

ระบบการสื่อสารด้วยท่าทางมือ
HAND GESTURE APPLICATION

ปณชิตยา คุณาวุฒิ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบการสื่อสารด้วยท่าทางมือ

HAND GESTURE APPLICATION

ปภาณทิยา คุณาวุฒิ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ปริญญาโทปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบการสื่อสารด้วยท่าทางมือ

HAND GESTRUE APPLICATION

ผู้จัดทำ

1. นางสาวปณิธา คุณาวุฒิ รหัสนักศึกษา 56010745



อาจารย์ที่ปรึกษา

(ผศ. ธนา หงษ์สุวรรณ)

ระบบการสื่อสารด้วยท่าทางมือ

นางสาวปณทิตยา คุณาวุฒิ 56010745
ผู้ช่วยศาสตราจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2560

บทคัดย่อ

ปัญหาที่ท้าทายอย่างหนึ่งของการเป็นสังคมผู้สูงอายุ คือ จะให้ผู้สูงอายุมีคุณภาพชีวิตที่ดีได้อย่างไร ความเสี่ยงด้านสุขภาพเป็นปัญหาที่สำคัญประการหนึ่ง เนื่องจากผู้สูงอายุมีแนวโน้มจะเกิดการเสื่อมสภาพของอวัยวะต่างๆ ในร่างกาย และเกิดโรคได้ จึงได้คิด โครงการงานขึ้น มานำเสนอการนำเทคโนโลยีด้านสารสนเทศมาช่วยลดความเสี่ยงด้านสุขภาพของผู้สูงอายุที่อยู่ตามลำพัง ซึ่งเมื่อเกิดเหตุฉุกเฉินแล้ว หรืออาจจะไม่ได้เกิดเหตุฉุกเฉินแต่ต้องการความช่วยเหลือทางใดทางหนึ่ง แล้วไม่สามารถขอความช่วยเหลือผ่านโทรศัพท์ได้

ระบบการสื่อสารด้วยท่าทางมือนี้อาจมีจุดมุ่งหมายคือเราจะติดกล้องไว้ตามบ้านแล้วพอผู้สูงอายุต้องการจะทำสิ่งใดสิ่งหนึ่ง หรือมีเหตุฉุกเฉิน ผู้สูงอายุสามารถขมู่มือขึ้นแล้วทำท่า 4 ท่าที่เรากำหนดไว้จากนั้นกล้องจะจับภาพทำได้

HAND GESTURE APPLICATION

Miss Pantiya Khunawut 56010745

Asst.Prof.Thana Hongsuwan Advisor

Academic Year 2017

ABSTRACT

Having a good quality of life at the older ages is one of the challenges for coming aging society in Thailand. The age-related health problem is becoming a big issue due to elderly's body degeneration and causes some degenerative diseases. As a result, this project is initiated to study of how to facilitate those elderly people suffering from their health problems to have a good quality of life with IT Technology. Especially for those who stay alone and would like to contact their family members in case of either urgent or non-urgent situation by showing their hand gestures and then the application will automatically send their request via mobile notification to their family members.

กิตติกรรมประกาศ

กิตติกรรมประกาศโครงการวิจัย ระบบการสื่อสารท่าทางมือนี้สำเร็จลุล่วงได้ด้วยความรู้และความช่วยเหลือ จากอาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษาและชี้แนะแนวทางการแก้ไขปัญหา ข้อบกพร่องต่างๆ และคณะผู้จัดทำโครงการวิจัยนี้ กราบขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ บิดา มารดาและ ผู้อุปการะคุณทุกท่าน ที่ได้ให้ความช่วยเหลือ แม้ไม่มากนัก แต่ทางผู้จัดมีความซาบซึ้งในความช่วยเหลือนั้น จึงขอขอบพระคุณอย่างสูง

กราบขอขอบพระคุณทุกท่าน มา ณ ที่นี้ค่ะ

ด้วยความเคารพอย่างสูง

ผู้จัดทำ

ปาณชิตยา คุณาวุฒิ

สารบัญ

หน้า

| | |
|---|-----|
| บทคัดย่อภาษาไทย..... | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VII |
| สารบัญภาพ..... | III |
| | |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ที่มาและความสำคัญของโครงการ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ประโยชน์ที่คาดว่าจะได้รับ..... | 2 |
| 1.4 ขอบเขตของโครงการ..... | 2 |
| | |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง..... | 3 |
| 2.1 หลักการเบื้องต้นในการประมวลผลภาพ..... | 3 |
| 2.1.1 ภาพดิจิทัล..... | 3 |
| 2.1.2 องค์ประกอบภาพดิจิทัล..... | 3 |
| 2.1.3 ประเภทของภาพดิจิทัล..... | 4 |
| 2.1.3.1 ภาพไบนารี..... | 4 |
| 2.1.3.2 ภาพระดับเทา..... | 5 |
| 2.1.3.3 ภาพดัชนี..... | 5 |
| 2.1.3.4 ภาพสี..... | 6 |
| 2.2 หลักการประมวลผลภาพ..... | 8 |
| 2.2.1 สีคืออะไร..... | 8 |
| 2.2.1.1 ลักษณะทางกายภาพของสี..... | 8 |
| 2.2.1.2 ลักษณะการมองเห็นของสี..... | 9 |
| 2.2.2 แบบจำลองสี..... | 9 |
| 2.2.2.1 แบบจำลองสี RGB..... | 9 |
| 2.2.2.2 แบบจำลองสี HSV..... | 10 |
| 2.2.3 การเปลี่ยนค่าสี RGB เป็นสี HSV..... | 10 |

สารบัญ(ต่อ)

หน้า

| | |
|---|----|
| 2.3 การประมวลผลภาพเฉพาะบริเวณ..... | 12 |
| 2.3.1 การกรองข้อมูลภาพ..... | 13 |
| 2.3.1.1 คอนโวลูชันแบบสองมิติ..... | 13 |
| 2.4 การคืนสภาพของภาพ..... | 16 |
| 2.4.1 ตัวกรองเชิงพื้นที่..... | 17 |
| 2.4.1.1 ตัวกรองค่าเฉลี่ย..... | 17 |
| 2.4.1.2 ตัวกรองสถิติเชิงอันดับ..... | 18 |
| 2.5 ฮิสโทแกรม..... | 19 |
| 2.6 การแบ่งส่วนของภาพ..... | 19 |
| 2.6.1 การแบ่งส่วนภาพโดยการขีดแบ่ง (Threshold) | 19 |
| 2.6.2 การประมวลผลภาพด้วยมอโฟร์โลยีพื้นฐาน..... | 20 |
| 2.6.2.1 การกร่อน..... | 20 |
| 2.7 ฟังก์ชันเพิ่มเติมของ OPENCV และ NUMPY..... | 22 |
| 2.7.1 Contours และอัลกอริทึมที่ใช้..... | 21 |
| 2.7.1.1 การวิเคราะห์โครงสร้างเชิงโครงสร้างของภาพไบนารี โดยใช้เส้นขอบ..... | 22 |
| 2.7.2 Convex Hull และอัลกอริทึมที่ใช้..... | 23 |
| 2.7.2.2 อัลกอริทึม Sklansky's..... | 24 |
| 2.7.3 Background Subtraction | 26 |
| 2.8 กฎการจำแนก..... | 26 |
| 2.8.1 การสร้างต้นไม้ตัดสินใจ..... | 27 |
| 2.8.1.1 อัลกอริทึม ID3..... | 29 |
| บทที่ 3 การออกแบบการทำงาน..... | 31 |
| 3.1 ซอฟต์แวร์ที่เกี่ยวข้องกับการพัฒนา..... | 31 |
| 3.2 ฮาร์ดแวร์ที่เกี่ยวข้องกับการพัฒนา..... | 31 |
| 3.3 ภาพรวมของระบบ..... | 31 |
| 3.4 ขั้นตอนการทำงาน..... | 32 |
| 3.5 ออกแบบการดำเนินงาน..... | 32 |
| 3.5.1 ส่วนเบ็กกราวนด์..... | 32 |

