องมองวัตถุในการมีค่าเท่ากับ 177 เซนติเมตร

เมื่อได้ค่าเส้นตั้งฉากของมุมกล้องกับตำแหน่งดวงตาและมุมองศาของตำแหน่งกึ่งกลางการมองเห็นกับกล้องแล้ว จึงใช้สมการที่ (1) ในการคำนวณหาค่าระยะห่างตามหลักตรีโกณมิติ ซึ่งจะได้ระยะห่างของดวงตาผู้ใช้กับหน้าจอ รูปที่ 10 แสดงตัวอย่างการการตรวจหาระยะห่างด้วยภาพใบหน้า

รูปที่ 11 ตัวอย่างการแจ้งเตือน



รูปที่ 12 ตัวอย่างซอฟต์แวร์

$$\theta = \arctan\left(\frac{p\left(\frac{m}{l}\right)}{t}\right) \quad (4)$$

2.4 ขั้นตอนการแจ้งเตือน

ซอฟต์แวร์ที่พัฒนาขึ้นในงานวิจัยนี้ สามารถทำงานในระบบปฏิบัติการ Windows 10 ดังนั้น ตัวอย่างการทำงานและการแจ้งเตือนจะอยู่ในขอบเขตของระบบนั้น โดยเงื่อนไขการแจ้งเตือนจะเตือนเมื่อผู้ใช้งานมีระยะการเพ่งที่ความเสี่ยงต่อการเกิดอาการคอมพิวเตอร์วิชันซินโดรม ซึ่งระยะดังกล่าวอ้างอิงจากการป้องกันอาการขั้นพื้นฐาน โดยระยะห่างที่เหมาะสมในการเพ่งมองจอจะอยู่ประมาณ 40–75 เซนติเมตร ซึ่งซอฟต์แวร์จะคำนวณระยะห่างการมองจอของผู้ใช้ทุกๆ 3 วินาที เมื่อผู้ใช้มีระยะการมองไม่เหมาะสม จะทำการแจ้งเตือนในรูปแบบป๊อปอัพ (Popup Notification) ตัวอย่างการแจ้งเตือนแสดงในรูปที่ 11 รูปที่ 12 แสดงตัวอย่างซอฟต์แวร์ที่ได้พัฒนาขึ้น

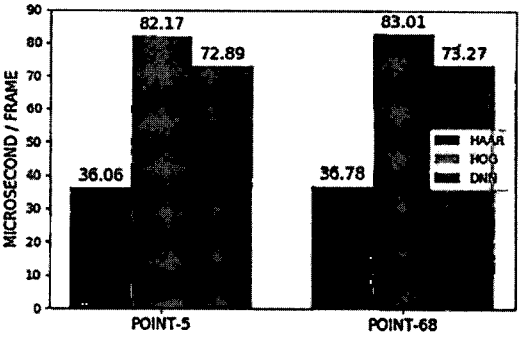
3. ผลการทดลองและอภิปรายผล

งานวิจัยนี้ได้ประเมินซอฟต์แวร์ที่พัฒนาขึ้นในหลายมิติ ประกอบด้วย การทดสอบความเร็วในการตรวจหาตำแหน่งของดวงตา การทดสอบความถูกต้องการตรวจหาระยะห่างของผู้ใช้ การประเมินการใช้งานได้ของระบบ เครื่องที่ใช้ทดสอบคือ เครื่องคอมพิวเตอร์ติดตั้งระบบปฏิบัติการวินโดวส์ (Windows 10) ซีพียูอินเทล ไอ3-2100 3.10 กิกะเฮริต์ซ สองแกน หน่วยความจำ 8 กิกะไบต์ จีพียู (GPU) รุ่นเอ็นวีเดีย จีฟอร์ซ จีทีเอ็กซ์ (Nvidia Geforce GTX) 1050 โดยผลการทดสอบมีดังนี้

3.1 การทดสอบความเร็ว

เนื่องจากการทำงานของระบบเป็นแบบเรียลไทม์ ความเร็วที่ใช้ต้องเหมาะสม ซึ่งการตรวจหาตำแหน่งของตาเป็นตัวแปรที่ส่งผลต่อความเร็วในการคำนวณ โดยในงานวิจัยนี้ ได้ใช้โมเดลหรือวิธีการเปรียบเทียบ 3 วิธี ได้แก่ ฮาร์ ฮอก และซีเอ็นเอ็น ทดสอบด้วยการตรวจหาดวงตาจากภาพเคลื่อนไหวขนาด 640*480 จำนวน 1,000 เฟรม

