



การพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ถูกปิดบางส่วน



วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

การพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ถูกปิดบางส่วน



วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

วิทยานิพนธ์ เรื่อง "การพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ถูกปิด
บางส่วน"

ของ สิทธิศักดิ์ ทีโน

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบวิทยานิพนธ์
(ดร.ชูชาติ หลุยไชยศักดิ์)

..... ประธานที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ธิยะมงคล)

..... กรรมการที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.สุเมธ เหมะวัฒนะชัย)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(รองศาสตราจารย์ ดร.พนัส นัถฤทธิ์)

อนุมัติ

.....
(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง	การพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ ถูกปิดบางส่วน
ผู้วิจัย	สิทธิศักดิ์ ทีโน
ประธานที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล
กรรมการที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุเมธ เหมะวิฒนะชัย
ประเภทสารนิพนธ์	วิทยานิพนธ์ วศ.ม. วิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2565
คำสำคัญ	ตรวจจับทิศทางใบหน้า, การควบคุมด้วยท่าทางใบหน้า, สถานะดวงตา

บทคัดย่อ

งานวิจัยนี้เป็นการพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและดวงตาของผู้ใช้งาน โดยในงานวิจัยนี้แบ่งการพัฒนาออกเป็น 2 รูปแบบ โดยรูปแบบแรก เป็นการพัฒนาโปรแกรมเพื่อตรวจจับทิศทางของใบหน้าเช่นการหันซ้าย หันขวา ก้มหน้าและเงยหน้า รูปแบบที่ 2 เป็นการตรวจจับสถานะของดวงตาผู้ใช้ เช่น ตาปิดหรือตาเปิดและทิศทางของลูกตาวามองไปทางซ้ายหรือทางขวา โดยหลักการทำงานของโปรแกรมจะเริ่มจากการตรวจจับใบหน้าและหาจุดสำคัญของใบหน้า โดยในงานวิจัยนี้ใช้ไลบรารีของ Mediapipe ซึ่งจะทำให้ได้จุดสำคัญต่างๆบนใบหน้าจำนวน 468 จุด โดยการหาทิศทางของใบหน้าจะใช้จุดสำคัญที่ได้จำนวน 5 จุดคือบริเวณจุดมุมดวงตาซ้าย ขวาและปลายจมูก เพื่อนำมาคำนวณหาทิศทางของเวกเตอร์ที่ชี้ตั้งแต่ตรงกลางระหว่างตาสองข้างไปถึงปลายจมูกจากนั้นทำการหาทิศทางของใบหน้าด้วยการแปลงมุมมองของระบบพิกัด จะทำให้ได้องศาการหมุนของใบหน้าเทียบกับขอบเขตมุมมองอ้างอิงจะได้ผลออกมา 5 ลักษณะคือ หน้าตรง หันซ้าย หันขวา ก้มหน้าและเงยหน้า

ในกรณีของการตรวจจับดวงตาในงานวิจัยนี้ใช้จุดพิกัดขอบบนและขอบล่างของดวงตาแต่ละข้างมาเปรียบเทียบระยะห่างว่าเป็นตาเปิดหรือตาปิด จากนั้นเอาพิกัดจุดกลางลูกตามาเปรียบเทียบกับมุมดวงตาด้านซ้ายและขวาเพื่อหาทิศทางการมองของลูกตา

ในการทดสอบความถูกต้องของโปรแกรมที่พัฒนาขึ้น โดยการทดลองมีจำนวนอาสาสมัครจำนวน 61 คน ผู้ชาย 47 คน และผู้หญิง 14 คน โดยทำการบันทึกวิดีโอของใบหน้าทั้งกรณีไม่ใส่และใส่หน้ากากอนามัยในระหว่างการบันทึกมีการหันหน้าซ้าย ขวา ก้มหน้าและเงยหน้า ผลที่ได้พบว่า

โปรแกรมตรวจจับทิศทางใบหน้ามีความถูกต้องเฉลี่ยร้อยละ 95.08 จึงมีศักยภาพในการนำไปประยุกต์ใช้ในงานต่างๆได้ เช่น การควบคุมแก๊วฮีเซ็นไฟฟ้าด้วยใบหน้าและดวงตาได้



Title	DEVELOPING A PROGRAM TO DETECT FACE DIRECTION AND THE STATE OF PARTIALLY CLOSED EYES
Author	Sitthisak Thino
Advisor	Assistant Professor Panomkhawn Riyamongkol, Ph.D.
Co-Advisor	Assistant Professor Sumet Heamawatanachai, Ph.D.
Academic Paper	M.Eng. Thesis in Computer Engineering, Naresuan University, 2022
Keywords	face-direction detection, facial gesture control, eye status

ABSTRACT

This research focuses on the development of software for detecting facial direction and eye status of users. The software development is divided into two parts. The first part involves the creation of a program that can accurately detect the direction of the face, including turning left, turning right, and raising or lowering the face. The second part focuses on detecting the status of the user's eyes, such as whether they are open or closed, as well as determining the direction of the gaze, whether it is towards the left or right. The developed program of this research utilizes the Mediapipe library, which offers 468 key points on the face, as the foundation for its working principle. In this research, a specific subset of five key points is employed, consisting of the corners of the left and right eyes, as well as the tip of the nose. These key points are crucial for calculating the direction vector from the center between the two eyes to the tip of the nose, enabling determination of the facial direction. By converting the perspective angles of the coordinate system, the program categorizes the rotation angle of the face into five distinct categories: straight face, turned left, turned right, raised face, and lowered face.

In the case of eye detection, the program utilizes the coordinates of the top and bottom edges of each eye to determine their status (open or closed). Additionally,

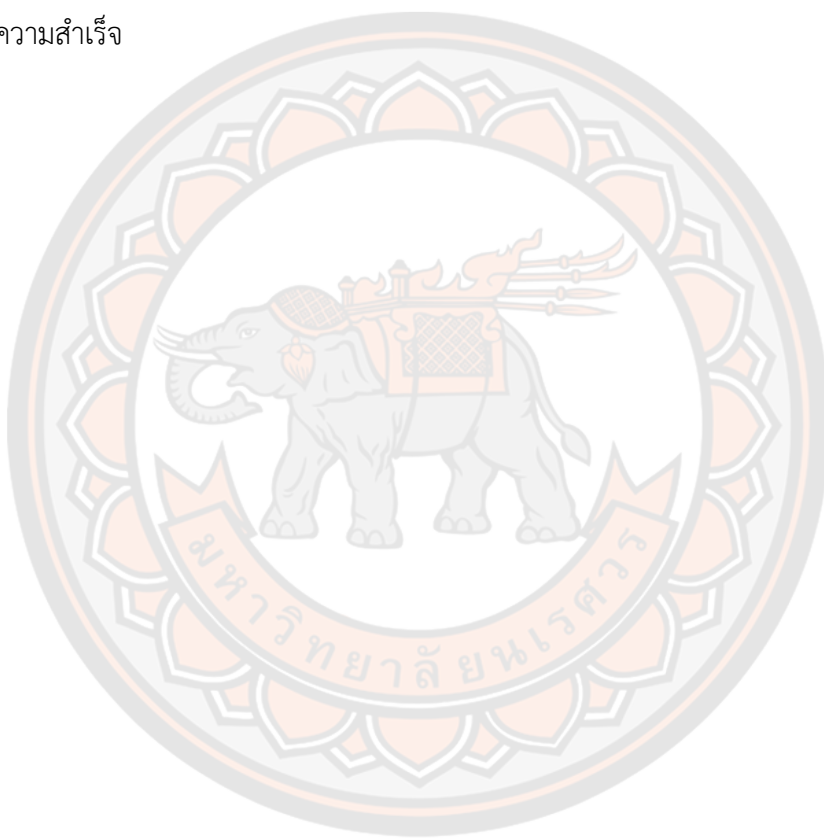
by comparing the coordinates of the center of the iris with the coordinates of the left and right eye corners, the program is able to determine the direction of gaze.

The accuracy of the developed program was evaluated during the testing phase, involving a total of 61 volunteer participants, consisting of 47 males and 14 females. The participants' faces were recorded in video format under two conditions: without wearing a face mask and while wearing a face mask. Throughout the recording process, participants were instructed to perform specific facial movements, including turning their faces to the left, right, and raising or lowering their faces. The evaluation results indicated that the program achieved an impressive average accuracy of 95.08% in detecting facial direction. These findings highlight the program's potential for diverse applications, such as the utilization of facial direction and eye movements for controlling electric wheelchairs.



ประกาศคุณูปการ

ในการดำเนินการวิจัยในครั้งนี้ได้รับความร่วมมือเป็นอย่างดีจาก อาจารย์รัฐภูมิ วรรณุศาสน์ หน่วยวิจัยด้านเทคโนโลยีที่ศนศาสตร์คอมพิวเตอร์และการโต้ตอบกับมนุษย์ (NU Vision Lab) หน่วยวิจัยเทคโนโลยีด้านวิศวกรรมความเที่ยงตรงและการแพทย์ (PMET) คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร รวมทั้งได้รับความช่วยเหลือในการประสานงานและใช้สถานที่ในการจัดทดลองจากเจ้าหน้าที่จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ขอขอบคุณที่ให้ความร่วมมือเป็นอย่างดีจนการวิจัยประสบความสำเร็จ



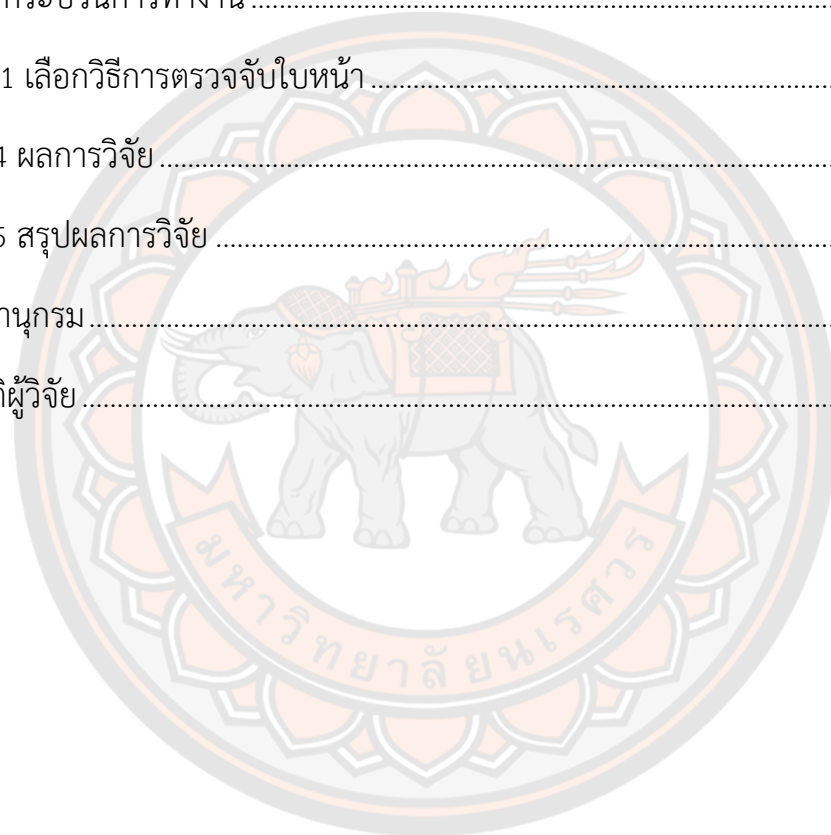
สิทธิศักดิ์ ทีโน

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	จ
ประกาศคุณูปการ.....	ช
สารบัญ.....	ช
สารบัญตาราง.....	ฉ
สารบัญรูปภาพ.....	ฉ
บทที่ 1 บทนำ.....	1
ที่มาและปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	2
งานวิจัยนี้มุ่งเน้นส่วนของการหาทิศทางของใบหน้าและสถานะของดวงตาเป็นส่วนใหญ่ เพื่อความถูกต้องและแม่นยำในการใช้งาน โดยมีวัตถุประสงค์หลักที่สำคัญดังนี้.....	2
ขอบเขตการวิจัย.....	2
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	3
แนวคิดและทฤษฎีที่เกี่ยวข้อง.....	3
การประมวลผลภาพ (Image Processing).....	3
ภาพระดับสีเทา (Grayscale Image).....	4
Color Models.....	4
RGB Color model.....	4
CMYK Color model.....	5

HSI Color model	5
Morphological.....	6
Line, and Edge Detection.....	7
Data Classification.....	7
SVM (Support Vector Machine).....	8
KNN (k-nearest neighbors).....	8
Neural Network.....	9
มุมออยเลอร์ (Euler's Angle).....	9
เครื่องมือและซอฟต์แวร์ที่ใช้งาน.....	10
Python.....	10
Open CV.....	10
Dlib 11	
MediaPipe.....	11
งานวิจัยที่เกี่ยวข้อง.....	12
Rapid Object Detection using a Boosted Cascade of Simple Features [6]	12
Partial Face Detection for Continuous Authentication [7]	12
Face Recognition from Partial Face Data [8].....	13
Face Alignment Across Large Poses: A 3D Solution [9].....	13
How far are we from solving the 2D & 3D Face Alignment problem? (and a....	14
dataset of 230,000 3D facial landmarks) [10].....	14
GlassGesture: Exploring Head Gesture Interface of Smart Glasses [11].....	15

Nod to Auth: Fluent AR/VR Authentication with User Head-Neck Modeling [12]	15
.....
บทที่ 3 วิธีดำเนินการวิจัย.....	17
3.1 ภาพการทำงานโดยรวมของของโปรแกรมที่พัฒนาขึ้น	17
3.2 วิธีการเก็บข้อมูล	17
3.3 กระบวนการทำงาน	18
3.3.1 เลือกวิธีการตรวจจับใบหน้า	18
บทที่ 4 ผลการวิจัย	28
บทที่ 5 สรุปผลการวิจัย	41
บรรณานุกรม	42
ประวัติผู้วิจัย	45



สารบัญตาราง

	หน้า
ตาราง 1 ผลการตรวจจับทิศทางการหันหน้า.....	28
ตาราง 2 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย	29
ตาราง 3 ผลการตรวจจับทิศทางการหันหน้า (1 เมตร)	30
ตาราง 4 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (1 เมตร).....	30
ตาราง 5 ผลการตรวจจับทิศทางการหันหน้า (2 เมตร)	31
ตาราง 6 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (2 เมตร).....	31
ตาราง 7 ผลการตรวจจับทิศทางการหันหน้า (ลดแสง)	32
ตาราง 8 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (ลดแสง).....	33
ตาราง 9 ผลการตรวจจับทิศทางการหันหน้า (เพิ่มแสง)	33
ตาราง 10 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (เพิ่มแสง).....	34
ตาราง 11 ผลการตรวจจับทิศทางการหันหน้าเฉพาะคนใส่แว่นตา	34
ตาราง 12 ผลการตรวจจับสถานะดวงตาซ้าย (1).....	35
ตาราง 13 ผลการตรวจจับสถานะดวงตาซ้าย (2).....	36
ตาราง 14 ผลการตรวจจับสถานะดวงตาขวา (1).....	36
ตาราง 15 ผลการตรวจจับสถานะดวงตาขวา (2).....	36
ตาราง 16 คำสั่งที่ใช้ในการส่งไปให้รถเซ็นไฟฟ้า.....	38
ตาราง 17 ผลที่ได้จากการทดสอบการหันหน้า.....	38