สารบัญ(ต่อ)

| | หน้า |
|---|------|
| 3.5.2 ส่วนสีของมือ..... | 33 |
| 3.5.3 ส่วนของมือที่เหลือในการเตรียมหาทำทาง..... | 34 |
| 3.5.3.1 หาจุดศูนย์กลางมือ..... | 37 |
| 3.5.3.2 หา Convex Hull..... | 37 |
| 3.6 หาทำทางมือโดยใช้ Decision Tree Classifier..... | 38 |
| 3.6.1 การสร้างต้นไม้ตัดสินใจด้วยอัลกอริทึม ID3..... | 38 |
| 3.6.1.1 การหาค่าคาดคะเนของข้อมูล..... | 38 |
| 3.6.1.2 ต้นไม้ตัดสินใจด้วยอัลกอริทึม ID3..... | 48 |
| บทที่ 4 การทดลอง..... | 49 |
| 4.1 การเรียนรู้..... | 49 |
| 4.2 วิเคราะห์การเรียนรู้..... | 50 |
| 4.3 การทดสอบ..... | 51 |
| 4.3.1 สมมติฐาน..... | 51 |
| 4.3.2 ผลการทดลอง..... | 51 |
| 4.4 วิเคราะห์ผล..... | 54 |
| 4.5 สรุปผลการทดลอง..... | 54 |
| บทที่ 5 บทสรุปและข้อเสนอแนะ..... | 56 |
| 5.1 บทสรุปของโครงการ..... | 56 |
| 5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข..... | 56 |
| 5.3 แนวทางการพัฒนาต่อ..... | 56 |
| บรรณานุกรม..... | 57 |

สารบัญตาราง

| ตาราง | หน้า |
|---|------|
| 2.1 แสดงค่าระดับสีของแบบจำลองสี HSV..... | 11 |
| 2.2 แสดงค่าสี HSV และสี RGB..... | 12 |
| 2.3 ตัวอย่างข้อมูลการตัดสินใจเล่นกอล์ฟ..... | 28 |
| 3.1 ข้อมูลสำหรับแอคทริบิวต์ “มุม” | 39 |
| 3.2 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่หนึ่ง”..... | 40 |
| 3.3 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่สอง”..... | 40 |
| 3.4 ข้อมูลสำหรับแอคทริบิวต์ “มุม” ในกรณีที่มีมุมน้อยกว่าหรือเท่ากับ 58.668..... | 41 |
| 3.5 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่หนึ่ง” กรณีมุมน้อยกว่าหรือเท่ากับ 58.668 | 42 |
| 3.6 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่สอง” กรณีมุมน้อยกว่าหรือเท่ากับ 58.668 | 42 |
| 3.7 ข้อมูลสำหรับแอคทริบิวต์ “มุม มากกว่า 58.668”..... | 43 |
| 3.8 ข้อมูลสำหรับแอคทริบิวต์ “มุม” ในกรณีที่มีมุมอยู่ระหว่าง 56.668 ถึง 80.034 | 43 |
| 3.9 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่หนึ่ง” ในกรณีที่มีมุมมีค่ามากกว่า 58.668 | 44 |
| 3.10 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่สอง” ในกรณีที่มีมุมมีค่ามากกว่า 58.668 | 44 |
| 3.11 ข้อมูลสำหรับแอคทริบิวต์ มุม > 80.034 | 45 |
| 3.12 ข้อมูลสำหรับแอคทริบิวต์ มุม 80.034 < มุม < 136.202 | 45 |
| 3.13 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่หนึ่ง” ในกรณีที่มีมุมน้อยกว่า 80.034 | 46 |
| 3.14 ข้อมูลสำหรับแอคทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขาวนิ้วที่สอง” ในกรณีที่มีมุมน้อยกว่า 80.034 | 46 |

สารบัญรูป

| รูป | หน้า |
|--|------|
| 2.1 แสดงตัวอย่างภาพดิจิทัล..... | 3 |
| 2.2 แสดงตัวอย่างภาพดิจิทัลขนาด 906×510 พิกเซล..... | 4 |
| 2.3 แสดงตัวอย่างภาพไบนารี..... | 4 |
| 2.4 แสดงตัวอย่างภาพระดับเทา และค่าโทนสีเทาของแต่ละพิกเซล..... | 5 |
| 2.5 แสดงตัวอย่างภาพดัชนี และค่าของแต่ละพิกเซล..... | 5 |
| 2.6 แสดงตัวอย่างภาพสี RGB..... | 6 |
| 2.7 แสดงตัวอย่างของภาพสี ภาพต้นฉบับ ภาพสีแดง ภาพสีเขียว และภาพสีน้ำเงิน..... | 6 |
| 2.8 แสดงตัวอย่างภาพมี 24 บิตเดปท์..... | 7 |
| 2.9 ความละเอียดของภาพสีเทา 1 บิต 2 บิต และ 4 บิต..... | 7 |
| 2.10 จำนวนสีกับบิตเดปท์หรือความละเอียดของภาพ..... | 8 |
| 2.11 ค่าระดับความเข้มเทาของภาพแต่ละพิกเซล..... | 8 |
| 2.12 แสดงไดอะแกรมของสี..... | 9 |
| 2.13 แบบจำลอง HSV..... | 10 |
| 2.14 การประมวลผลภาพเฉพาะบริเวณ..... | 13 |
| 2.15 การคอนโวลูชันแบบสองมิติ..... | 14 |
| 2.16 (ก) ตัวกรองความถี่ต่ำเกาส์เซียน (ข) มาส์กสำหรับตัวกรองความถี่ต่ำเกาส์เซียน ขนาด 3×3 | 18 |
| 2.17 ภาพและฮิสโทแกรมของภาพ..... | 19 |
| 2.18 ตัวอย่างการแบ่งด้วยการขีดแบ่งด้วยค่า Threshold..... | 20 |
| 2.19 ตัวอย่างการกร่อน..... | 21 |
| 2.20 ตัวอย่างการใช้ฟังก์ชัน cv2.findContours..... | 22 |
| 2.21 รูปของขั้นตอนของการวิเคราะห์โครงสร้างเชิงโครงสร้างของภาพไบนารี โดยใช้เส้นขอบ..... | 23 |
| 2.22 ตัวอย่าง Convex hull เมื่อ returnPoints เป็น True..... | 23 |
| 2.23 ตัวอย่าง Sklansky's square | 24 |
| 2.24 รูปแสดงขั้นตอนการสร้างโมเดลการจำแนก..... | 27 |
| 2.25 ตัวอย่างต้นไม้ตัดสินใจจากชุดข้อมูลการตัดสินใจเล่นกอล์ฟ..... | 29 |
| 3.1 ภาพรวมของระบบ..... | 31 |
| 3.2 ภาพของขั้นตอนการทำงาน..... | 32 |
| 3.3 รูปหน้าเริ่มต้นของโปรแกรม..... | 33 |

สารบัญรูป(ต่อ)

| รูป | หน้า |
|--|------|
| 3.4 ตัวอย่างภาพการเรียนรู้สีของมือ..... | 33 |
| 3.5 ตัวอย่างภาพที่ถูกเปลี่ยนเป็นสี HSV..... | 34 |
| 3.6 ตัวอย่างอาเรย์ 9 ช่องที่จะนำไปหา Histogram..... | 34 |
| 3.7 ตัวอย่างจากผลลัพธ์จากส่วนแบ็กกราวน์ | 34 |
| 3.8 ตัวอย่างการครอบตัดภาพแล้วเปลี่ยนเป็นสี HSV | 35 |
| 3.9 ผลจากการนำฮิสโทแกรมมา Match กับสีของมือที่เราต้องการ..... | 35 |
| 3.10 แสดงตัวอย่างการทำคอนโวลูชันด้วย Ellipse (13,13) | 35 |
| 3.11 แสดงตัวอย่างรูปที่ทำการเทรชโฮล | 36 |
| 3.12 แสดงตัวอย่างการหาเส้นขอบ | 36 |
| 3.13 แสดงตัวอย่างการหาจุดศูนย์กลางมือ..... | 37 |
| 3.14 แสดงตัวอย่าง Convex hull..... | 37 |
| 3.15 แอดทริบิวต์ของ Tree..... | 38 |
| 3.16 แสดง Decision Tree..... | 48 |
| 4.1 ตัวอย่างการเรียนรู้ 4 ท่า..... | 49 |
| 4.2 แอดทริบิวต์ 3 แอดทริบิวต์ของท่า L และท่า V | 50 |
| 4.3 แอดทริบิวต์ 3 แอดทริบิวต์ของท่าคาราบาว และท่าความรัก | 50 |
| 4.4 รูปทดสอบคนที่หนึ่ง..... | 51 |
| 4.5 รูปทดสอบคนที่สอง..... | 52 |
| 4.6 รูปทดสอบคนที่สาม..... | 52 |
| 4.7 รูปทดสอบคนที่สี่..... | 53 |
| 4.8 รูปทดสอบคนที่ห้า..... | 53 |