ผลการประเมินพบว่า วิธีการที่มีความเร็วมากที่สุดคือ ฮาร์ มีความเร็วประมาณ 36 มิลลิวินาทีต่อเฟรม รองลงมา คือ ซีเอ็นเอ็น มีความเร็วประมาณ 73 มิลลิวินาทีต่อเฟรม และฮอก มีความเร็วประมาณ 82 มิลลิวินาทีต่อเฟรม ตามลำดับ ซึ่งการตรวจหาจุดบนใบหน้านั้น การตรวจหาแบบ 68 จุด กับ 5 จุด มีความเร็วไม่แตกต่างกันแสดงในรูปที่ 13



รูปที่ 13 ผลการทดสอบความเร็ว

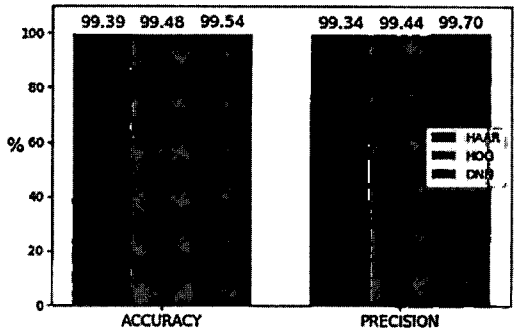
3.2 การทดสอบความถูกต้อง

การประเมินผลจะตรวจหาระยะห่างของด้วยตากับหน้าจอ ทดสอบในระยะ 40, 60 และ 80 ซม. ตรวจหาภาพทั้งหมด 1,000 เฟรม ใช้โมเดลการตรวจหาใบหน้าทั้ง 3 แบบก่อนหน้านี้ และการตรวจหาดวงตาใช้การตรวจหาคุณลักษณะบนใบหน้าแบบ 5 จุด ผลการประเมินแสดงได้ดังตารางที่ 2

ตารางที่ 2 ผลการทดสอบทั้งสามระยะและสามโมเดล

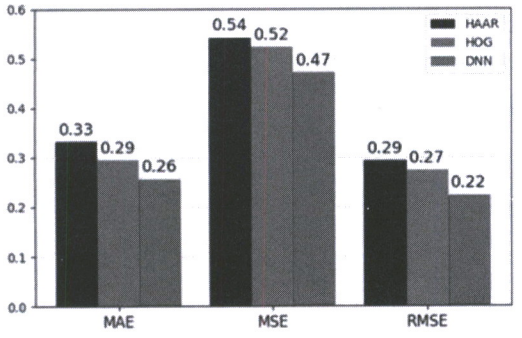
DIS.	ACC%	PRE%	MAE.	MSE.	RMSE.
HAAR40	0.99	0.99	0.26	0.42	0.17
HAAR60	0.99	0.99	0.62	0.66	0.44
HAAR80	1.00	1.00	0.11	0.52	0.27
HOG40	1.00	0.99	0.16	0.32	0.10
HOG60	0.99	0.99	0.63	0.69	0.47
HOG80	1.00	1.00	0.09	0.49	0.24
DNN40	1.00	0.99	0.14	0.30	0.09
DNN60	0.99	1.00	0.59	0.69	0.47
DNN80	1.00	1.00	0.04	0.33	0.11

ในแต่ละโมเดลการตรวจจับใบหน้ามีผลลัพธ์การประเมินที่ไม่แตกต่างกันมาก โดยความถูกต้องและความ