สารบัญรูปภาพ

	หน้า
รูปภาพ 1 ขั้นตอนการรับภาพสู่ข้อมูลดิจิทัล.....	3
รูปภาพ 2 แสงสีขาวเมื่อผ่าน prism จะเป็น color spectrum.....	4
รูปภาพ 3 กล้องแสดงค่าสี RGB.....	4
รูปภาพ 4 สูตรการแปลงค่าสีจาก RGB ไปเป็น CMY.....	5
รูปภาพ 5 วงกลมแสดงค่าสี (Hue) ด้านขวาแนวตั้งเป็นแนวที่แสดงความเข้มของแสง.....	6
รูปภาพ 6 ตัวอย่างการใช้ Morphological.....	7
รูปภาพ 7 Hard support vector machines.....	8
รูปภาพ 8 ตัวอย่าง k-nn.....	8
รูปภาพ 9 Single and multilayer neural networks.....	9
รูปภาพ 10 มุมออยเลอร์บนใบหน้า.....	10
รูปภาพ 11 ML solutions in MediaPipe.....	11
รูปภาพ 12 2 features ที่ใช้สำหรับ training face.....	12
รูปภาพ 13 ผลของการใช้ FACIAL SEGMENT BASED FACE DETECTOR.....	13
รูปภาพ 14 ภาพรวมการทำงานของ Face Recognition from Partial Face Data.....	13
รูปภาพ 15 ผลลัพธ์ของ 3DDFA.....	14
รูปภาพ 16 ผลลัพธ์ที่ได้จากการใช้ 2D & 3D Face Alignment.....	14
รูปภาพ 17 google glass.....	15
รูปภาพ 18 รูปภาพการทำงานของ Nod to Auth user identification.....	16
รูปภาพ 19 แผนภาพแสดงการทำงาน.....	17

รูปภาพ 20 จุดสำคัญบนในหน้าที่ผ่าน model ของ dlib.....	19
รูปภาพ 21 จุดสำคัญบนในหน้าที่โดยใช้ mediapipe facemesh.....	20
รูปภาพ 22 ภาพแสดงจุดบน 2D และ 3D	21
รูปภาพ 23 แสดงตำแหน่งต่างๆบน World Coordinates และ Camera Coordinates.....	21
รูปภาพ 24 vector ที่กำหนดขึ้นมาและแสดงในมุมมองต่างๆ.....	23
รูปภาพ 25 เวกเตอร์ในพิกัด c และ พิกัด 0	24
รูปภาพ 26 แสดงมุมของเวกเตอร์ v เทียบกับแกน Z_0	25
รูปภาพ 27 A. ระบบพิกัด 0 ของเวกเตอร์ v B. ระบบพิกัด 2 ของเวกเตอร์ v.....	25
รูปภาพ 28 พิกัด 2 เทียบกับมุมมองด้านข้าง.....	26
รูปภาพ 29 ระยะการหันหน้าที่กำหนดไว้ระหว่าง หันขวา หน้าตรง และ หันซ้าย.....	26
รูปภาพ 30 ระยะการหันหน้าที่กำหนดไว้ระหว่าง เงยหน้า หน้าตรง และ ก้มหน้า	27
รูปภาพ 31 ผลการตรวจจับดวงตา.....	27
รูปภาพ 32 การตรวจจับม่านตาที่มองทางซ้าย.....	27
รูปภาพ 33 ระยะห่าง 0.5 เมตร 1 เมตร และ 2 เมตร ตามลำดับ.....	29
รูปภาพ 34 ภาพที่ปรับปรุงแสง ลดแสง ปกติ และเพิ่มแสงตามลำดับ.....	32
รูปภาพ 35 การส่งคำสั่งไปให้รถเซ็นไฟฟ้าผ่าน Bluetooth	37
รูปภาพ 36 ตัวอย่างโปรแกรมการหันหน้าสำหรับใช้กับรถเซ็นไฟฟ้า	37
รูปภาพ 37 รูปที่ไม่สามารถตรวจจับใบหน้าได้.....	39
รูปภาพที่ 38 ตัวอย่างการตรวจจับจุดสำคัญบนใบหน้าที่ผิดปกติ.....	39
รูปภาพที่ 39 ตัวอย่างภาพที่มุมของกล้องต่ำเกินไป	40

บทที่ 1

บทนำ

ที่มาและปัญหา

การพัฒนาเทคโนโลยีตรวจจับใบหน้าปัจจุบันมีการพัฒนาอย่างต่อเนื่อง รวดเร็วและแม่นยำขึ้นเป็นอย่างมาก มีการนำไปใช้อย่างแพร่หลายทั้งอุปกรณ์ในชีวิตประจำวันหรือตามองค์กรต่างๆ และแนวโน้มในอนาคตจะมีการใช้เทคโนโลยีการตรวจจับใบหน้ามากขึ้นด้วยเช่นกัน

ตอนนี้สามารถพบเห็นการตรวจจับใบหน้าได้ตามอุปกรณ์ต่างๆ เช่น กล้องถ่ายรูปที่นำการตรวจจับใบหน้ามาใช้งานเพื่อโพกสหน้าคนที่ถูกถ่ายได้ดียิ่งขึ้น กล้องถ่ายรูปในสมาร์ตโฟนก็เช่นกัน นอกจากนั้นในสมาร์ตโฟนบางรุ่นยังนำมาใช้อย่างอื่นอีกเช่นใช้ในการปลดล็อคแทนการใส่รหัส, การวาดรหัสผ่าน หรือการใช้ลายนิ้วมือ นอกจากนั้นยังสามารถเอามาใช้ในการเพิ่มแสงหน้าจอสมาร์ตโฟนอันโนมิติได้ด้วย บางแอปพลิเคชันในสมาร์ตโฟนนำมาใช้เพื่อความบันเทิงเช่นตกแต่งใบหน้าให้ดูดีขึ้นตกแต่งใบหน้าเพื่อเพิ่มความสนุกสนานหรือแม้กระทั่งระบุตัวตนในรูปแบบที่เราอัปโหลดบนโซเชียลอันโนมิติโดยที่เราไม่ต้องมากดเอง บางองค์กรนำมาใช้ในการเข้างานแทนการต่อบัตรเพื่อความสะดวกระบายในการเข้างาน

การตรวจจับใบหน้าจะมีปัญหาใหญ่อยู่บางอย่างเช่น นำภาพถ่ายมาใช้แทนคนจริงทำให้ไม่สามารถแยกแยะคนจริงกับรูปภาพได้เพราะใช้อัลกอริทึมใช้แค่รูปเดียวในการทำงานทำให้ไม่สามารถแยกได้ ถึงแม้ปัจจุบันมีกล้องที่สามารถรับรู้ความรู้สึกได้สามารถแยกรูปภาพกับคนจริงได้ด้วยการดูที่ความรู้สึกหรือระยะห่างจากกล้อง แต่ปัจจุบันกล้องที่สามารถรับรู้ความรู้สึกได้นั้นราคายังสูงอยู่ ปัญหาอีกอย่างคือการใช้วัตถุมาปิดบังบนใบหน้าอย่างเช่นแมสแทบทุกคนที่ออกมาที่สาธารณะ ซึ่งปิดใบหน้าไปเกือบครึ่งหน้า ปัจจุบันมีการปรับปรุงให้สามารถตรวจจับใบหน้าที่สวมใส่แมสได้และมีความแม่นยำมากขึ้นแต่ถ้าจะหาจุดและบนตำแหน่งบนใบหน้าอาจจะมีการคลาดเคลื่อนได้อยู่ นอกจากนี้ยังมีปัญหาเรื่องการหันหน้าไม่ตรงก็มีส่วนทำให้การตรวจจับใบหน้าคลาดเคลื่อนได้

จะเห็นได้ว่าการตรวจจับใบหน้านั้นยังได้รับความนิยมและยังมีปัญหาในบางเรื่องอยู่ แต่ถ้าสามารถรู้ว่าภาพที่เรามาตรวจจับหาใบหน้าที่มีปัญหาเรื่องใดอยู่ก็จะทำให้สามารถหลีกเลี่ยงวิธีการไม่ให้เกิดปัญหาหรือสามารถนำมาแก้ปัญหาก่อนได้ จะทำให้การตรวจจับใบหน้ามีความแม่นยำขึ้นและยังรับรู้สถานะบางอย่างบนใบหน้าได้ด้วยเช่นตรงบริเวณปากถูกปกปิดอยู่ เป็นต้น

ดังนั้นในงานวิจัยนี้จึงมุ่งเน้นในการศึกษาและพัฒนาโปรแกรมเพื่อตรวจจับทิศทางของใบหน้า และดวงตาด้วยการต่อยอดของผลที่ได้จากเทคนิคเดิมและนำมาต่อยอดเพิ่มเติม เพื่อประยุกต์ใช้กับ

งานที่ใช้ประโยชน์จากข้อมูลบนใบหน้าเช่น การนำไปใช้ในการควบคุมเช่นควบคุมรถเข็นด้วยทิศทางการหันหน้า

วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มุ่งเน้นส่วนของการหาทิศทางของใบหน้าและสถานะของดวงตาเป็นส่วนใหญ่ เพื่อความถูกต้องและแม่นยำในการใช้งาน โดยมีวัตถุประสงค์หลักที่สำคัญดังนี้

1. พัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ถูกปิดบางส่วน
 - บอกทิศทางการหมุนของใบหน้าได้
 - บอกสถานะของดวงตาได้ว่าตอนนี้ปกติ หลับตา
 - สามารถตรวจจับใบหน้าปกติและใบหน้าที่มีอะไรมาบังบางส่วนได้

2. สามารถนำสถานะของใบหน้าที่ได้มาเลือกใช้กับการประยุกต์ 1 อย่างได้ เช่น นำไปควบคุมอุปกรณ์ที่ติดกับรถเข็นให้รถเข็นเดินตามทิศทางที่กำหนดได้

ด้วยวัตถุประสงค์ดังกล่าวหวังว่างานวิจัยนี้จะสามารถนำไปพัฒนาและประยุกต์ใช้ให้มีประสิทธิภาพมากขึ้น

ขอบเขตการวิจัย

1. ใน 1 รูปมีแค่ใบหน้าเดียวเท่านั้น ถ้าในรูปมีหลายหน้าให้เลือกใบหน้าที่อยู่ตรงกลาง
2. สิ่งที่ปิดบังใบหน้าต้องไม่ปิดบังดวงตาทั้งสองข้าง แต่ใส่แว่นได้
3. การใส่หน้ากากอนามัยต้องไม่มีสิ่งปิดบังใบหน้าอย่างอื่นเพิ่ม
4. การหันหน้าไม่เกิน 45 องศา จากหน้าตรง

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

แนวคิดและทฤษฎีที่เกี่ยวข้อง

การประมวลผลภาพ (Image Processing)

กระบวนการจัดการและวิเคราะห์รูปภาพให้เป็นข้อมูลในแบบดิจิทัล โดยใช้คอมพิวเตอร์การ เพื่อให้ได้ข้อมูลที่เรากำลังต้องการทั้งในเชิงคุณภาพและปริมาณ (ขนาด รูปร่าง) สามารถนำข้อมูลไปวิเคราะห์ และไปใช้ประโยชน์ได้ ด้านการแพทย์ เครื่องมือที่ใช้งาน เช่น เครื่อง MRI (Magnetic Resonance Imaging) หรือ การถ่ายภาพด้วยอัลตราซาวด์ (Ultrasound), ระบบดูแลการจราจรบนท้องถนน, ระบบอ่านป้ายทะเบียนรถ, ระบบตรวจสอบคุณภาพของผลิตภัณฑ์ในกระบวนการผลิต, การวิเคราะห์ภาพถ่ายดาวเทียม, ระบบตรวจจับใบหน้า [1]

คลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นต่างๆ นั้นสามารถนำไปใช้สร้าง ภาพได้ แต่การที่จะรับภาพเหล่านั้นจำเป็นต้องมีเซนเซอร์ที่เหมาะสมเพื่อตรวจรับสัญญาณตอบสนอง (Image Sensing) ที่ได้รับมาและแปลงให้เป็นภาพสองมิติหรือที่เรียกว่า “เทคนิคการได้มาของภาพ (Image Acquisition)”

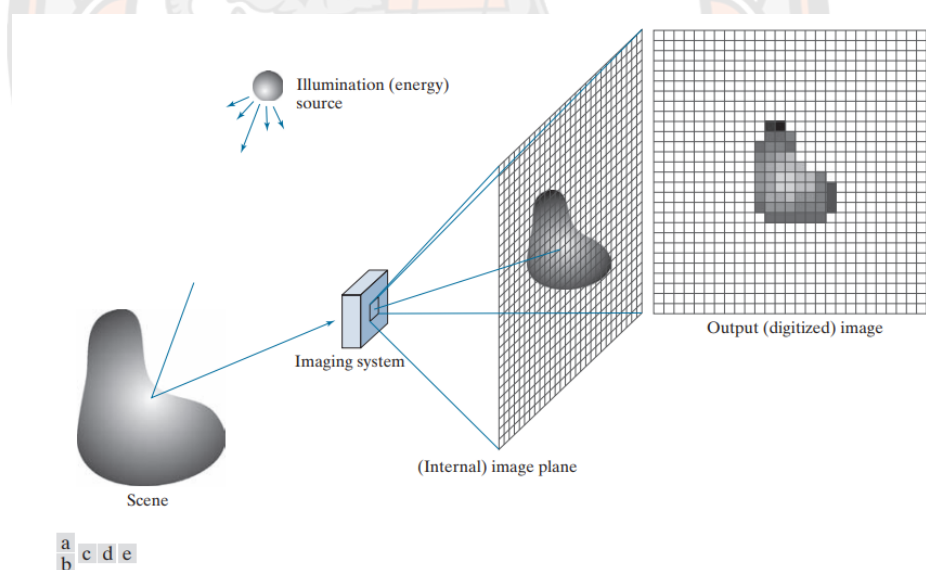


FIGURE 2.15 An example of digital image acquisition. (a) Illumination (energy) source. (b) A scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

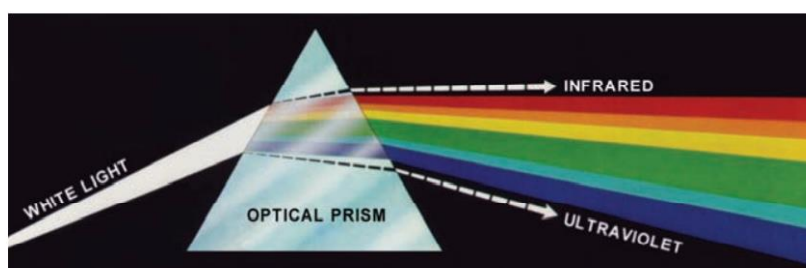
รูปภาพ 1 ขั้นตอนการรับภาพสู่ข้อมูลดิจิทัล

ภาพระดับสีเทา (Grayscale Image)