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันสังคมไทยเริ่มเข้าสู่สังคมผู้สูงอายุ โดยคำว่า สังคมผู้สูงอายุนั้นหมายถึงสัดส่วนของประชากรผู้สูงอายุ (อายุ 60 ปีขึ้นไป) มีมากกว่า 20 เปอร์เซ็นต์ของประชากรรวมของประเทศ ซึ่งจากการคาดการณ์ของสภาพพัฒนาเศรษฐกิจและสังคมแห่งชาติ พบว่าในปี พ.ศ. 2563 หรืออีก 3 ปีข้างหน้า ประเทศไทยจะมีประชากรสูงวัยถึง 12.6 ล้านคน จากจำนวนประชากรรวม 66 ล้านคน หรือคิดเป็น 19.1 เปอร์เซ็นต์ของประชากรรวม

ปัญหาที่ท้าทายอย่างหนึ่งของการเป็นสังคมผู้สูงอายุ คือ จะให้ผู้สูงอายุมีคุณภาพชีวิตที่ดีได้อย่างไร ความเสี่ยงด้านสุขภาพเป็นปัญหาที่สำคัญประการหนึ่ง เนื่องจากผู้สูงอายุมิแนวโน้มจะเกิดการเสื่อมสภาพของอวัยวะต่างๆ ในร่างกาย และเกิดโรค โดยโรคที่พบบ่อย คือ เบาหวาน ความดันโลหิตสูง โรคหลอดเลือดหัวใจ โรคหลอดเลือดสมอง ซึ่งโรคเหล่านี้บางครั้งเกิดอย่างเฉียบพลัน และต้องการความช่วยเหลืออย่างรีบด่วน

นอกจากนั้นแนวโน้มการอยู่ตามลำพังคนเดียวหรืออาศัยกับคู่สมรสโดยไม่มีลูกหลานคอยดูแลของผู้สูงอายุมิแนวโน้มเพิ่มขึ้นอย่างต่อเนื่อง โดยในปี 2529 สัดส่วนผู้สูงวัยที่อยู่คนเดียวมี ร้อยละ 4.3 เพิ่มขึ้น ร้อยละ 7.6 ในปี 2550 ส่วนกลุ่มผู้สูงวัยที่อยู่กันเฉพาะคู่สมรส ในปี 2529 มีอัตราส่วน ร้อยละ 6.7 เพิ่มขึ้นเป็น ร้อยละ 16.3 ในปี 2550 ความเสี่ยงด้านสุขภาพของผู้สูงอายุที่อยู่ตามลำพังคนเดียว หรือกรณีที่อยู่ตามลำพังในเวลากลางวันที่ถูกหลานไปทำงาน จึงเพิ่มสูงขึ้นมากขึ้นไปอีก

โครงการนี้นำเสนอการนำเทคโนโลยีด้านสารสนเทศมาช่วยลดความเสี่ยงด้านสุขภาพของผู้สูงอายุที่อยู่ตามลำพัง ซึ่งเมื่อเกิดเหตุฉุกเฉินแล้ว อาจไม่สามารถขอความช่วยเหลือผ่านช่องทางปกติ เช่น โทรศัพท์ได้ นอกจากนั้นยังใช้ช่วยอำนวยความสะดวกบางอย่างแก่ผู้สูงอายุ โดยโครงการนี้เลือกการประยุกต์ใช้ท่าทางมือ (Hand Gesture) มาใช้ เนื่องจากเห็นว่าการใช้ท่าทางมือ สามารถสอนให้ผู้สูงอายุเรียนรู้ได้ง่าย

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อสร้างแอปพลิเคชัน (Application) สำหรับตรวจจับท่าทางมือ ณ เวลาจริง (Real Time)
- 2) สามารถนำไปประยุกต์ใช้เพื่อช่วยลดความเสี่ยงด้านสุขภาพ หรืออำนวยความสะดวกแก่ผู้สูงอายุได้
- 3) เพื่อศึกษาแนวทางการนำเทคโนโลยีด้านการประมวลผลภาพ (Image Processing) มาใช้-งาน

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) มีระบบที่สามารถตรวจจับท่าทางมือได้
- 2) ช่วยอำนวยความสะดวกแก่ผู้สูงอายุได้
- 3) เข้าใจการพัฒนาโดยการนำเทคโนโลยีด้านการประมวลผลภาพ (Image Processing)

1.4 ขอบเขตของโครงการ

สร้างโปรแกรมประยุกต์ท่าทางมือ บนเครื่องคอมพิวเตอร์ขนาดเล็กกราฟิกราสเบอร์รี่พาย ที่ติดกล้องวิดีโอ ซึ่งจะคอยตรวจจับท่าทางมือและทำงานตามที่กำหนดไว้ โดยจะมีการทำงาน ดังนี้

- 1) พัฒนาโปรแกรมที่จับภาพเคลื่อนไหว และแยกภาพเคลื่อนไหวออกเป็นเฟรมภาพ
- 2) ศึกษาวิธีการทางการประมวลผลภาพต่างๆ เพื่อหาวิธีการที่เหมาะสม ที่ใช้แยกภาพมือและตรวจจับท่าทางมือจากภาพ
- 3) พัฒนาโปรแกรมที่สามารถตรวจจับท่าทางมือ และสามารถบอกได้ว่าขณะนั้นผู้ใช้กำลังทำท่าทางมือที่กำหนดหรือไม่ (กำหนดท่าทางมือไว้ไม่เกิน 5 ท่า)

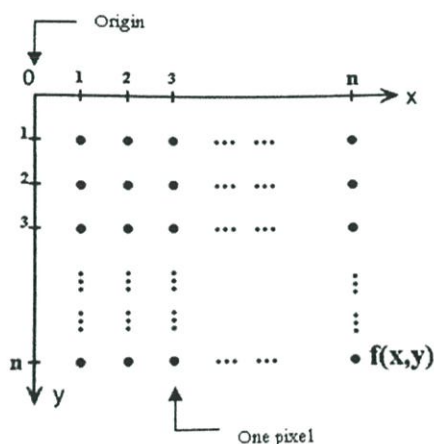
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 หลักการเบื้องต้นในการประมวลผลภาพ

2.1.1 ภาพดิจิทัล

ภาพดิจิทัลเป็นฟังก์ชันของข้อมูล 2 มิติ ซึ่งมีการอ้างอิงค่าตามตำแหน่งของพิกเซล ดังแสดงในรูป 2.1 โดยค่าที่อ้างอิงในแต่ละตำแหน่งของพิกเซลส่วนใหญ่อยู่ในรูปของค่าความเข้มแสง และสี ซึ่งค่าของพิกเซลนี้สามารถแทนได้ในหลายรูปแบบตามประเภทของภาพ ดังจะกล่าวไปหัวข้อต่อไป



รูปที่ 2.1 แสดงตัวอย่างภาพดิจิทัล

2.1.2 องค์ประกอบของภาพดิจิทัล

ภาพดิจิทัลสามารถนิยามเป็นฟังก์ชันสองมิติ $f(x, y)$ โดยที่ x, y เป็นพิกัดของภาพ และแอมพลิจูดของ f ที่พิกัด (x, y) ใดๆ ภายในภาพ กล่าวคือค่าความเข้มแสงของภาพที่ตำแหน่งนั้นๆ เมื่อ (x, y) และแอมพลิจูดของ f เป็นค่าจำกัด (Finite Value)

กำหนดให้ภาพ $f(x, y)$ เป็นภาพดิจิทัลที่มีขนาด M แถว N หลัก และมีพิกัดจุดกำเนิด ภาพที่ตำแหน่ง $(x, y) = (0, 0)$ ดังนั้นจะสามารถเขียนสมการให้อยู่ในรูปของเมตริกซ์ได้ ดังนี้

$$f(x, y) = \left(\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \right) \quad (2.1)$$

จากสมการที่ 2.1 ค่าแต่ละค่าที่อยู่ในเมตริกซ์จะเรียกว่า “องค์ประกอบภาพ” หรือ พิกเซล โดยตำแหน่ง (0,0) จะอยู่ทางด้านซ้ายมือบนของภาพ การจัดลำดับตำแหน่งจุดภาพ จะเรียงจากซ้ายไปขวาในแต่ละเส้นภาพและจัดลำดับของเส้นภาพจะเรียงจากบนลงล่าง ดังแสดงตัวอย่างในรูปที่ 2.2