รูปที่ 14 ค่าความถูกต้องและแม่นยำ

เที่ยงตรงมีค่าเท่ากับร้อยละ 99 ทั้งคู่ แสดงในรูปที่ 14 ซึ่งถือว่าระบบมีแม่นยำสูงพอสมควร และความคาดเคลื่อนเฉลี่ยที่ดีที่สุดคือโมเดลซีเอ็นเอ็น มีค่าเฉลี่ยความคาดเคลื่อนสัมบูรณ์ (MAE) เท่ากับ 0.26 ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (MSE) เท่ากับ 0.47 ค่ารากที่สองของความคาดเคลื่อนเฉลี่ย (RMSE) เท่ากับ 0.22 รองลงมาคือโมเดล ฮอก (HOG) มีค่าเฉลี่ยความคาดเคลื่อนสัมบูรณ์ (MAE) เท่ากับ 0.29 ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (MSE) เท่ากับ 0.52 ค่ารากที่สองของความคาดเคลื่อนเฉลี่ย (RMSE) เท่ากับ 0.27 และฮาร์ (HAAR) มีค่าเฉลี่ยความคาดเคลื่อนสัมบูรณ์ (MAE) เท่ากับ 0.33 ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (MSE) เท่ากับ 0.54 ค่ารากที่สองของความคาดเคลื่อนเฉลี่ย (RMSE) เท่ากับ 0.29 ตามลำดับ แสดงในรูปที่ 15 ซึ่งจะเห็นได้ว่าวิธีการทั้งหมดมีคาดเคลื่อนในการตรวจหาระยะห่างดวงตากับหน้าจอที่น้อยมาก ดังนั้นในงานวิจัยนี้จึงเลือกใช้วิธีอาร์ (HAAR) ซึ่งมี



รูปที่ 15 ค่าความคาดเคลื่อน

ความเร็วมากที่สุด โดยมีความถูกต้องแม่นยำไม่แตกต่างจากวิธีการอื่น

3.3 การประเมินความสามารถในการใช้งานได้

การประเมินนี้เป็นการทดสอบการใช้งานได้จริงของซอฟต์แวร์ โดยใช้แบบสอบถามซูมิ (SUMI) [25] ในการตั้งคำถามการทำงานได้ของระบบ โดยกลุ่มตัวอย่างเป็นผู้ใช้งานคอมพิวเตอร์ทั่วไป 100 คน คำถามประกอบด้วยคำถาม 5 ด้าน ได้แก่ ประสิทธิภาพ ผลกระทบ ความช่วยเหลือ การควบคุม และการเรียนรู้ได้ ด้านละ 10 คำถาม แต่ละคำถามมีทั้งแง่บวกและลบ ระดับคะแนนของคำถามมีสามระดับ คือคำถามในแง่บวก เห็นด้วย (3) ไม่ตัดสินใจ (2) ไม่เห็นด้วย (1) และคำถามแง่ลบ เห็นด้วย (1) ไม่ตัดสินใจ (2) ไม่เห็นด้วย (3) การตีความค่าเฉลี่ยของคะแนนที่ได้มีห้าระดับ ได้แก่ แย่มาก (0-0.60) แย่ (0.61-1.20) ปกติ (1.21-1.80) ดี (1.81-2.4) และดีมาก (2.41-3.00)

การประเมินความสามารถในการใช้งานได้ของซอฟต์แวร์ทั้ง 5 ด้าน แสดงได้ดังตารางที่ 3 ซึ่งมีระดับการใช้งานที่ดีขึ้นไป ด้านผลกระทบ ความช่วยเหลือ และการควบคุม อยู่ในระดับดีมาก และด้านประสิทธิภาพ การเรียนรู้ได้อยู่ในระดับดี ซึ่งสามารถสรุปได้ว่าซอฟต์แวร์ที่พัฒนาขึ้นมีศักยภาพในการทำงานได้จริง

เนื่องจากงานวิจัยนี้เป็นการปรับปรุงวิธีการของซอฟต์แวร์อายการ์ดที่มีอยู่ก่อนแล้ว ผู้วิจัยจึงได้เปรียบเทียบความถูกต้องระหว่างทั้งสองวิธีการ ผลปรากฏว่า วิธีการของงานวิจัยนี้ได้รับค่าความผิดผลาดน้อยกว่าอายการ์ดอย่างเห็นได้ชัด ดังแสดงในตารางที่ 4 โดยมีค่าเฉลี่ยความคาดเคลื่อนสัมบูรณ์โดยรวม อายการ์ดเท่ากับ 1.67 งานวิจัยนี้เท่ากับ 0.29 ตามลำดับ ค่าเฉลี่ยความคลาดเคลื่อนกำลังสองโดยรวม อายการ์ดเท่ากับ 2.17 งานวิจัยนี้เท่ากับ 0.49 ตามลำดับ ค่ารากที่สองของความคาดเคลื่อนเฉลี่ยโดยรวม อายการ์ดเท่ากับ 2.06 งานวิจัยนี้เท่ากับ 0.26 ตามลำดับ