ภาพระดับสีเทาเป็นภาพซึ่งค่าในแต่ละจุดภาพคือค่าความเข้มของสีแต่ ละตำแหน่งของ จุดภาพนั้น ซึ่งค่าที่เป็นได้ของภาพระดับสีเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่าง เช่น ภาพ ระดับสี เทา 8 บิตที่ระดับสี ทั้งหมด 256 ระดับ โดยนิยมระบุในช่วง 0-1 หรือ 0-255

Color Models

เมื่อแสงสีขาวผ่าน prism จะปรากฏ spectrum ของสีตั้งแต่สีแดงไปจนถึงสีม่วง สีแต่ละสีจะ มีความถี่ที่แตกต่างกัน



รูปภาพ 2 แสงสีขาวเมื่อผ่าน prism จะเป็น color spectrum

RGB Color model

เกิดจากการรวมตัวกันของสเปกตรัมของ แสงสีแดง (Red), เขียว (Green) และน้ำเงิน (Blue) ในสัดส่วน ความเข้มขั้นที่แตกต่างกัน ตาม Figure 3 เมื่อนำ RGB มาใส่ใน รูปแบบ cube จะแสดงการไล่ระดับสี สีหลักทั้ง 3 สีจะมีค่า 0-255 จะเห็นมุมหนึ่งเป็นสีขาว เพราะทุกสีรวมกันจะได้เป็นสีขาว (255,255,255)



รูปภาพ 3 กล่องแสดงค่าสี RGB

CMYK Color model

CMYK ย่อมาจาก Cyan (ฟ้า) Magenta (แดงอมม่วง) Yellow (เหลือง) Key (สีดำ) ส่วนมากเหมาะสำหรับใช้ในการพิมพ์ต่างๆ C, M, Y นั้นจะอยู่ตรงข้ามกับ R, G, B สามารถแปลงจาก RGB ได้จากสมการข้างล่าง ซึ่งค่า RGB จะต้องถูก normalized ให้อยู่ในช่วง 0-1 ก่อน และค่า K สามารถหาได้จาก $K = \min(C,M,Y)$

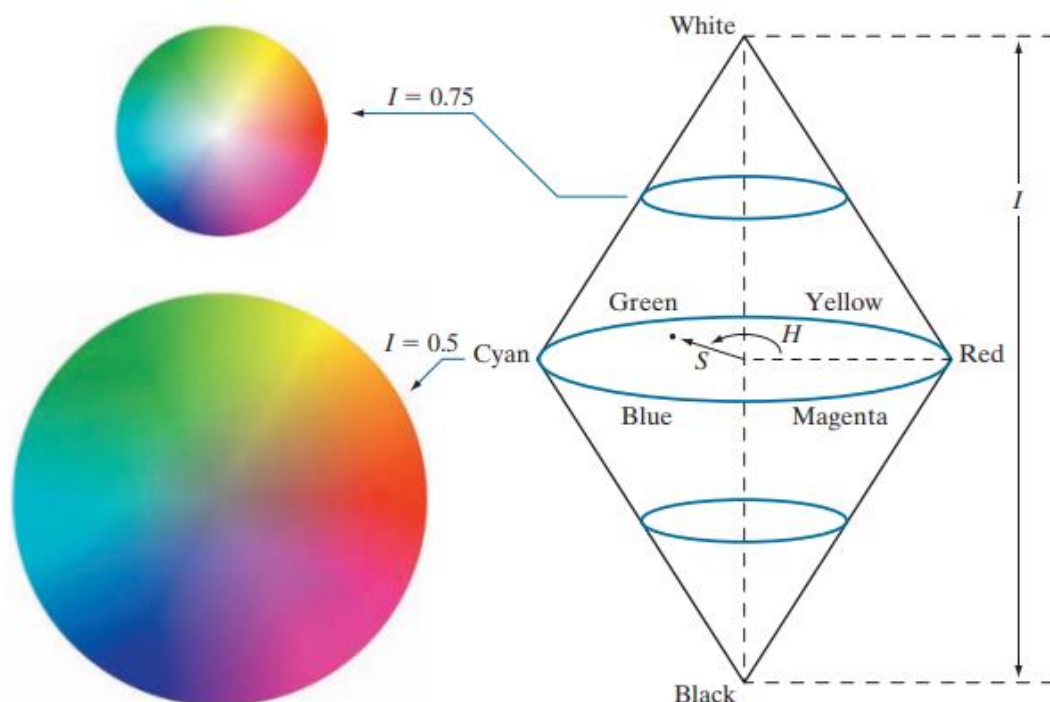
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

รูปภาพ 4 สูตรการแปลงค่าสีจาก RGB ไปเป็น CMY

ระบบสี CMYK จะเป็นระบบที่ถูกนำมาใช้ในงานพิมพ์ต่าง ๆ และการแสดงสีในรูปแบบจำลองกราฟฟิคสามมิติ โดยการใช้งานจะมีอยู่ด้วยกัน 2 แบบคือ CMYK และ RGB ที่มีการแบ่งแยกตามการใช้งาน ในการพิมพ์งานแต่ละครั้งจะต้องตั้งค่าการทำงานในโปรแกรมด้วยการใช้ค่าสีระบบ CMYK แต่ถ้าต้องการสีที่แสดงผลออกทางหน้าจอจะเลือกใช้ระบบ RGB การผสมทั้ง 4 สีนี้จะทำให้เกิดสีได้อีกหลากหลายสี ระบบสี CMYK จะถูกนำมาใช้ในการพิมพ์สื่อสิ่งพิมพ์ต่าง ๆ เช่น พิมพ์สมุด พิมพ์กล่องกระดาษ พิมพ์ถุงกระดาษ พิมพ์การ์ด พิมพ์หนังสือโหมดสี CMYK เป็นส่วนที่มีความสำคัญและจำเป็นอย่างมากในการออกแบบสำหรับงานพิมพ์ การจะทำไฟล์งานแต่ละครั้งจะต้องตั้งค่าไฟล์งานอาร์ตเวิร์คให้เป็นโหมดสี CMYK Color เสมอ CMYK เป็นระบบสีที่ใช้กับเครื่องพิมพ์ออกทางกระดาษหรือวัสดุผิวเรียบ ประกอบด้วยสีหลัก 4 สี คือสีฟ้า Cyan สีม่วงแดง Magenta สีเหลือง Yellow และสีดำ Black เมื่อผสมกันจะเกิดเป็นสีดำเนื่องจากหมึกพิมพ์มีความไม่บริสุทธิ์จึงเป็นการผสมสีแบบลบ Subtractive ซึ่งหลักการเกิดสีของระบบนี้ คือ หมึกสีหนึ่งจะดูดกลืนแสงจากสีหนึ่งแล้วสะท้อนกลับออกมาเป็นสีต่าง ๆ

HSI Color model

HSL Color เป็นระบบสีที่มนุษย์ เข้าใจได้ง่าย เพราะไม่ได้นิยามสีจากการผสมสี แต่เป็นระบบที่สร้างสีขึ้นมาจาก Hue (เนื้อสี), Saturation (ความอิ่ม) และ Lightness (ความสว่าง)



รูปภาพ 5 วงกลมแสดงค่าสี (Hue) ด้านขวาแนวตั้งเป็นแนวที่แสดงความเข้มของแสง

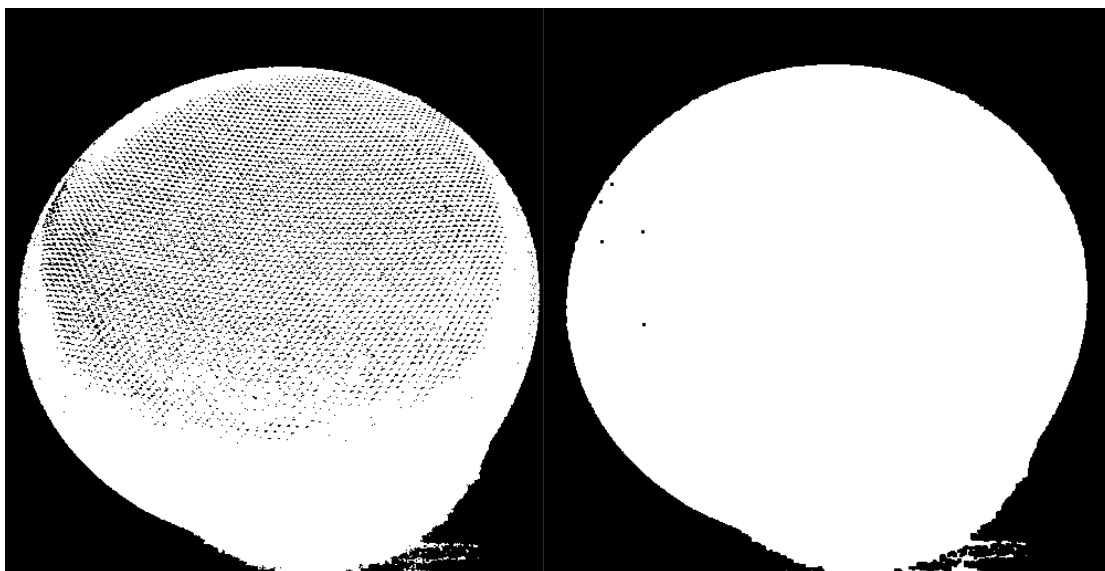
Morphological

การเปลี่ยนแปลงรูปร่างของวัตถุในภาพ เช่นการลด Noise หรือการลบ Hole ใน Foreground หรือ Background เปลี่ยนแปลงวัตถุในรูปภาพให้สมบูรณ์ขึ้น เช่นในภาพมีรูปวงกลมอยู่แต่ในวงกลมมีรูอยู่เล็กน้อย สามารถใช้เทคนิคนี้ทำให้รูในรูปวงกลมหายไป

Structuring element เปรียบเสมือนตัวกรองที่มีไว้สำหรับเปรียบเทียบระหว่าง pixel นั้นกับรอบๆขนาดเท่ากับ Structuring element ว่าเหมือนกันหรือไม่

Erosion and Dilation, erosion เปรียบเสมือนการลดขนาดของวัตถุ ทุกตำแหน่งของวัตถุจะถูกเปรียบเทียบโดย Structuring element ถ้าสอดคล้องกันก็จะทำการลบตำแหน่งนั้นออกไป ส่วน dilation ก็ตรงข้ามกับ erosion ทำให้วัตถุดูเหมือนขนาดใหญ่ขึ้นและสามารถลบรูตรงกลางวัตถุได้ด้วย

Opening and Closing จะเป็นการนำ erosion และ dilation มาใช้เช่น Opening จะใช้ erosion ก่อน 1 ครั้งแล้วตามด้วย closing จะทำให้ภาพตรงขบมีความสมบูรณ์ขึ้นตามรูปทรงของ Structuring element ส่วน closing ก็จะใช้ dilation ก่อน 1 ครั้งแล้วตามด้วย erosion จะทำให้รูตรงวัตถุหายไป



รูปภาพ 6 ตัวอย่างการใช้ Morphological

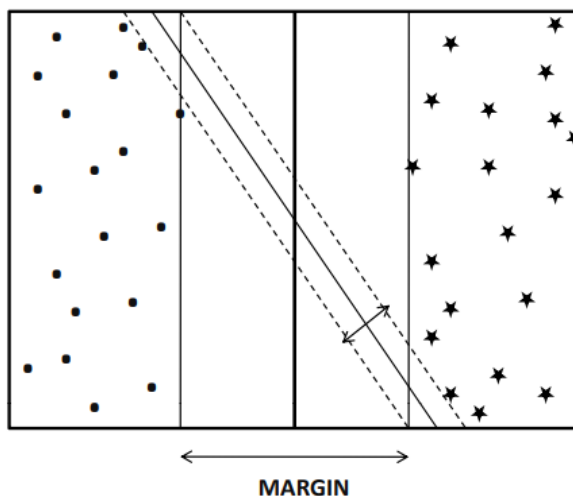
Line, and Edge Detection

Data Classification

เป็นกระบวนการสร้างโมเดลจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ เพื่อแสดงให้เห็นความแตกต่างระหว่าง class หรือ กลุ่มของข้อมูลได้ และเพื่อทำนายว่าข้อมูลนี้ ควรจัดอยู่ใน classใด ซึ่งโมเดลที่ใช้จำแนกข้อมูลออกเป็นกลุ่มตามที่ได้กำหนดไว้ จะขึ้นอยู่กับการวิเคราะห์เซตของข้อมูลทดลอง (Training data) โดยนำ Training data มาสอนให้ระบบเรียนรู้ว่ามีข้อมูลใดอยู่ใน class เดียวกันบ้าง ผลคือ Classifier model และส่วนข้อมูลสำหรับทดสอบ Testing Data ซึ่งเป็นกลุ่มที่มีข้อมูลจริงที่ใช้ทดสอบนี้จะถูกนำมาเปรียบเทียบกับกลุ่มที่หามาได้จากโมเดลเพื่อทดสอบความถูกต้องทั่วไปจะใช้อัตราส่วน 80 ต่อ 20 แล้วจะทำการปรับปรุงโมเดลให้ได้ผลที่อยู่ในระดับที่พอใจ เมื่อนำไปใช้จะต้องป้อนข้อมูลใหม่ผ่านโมเดล โดยโมเดลจะสามารถทำนายกลุ่มของข้อมูลนี้ได้ [2]

SVM (Support Vector Machine)

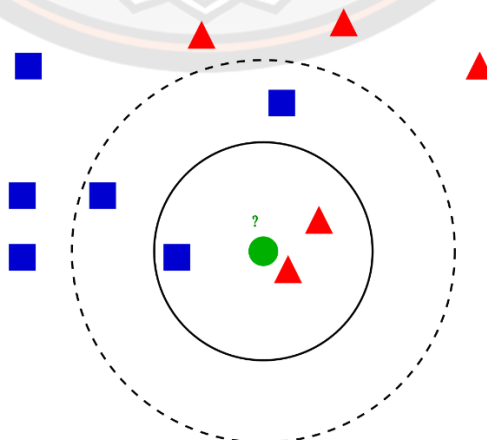
Support Vector Machine: (SVM) เป็นตัวจำแนกเชิงเส้น (Linear Classifier) แบบ 2 คลาส นิยมใช้กันมากเพราะมีประสิทธิภาพในการจำแนกข้อมูลที่มีจำนวนมิติได้เยอะ สามารถจำแนกข้อมูลที่มีความคลุมเครือได้อย่างมีประสิทธิภาพ หลักการของ SVM คือการหาเส้นตรงที่มี Margin ที่โตที่สุด (Maximum Margin) สามารถแบ่งข้อมูลออกเป็น 2 คลาส ดังตัวอย่างในภาพที่ 7 เป็นข้อมูลขนาด 2 มิติ โดนถูกจำแนกออกเป็น 2 คลาส ได้แก่ + (O) และคลาส - (Δ) โดยเส้นตรงที่ใช้แบ่งข้อมูลมี Margin เท่ากับ $M=2w$



รูปภาพ 7 Hard support vector machines

KNN (k-nearest neighbors)