รูปที่ 2.2 แสดงตัวอย่างภาพดิจิทัล ขนาด 906×510 พิกเซล

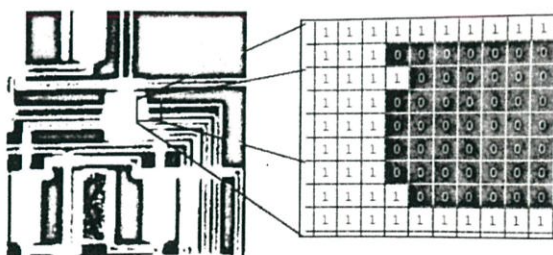
การใช้หน่วยความจำเพื่อจัดเก็บภาพดิจิทัล โดยเนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก จำนวนแถว \times จำนวนหลัก \times จำนวนบิตของข้อมูลในแต่ละจุดภาพ

2.1.3 ประเภทของภาพดิจิทัล

ภาพดิจิทัลมีอยู่หลากหลายประเภท เช่น ไบนารี (Binary Image) ภาพระดับเทา (Intensity Image) ภาพดัชนี (Indexed Image) และภาพสี (RGB Image) โดยภาพแต่ละประเภทนั้นจะมีค่าความเข้มแสงของภาพที่แตกต่างกัน

2.1.3.1 ภาพขาวดำหรือภาพไบนารี

สำหรับแต่ละจุดภายในภาพไบนารีนั้น สามารถแสดงค่าที่แตกต่างกันได้ 2 ระดับ คือ 0 สีดำ และ 1 สีขาวเท่านั้น ดังรูปที่ 2.3

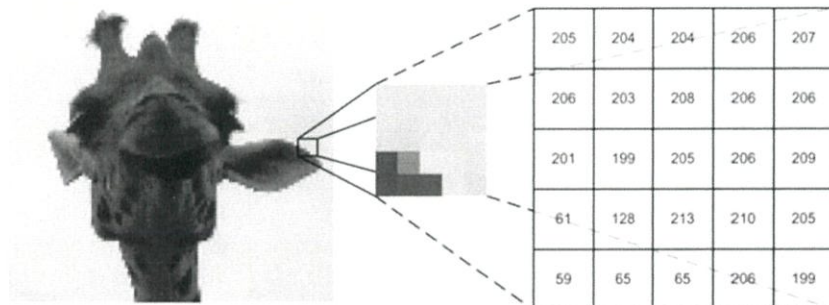


รูปที่ 2.3 แสดงตัวอย่างภาพไบนารี

2.1.3.2 ภาพระดับเทา

ภาพระดับเทาเป็นภาพที่แสดงค่าความเข้มแสง ณ แต่ละตำแหน่งของจุดภาพภายในภาพ ซึ่งภาพจะมีลักษณะเป็น โทนสีเทา (Gray Scale) โดยค่าความเข้มแสงที่สามารถแสดงได้จะขึ้นอยู่กับจำนวนบิตที่ใช้แทนภาพในแต่ละภาพได้จาก $L = 2^k$ เมื่อ L คือระดับสีเทา และ k คือค่าจำนวนบิตที่ใช้แทนภาพ เช่น ภาพระดับเทาขนาด 8 บิต สามารถแสดงค่าระดับสีเทาได้ทั้งหมด 256 ระดับ ค่าที่เป็นไปได้คือ 0-255

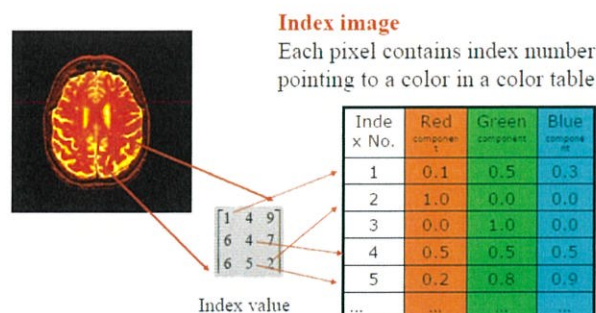
ภาพระดับเทา (Gray Level Image) หมายถึง ภาพที่มีค่าความสว่างของแต่ละจุดภาพอยู่ในช่วงสีดำ สีเทา เรื่อยไปจนถึงสีขาวซึ่งเป็นค่าความสว่างที่มากที่สุด สำหรับค่าของระดับสีเทานั้น ปกติแล้วโดยทั่วไปจะมีความละเอียด 8 บิต หรือ 256 ระดับ หรืออาจใช้จำนวนบิตมากหรือน้อยกว่านั้นขึ้นอยู่กับความละเอียดของภาพที่ต้องการ ว่ามากหรือน้อยเพียงใด รูปที่ 2.4 แสดงภาพระดับเทาที่เก็บด้วยจำนวนบิตต่างๆกัน



รูปที่ 2.4 แสดงตัวอย่างภาพระดับเทา และค่าโทนสีเทาของแต่ละพิกเซล

2.1.3.3 ภาพดัชนี

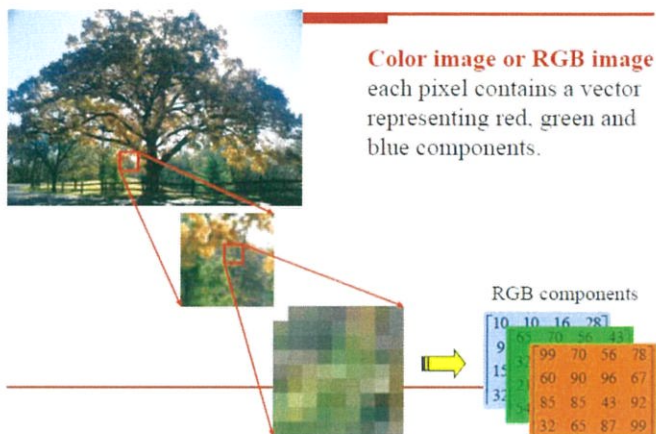
ภายในแต่ละจุดในภาพที่มีการทำดัชนีจะเก็บค่าดัชนีตัวเลข ซึ่งเป็นค่าที่ชี้ไปยังสีในตารางสี ดังนั้นถ้าต้องการจะทราบสีในแต่ละจุดภาพจะต้องไปดูค่าสีในตารางสีที่ดัชนีอ้างอิงตรงกับค่าในจุดภาพ



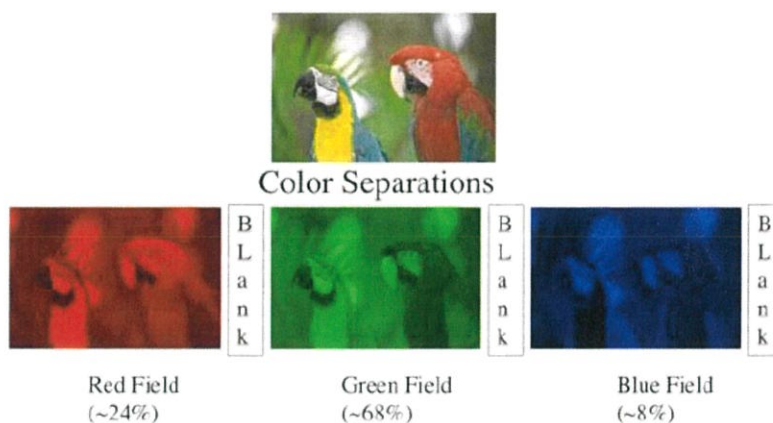
รูปที่ 2.5 แสดงตัวอย่างภาพดัชนี และค่าของแต่ละพิกเซล

2.1.3.4 ภาพสี

สำหรับแต่ละพิกเซลในภาพสีประกอบด้วยค่าสี 3 ค่า นั่นก็คือค่าของสีแดง สีเขียว และสีน้ำเงิน ดังรูปที่ 2.6 และรูปที่ 2.7



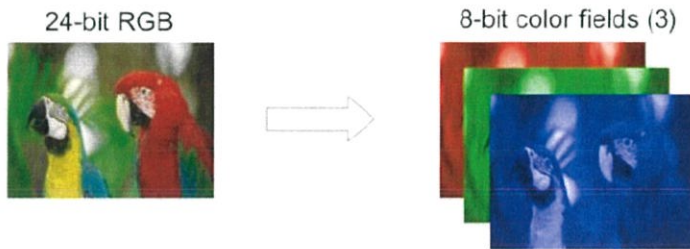
รูปที่ 2.6 แสดงตัวอย่างภาพสี RGB และค่าในแต่ละพิกเซลของภาพสี