ตารางที่ 3 ผลการประเมินการใช้งานได้ของซอฟต์แวร์

ด้านการประเมิน	ผลการประเมิน
ประสิทธิภาพ	ดี (2.35)
ผลกระทบ	ดีมาก (2.58)
ความช่วยเหลือ	ดีมาก (2.46)
การควบคุม	ดีมาก (2.75)
การเรียนรู้ได้	ดี (2.12)

ตารางที่ 4 การเปรียบเทียบความถูกต้องระหว่างอายการ์ดและงานวิจัยนี้

Error Test	Eyegrade	This work
MAE	1.67	0.29
MSE	2.17	0.49
RMSE	2.06	0.26

4. สรุป

บทความนี้นำเสนอการพัฒนาซอฟต์แวร์เพื่อป้องกันอาการคอมพิวเตอร์วิชันซินโดรม โดยการตรวจหาระยะห่างของดวงตากับหน้าจอด้วยภาพใบหน้าของผู้ใช้ โดยซอฟต์แวร์จะแจ้งเตือนเมื่อผู้ใช้มีระยะการมองหน้าจอที่ไม่เหมาะสม เพื่อให้ผู้ใช้ปรับระยะที่เหมาะสมไม่เสี่ยงต่อการเกิดอาการคอมพิวเตอร์วิชันซินโดรม การตรวจหาระยะห่างระหว่างผู้ใช้งานกับหน้าจอของซอฟต์แวร์ ได้ประยุกต์ใช้เทคโนโลยีคอมพิวเตอร์วิชันร่วมกับหลักการคำนวณรูปสามเหลี่ยมมุมฉาก (Trigonometry)

การประเมินผลงานวิจัยนี้แสดงให้เห็นว่าระบบมีการตรวจหาระยะห่างที่ถูกและแม่นยำ โดยมีค่าความถูกต้องและเที่ยงตรงที่สูงถึงร้อยละ 99 ขึ้นไป และมีคาดเคลื่อนประมาณ 0.4 ซึ่งถือว่าน้อยมาก การประเมินความสามารถในการใช้งานได้บ่งบอกถึงความสามารถในการใช้งานได้จริงของซอฟต์แวร์

ในงานวิจัยนี้ได้ใช้วิธีการของซอฟต์แวร์อายการ์ดมาปรับปรุงและสร้างเป็นซอฟต์แวร์ตัวใหม่ขึ้นมาโดยนำเฉพาะมอดูลการตรวจหาระยะมาใช้ การปรับแต่งในงานวิจัยนี้ทำการทดลองหาโมเดลการตรวจหาใบหน้าที่เหมาะที่สุด และปรับแก้วิธีการคำนวณการหาระยะด้วยการปรับขนาดตัวแปรในการคำนวณ ได้แก่ ความสูงของการมองเห็น และระยะการมองเห็นวัตถุ ซึ่งได้อธิบายไว้ในสมการที่ (4) โดยการทดสอบการตรวจหาใบหน้านั้น เป็นวิธีการที่ได้เป็นวิธีการเดียวกับอายการ์ด แต่อย่างไรก็ตาม การคำนวณระยะห่างในงานวิจัยนี้มีค่าความถูกต้องที่ดีกว่าอายการ์ดอย่างเห็นได้ชัด

หากนำการปรับปรุงการตรวจหาระยะในงานวิจัยนี้ไปใช้ในซอฟต์แวร์อายการ์ด จะไม่กระทบต่อประสิทธิภาพของมอดูลอื่นในอายการ์ดแต่อย่างใด เนื่องจากมอดูลแต่ละมอดูลจะมีการทำงานหรือการคำนวณค่าต่างๆ ที่แยกส่วนกันอยู่แล้ว

งานวิจัยในอนาคต ซอฟต์แวร์ที่พัฒนาขึ้นจะนำภาพใบหน้าหรือตำแหน่งของจุดใบใบหน้า ไปประยุกต์ใช้งานในปัจจัยอื่นๆ เพิ่มเติมเพื่อป้องกันอาการคอมพิวเตอร์วิชันซินโดรมนอกเหนือจากการตรวจหาระยะห่าง

เอกสารอ้างอิง

[1] M. Rosenfield, "Computer vision syndrome (a.k.a. digital eye strain)," *Optometry in Practice*, vol. 17, no. 1, pp. 1–10, 2016.