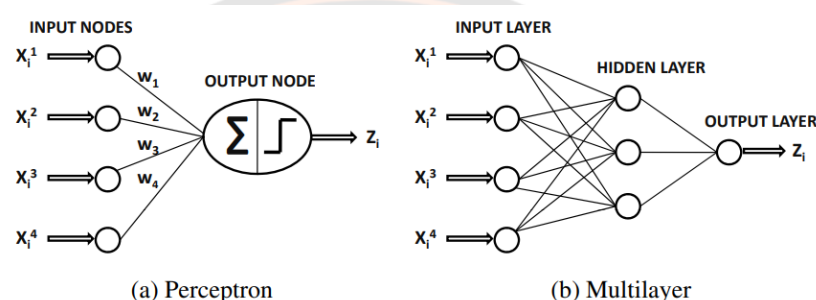
เป็นวิธีที่ใช้ในการจัดแบ่งคลาส โดยเทคนิคนี้จะตัดสินใจว่าใกล้เคียงกับคลาสใดโดยหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K แล้วในจำนวนนั้นมีคลาสใดที่มากที่สุดดังรูปภาพที่ 8 มีค่า K เท่ากับ 3 จำนวนของคลาสสามเหลี่ยมคือ 2 และคลาสสี่เหลี่ยมคือ 1 จึงตัดสินใจได้ว่าเป็นคลาสสามเหลี่ยม



รูปภาพ 8 ตัวอย่าง k-nn

Neural Network

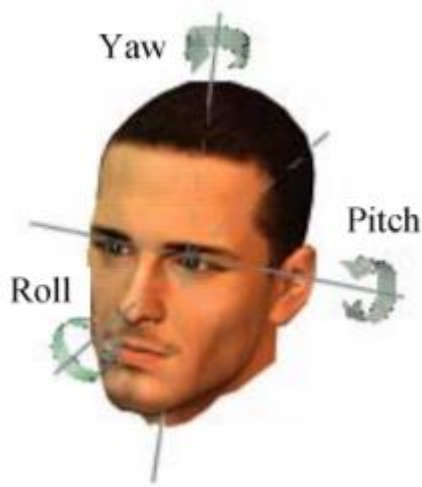
เป็นแนวคิดที่ออกแบบระบบโครงข่ายคอมพิวเตอร์ ให้เลียนแบบการทำงานของสมองมนุษย์ สถาปัตยกรรมพื้นฐานที่สุดของโครงข่ายประสาทเทียมคือ Perceptron ซึ่งประกอบด้วยชุดของ Input Layer ชั้นที่เป็นส่วนของการรับข้อมูล, Output Layer ชั้นที่เป็นส่วนต่อสิ่งที่ประมวลผลออกมา ซึ่งในส่วน hidden layer ของ perceptron นั้นจะมีแค่ layer เดียวเป็นส่วนไว้สำหรับคำนวณผลลัพธ์จาก input แล้วส่งไปให้ output เพื่อที่จะทำให้ Neural Network สามารถคิดและประมวลผลซับซ้อนได้เหมือนสมองมนุษย์ ชั้นที่เป็น Hidden Layer จึงต้องมีหลายๆชั้น ให้มันส่งข้อมูลประมวลผลต่อกันไป ทำให้มันสามารถคำนวณอะไรที่ซับซ้อนได้เยอะขึ้น



รูปภาพ 9 Single and multilayer neural networks

มุมออยเลอร์ (Euler's Angle)

มุมออยเลอร์ (Euler angles) คือมุมสมมุติใช้ กำหนดการเปลี่ยนจากแกนอ้างอิงใดๆเช่นแกนอ้างอิง navigation frame ไปเป็นแกนอ้างอิง body frame ประเด็นสำคัญเกี่ยวกับ Euler angles คือแกนอ้างอิง body frame จะเป็นผลลัพธ์ของการหมุนในลำดับดังนี้เสมอ yaw ไป pitch ไป roll ลำดับของการหมุนแกนมี ความสำคัญกำหนดให้ yaw หมุนรอบแกน up – z axis, pitch หมุนรอบแกน east – x axis และ roll หมุนรอบแกน north – y axis



รูปภาพ 10 มุมออยเลอร์บนใบหน้า

เครื่องมือและซอฟต์แวร์ที่ใช้งาน

Python

ภาษาไพธอน (Python) คือ ภาษาโปรแกรมคอมพิวเตอร์ระดับสูง ซึ่งถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เขียนให้เป็นภาษาเครื่องไพธอน (Python) มีการทำงานแบบ Interpreter คือ เป็นการแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการ นอกจากนี้ภาษาไพธอน (Python) ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม คือ สามารถรันภาษาไพธอน ได้ทั้งบนระบบ Unix, Linux , WindowsNT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD โดยภาษาไพธอน (Python) นี้เป็น Open Source โค้ดของไพธอน ถูกสร้างขึ้นมาจากภาษาซี ทำให้สามารถที่จะนำ Python มาพัฒนาโปรแกรมได้โดยไม่ต้องเสียค่าใช้จ่าย [3]

Open CV

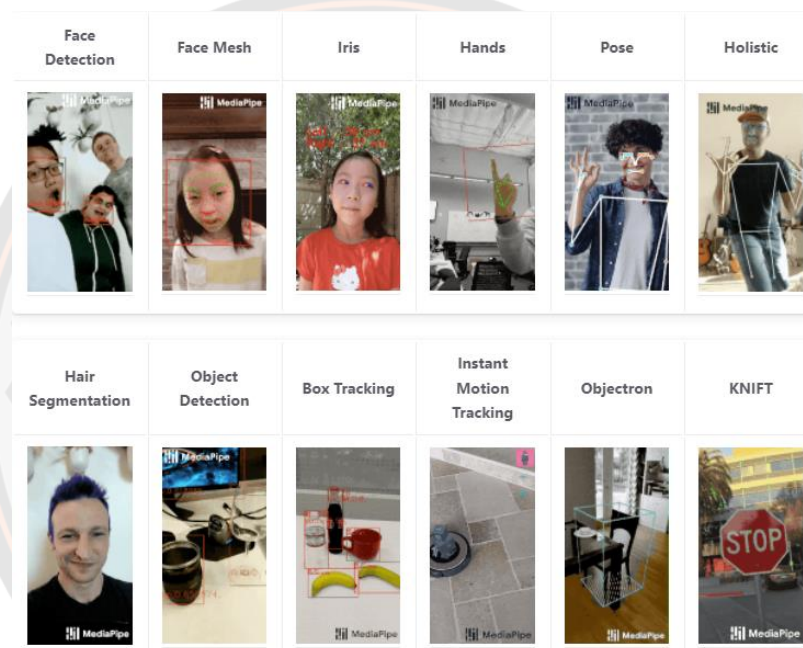
OpenCV คือ library ที่ทำขึ้นมาเพื่อแก้ปัญหาด้าน computer vision สร้างขึ้นมาตั้งแต่ปี 1999 โดย Intel สามารถทำงานบนระบบปฏิบัติการได้หลากหลาย เช่น Windows, Linux และ MacOS ทั้งยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe

Dlib

Dlib คือเครื่องมือสำหรับภาษา C++ ไว้สำหรับแก้ปัญหาที่ซับซ้อน มีเครื่องมือสำหรับ machine learning และ image processing สามารถใช้งานร่วมกับภาษา python ได้ และยังเป็นเปิดให้ใช้งานฟรี [4]

MediaPipe

เป็นเครื่องมือฟรีที่ถูกพัฒนาโดย Google สามารถใช้งานได้หลายแพลตฟอร์ม มีเครื่องมือหลายชนิดที่ใช้ machine learning ในการแก้ปัญหา ไม่ว่าจะเป็น Face Detection, Face Mesh, Hand Tracking, Human Pose Detection and Tracking สามารถทำงานได้รวดเร็ว[5]



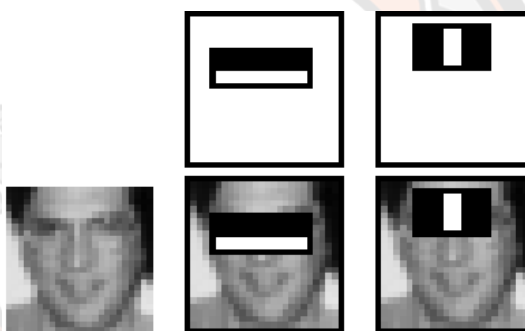
รูปภาพ 11 ML solutions in MediaPipe

งานวิจัยที่เกี่ยวข้อง

Rapid Object Detection using a Boosted Cascade of Simple Features [6]

เป็น Object Detection Algorithm ใช้ในการตรวจจับวัตถุแบบ real-time หลักการทำงานของ Haar Cascades วิธีการคือแบ่งพื้นที่ของภาพด้วยรูปแบบสำเร็จต่างๆ ในแต่ละรูปแบบก็จะมีสีขาวและสีดำ นำพื้นที่ของสีขาวและสีดำมาหาความต่างจากนั้นนำผลลัพธ์ไปเทียบกับค่า threshold เพื่อกำหนดว่าเป็นรูปภาพของใบหน้าหรือเป็นภาพพื้นหลัง นำข้อมูลที่ได้ไปผ่านขั้นตอน Adaptive Boost จะทำให้การแยกประเภทมีประสิทธิภาพที่ดีขึ้น และนำไปเข้ากระบวนการ Cascaded Classifier แล้วนำมาเก็บข้อมูลไว้ในรูปของไฟล์ xml เป็นงานวิจัยเกี่ยวกับการ detection แรกๆที่นำมาใช้สำหรับการตรวจจับใบหน้า

วิธีการนำมาใช้งานง่าย, จำนวนของข้อมูลน้อยก็สามารถทำงานได้ดี แต่ในเรื่องความแม่นยำในปัจจุบันถือว่ามีความแม่นยำน้อยกว่าวิธีการอื่นพอสมควร

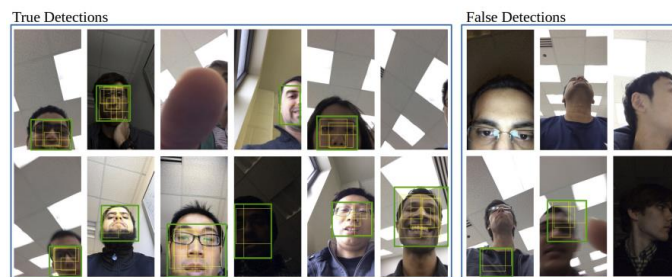


รูปภาพ 12 2 features ที่ใช้สำหรับ training face

Partial Face Detection for Continuous Authentication [7]

ตรวจจับใบหน้าที่ถูกบดบังบางส่วน เพื่อใช้กับกล้องหน้าบนสมาร์ทโฟน โดยการแบ่งส่วนของใบหน้าออกเป็นหลายๆส่วนก่อนเข้าสู่ขั้นตอน Machine Learning โดย SVM ผลที่ได้เหมาะกับการทำงานบนสมาร์ทโฟน คือสามารถทำงานได้รวดเร็ว สามารถนำไปใช้กับระบบความปลอดภัยบนมือถือได้ เช่นระบบ authentication

เน้นภาพที่มาจากกล้องสมาร์ทโฟนทำให้เหมาะกับการใช้งานบนสมาร์ทโฟน แต่ภาพที่ได้ส่วนมากเป็นภาพที่กล้องอยู่ต่ำกว่าหน้าแทบทุกภาพ

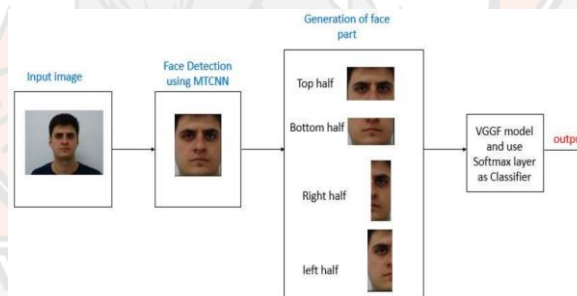


รูปภาพ 13 ผลของการใช้ FACIAL SEGMENT BASED FACE DETECTOR

Face Recognition from Partial Face Data [8]

ทำการรู้จำใบหน้าโดยที่มีสิ่งที่ปกปิดบนใบหน้าไม่ว่าจะเป็น แมส, แว่นกันแดด หรือหมวกใบใหญ่ที่สามารถปิดใบหน้าบางส่วนได้ งานวิจัยนี้ทำการตรวจจับใบหน้าที่ก่อน แล้วทำการแบ่งส่วนของใบหน้าเป็น 4 ส่วน คือ Top half, Bottom half, right half, left half นำไปเข้าสู่ขั้นตอนการรู้จำใบหน้าโดยใช้ VGGF Model ผลที่ได้แบ่งตามส่วนของใบหน้า คือ Full Face 98%, Top half 90%, Bottom half 83%, Right half 77%, Left half 78%

เป็นเทคนิคที่ง่ายไม่ซับซ้อน แต่ก็จะได้เฉพาะส่วนที่แบ่งภาพออกมาเท่านั้น เนื่องจากแบ่งภาพเป็นหลายส่วนการทำงานก็จะเยอะขึ้นตามไปด้วย



รูปภาพ 14 ภาพรวมการทำงานของ Face Recognition from Partial Face Data

Face Alignment Across Large Poses: A 3D Solution [9]

การจัดตำแหน่งใบหน้าให้พอดีกับรูปหน้าและการหาตำแหน่งบนใบหน้า เป็นหัวข้อสำคัญใน computer vision อัลกอริทึมส่วนใหญ่จะใช้กับใบหน้าที่หันเพียงเล็กน้อย (น้อยกว่า 45 องศา) และจะไม่สามารถนำมาใช้กับใบหน้าที่หันด้านข้างมากๆได้ ในงานวิจัยนี้มีความท้าทายอยู่ 3 อย่างคือ อย่างแรก model face landmark ทั่วไปจะใช้กับใบหน้าที่ตำแหน่ง landmark เห็นครบทุกตำแหน่ง อย่างที่สองใบหน้าที่มีความหลายหลายของการหันหน้า ตั้งแต่การหันหน้าตรงไปจนถึงการหันด้านข้างเยอะเกินไป อย่างที่ 3 คือ การให้ชื่อตำแหน่งของ landmark ในภาพที่หันด้านข้างเยอะไปเป็นสิ่งที่ยาก เพราะเนื่องจากจะเกิดจุด landmark ที่มองไม่เห็นเกิดขึ้น ในงานวิจัยนี้จึงได้เสนอการจัดตำแหน่ง

บนใบหน้าใหม่โดยใช้ชื่อว่า 3D Dense Face Alignment (3DDFA) โดยใช้ 3D face model แล้วใช้ convolutional neural network (CNN). นอกจากนี้ยังทำการ train ข้อมูลขึ้นมาใหม่เพื่อใช้ในการแก้ปัญหาอย่างอื่นอีกด้วย

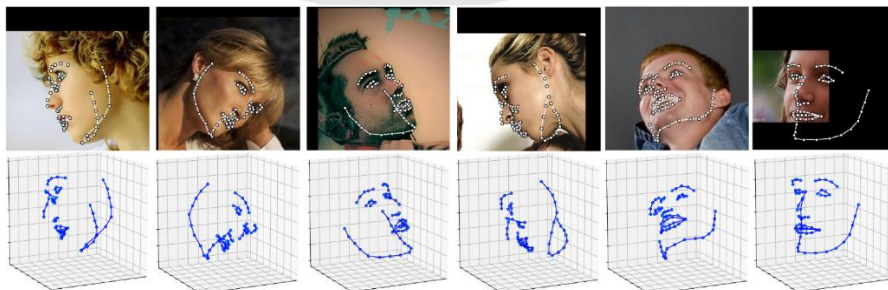


รูปภาพ 15 ผลลัพธ์ของ 3DDFA

How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks) [10]