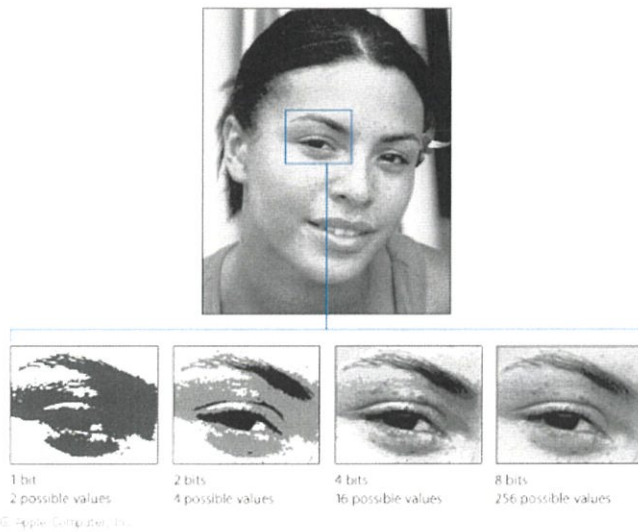
รูปที่ 2.7 แสดงตัวอย่างของภาพสี ภาพต้นฉบับ ภาพสีแดง ภาพสีเขียว และสีน้ำเงินตามลำดับ

การไล่เฉดสี ถ้าภาพขาว-ดำ จะเห็นว่ามีแค่สองสี คือ ขาวกับดำ ฉะนั้นความละเอียดของภาพนั้นคือ 1 บิตเดปท์ หรือเท่ากับ $2^1 = 2$ สี นั่นคือขาวกับดำ แต่โดยทั่วไปดังรูปที่ 3.6 จะไล่เฉดสีออกเป็นสามสี คือ สีแดง สีเขียวและสีน้ำเงิน แต่สีแดงมีความละเอียดเท่ากับ 2^8 บิต สีเขียวก็ความละเอียดเท่ากับ 2^8 บิต สีน้ำเงินก็มีความละเอียดเท่ากับ 2^8 บิตเช่นกัน จึงเอาสามรูปนี้มาซ้อนทับกันจึงเรียกรูปประเภทนี้ว่า RGB และมีจำนวนสีเท่ากับ $2^8 \times 2^8 \times 2^8 = 2^{24}$ เรียกรูปประเภทนี้ว่า 24 Bit Depth

ภาพสีจะมีความแตกต่างกับภาพระดับเทาที่ได้กล่าวมาข้างต้น เพราะนอกจากค่าความเข้มแสงแล้ว ภาพสียังประกอบไปด้วยสัญญาณสีต่างๆ ที่จะต้องนำมาแยกก่อนการวิเคราะห์ห้วงค์ประกอบของภาพต่อไป แม้ว่าคุณลักษณะสีเป็นสิ่งที่สายตามนุษย์มองเห็นและรับรู้ได้ดีที่สุด อย่างไรก็ตามจำนวนบิตที่ใช้เป็นแทนค่าสีแต่ละพิกเซล ก็มีผลต่อการมองเห็นของมนุษย์ ดังแสดงในรูปที่ 2.8 ถึงผลของการแทนระดับสีด้วยจำนวนบิตที่แตกต่างกัน จะเห็นว่าการไล่ของเฉดสีของการแทนสีด้วยจำนวนบิตเดปท์สูงๆ ก็ย่อมให้ความต่อเนื่องของการไล่เฉดสีได้ดีกว่าการแทนสีด้วยบิตเดปท์ต่ำๆ



รูปที่ 2.8 ภาพ มี 24 บิตเดปท์ออกเป็น 3 แชนแนลหรือ 3×2^8 คือ น้ำเงิน เขียว แดง



รูปที่ 2.9 ความละเอียดของภาพสีเทา 1 บิต 2 บิต 4 บิต และ 8 บิตตามลำดับ

| Color depth | Red | | Green | | Blue | | Total colors |
|-------------|--------|---|--------|---|--------|---|---------------------|
| 1 bit | 2 | x | 2 | x | 2 | = | 8 |
| 2 bit | 4 | x | 4 | x | 4 | = | 64 |
| 3 bit | 8 | x | 8 | x | 8 | = | 512 |
| 4 bit | 16 | x | 16 | x | 16 | = | 4,096 |
| 5 bit | 32 | x | 32 | x | 32 | = | 32,768 |
| 6 bit | 64 | x | 64 | x | 64 | = | 262,144 |
| 7 bit | 128 | x | 128 | x | 128 | = | 2,097,152 |
| 8 bit | 256 | x | 256 | x | 256 | = | 16,777,216 |
| 9 bit | 512 | x | 512 | x | 512 | = | 134,217,728 |
| 10 bit | 1,024 | x | 1,024 | x | 1,024 | = | 1,073,741,824 |
| 11 bit | 2,048 | x | 2,048 | x | 2,048 | = | 8,589,934,592 |
| 12 bit | 4,096 | x | 4,096 | x | 4,096 | = | 68,719,476,736 |
| 13 bit | 8,192 | x | 8,192 | x | 8,192 | = | 549,755,813,888 |
| 14 bit | 16,384 | x | 16,384 | x | 16,384 | = | 4,398,046,511,104 |
| 15 bit | 32,768 | x | 32,768 | x | 32,768 | = | 35,184,372,088,832 |
| 16 bit | 65,536 | x | 65,536 | x | 65,536 | = | 281,474,976,710,656 |

รูปที่ 2.10 จำนวนสีทั้งหมดในภาพกับบิตเดปท์หรือความละเอียดของภาพ

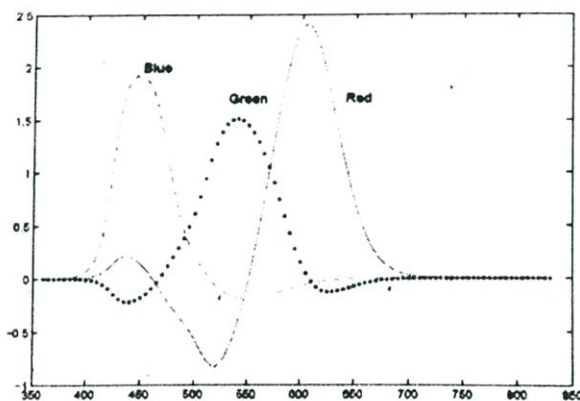
2.2 หลักการประมวลผลภาพ

2.2.1 สีคืออะไร

สีเกิดจากคุณสมบัติทางกายภาพของแสงซึ่งเป็นส่วนสำคัญที่ทำให้เกิดสีขึ้น และสีเป็นกรากฏการณ์ของการรับรู้เกี่ยวกับการมองเห็นอย่างหนึ่งของมนุษย์ และจำแนกสีต่างๆได้เช่นกัน

2.2.1.1 ลักษณะทางกายภาพของสี

แสงที่เรามองเห็นนั้นเป็นส่วนหนึ่งของพลังงานคลื่นแม่เหล็กไฟฟ้า โดยที่ค่าของความยาวคลื่นของสีน้ำเงิน สีเขียว และสีแดง ได้ถูกกำหนดขึ้นเมื่อปีคริสต์ศักราช 1931 โดย CIE (Commission International d'Eclairage) และองค์กรที่รับผิดชอบเกี่ยวกับมาตรฐานสีโดยเฉพาะ รูปที่ 2.11 แสดงถึงค่าความยาวคลื่นของแม่สีหลักทั้งสามสี คือ สีน้ำเงิน สีเขียว และสีแดง



รูปที่ 2.11 ค่าระดับความเข้มเทาของภาพแต่ละพิกเซล

2.2.1.2 ลักษณะการมองเห็นของสี

เป็นปรากฏการณ์ของการรับรู้เกี่ยวกับการมองเห็นอย่างหนึ่งของมนุษย์ เช่น การรับรู้ว่าดอกดาวเรืองเป็นสีเหลือง ใบไม้เป็นสีเขียว เป็นต้น ซึ่งโดยปกติแล้วมนุษย์จะมองเห็นสีต่างๆที่เกิดจากการผสมสีของสีแดง สีเขียว และสีน้ำเงินเป็นหลัก