[2] L. Mowatt, C. Gordon, A. B. R. Santosh, and T. Jones, "Computer vision syndrome and ergonomic practices among undergraduate university students," *International Journal of Clinical Practice*, vol. 72, no. 1, 2017.

[3] HootSuite. (2019). *The global state of digital in 2019.* [Online]. Available: https://hootsuite.com/resources/digital-in-2019

[4] M. Salehan and A. Negahban, "Social networking on smartphones: When mobile phones become addictive," *Computers in Human Behavior*, vol. 29, no. 6, pp. 2632-2639, 2013.

[5] S. Munshi, A. Varghese, and S. Dhar-Munshi, "Computer vision syndrome–A common cause of unexplained visual symptoms in the modern era," *International Journal of Clinical Practice*, vol. 71, no. 7, 2017,

[6] American Optometric Association. (2020). *Computer Vision Syndrome.* [Online]. Available: https://www.aoa.org/patients-and-public/caring-for-your-vision/protecting-your-vision/computer-vision-syndrome

[7] J. S. Jennifer and T. S. Sharmila, "Edge based eye-blink detection for computer vision syndrome," *in Proceedings 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP),* 2017, pp. 1–5.

[8] F. Vieira, E. Oliveira, and N. Rodrigues, "iSVC – digital platform for detection and prevention of computer vision syndrome," *in Proceedings 2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH),* 2019, pp. 1–7.

[9] N. Julius and E. E. Mustapha, "Take-A-Break notification: An ergonomic application," in *Proceedings of the 6th International Conference on Information Technology and Multimedia,* 2014, pp. 390–395.

[10] J. Ho, R. Pointner, H.-C. Shih, Y.-C. Lin, H.-Y. Chen, W.-L. Tseng, and M. Y. Chen, "EyeProtector: Encouraging a healthy viewing distance

when Using smartphones," presented at the Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, Copenhagen, Denmark, 2015.

[11] T. Toda, M. Nakai, and L. Xinxin, "A close face-distance warning system for straightend neck prevention," in *Proceedings IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society,* 2015, pp. 003347–003352.

[12] P. Wasnik and A. Jeyakumar, "Monitoring stress level parameters of frequent computer users," in *Proceedings 2016 International Conference on Communication and Signal Processing (ICCSP),* 2016, pp. 1753–1757.

[13] W. Pookhuntod, J. Angskun, and T. Angskun, "EyeGuard: A software module for reducing factors causing computer vision syndrome," *Journal of Science and Technology Mahasarakham University,* vol. 37, no. 6, pp. 747–758, 2018 (in Thai).

[14] I. König, P. Beau, and K. David, "A new context: Screen to face distance," in *Proceedings 2014 8th International Symposium on Medical Information and Communication Technology (ISMICT),* 2014, pp. 1–5.

[15] Z. Li, W. Chen, and K. Bian, "Look into my eyes: Fine-grained detection of face-screen distance on smartphones," in *Proceedings 2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN),* 2016, pp. 258–265.

[16] M. A. Hossain and M. Mukit, "A real-time face to camera distance measurement algorithm using object classification," in *Proceedings 2015 International Conference on Computer and Information Engineering (ICCIE),* 2015, pp. 107–110.

[17] X. Dong, F. Zhang, and P. Shi, "A novel approach for face to camera distance estimation by monocular vision," *International Journal of Innovative Computing, Information and Control,* vol. 10, pp. 659–669, 2014.

[18] C. Godard, O. Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017, pp. 270–279.

[19] S. Bianco, M. Buzzelli, and R. Schettini, "A unifying representation for pixel-precise distance estimation," *Multimedia Tools and Applications,* vol. 78, no. 10, pp. 13767–13786, 2019.

[20] G. Bradski, "The openCV library," *Dr. Dobb's Journal of Software Tools,* vol. 120, pp. 122–125, 2000.

[21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001,* 2001, vol. 1, pp. I-I.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," presented at the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, USA, 2005.

[23] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research,* vol. 10, no. 60, pp. 1755–1758, 2009.

[24] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 Faces in-the-

wild challenge: Database and results," *Image and Vision Computing*, vol. 47, pp. 3–18, 2016.

[25] J. Kirakowski and M. Corbett, "SUMI: The software usability measurementinventory," *British Journal of Educational Technology*, vol. 24, no. 3, pp. 210–212, 2006.