งานวิจัยนี้ดูว่าปัญหาของ 2D & 3D Face Alignment นั้นอีกไกลแค่ไหนถึงจะแก้ปัญหานี้จะถึงจุดอิ่มตัว งานวิจัยนี้ใช้ dataset จำนวนมาก และทำการสร้าง model ใหม่โดยใช้ชื่อว่า LS3D-W ซึ่งนำ neural network มาใช้ ศึกษา face alignment อื่นๆและพิจารณาที่มีผลกระทบต่อประสิทธิภาพของ face alignment นำไปเพิ่มลงใน network ใหม่อีกที จึงทำให้เห็นว่าประสิทธิภาพที่ได้นั้นแม่นยำขึ้นอย่างน่าทึ่ง แต่ด้วยเทคโนโลยีปัจจุบันนั้นการ train ข้อมูลจำนวนมากใช้พื้นที่ความจำจำนวนมากทำให้ยังไม่พอสำหรับข้อมูลเหล่านี้

ผลลัพธ์ที่ได้อยู่ในระดับที่ดีมาก และได้ model 3D ออกมาด้วย แต่การประมวลผลยังนานอยู่เพราะด้วยตัวข้อมูลจำนวนมาก



รูปภาพ 16 ผลลัพธ์ที่ได้จากการใช้ 2D & 3D Face Alignment

GlassGesture: Exploring Head Gesture Interface of Smart Glasses [11]

งานวิจัยนี้สำหรับ smart glasses ซึ่งคล้ายๆกับแว่นตาสวมอยู่บนใบหน้าใช้อุปกรณ์ที่ชื่อ Google Glass มุ่งเน้น 2 อย่างคือจดจำท่าทางเช่นหันซ้ายหันขวาและระบบตรวจสอบผู้ใช้งานโดยใช้วิธีการดูจากการเคลื่อนที่ของส่วนหัว ใช้ข้อมูลจาก sensor gyroscope มาทำการหา feature ข้อมูลนั้นแบ่งเป็น 3 ระดับความยากคือ ตอนนั้น ตอนเดิน และตอนวิ่ง ส่วนท่าทางเช่น left3 right3 shake ผลที่ได้คือระบบจดจำท่าทางได้ความแม่นยำที่ 96% ระบบตรวจสอบผู้ใช้งานโดยใช้วิธีการดูจากการเคลื่อนที่ของส่วนหัวได้ 92%

ใช้เซนเซอร์ตรวจจับทำให้ได้ความแม่นยำที่มาก แต่ก็ต้องมีพื้นที่ให้อุปกรณ์เช่นติดไว้ตรงหน้าแว่นตา

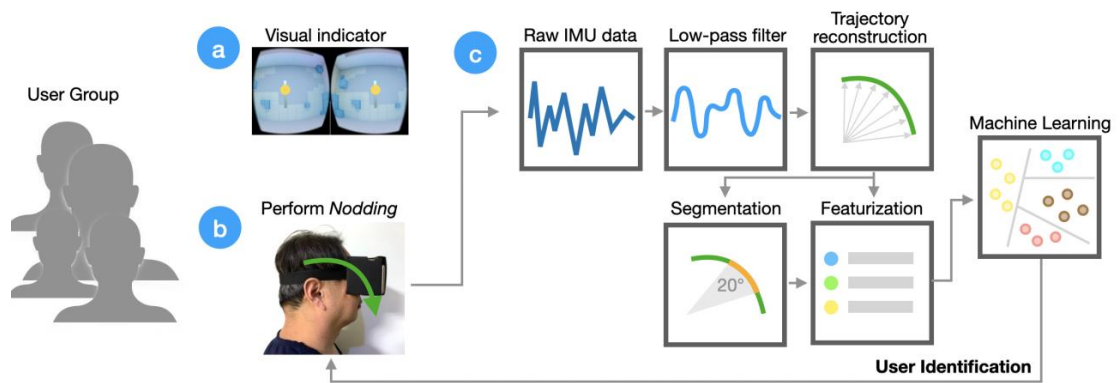


รูปภาพ 17 google glass

Nod to Auth: Fluent AR/VR Authentication with User Head-Neck Modeling [12]

เครื่องมือสำหรับ VR นั้นจำเป็นจะต้องสวมใส่ไว้อยู่บนหัวและครอบตาให้ตาเราองในโลกของ VR เท่านั้นจำทำให้ไม่สามารถมองเห็นข้างนอกนั้น ส่วนวิธีการตรวจสอบผู้ใช้งาน (Authentication) ถ้ายังใช้ระบบพาสเวิร์ดอยู่ก็ถือว่าเป็นอุปสรรคเพราะการพิมพ์บน VR นั้นยาก งานวิจัยนี้เลยใช้เครื่องมืออยู่ติดมากับ VR อยู่แล้วนั่นก็คือมือถือซึ่งมี acceleration และ gyroscope อยู่แล้ว จึงสามารถนำข้อมูลตรงนี้มาใช้งานได้ นำข้อมูลที่ได้มาหา feature จากนั้นนำเข้าสู่ machine learning ผลที่ได้คือความแม่นยำอยู่ที่ 97.1%

ใช้เซนเซอร์ที่มีอยู่แล้วใน VR เช่น acceleration และ gyroscope ทำให้ได้ข้อมูลที่แม่นยำ แต่ข้อจำกัดก็คืออยู่ที่สมาร์ทโฟน



รูปภาพ 18 รูปภาพการทำงานของ Nod to Auth user identification



บทที่ 3

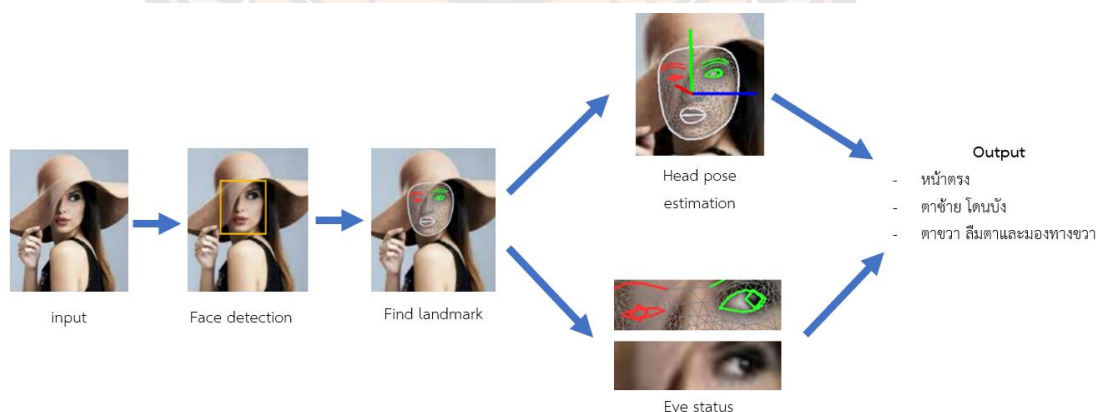
วิธีดำเนินการวิจัย

งานวิจัยนี้เป็นหนึ่งในการประยุกต์ใช้เทคโนโลยีการรู้จำและช่วยให้สามารถนำไปใช้งานได้หลากหลายขึ้น ซึ่งงานวิจัยนี้สามารถตรวจจับและระบุทิศทางและสถานะของใบหน้าและดวงตาที่ถูกปิดบางส่วนได้อย่างแม่นยำ ในขั้นตอนแรกคือการเก็บข้อมูลเก็บเป็นไฟล์วิดีโอในท่าทางต่างๆ หลังจากนั้นกระบวนการทำงานใช้เทคโนโลยีที่มีในปัจจุบันมาช่วยเช่น การตรวจจับใบหน้าและการหาจุดสำคัญของใบหน้า เนื่องจากรู้จุดสำคัญของใบหน้าแล้วขั้นตอนต่อไปคือการหาทิศทางของใบหน้าและการหาสถานะของดวงตาต่อไป

3.1 ภาพการทำงานโดยรวมของของโปรแกรมที่พัฒนาขึ้น

การทำงานของโปรแกรมตามรูปภาพที่ 19 สามารถแบ่งรายละเอียดออกเป็น 5 ขั้นตอน ดังนี้

- 3.1.1 input รับภาพที่มีใบหน้าคนเข้ามา
- 3.1.2 ทำการตรวจจับใบหน้าเพื่อหาตำแหน่งหน้าที่อยู่ในรูป
- 3.1.3 ค้นหาจุดสำคัญของใบหน้า
- 3.1.4 ทำการประมวลผล
 - ทิศทางของใบหน้าว่าหันไปทางไหน
 - สถานะดวงตา
- 3.1.5 output รวบรวมผลออกมาเป็นผลลัพธ์ของโปรแกรม



รูปภาพ 19 แผนภาพแสดงการทำงาน

3.2 วิธีการเก็บข้อมูล

ใช้ทั้ง dataset ที่มีอยู่แล้วและ data ที่เก็บเพิ่มขึ้นเพิ่มเติม

วิดีโอข้อมูลที่เก็บเพิ่มและจะใช้เป็นการทดสอบหลักใช้อาสาสมัครจำนวน 61 คน ผู้ชาย 47 คน หญิง 14 คน ช่วงอายุ 24-47 ปี เฉลี่ย 25 ปี สวมแว่นตา 19 คน ส่วนใหญ่เป็นคนไทย และมี

ต่างชาติ 1 คนจากภูฏาน เก็บเป็นไฟล์วิดีโอความละเอียด 720x1280 pixel เห็นใบหน้าชัดเจนและ
กล้องห่างจากใบหน้าอย่างน้อย 0.5-1 เมตร ในวิดีโอมีการหันหน้าตามทิศทางต่างๆ เช่น หันซ้าย หัน
ขวา ก้มหน้า เงยหน้า

3.3 กระบวนการทำงาน

3.3.1 เลือกวิธีการตรวจจับใบหน้า

การตรวจจับใบหน้าที่มีการพัฒนาอย่างต่อเนื่อง มีการใช้เทคนิคต่างๆมาช่วยเช่น Haar
Cascades, Histogram of Oriented Gradients, Deep Learning Based มาช่วยให้ได้ผลลัพธ์การ
ตรวจจับใบหน้าได้ดีขึ้น ตัวอย่างอัลกอริทึมในการตรวจจับใบหน้า

Haar Cascades (2001)

นำเสนอโดย Paul Viola และ Michael Jones ในปี 2001 เป็นเทคนิคของ face detection
แรกๆที่ใช้กันอย่างแพร่หลาย ผลการตรวจจับใบหน้าถือว่าแม่นยำในระดับหนึ่ง แต่จะมีข้อเสียเห็นพื้น
หลังเป็นใบหน้าซึ่งเห็นได้เยอะเมื่อมีการใช้งานกับข้อมูลที่เป็นวิดีโอ

DLib HOG (2005)

ใช้เทคนิค Histogram of Oriented Gradients ในการหา feature ร่วมกับใช้ linear
classifier ผลลัพธ์ที่ได้เร็วกว่าและแม่นยำกว่าใช้ Haar Cascades ยังมีปัญหากับใบหน้าแปลกๆจะ
ตรวจจับใบหน้าไม่เจอ แต่ก็สามารถเพิ่มประสิทธิภาพโดยใช้ cnn (Convolutional Neural
Network) เข้ามาช่วย เนื่องจากเป็น neural network ก็จะทำให้การประมวลผลนานขึ้น

Deep Learning Based

ช่วงหลังปี 2011 เป็นต้นมาส่วนใหญ่เป็นการใช้ deep learning ผลลัพธ์ที่ได้ก็ดีขึ้นมาก
ขึ้นอยู่กับวิธีการ train data และ model ที่ใช้ ตัวอย่างที่ใช้กับ face detection เช่น SSD (Single
Shot Multibox Detector) ปี 2015, MTCNN (Multi-Task Cascaded Convolutional Neural
Network) ปี 2016, RetinaFace ปี 2019 เนื่องจากเป็น deep learning based จะทำงานกับ gpu
ได้ดีถ้าใช้ cpu จะช้าอย่างเห็นได้ชัด

MediaPipe (2019)

ถูกพัฒนาขึ้นโดย google ออกแบบมาให้กับบนมือถือ ใช้ model ที่ชื่อ MobileNet ทำให้
การประมวลผลไวขึ้น แต่ก็ยังมีข้อจำกัดถ้าเจอหลายคนในรูปเดียว สามารถทำงานได้หลาย platform
ทั้งในรูปแบบเว็บไซต์, android, ios, และบนคอมพิวเตอร์

YuNet (2021)

เป็น library ที่ติดมากับ OpenCV เป็น Convolutional Neural Network (CNN)-based
มี model ที่เล็กมาก มีการประมวลผลที่ไวและยังได้ประสิทธิภาพที่ดี

อัลกอริทึมที่เลือกมาใช้ทดสอบจะต้องมีคุณสมบัติดังต่อไปนี้

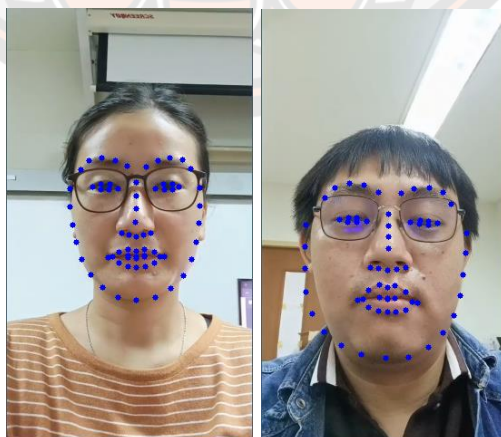
1. สามารถทำงานบนภาษา python ได้
2. สามารถทำงานบนสมาร์ตโฟน android ได้และสามารถแก้ไข code เพื่อทำการทดสอบบนสมาร์โฟนได้
3. สามารถให้แสดงจุดสำคัญบนใบหน้าได้

อัลกอริทึมที่เลือกมามีอยู่ 2 ตัวคือ dlib ที่ใช้เทคนิค HOG และ mediapipe (facemesh) ******* ที่ถูกพัฒนาให้สามารถใช้งานกับมือถือได้ดี ทั้งสองอัลกอริทึมก็เป็นที่ยอมรับและนิยมใช้กันมาก โดยเฉพาะ mediapipe ที่เพิ่งจะมาเมื่อปี 2019 จากการทดสอบจุดบนใบหน้ามีค่าความผิดพลาดอยู่ที่ประมาณ 4% สามารถประยุกต์ใช้บนแพลตฟอร์มอื่นๆได้เช่น บนเว็บ หรือบนสมาร์ตโฟนได้ทั้ง Android และ IOS ในส่วนการประมวลผลใช้ทรัพยากรน้อยถึงแม้ว่าจะอยู่ในหมวดของ deep learning base ก็ตาม แต่ก็มีข้อเสียตรงที่เจอหลายใบหน้าในรูปภาพเดียว mediapipe จะทำงานไม่ได้ ซึ่งในงานวิจัยนี้มีข้อจำกัดให้ในรูปที่ใช้ทดสอบมีแค่หน้าเดียวอยู่แล้ว