การรับรู้สีเกิดขึ้นได้ก็ต่อเมื่อมีแสงเดินทางเข้าไปสู่ตา โดยตาของมนุษย์จะทำหน้าที่เป็นส่วนรับแสง และส่งสัญญาณไปยังสมองเพื่อแปลสัญญาณดังกล่าวเป็นการรับรู้สีต่างๆ ซึ่งภายในตาจะมีส่วนที่เรียกว่า เรตินา มีหน้าที่รับแสง และเปลี่ยนแสงเป็นกระแสประสาท โดยมีเซลล์รับแสงอยู่สองประเภท คือเซลล์รูปแท่ง และเซลล์รูปกรวย เซลล์รูปแท่งจะทำงานเมื่อมีแสงน้อย แต่เซลล์รูปกรวยจะทำงานเมื่อมีแสงมากเป็นเซลล์ที่ทำให้เกิดการรับรู้สี

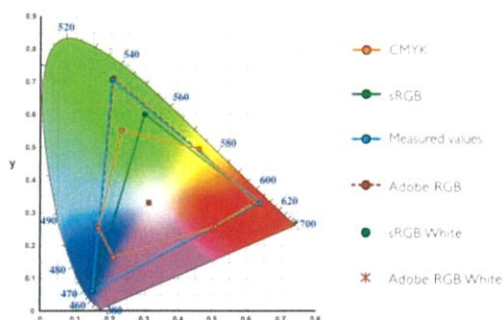
แม่สีหลักจะมีสามสี คือ สีแดง (R) สีเขียว (G) และสีน้ำเงิน (B) และแม่สีรองจะเกิดจากเอาสีแดง สีเขียว และสีน้ำเงินมาผสมกัน จะได้ สีน้ำเงินเขียว (Cyan) สีม่วงแดง (Magenta) สีเหลือง (Yellow) และสีขาว (White)

2.2.2 แบบจำลองสี

แบบจำลองสีเป็นวิธีที่กำหนดสีต่างๆให้เป็นแบบมาตรฐาน ซึ่งแบบจำลองสีในแต่ละแบบก็จะมีคุณสมบัติแตกต่างกันออกไป จึงเหมาะสำหรับการใช้งานที่แตกต่างกัน ในแบบจำลองของสีนั้นจะใช้สีหลักๆที่เรียกว่าแม่สีมาผสมผสานกลมกลืนกันเพื่อใช้ในการแสดงค่าสีอื่นๆ โดยแม่สีหลักนั้นจะแตกต่างกันไปตามแบบจำลองสีแต่ละชนิด

2.2.2.1 แบบจำลองสี RGB

แบบจำลองสีนี้เป็นแบบจำลองของแม่สีหลัก 3 สี ซึ่งเป็นสีที่เกิดจากการรวมกันของแสง ประกอบด้วยสีที่สำคัญ 3 สีด้วยกัน ได้แก่ สีแดง สีเขียว และสีน้ำเงิน ถูกนำมาใช้งานในด้านการแสดงผลข้อมูลบนจอภาพ รวมไปถึงการเก็บข้อมูลภาพในระบบคอมพิวเตอร์ เมื่อทำการพิจารณาสีโดยเปรียบเทียบกับความยาวคลื่น โดยใช้ไดอะแกรมของสีซึ่งแสดงไว้ดังรูปที่ 2.12



รูปที่ 2.12 แสดงไดอะแกรมของสี

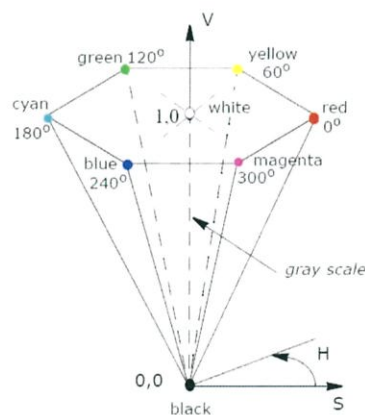
2.2.2.2 แบบจำลองสี HSV

แบบจำลองสีเป็นแบบจำลองสีที่ถูกกำหนดขึ้นมาเพื่อใช้งานศิลปะในระบบคอมพิวเตอร์ โดยที่

- 1) ค่าสี (Hue) หมายถึง ค่าสีต่างๆ เช่น สีแดง สีเขียว สีน้ำเงิน สีส้ม เป็นต้น
- 2) ค่าความอิ่มตัว (Saturation) หมายถึงปริมาณของสีที่ถูกทำให้เจือจางด้วยสีขาว โดยยังมีปริมาณสีขาวมากขึ้นเท่าใด ความอิ่มตัวยิ่งมีน้อย เช่น สีแดงมีความอิ่มตัวมากกว่าสีชมพู
- 3) ค่าความสว่าง (Value) คือ ระดับความสว่างของสีนั้น

แบบจำลองนี้มีความเป็นไปได้กับความเป็นจริงมากกว่าแบบจำลองอื่น เพราะว่าความเข้มข้นของสีนั้นไม่ได้ขึ้นอยู่กับตัวเนื้อสีเพียงอย่างเดียว แต่ยังสามารถเกี่ยวกับเรื่องแสงสว่างอีกด้วย ดังนั้นแบบจำลองนี้จะสามารถเพิ่มหรือลดความสว่างของสีใดสีหนึ่งได้ และแบบจำลองนี้ค่อนข้างที่จะมีประโยชน์สำหรับการประมวลผลภาพเป็นอย่างมาก

หลักการของแบบจำลองสีจะเสมือนว่าเรามองลูกบาศก์ใบแบบจำลองสีRGB ในทิศทางขนานกับแกนระดับสีเทา จากตำแหน่งสีขาวเข้าไป เราสามารถที่จะมองภาพของแบบจำลองนี้เป็นรูปกรวยดังรูปที่ 2.13 ซึ่งทุกๆจุดบนผิวหน้าของรูปได้แสดงถึงค่าความอิ่มตัว ดังนั้นแกนรัศมีจึงบอกถึงค่าความอิ่มตัว ส่วนแกนความสูงบอกถึงค่าความสว่าง และสุดท้ายมุมของแกนรัศมีบ่งบอกถึงค่าของสี



รูปที่ 2.13 แบบจำลองสี HSV

2.2.3 การเปลี่ยนค่าแบบจำลองสี RGB เป็นค่าแบบจำลองสี HSV

สมมติว่าสีได้ถูกกำหนดไว้ด้วยแบบจำลอง RGB ถ้าค่าของสามสีเท่านั้น ดังนั้นสีที่จะได้เป็นภาพสีใน โทนเทาหรือภาพขาวดำนั่นเอง นั่นคือความเข้มข้นของสีขาวนั่นเอง สีนั้นได้ประกอบไปด้วยสีขาวซึ่งมีความอิ่มตัวเท่ากับศูนย์ ในทางกลับกัน ถ้าค่าของ RGB ทั้งสามมีค่า






แตกต่างกันมากๆ เราจะหวังได้ว่าผลลัพธ์ของสีจะมีค่าความอิ่มตัวสูง โดยเฉพาะ ถ้าหนึ่งหรือสองค่าของ RGB เป็นศูนย์ ค่าความอิ่มตัวจะเท่ากับหนึ่ง ซึ่งเป็นค่ามากที่สุดเท่าที่จะเป็นไปได้ ค่าความสว่างได้ถูกกำหนดเป็นเศษส่วนรอบๆวงกลมโดยเริ่มจากสีแดงซึ่งเท่ากับว่ามีค่าความสว่างเป็นศูนย์ และดูรอบๆวงกลมในรูปที่ 2.13 ได้ค่าระดับสีต่างๆ ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงค่าระดับสีของแบบจำลองสี HSV

| สี | ค่าระดับสี (Hue) |
|--------------|------------------|
| แดง | 0 |
| เหลือง | 0.1667 |
| เขียว | 0.333 |
| น้ำเงินเขียว | 0.5 |
| น้ำเงิน | 0.6667 |
| ม่วงแดง | 0.8333 |