3.3.2 หาจุดสำคัญบนใบหน้า

3.3.2.1 หาจุดสำคัญด้วย Dlib

เมื่อทำการตรวจหาใบหน้าด้วย Dlib จะได้ผลลัพธ์ออกมาเป็น x, y, w, h เป็นตำแหน่งและขนาดของใบหน้าที่ตรวจเจอ แล้วนำไปหาจุดสำคัญบนใบหน้าด้วยฟังก์ชันของ dlib ที่ชื่อ shape_predictor และใช้ model ชื่อ shape_predictor_68_face_landmarks.dat หลังจากประมวลผลเสร็จแล้วจะได้จุดบนใบหน้าทั้งหมด 68 จุดดังในรูปภาพ 20



รูปภาพ 20 จุดสำคัญบนใบหน้าที่ผ่าน model ของ dlib

3.2.2.2 หาจุดสำคัญด้วย mediapipe

Mediapipe จะใช้วิธีการเหมือนกันกับ Dlib คือหาดำแหน่งของใบหน้าก่อนแล้วค่อยหาจุดสำคัญของใบหน้า ทั้งการตรวจจับใบหน้าและหาจุดสำคัญจะใช้ model ที่ประมวลผลผ่าน TensorFlow Lite ทั้งหมด จุดสำคัญที่ได้จะมีทั้งหมด 468 จุด และแต่ละจุดมีทั้งค่า x, y และ z นั่นคือได้ผลลัพธ์ในรูปแบบของ 3D ด้วย แสดงผลลัพธ์ตามในรูปภาพที่ 21



รูปภาพ 21 จุดสำคัญบนใบหน้าโดยใช้ mediapipe facemesh

3.3.3 ทิศทางการหมุนของใบหน้า

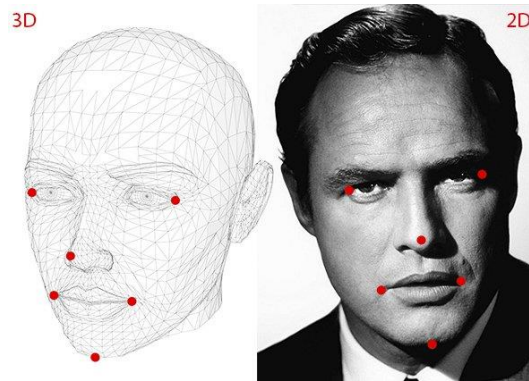
- การหาทิศทางของใบหน้าโดยวิธีการแปลงจาก 2D เป็น 3D

เป็นวิธีการโดยใช้หลักการจาก “การพัฒนาแอปพลิเคชันควบคุมรถเข็นไฟฟ้าด้วยทิศทางของใบหน้า”[13] ซึ่งเป็นการจำลองโมเดลสามมิติของใบหน้าขึ้นมาแล้วนำมาเทียบกับจุดบนใบหน้า วิธีการทำงานมีดังนี้

จากที่ได้จุดสำคัญ (landmark) บนใบหน้ามา แต่ละจุดที่ได้จะมีค่า x, y กำกับไว้อยู่ในรูปแบบ 2D จากนั้นทำการแปลงให้อยู่ในรูปแบบของ 3D ดังตัวอย่างในรูปภาพที่ 22 โดยจะต้องมี model ของใบหน้าไว้สำหรับเปรียบเทียบ

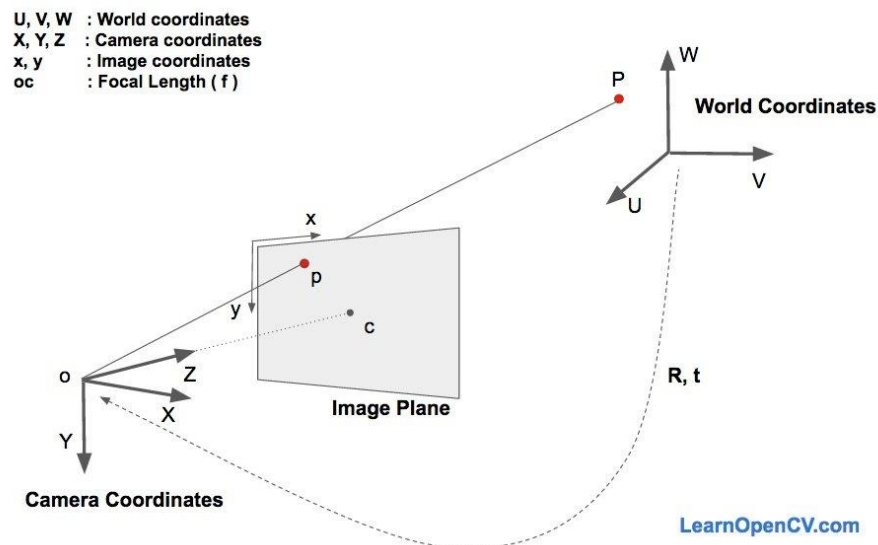
model ที่สร้างขึ้นมาใช้มีจุดดังนี้

1. ปลายจมูก : (0.0, 0.0, -5.0)
2. คาง : (0.0, -330.0, -65.0)
3. มุมซ้ายสุดของตาซ้าย : (-225.0, 170.0, -135.0)
4. มุมขวาสุดของตาขวา: (225.0, 170.0, -135.0)
5. มุมซ้ายสุดของปาก : (-150.0, -150.0, -125.0)
6. มุมขวาสุดของปาก : (150.0, -150.0, -125.0)



รูปภาพ 22 ภาพแสดงจุดบน 2D และ 3D

เมื่อรู้ตำแหน่ง landmark ของ model และ ใบหน้าจากรูปแล้ว ต่อไปว่าใบหน้ามีการหมุนเป็นยังไง ใน รูปภาพที่ 23 จุดที่อยู่ world coordinates รู้ตำแหน่งอยู่แล้วเพราะเป็นตำแหน่งจริง และ ถ้ารู้ rotation และ translation จะสามารถคำนวณหาจุดบน camera coordinates ได้



รูปภาพ 23 แสดงตำแหน่งต่างๆบน World Coordinates และ Camera Coordinates

ถ้าสมมุติว่ารู้ตำแหน่งของ (U, V, W) 3D จาก world coordinates และถ้ารู้ rotation R (3×3 matrix) และ translation t (3×1 matrix) จะสามารถหาตำแหน่ง (X, Y, Z) ของ camera coordinates ได้ดังสูตรต่อไปนี้

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \mathbf{R} \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \mathbf{t} \\ \Rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= [\mathbf{R} \mid \mathbf{t}] \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \end{aligned} \quad \text{เปลี่ยนรูปได้เป็น} \quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}$$

Direct Linear Transform

ถ้ารู้จุดของ 3D model (U, V, W) แต่ยังไม่รู้ (X, Y, Z) รู้แค่ที่เป็น 2D (x, y) ในกรณีที่ไม่มี radial distortion ตำแหน่งของ (x, y) จาก image coordinates จะเป็นดังนี้

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

ให้ f_x และ f_y เป็น focal length และ c_x c_y เป็นตำแหน่งตรงกลาง และ s เป็น scale factor เป็นค่าที่ไม่สามารถรู้ได้เพราะรูปทุกรูปไม่สามารถบอกความลึกมาได้ สุดท้ายจะได้สูตรดังนี้

$$s \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}$$

ใช้ opencv ช่วยในการคำนวณจากฟังก์ชัน solvePnP เพื่อหา rotation vector ฟังก์ชัน solvePnP สามารถคำนวณหา rotation และ translation ได้ใช้สูตรจากข้างบนที่กล่าวมา ฟังก์ชัน solvePnP มี syntax ดังนี้

`cv2.solvePnP(objectPoints, imagePoints, cameraMatrix, distCoeffs[, rvec[, tvec[, useExtrinsicGuess[, flags]]]) → retval, rvec, tvec`

input ที่ solvePnP ต้องการคือ objectPoints เป็นเหมือนจุดบน model 3D ที่สร้างขึ้นมา และ imagePoints เป็นจุดบนรูปภาพ ส่วน cameraMatrix กับ distCoeffs จะใส่ค่าอัตโนมัติตามรูปที่ได้ไปก่อน

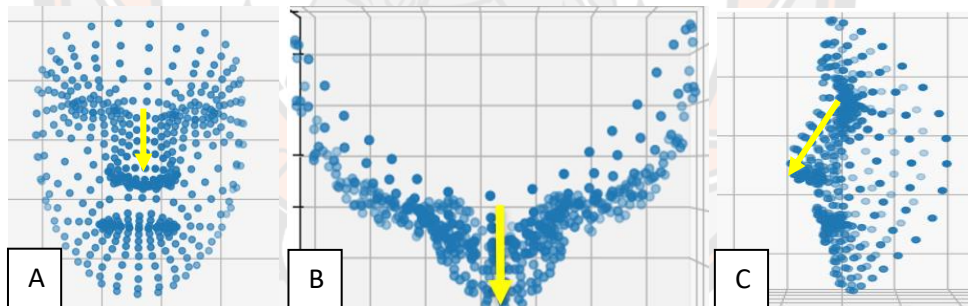
ผลลัพธ์ที่ได้คือ rotation vector และ translation vector ค่าที่จะนำไปใช้ต่อคือ rotation vector ทำการแปลงให้อยู่ในรูปแบบของออยเลอร์ มุมออยเลอร์เป็นค่าที่แสดงถึงการหมุนสามมิติและสามมุมที่แยกกันที่ประกอบกรการหมุน ผลลัพธ์ของการหมุนในลำดับดังนี้เสมอ yaw ไป pitch ไป roll

เมื่อรู้องศาการหมุนของใบหน้าทั้ง 3 ค่าสามารถทำผลที่ได้มาแปลงเป็นอินพุตแล้วไปให้รถเซ็นไฟฟ้าต่อได้ โดรนการหมุนของ yaw จะแปลงเป็นการหมุนทางซ้ายขวา, pitch แปลงเป็นการก้มหน้าและเงยหน้า ผลจากการรู้มุมก็จะตีความออกมาเป็น หันซ้าย หันขวา ก้มหน้า เงยหน้า และหน้าตรง

- การหาทิศทางของใบหน้าโดยเมื่อจุดสำคัญ (landmark) อยู่ในรูปแบบ 3D

การหาจุดสำคัญบนใบหน้าปกติแล้วแต่จุดมีแค่ x, y แต่ในกรณีของ facemesh mediapipe จุดสำคัญที่ได้จะมี x, y, z ซึ่งอยู่ในรูปแบบ 3D อยู่แล้ว วิธีการหาทิศทางของใบหน้าจะกำหนด vector ไว้หนึ่งตัว และทำการหมุนแกนต่างๆ แล้วหามุมออกมา วิธีการโดยละเอียดมีดังนี้

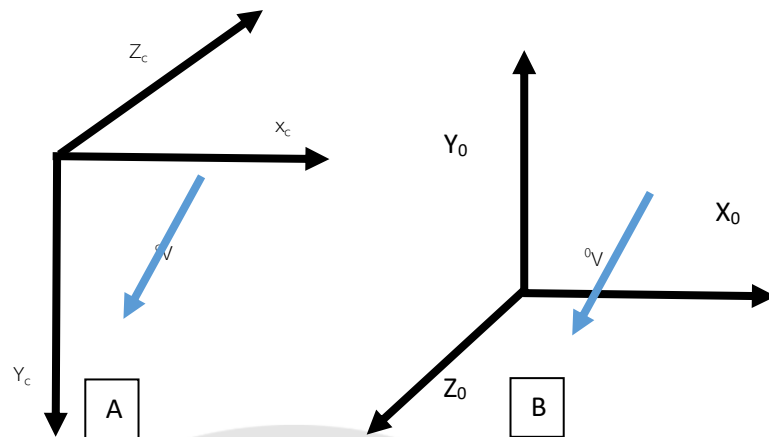
ทำการสร้างเวกเตอร์ที่ชี้ตั้งแต่ตรงกลางระหว่างตาทั้งสองข้างไปถึงปลายจมูกและลองเปลี่ยนมุมมองของภาพเป็นมุมมองด้านบนและมุมมองด้านข้างจะเห็นว่าเวกเตอร์ที่สร้างขึ้นมามีมุมกับมุมมองต่างๆ จากมุมมองด้านบนถ้าหันหน้าไปทางซ้ายเวกเตอร์ก็จะชี้ไปทางขวาถ้าหน้าตรงเวกเตอร์จะชี้ลงข้างล่างทำมุม 90 องศากับพื้น จากมุมมองด้านบนทำให้สามารถรู้ถึงมุมการหันซ้ายขวาได้ และจากมุมมองด้านข้างเมื่อก้มหน้าและเงยหน้าเวกเตอร์จะทำมุมที่สัมพันธ์กันทำให้สามารถคำนวณหามุมในการก้มเงยได้ แสดงภาพใบหน้าการหันดังรูปภาพที่ 24



รูปภาพ 24 vector ที่กำหนดขึ้นมาและแสดงในมุมมองต่างๆ

A. มุมมองด้านบนปกติ B. มุมมองจากด้านบน C. มุมมองจากด้านข้าง

วิธีการคำนวณหามุมเมื่อมองจากด้านบน ทำการแปลงระบบพิกัด (Coordinate System) ให้สามารถคำนวณจากมุมมองด้านข้างง่ายขึ้นดังรูปภาพที่ 25 แปลงจากระบบพิกัด c มาเป็นในรูปแบบระบบพิกัด 0



รูปภาพ 25 เวกเตอร์ในพิกัด c และ พิกัด 0

A. ระบบพิกัด c ตามจุดที่ได้จาก mediapipe B. ระบบพิกัด 0

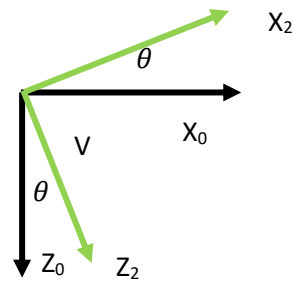
หาค่า 0R_c (rotation matrix) เพื่อทำการแปลงจากระบบพิกัดแบบ c ไปเป็นระบบพิกัดแบบ 0 จากภาพที่ 3 จะเห็นได้ว่าแกน X เหมือนเดิม แกน Y ซี่ตรงข้ามกัน และแกน Z ซี่ตรงข้ามกัน ค่า rotation จะได้ดังนี้

$${}^0R_c = [{}^0X_c \quad {}^0Y_c \quad {}^0Z_c] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

หาค่าเวกเตอร์ 0V ที่มองจากระบบพิกัด 0 จะได้ดังนี้

$${}^0V = {}^0R_c {}^cV \quad \text{โดย} \quad {}^0V = \begin{bmatrix} {}^0V_x \\ {}^0V_y \\ {}^0V_z \end{bmatrix}$$

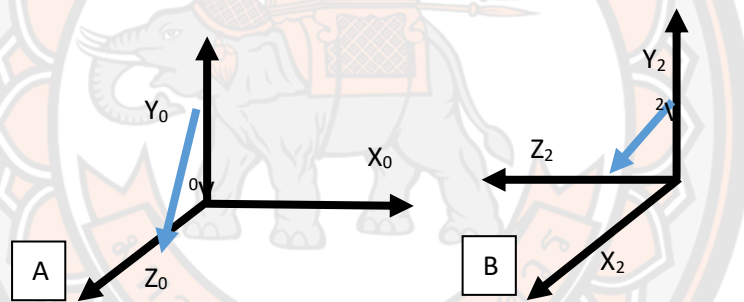
หามุมของเวกเตอร์ V การหันหน้าซ้ายขวาซึ่งเป็นการหมุนรอบแกน Y โดยหามุมที่ทำกับแกน z แล้วหาค่า θ_1 ออกมา ค่า θ_1 ที่ได้ก็จะเป็นมุมของการหันหน้าซ้าย, หน้าตรง, หันขวา ตามภาพที่