ตารางที่ 2.2 แสดงค่าสี HSV และสี RGB

HSV to RGB color table

| Color | Color name | (H,S,V) | Hex | (R,G,B) |
|---|------------|------------------|---------|---------------|
|  | Black | (0°,0%,0%) | #000000 | (0,0,0) |
|  | White | (0°,0%,100%) | #FFFFFF | (255,255,255) |
|  | Red | (0°,100%,100%) | #FF0000 | (255,0,0) |
|  | Lime | (120°,100%,100%) | #00FF00 | (0,255,0) |
|  | Blue | (240°,100%,100%) | #0000FF | (0,0,255) |
|  | Yellow | (60°,100%,100%) | #FFFF00 | (255,255,0) |
|  | Cyan | (180°,100%,100%) | #00FFFF | (0,255,255) |
|  | Magenta | (300°,100%,100%) | #FF00FF | (255,0,255) |
|  | Silver | (0°,0%,75%) | #C0C0C0 | (192,192,192) |
|  | Gray | (0°,0%,50%) | #808080 | (128,128,128) |
|  | Maroon | (0°,100%,50%) | #800000 | (128,0,0) |
|  | Olive | (60°,100%,50%) | #808000 | (128,128,0) |
|  | Green | (120°,100%,50%) | #008000 | (0,128,0) |
|  | Purple | (300°,100%,50%) | #800080 | (128,0,128) |
|  | Teal | (180°,100%,50%) | #008080 | (0,128,128) |
|  | Navy | (240°,100%,50%) | #000080 | (0,0,128) |

สมมติว่าเราให้ค่าของสามสีอาร์ จี และบี อยู่ระหว่าง 0 ถึง 1 เท่านั้น แต่ถ้าค่าเหล่านั้นมีค่าอยู่ระหว่าง 0 ถึง 255 เราจะต้องทำการหารค่าต่างๆด้วย 255 เป็นอันดับแรกเพื่อเป็นการทำให้ค่าอยู่ระหว่าง 0 ถึง 1 จากนั้นเราจะทำการกำหนดค่าต่างๆ ตามสมการที่ 2.10

$$\left. \begin{aligned} V &= \max \{ R, G, B \} \\ \delta &= V - \min \{ R, G, B \} \\ S &= \frac{\delta}{V} \end{aligned} \right\} \dots(2.10)$$

โดยที่ $\min(R,G,B)$ และ $\max(R,G,B)$ คือค่าน้อยที่สุดและมากที่สุด เมื่อทำการเปรียบเทียบค่าระหว่างสีแดง สีเขียว และสีน้ำเงินตามลำดับ

ผลลัพธ์สำหรับค่า V ที่ได้ เราจะต้องพิจารณาจากกรณีในสมการ 2.11

$$\left. \begin{aligned} 1. \text{ ถ้า } R = V \text{ ดังนั้น } H &= \frac{1}{6} \left(\frac{G-B}{\delta} \right) \\ 2. \text{ ถ้า } G = V \text{ ดังนั้น } H &= \frac{1}{6} \left(2 + \frac{B-R}{\delta} \right) \\ 3. \text{ ถ้า } B = V \text{ ดังนั้น } H &= \frac{1}{6} \left(4 + \frac{R-G}{\delta} \right) \end{aligned} \right\} \dots(2.11)$$

ถ้าผลลัพธ์ของค่า H ที่ได้เป็นจำนวนติดลบ เราจะต้องทำการบวกค่า 1 เพิ่มเข้าไป ในกรณี
ที่ $(R,G,B) = (0,0,0)$ ค่า $v = \delta = 0$ ดังนั้น $(H,S,V) = (0,0,0)$

ตัวอย่างการเปลี่ยน RGB เป็น HSV

เมื่อสมมติค่า $(R,G,B) = (0.2, 0.4, 0.6)$

ทำให้ได้ค่า

$$V = \max\{0.2, 0.4, 0.6\} = 0.6$$

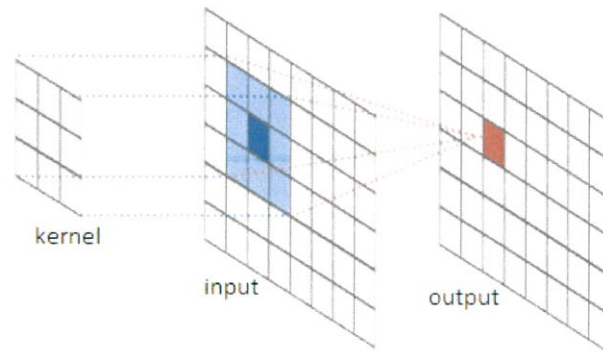
$$\delta = V - \min\{0.2, 0.4, 0.6\} = 0.6 - 0.2 = 0.4$$

$$S = \frac{0.4}{0.6} = 0.6667$$

$$\text{เมื่อ } B = V \text{ ดังนั้น เราจะได้ } H = \frac{1}{6} \left(4 + \frac{0.2-0.4}{0.4} \right) = 0.5833$$

2.3 การประมวลผลภาพเฉพาะบริเวณ

การประมวลผลภาพเฉพาะบริเวณ ทำให้ค่าระดับความเข้มเทาของพิกเซลในแต่ละจุดในภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของกลุ่มพิกเซลอยู่ในบริเวณข้างเคียงกันของภาพ ดังรูป 2.14 แสดงลักษณะการกระทำกับภาพเฉพาะบริเวณ ตัวอย่างในการกรองเชิงพื้นที่ (Spatial Domain filtering) หรือที่นิยมเรียกว่าคอนโวลูชัน (Convolution)



รูปที่ 2.14 การประมวลผลภาพเฉพาะบริเวณ

2.3.1 การกรองข้อมูลภาพ

การกรองข้อมูลภาพ (Image Filtering) คือการประมวลผลภาพแบบบริเวณอย่างหนึ่ง โดยหลักการคือ การนำภาพต้นฉบับไปผ่านตัวกรองสัญญาณที่จะกล่าวในหัวข้อ 2.5.2 เพื่อให้ได้ภาพผลลัพธ์ออกมา โดยภาพผลลัพธ์ที่ได้จะมีคุณสมบัติแตกต่างจากต้นฉบับ วัตถุประสงค์หลักการกรองข้อมูลภาพคือกรองสิ่งที่ไม่ต้องการออกไปเพื่อให้มีคุณสมบัติของภาพที่เราต้องการ

ในการกรองข้อมูลภาพ เราจะพิจารณาภาพดิจิทัล ภาพดิจิทัลคือสัญญาณสองมิติที่ประกอบขึ้นจากสัญญาณความถี่ต่างๆผสมกันอยู่ในสัดส่วนที่ต่างกัน การออกแบบตัวกรองจึงเป็นการกำหนดว่าเราต้องการจำกัดสัญญาณความถี่ใดออกไป หรือต้องการเลือกสัญญาณความถี่ใดบ้าง ดังนั้นในการกรองสัญญาณใดๆ เราจะต้องทราบความถี่ หรือช่วงความถี่นั้น จากนั้นเราจะเลือกตัวกรองที่เหมาะสมมาใช้เพื่อกำจัดสัญญาณที่ไม่ต้องการออก แบบที่เราใช้ในโครงการนี้คือการคอนโวลูชันแบบสองมิติ (Two-dimensional Convolution)

2.3.1.1 คอนโวลูชันแบบสองมิติ

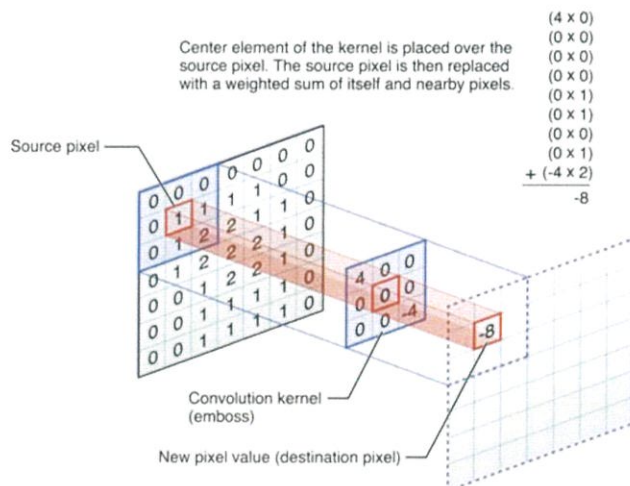
ในงานทางด้านประมวลผลภาพ การกรองข้อมูลภาพ จะใช้วิธีการประมวลผลภาพแบบบริเวณหรือคอนโวลูชัน ซึ่งการคอนโวลูชันคือการกระทำกันระหว่างภาพ และมาสก์ ซึ่งในหนังสือบางเล่มอาจเรียกแทนมาสก์ $M(i, j)$ เป็นหน้าต่างขนาด $n \times m$ และภาพ $F(x, y)$ ต้นฉบับมีขนาด $n \times m$ การคอนโวลูชันระหว่างมาสก์กับภาพสามารถแสดงได้ดังสมการที่ 2.12

$$G(x, y) = M * F = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} M(i, j) \cdot F(x - i, y - j) \quad \dots (2.12)$$