รูปภาพ 26 แสดงมุมของเวกเตอร์ V เทียบกับแกน Z_0

$$\theta_1 = \text{atan} \left(\frac{{}^0V_x}{{}^0V_z} \right)$$

วิธีการคำนวณหามุมเมื่อมองจากด้านข้าง คำนวณเริ่มจากระบบพิกัด 2 ซึ่งหมุนรอบแกน y_0 เป็นมุม θ ทำให้แกน Z_2 ตรงกับ vector V rotation matrix ในการแปลงจากพิกัด 2 มาพิกัด 0 ตามรูปภาพที่ 27



รูปภาพ 27 A. ระบบพิกัด 0 ของเวกเตอร์ V B. ระบบพิกัด 2 ของเวกเตอร์ V

A. ระบบพิกัด 0 ของเวกเตอร์ V B. ระบบพิกัด 2 ของเวกเตอร์ V

rotation matrix ในการแปลงจากพิกัด 2 ไป 0 คือ

$${}^0R_2 = \begin{bmatrix} {}^0X_2 & {}^0Y_2 & {}^0Z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & -1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

rotation matrix ในการแปลงจากพิกัด 0 ไป 2 คือ

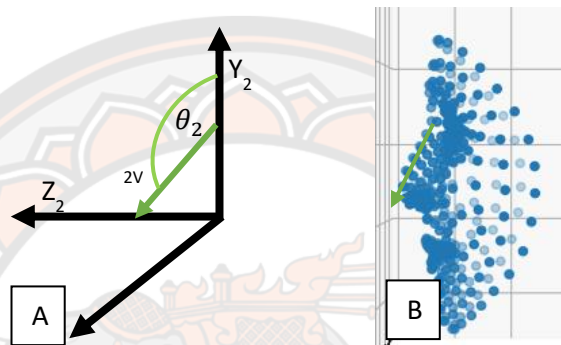
$${}^2R_0 = {}^0R_2^T = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & -1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

หา vector V เมื่อมองด้วยพิกัด 2 จะได้ว่า

$${}^2V = {}^2R_0 {}^0V$$

หามุมด้านข้างของใบหน้าเพื่อหาทิศทางก้มเงยของ vector V ได้จาก

$$\theta_2 = \text{atan2}({}^2V_z, {}^2V_y)$$

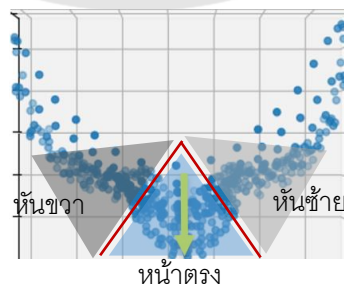


รูปภาพ 28 พิกัด 2 เทียบกับมุมมองด้านข้าง

A. ภาพแสดงมุมระหว่าง Y_2 กับ 2V จะได้มุม θ_2 B. ภาพแสดงเวกเตอร์จากมุมมองใบหน้าด้านข้าง

- การหาทิศทางของใบหน้า

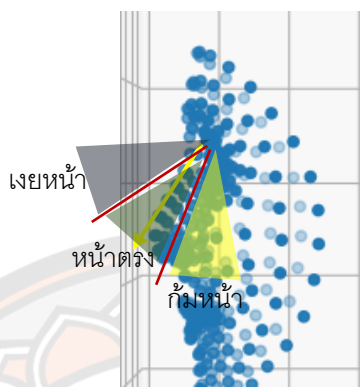
การหันหน้าซ้ายหรือหันหน้าขวา เมื่อมองมุมมองใบหน้าจากด้านบน ทำการเชื่อมมุมระหว่างเวกเตอร์กับแกน Z สำหรับหน้าตรงถ้าไม่เกินที่ขนาดมุมที่กำหนดไว้ให้เป็นหน้าตรง ถ้าหันซ้ายเกินมุมที่กำหนดให้แสดงผลเป็นหันซ้าย ถ้าหันขวาเกินมุมที่กำหนดก็ให้แสดงผลเป็นหันขวา แสดงดังภาพที่ 29



รูปภาพ 29 ระยะการหันหน้าที่กำหนดไว้ระหว่าง หันขวา หน้าตรง และ หันซ้าย

การก้มหน้าหรือเงยหน้า เมื่อมองมุมมองใบหน้าจากด้านข้าง ทำการเชื่อมมุมระหว่างเวกเตอร์กับแกน Y สำหรับหน้าตรงถ้าไม่เกินที่ขนาดมุมที่กำหนดไว้ให้เป็นหน้าตรง ถ้ามุนน้อยกว่าที่กำหนดให้

แสดงผลลัพธ์เป็นเงยหน้า ถ้ามุมเกินกว่าที่กำหนดก็ให้แสดงผลลัพธ์เป็นหันก้มหน้า แสดงดังรูปภาพที่ 30



รูปภาพ 30 ระยะการหันหน้าที่กำหนดไว้ระหว่าง เงยหน้า หน้าตรง และ ก้มหน้า

3.4 การตรวจจับสถานะของดวงตา

สถานะของดวงตาที่สนใจมีดังนี้ 1. ลืมตาหรือหลับตา 2. ทิศทางของม่านตา 3. มีอะไรมาบังดวงตาหรือไม่

- ลืมตาหรือหลับตา

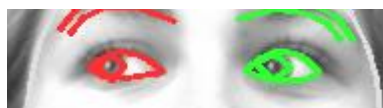
ทำการเช็คว่ายอดตาบนกับขอบตาล่างมีระยะห่างมากหรือน้อย ถ้าระยะห่างระหว่างขอบตาบนกับขอบตาล่างน้อยแสดงว่าเป็นไปได้มากที่ตอนนี้หลับตาอยู่



รูปภาพ 31 ผลการตรวจจับดวงตา

- ทิศทางของม่านตา

จากจุดสำคัญของใบหน้าที่ได้มาจะมีม่านตาด้วย (iris) ทำให้สามารถเช็คทิศทางของม่านตาได้ง่าย ทิศทางของม่านตาที่สนใจคือ มองทางซ้าย มองตรง และมองทางขวา



รูปภาพ 32 การตรวจจับม่านตาที่มองทางซ้าย

- มีอะไรมาบังดวงตาหรือไม่

สถานะที่สนใจคือ ดวงตาปกติหรือมีอะไรมาบังดวงตา ทำการเช็คว่ายอดตาที่ตรวจจับเจอมีลักษณะคล้ายดวงตาหรือไม่ ถ้าไม่คล้ายดวงตาแสดงว่ามีบางสิ่งมาบังดวงตา

บทที่ 4

ผลการวิจัย

ผลการวิจัยเรื่อง การพัฒนาโปรแกรมเพื่อตรวจจับทิศทางใบหน้าและสถานะดวงตาที่ถูกปิดบางส่วน ผู้วิจัยได้วิเคราะห์และนำเสนอในรูปแบบของตารางประกอบการอธิบาย โดยแบ่งออกเป็น 3 ส่วนดังนี้

1. ผลการตรวจจับทิศทางการหันหน้า
2. ผลการตรวจจับสถานะของดวงตา
3. ผลการนำไปประยุกต์ใช้

4.1 ผลการตรวจจับทิศทางการหันหน้า

4.1.1 การทดสอบการหันหน้า

ภาพที่นำมาทดสอบมาจากวิดีโอที่มีการหันหน้าต่างๆ มีอาสาสมัคร 61 คนแยกเป็นผู้ชาย 47 คน หญิง 14 คน ช่วงอายุ 24-47 ปี เฉลี่ย 25 ปี สวมแว่นตา 19 คน ในวิดีโอมีการ หันซ้าย หันขวา ก้มหน้า เงยหน้าและหน้าตรง ทำการแคปรูปออกมา 2 รูปต่อการหันหน้า 1 คนจะได้ทั้งหมด 10 รูป รวมเป็น 610 รูป นำแต่ละรูปมาทำการทดสอบความถูกต้องได้ดังตารางที่ 1

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		118				
	เงยหน้า			115	2		
	ก้มหน้า				105		
	หน้าตรง	1	3	2	7	122	
	ไม่เจอ	1	1	5	8		

ตาราง 1 ผลการตรวจจับทิศทางการหันหน้า

ทดสอบอีกครั้งพร้อมเพิ่มการสวมหน้ากากอนามัย ผลการทดสอบดังตารางที่ 2

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		119				
	เงยหน้า			103			
	ก้มหน้า				95		
	หน้าตรง	2	3	5	6	122	
	ไม่เจอ			14	21		

ตาราง 2 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย

4.1.2 ทดสอบเพิ่มระยะของภาพ

เพิ่มระยะของภาพ จากภาพปัจจุบันระยะห่างประมาณ 0.5 เมตร เพิ่มเป็น 1 เมตร และ 2 เมตร ดังในรูปตัวอย่างที่ 33 นักระยะ 1 และ 2 เมตร มาทดสอบเพิ่มเติม



รูปภาพ 33 ระยะห่าง 0.5 เมตร 1 เมตร และ 2 เมตร ตามลำดับ

ระยะ 1 เมตร

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		118				
	เงยหน้า			112	1		
	ก้มหน้า				106		
	หน้าตรง	2	1	3	6	122	
	ไม่เจอ		3	7	9		

ตาราง 3 ผลการตรวจจับทิศทางการหันหน้า (1 เมตร)

ทดสอบอีกครั้งพร้อมเพิ่มการสวมหน้ากากอนามัย ผลการทดสอบดังตารางที่ 4

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		122				
	เงยหน้า			98			
	ก้มหน้า				90		
	หน้าตรง	1		7	10	122	
	ไม่เจอ	1		17	22		

ตาราง 4 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (1 เมตร)

ระยะ 2 เมตร

		actual					
Prediction		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
	หันซ้าย	117					
	หันขวา		118				
	เงยหน้า			112			
	ก้มหน้า				106		
	หน้าตรง	2	3	3	7	120	
	ไม่เจอ	3	1	7	9	2	

ตาราง 5 ผลการตรวจจับทิศทางการหันหน้า (2 เมตร)

ทดสอบอีกครั้งพร้อมเพิ่มการสวมหน้ากากอนามัย ผลการทดสอบดังตารางที่ 6

		actual					
Prediction		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
	หันซ้าย	115					
	หันขวา		113				
	เงยหน้า			98			
	ก้มหน้า				90		
	หน้าตรง	6	7	9	15	122	
	ไม่เจอ	1	2	15	17		

ตาราง 6 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (2 เมตร)

4.1.2 ทดสอบเพิ่มลดแสงสว่าง

จากรูปภาพที่มีนำมาเพิ่มลดแสงสว่าง ดังในภาพที่ 34 นำภาพปกติที่ระยะห่าง 0.5 เมตร นำมาลดแสงและเพิ่มแสง โดยใช้ OpenCV ฟังก์ชันชื่อ convertScaleAbs การลดแสงทำการปรับค่า alpha เท่ากับ 0.5 และ beta เท่ากับ -10 ส่วนการเพิ่มแสง alpha เท่ากับ 1.5 และ beta เท่ากับ 10



รูปภาพ 34 ภาพที่ปรับปรุงแสง ลดแสง ปกติ และเพิ่มแสงตามลำดับ

ลดแสง

actual

	หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
หันซ้าย	120					
หันขวา		118				
เงยหน้า			115	2		
ก้มหน้า				105		
หน้าตรง	1	3	2	7	122	
ไม่เจอ	1	1	5	8		

Prediction

ตาราง 7 ผลการตรวจจับทิศทางการหันหน้า (ลดแสง)

ทดสอบอีกครั้งพร้อมเพิ่มการสวมหน้ากากอนามัย ผลการทดสอบดังตารางที่ 8

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		119				
	เงยหน้า			103			
	ก้มหน้า				95		
	หน้าตรง	2	3	5	6	122	
	ไม่เจอ			14	21		

ตาราง 8 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (ลดแสง)

เพิ่มแสง

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		118				
	เงยหน้า			115	2		
	ก้มหน้า				105		
	หน้าตรง	1	3	2	7	122	
	ไม่เจอ	1	1	5	8		

ตาราง 9 ผลการตรวจจับทิศทางการหันหน้า (เพิ่มแสง)

ทดสอบอีกครั้งพร้อมเพิ่มการสวมหน้ากากอนามัย ผลการทดสอบดังตารางที่ 10

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	120					
	หันขวา		119				
	เงยหน้า			103			
	ก้มหน้า				95		
	หน้าตรง	2	3	5	6	122	
	ไม่เจอ			14	21		

ตาราง 10 ผลการตรวจจับทิศทางการหันหน้าเมื่อสวมหน้ากากอนามัย (เพิ่มแสง)

ทดสอบเฉพาะกลุ่มคนใส่แว่นตาทั้งหมด 19 คน คนละ 2 รูป ได้รูปแบบการหันหน้าละ 38 รูป ดังตารางที่ 11

		actual					
		หันซ้าย	หันขวา	เงยหน้า	ก้มหน้า	หน้าตรง	ไม่เจอ
Prediction	หันซ้าย	38					
	หันขวา		34				
	เงยหน้า			29			
	ก้มหน้า				28		
	หน้าตรง		4	4	5	38	
	ไม่เจอ			5	5		

ตาราง 11 ผลการตรวจจับทิศทางการหันหน้าเฉพาะคนใส่แว่นตา

ผลการวิจัยพบว่าความถูกต้องในการหาทิศทางของใบหน้าในทิศทางต่าง ๆ จากการหันซ้าย การหันขวา การเงยหน้าและการก้มหน้าของอาสาสมัครมีความถูกต้องร้อยละ 95.08 การหันหน้าผิดพลาดมากที่สุดคือการก้มหน้า เหตุเพราะตอนก้มหน้าจะตรวจจับใบหน้าไม่เจอทำให้ไม่สามารถหาทิศทางของใบหน้าได้ รองลงมาคือการเงยหน้าจะคล้ายส่วนของก้มหน้าแต่ว่าในการทดสอบเงยหน้ามากไม่ได้เพราะมือถือที่ทำการบินที่อยู่ต่ำกว่าใบหน้า ส่วนการหันซ้าย หันขวา แทบ จะถูกต้องทั้งหมด ความแตกต่างในช่วงอายุเนื่องจากกลุ่มตัวอย่างมีอายุ 23 ปี มากกว่าร้อยละ 80 ทำให้ไม่สามารถเปรียบเทียบกลุ่มช่วงอายุได้

ความแตกต่างของผลลัพธ์เมื่อแยกระหว่าง ชาย และ หญิง อันดับแรกผมที่ปิดใบหน้า ผู้หญิงจะมีผมปิดเยอะกว่าผู้ชาย แต่ในกลุ่มตัวอย่างผู้ชายบางคนไว้ผมข้างหน้ายาว ทำให้ไม่แตกต่างกันมาก สองผู้ชายจะมีหนวด ในกลุ่มตัวอย่างชายส่วนมากมีหนวดเล็กน้อยกับไม่มีเลยทำให้โปรแกรมยังสามารถตรวจจับได้ดี

4.2 ผลการตรวจจับสถานะของดวงตา

ภาพที่ใช้ในการทดสอบมาจากวิดีโอเกี่ยวกับการทดสอบหันหน้า ทั้งหมด 61 คน ภาพที่ไม่มีคือหลับตาก็มีสิ่งบดบังดวงตา ภาพหลับตาเลือกจับภาพในขณะที่หลับตาได้มาทั้งหมด 35 ภาพ ส่วนภาพที่มีสิ่งบดบังดวงตาทำการตัดรูปอื่นมาใส่แทนดวงตาแล้วทดสอบ

actual

	ม่านตามองทางซ้าย	ม่านตามองทางขวา	ม่านตามองตรง
Prediction	ม่านตามองทางซ้าย	118	
	ม่านตามองทางขวา		115
	ม่านตามองตรง	4	7
			122

ตาราง 12 ผลการตรวจจับสถานะดวงตาซ้าย (1)

actual

	ลืมตา	หลับตา	มีสิ่งบดบังดวงตา
Prediction	ลืมตา	103	
	หลับตา	15	35
			2

มีสิ่งบดบังดวงตา	4		120
------------------	---	--	-----

ตาราง 13 ผลการตรวจจับสถานะดวงตาซ้าย (2)

actual

	ม่านตามองทางซ้าย	ม่านตามองทางขวา	ม่านตามองตรง
Prediction			
ม่านตามองทางซ้าย	110		
ม่านตามองทางขวา		120	
ม่านตามองตรง	12	2	122

ตาราง 14 ผลการตรวจจับสถานะดวงตาขวา (1)

actual

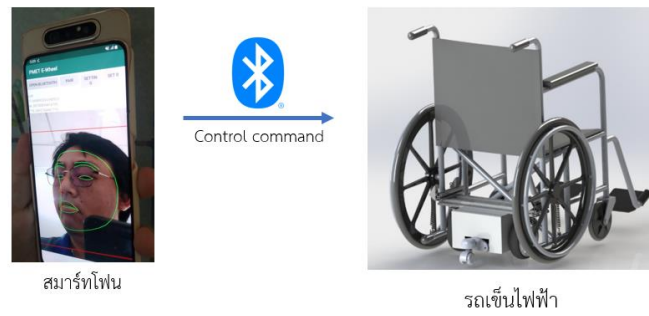
	ลืมตา	หลับตา	มีสิ่งบดบังดวงตา
Prediction			
ลืมตา	112		
หลับตา	7	35	
มีสิ่งบดบังดวงตา	3		122

ตาราง 15 ผลการตรวจจับสถานะดวงตาขวา (2)

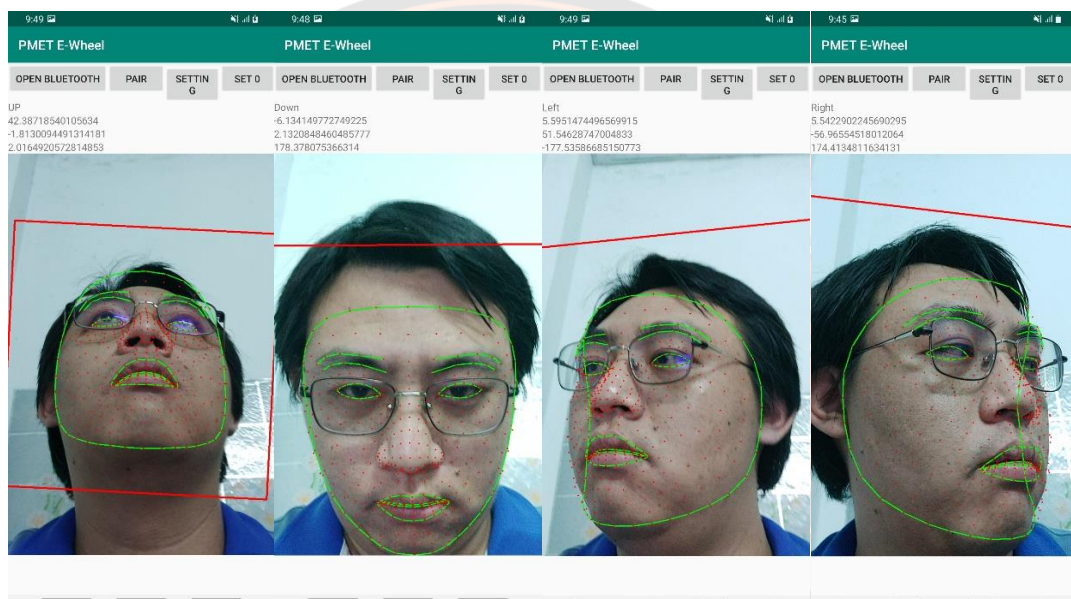
จากตารางที่ 12 13 14 และ 15 จะเห็นได้ว่าม่านตามองทางซ้าย-ขวา มีความผิดพลาดเยอะ เพราะการรูปที่ได้ส่วนมากเป็นการหันหน้าแล้วมองกล้องทำให้ไม่ได้รูปหน้าตรงมาทดสอบ ส่วนของลืมตามีส่วนคาบเกี่ยวกับคนตาเล็กกับหลับตา

4.3 ผลการนำไปประยุกต์ใช้

ในงานวิจัยนี้ นำไปทดสอบควบคุมรถเซ็นไฟฟ้าด้วยทิศทางของใบหน้า ทำการลงโปรแกรมทดสอบในสมาร์ตโฟนนำไปติดตั้งหน้ารถเซ็น ใช้การหันของใบหน้านำไปแปลงเป็นคำสั่งให้รถเซ็นไฟฟ้าเคลื่อนที่ คำสั่งที่ใช้คือ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา หน้าตรง นำคำสั่งที่ได้ส่งผ่าน Bluetooth ไปให้อุปกรณ์ที่อยู่ตรงรถเซ็นไฟฟ้าเพื่อทำการเคลื่อนที่ต่อไป



รูปภาพ 35 การส่งคำสั่งไปให้รถเข็นไฟฟ้าผ่าน Bluetooth



รูปภาพ 36 ตัวอย่างโปรแกรมการหันหน้าสำหรับใช้กับรถเข็นไฟฟ้า

การหันหน้า	ข้อความ	ทิศทางการเคลื่อนที่
หน้าตรง	ไม่มีการส่งข้อความ	ไม่มีการเคลื่อนที่
หันซ้าย	L	หันไปด้านซ้ายโดยอยู่ตำแหน่งเดิม
หันขวา	R	หันไปด้านขวาโดยอยู่ตำแหน่งเดิม
ก้มหน้า	F	เคลื่อนที่ไปข้างหน้า
เงยหน้า	B	เคลื่อนที่ไปข้างหลัง

ตาราง 16 คำสั่งที่ใช้ในการส่งไปให้รถเซ็นไฟฟ้า

การทดสอบจะดูว่าการหันหน้าถูกต้องมากน้อยแค่ไหน โดยทดสอบการหันหน้าไปทางซ้าย 100 ครั้ง หันหน้าไปทางขวา 100 ครั้ง ก้มหน้า 100 ครั้ง เงยหน้า 100 ครั้ง

การหันหน้า	ถูกต้องจาก 100 ครั้ง
หันซ้าย	96
หันขวา	89
เงยหน้า	83
ก้มหน้า	72

ตาราง 17 ผลที่ได้จากการทดสอบการหันหน้า

จากผลการทดลองจะเห็นได้ว่าการหันซ้ายหันขวานั้นสูงกว่า 90% แต่การก้มหน้ากับเงยหน้าที่ได้ความถูกต้องน้อยลงมาเพราะมุมมองของสมาร์ทโฟนที่วางจะอยู่ต่ำกว่าใบหน้าจริงไม่ได้อยู่ตรงหน้าของผู้ใช้ ทำให้บางครั้งการก้มหน้ากับเงยหน้าตรวจจับไม่เจอใบหน้า

4.4 ตัวอย่างที่ทำให้เกิดการควมผิดพลาด

ตัวอย่างข้อมูลที่ทำให้เกิดการประมวลผลผิดพลาดมีสามารถจัดออกตามขั้นตอนการทำงานได้ดังนี้

ตรวจจับไม่เจอใบหน้า เพราะเงยหน้ามากเกินไปหรือก้มมากเกินไป หรือหันซ้ายขวาที่มากเกินไปจนไม่เห็นดวงตาอีกข้าง จนทำให้ mediapipe (facemesh) ไม่สามารถตรวจจับหาใบหน้าได้ ดังรูปภาพที่ 37



รูปภาพ 37 รูปที่ไม่สามารถตรวจจับใบหน้าได้

การหาจุดสำคัญบนใบหน้าได้จุดที่ไม่ตรงกับใบหน้า ตัวอย่างในรูปภาพที่ 38 การก้มมากเกินไปให้จุดสำคัญที่ได้มาคลาดเคลื่อน การใส่หน้ากากอนามัยพร้อมก้มก็มีส่วนอยู่บ้างเพราะตำแหน่งจมูกบางครั้งจะไม่ตรง การหันหน้าซ้ายขวามากเกินไปก็ทำให้คลาดเคลื่อนได้มุมที่ทำให้คลาดเคลื่อนส่วนมากจะมากกว่า 50 องศา



รูปภาพที่ 38 ตัวอย่างการตรวจจับจุดสำคัญบนใบหน้าที่มีผิดพลาด

ตำแหน่งมุมของกล้องกับใบหน้าที่ต่างกันเกินไป ดังรูปภาพที่ 39 ภาพซ้ายจะเป็นการก้มหน้า ภาพขวาเป็นการเงยหน้า เนื่องจากกล้องอยู่ต่ำเกินไปทำให้มุมที่ใช้ในการตัดสินใจระหว่างเงยหน้ากับก้มหน้าจะไม่เหมือนกันภาพอื่นๆ



รูปภาพที่ 39 ตัวอย่างภาพที่มุมของกล้องต่ำเกินไป



บทที่ 5

สรุปผลการวิจัย

5.1 ผลการวิจัยสามารถสรุปดังนี้

5.1.1 บกทศทงการหมนของบอหน้า แบ่งเป็นการหันซ้าย หันขวา ก้มหน้า เงยหน้า ถ้ามุมของกล้องกับบอหน้าอยู่ในมุมที่เหมาะสมจะทำให้ได้ผลของการก้มหน้าและเงยหน้าดีขึ้น ส่วนการหันซ้ายและหันขวานั้นผลออกมาเข้าใกล้ 100% จะมีผิดพลาดตรงที่หันหน้ามากกว่า 45 องศา

5.1.2 บกสถานะของดวงตาได้ว่าตอนนี้ลืมตา หลับตา ผลที่ได้อยู่ในระดับที่ดี จะมีส่วนที่ผิดพลาดตรงเกณฑ์ที่แยกระหว่างลืมตากับหลับตาบางคนตาเล็กก็ทำให้ผิดพลาดได้ ส่วนสถานะของม่านตาวามองไปทางไหนเนื่องจากการคำนวณนั้นไม่ยากเลยทำให้ได้ผลดี แต่ก็มีข้อจำกัดคือความละเอียดของภาพถ้าน้อยไปจะมองไม่เห็นม่านตา

5.1.3 สามารถตรวจจับบอหน้าปกติและบอหน้าที่มีอะไรมาบังบางส่วนได้ เนื่องจากการทดสอบเป็นการจำลองว่ามีอะไรมาบังดวงตาโดยการวางรูปอื่นทับดวงตา ทำให้สามารถแยกประเภทได้ง่าย แต่ถ้าสิ่งที่มาบังดวงตาลักษณะคล้ายดวงตาก็อาจทำให้ผิดพลาดได้

5.1.4 นำไปควบคุมอุปกรณ์ที่ติดกับรถเข็นให้รถเข็นเดินตามทิศทางที่กำหนดได้ ความถูกต้องของการควบคุมมากกว่าร้อยละ 90 และใช้อัลกอริทึมที่เหมาะสมสำหรับสมาร์ทโฟน (facemesh mediapipe) ทำให้ส่งคำสั่งได้ทันไม่กระตุก แต่สิ่งที่เป็นปัญหาคือการวางตำแหน่งของสมาร์ทโฟนมีผลทำให้คำนวณการก้มหน้าและเงยหน้าผิดพลาดได้ง่าย ส่วนหันซ้ายหันขวามีความถูกต้องสูง

5.2 ข้อเสนอแนะเพื่อการทำวิจัยต่อไป

- สามารถบอกขนาดของมุมการหันหน้าได้ แต่การทดสอบจะต้องมีการวัดการหันหน้าให้ชัดเจนเพื่อเทียบกับโปรแกรมที่พัฒนาขึ้นมา
- การหันหน้าอาจจะมียากกว่านี้เช่น หันซ้ายพร้อมกับเงยหน้า เป็นต้น
- ประยุกต์ใช้กับงานชิ้นอื่นที่ใช้งานดวงตา เช่น การปลดล็อคด้วยบอหน้า

บรรณานุกรม

1. Rafael C. Gonzalez , R.E.W., *Digital Image Processing*, ed. 4. 2018: New York, NY : Pearson.
2. Aggarwal, C.C., *Data Classification Algorithms and Applications*. 2020: Chapman and Hall/CRC.
3. สังก้อง, ส., การเขียนโปรแกรมภาษา *Python GUI+Network+Database+Web*, ed. น. ปี ยะ. 2563, กรุงเทพฯ: จีไอว่า.
4. King, D.E., *Dlib-ml: A machine learning toolkit*. *Journal of Machine Learning Research* 10, 2009: p. 1755-1758.
5. *mediapipe*. Available from: <https://google.github.io/mediapipe/>.
6. Paul Viola, M.J., *Rapid Object Detection using a Boosted Cascade of Simple Features*, in *COMPUTER VISION AND PATTERN RECOGNITION*. 2001.
7. Upal Mahbub, V.M.P., Deepak Chandra, Brandon Barbello, Rama Chellappa, *Partial Face Detection for Continuous Authentication*. 2016 IEEE International Conference on Image Processing (ICIP), 2016: p. 2991-2995.
8. Bisoy, S.A.P.K.S.K., *Face Recognition from Partial Face Data*, in *2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*. 2021, IEEE: Bhubaneswar, India.
9. X. Zhu, Z.L., X. Liu, H. Shi, and S. Z. Li, *Face Alignment Across Large Poses: A 3D Solution*, in *In CVPR*. 2016.
10. Tzimiropoulos, A.B.a.G., *How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)*, in *International Conference on Computer Vision (ICCV)*. 2017.
11. Shanhe Yi, Z.Q., Ed Novak, Yafeng Yin, Qun Li, *GlassGesture: Exploring Head Gesture Interface of Smart Glasses*. 2016.
12. Xue Wang, Y.Z., *Nod to Auth: Fluent AR/VR Authentication with User Head-Neck Modeling*. 2021.

13. สิริศักดิ์ ทีโน, ส.เ., พนมขวัญ รียะมงคล, รัฐภูมิ วรานุศาสน์, การพัฒนาแอปพลิเคชันควบคุมรถเซ็นไฟฟ้ด้วยทิศทางของใบหน้า, in การประชุมวิชาการระดับชาติ ม.ทักษิณ ครั้งที่ 32. 2565.