โดยที่ $G(x, y)$ คือภาพผลลัพธ์ที่ได้จากการคอนโวลูชันที่จุดพิกัด (x, y) ใดๆของภาพ จากสมการที่ 2.12 จะเห็นว่าระดับความเข้มเทาที่จุดพิกัด (x, y) ในภาพผลลัพธ์สามารถคำนวณได้จากการหาผลรวมของผลคูณระหว่างค่าสัมประสิทธิ์หรือค่าถ่วง

น้ำหนักในมาสก์ กับค่าระดับความเข้มเทาของภาพในบริเวณที่มาสก์ซ้อนทับอยู่ หรืออาจกล่าวได้ว่า ค่าระดับความเข้มเทาของพิกเซลแต่ละจุดของภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของกลุ่มพิกเซลที่อยู่ในบริเวณข้างเคียงกัน (Neighborhood Pixels) โดยการทำคอนโวลูชันนั้น เราจะต้องทำการเคลื่อนย้ายมาสก์ไปทางแนวแกนอนและแกนตั้ง หรือในแนวแถวและแนวหลักของภาพตลอดทั่วทั้งภาพ

เพื่อให้เกิดความเข้าใจง่ายและชัดเจนยิ่งขึ้น วิธีการคอนโวลูชันแบบสองมิติสามารถแสดงได้ดังรูปที่ 2.15 ขั้นตอนของวิธีการคอนโวลูชันนั้นเกิดจากการทำมาสก์ไปวางทับกับภาพต้นฉบับ โดยนำจุดตำแหน่งกึ่งกลางของมาสก์ไปวางซ้อนทับกับจุดพิกัด (x, y) ใดๆ ของภาพต้นฉบับ จากนั้นทำการคูณค่าสัมประสิทธิ์ หรือกลุ่มตัวเลขในมาสก์กับระดับความเข้มเทาของแต่ละจุดภาพในแต่ละตำแหน่งที่ตรงกันแบบจุดต่อจุด ต่อจากนั้นนำผลคูณที่ได้ทั้งหมดมารวมกัน เป็นค่าผลลัพธ์ของการทำคอนโวลูชัน โดยมาสก์นี้จะทำการเคลื่อนที่ผ่านทีละจุดภาพตลอดทั่วทั้งภาพต้นฉบับ (จากซ้ายไปขวา จากบนลงล่าง) และค่าผลลัพธ์ที่ได้จากการเคลื่อนย้ายมาสก์ในแต่ละจุดภาพนั้นจะถูกนำไปเก็บไว้ในข้อมูลของภาพผลลัพธ์ตรงตำแหน่งเดิมของภาพต้นฉบับ



รูปที่ 2.15 การคอนโวลูชันแบบสองมิติ

เป็นไปดังรูปที่ 2.15 โดยปกติแล้วในการคอนโวลูชันนั้น เราจะไม่มีอมให้มีการเลื่อนมาสก์ออกมานอกเขตของภาพ เมื่อเราพิจารณาเลือกใช้มาสก์ขนาด 3×3 ดังนั้นข้อมูลภาพต้นฉบับแถวแรก และแถวสุดท้าย รวมถึงข้อมูลภาพหลักแรกและหลักสุดท้าย จะไม่มีการประมวลผล จึงมีผลทำให้ภาพผลลัพธ์จะมีขนาดเล็กกว่าภาพต้นฉบับเสมอ ดังนั้นวิธีการต่างๆคือเติมค่าศูนย์บริเวณรอบๆ ของภาพต้นฉบับ ซึ่งเรียกว่า การทำ ซีโร แพดดิ้ง (Zero-Padding) ซึ่งจุดประสงค์เพื่อทำให้ภาพต้นฉบับมีขนาดใหญ่ขึ้น หลังจากการทำคอนโวลูชันแล้วจะได้ภาพผลลัพธ์มีขนาดเท่ากับภาพต้นฉบับ

แม้กระบวนการคอนโวลูชันสามารถทำความเข้าใจได้ง่าย แต่การทำคอนโวลูชันนั้นจะต้องใช้เวลามาก ซึ่งถ้าภาพขนาด $M \times M$ มาส์กขนาด $n \times n$ จะต้องมีการคูณถึง M^2n^2 ครั้ง เพื่อเป็นลดขั้นตอนของจำนวนครั้งในการทำคอนโวลูชัน ขนาดของมาส์กที่เป็นที่นิยมใช้กันคือขนาด 3×3

ตัวอย่างต่อไปนี้เป็นตัวอย่างแสดงการคำนวณทีละขั้นตอนของกระบวนการกระทำการกับภาพแบบบริเวณข้างเคียง (Local Image Processing)

ตัวอย่างการทำคอนโวลูชันสองมิติ

$$A = \begin{bmatrix} 1 & 2 & 3 & -1 \\ 6 & -5 & 4 & 3 \\ -2 & 4 & -6 & 7 \\ 5 & 0 & 1 & -3 \end{bmatrix} \text{ เมทริกซ์ } A \text{ คือข้อมูลของภาพ}$$

$$B = \begin{bmatrix} 1 & -3 & 5 \\ -2 & 4 & 6 \\ 7 & 8 & -9 \end{bmatrix} \text{ เมทริกซ์ } B \text{ คือมาส์ก}$$

ขั้นตอนการคอนโวลูชันสองมิติ

1. ทำการหมุนเมทริกซ์ B (Mask) 180 องศา ในทิศทางทวนเข็มนาฬิกา ผลลัพธ์ที่ได้ กำหนดให้เป็นเมทริกซ์ C
2.
$$B = \begin{bmatrix} -9 & 8 & 7 \\ 6 & 4 & -2 \\ 5 & -3 & 1 \end{bmatrix}$$
3. ทำการกำหนดตำแหน่งพิกเซลที่อยู่ตรงกลางของเมทริกซ์ C โดยหาได้จากการนำขนาดของเมทริกซ์ C บวกด้วย 1 แล้วหารด้วย 2
4. เมทริกซ์ 3×3 พิกเซล ตำแหน่งตรงกลางของเมทริกซ์คือ $((3+1/2), (3+1/2))$ นั่นก็คือตำแหน่ง (2, 2)
5. นำเมทริกซ์ C วางทับลงบนเมทริกซ์ A ที่เป็นภาพ โดยเริ่มที่ตำแหน่งแถวที่ 1 หลักที่ 1 (หรือว่าพิกเซล (0,0)) โดยให้พิกเซลที่อยู่ตรงกลางของมาส์กของเมทริกซ์ C ซึ่งก็คือตำแหน่งพิกเซล (2,2) อยู่ตรงกันกับแถว 1 หลัก 1 ของภาพ (เมทริกซ์ A)
6. ทำการเลื่อนมาส์ก (เมทริกซ์ C) วางทับลงบนเมทริกซ์ A ที่ตำแหน่งบนแถว 1 หลัก 2 ทำการหาค่าผลรวมของผลคูณ ผลรวมที่ได้นี้จะป็นค่าใหม่ที่อยู่ที่ตำแหน่งพิกเซล แถว 1 หลัก 2 ของเมทริกซ์ D
7. จากนั้นทำการเลื่อนมาส์ก (เมทริกซ์ C) ผ่านไปที่ละจุดภาพตลอดทั่วทั้งภาพต้นฉบับ (จากซ้ายไปขวาและบนลงล่าง) และค่าผลลัพธ์ที่ได้จากการเคลื่อนย้ายมาส์กในแต่ละจุดภาพนั้นจะถูกนำไปเก็บไว้ในเมทริกซ์

D ตรงตำแหน่งที่เดียวกับเมทริกซ์ A ทำเช่นนี้ไปเรื่อยๆ จนครบทุกตำแหน่งพิกเซลของเมทริกซ์ A

8. เมื่อทำครบทั้ง 5 ขั้นตอนดังกล่าวแล้ว จะทำให้ได้ผลลัพธ์ ที่เกิดจากการทำคอนโวลูชันระหว่างเมทริกซ์ A และ B ดังนี้

$$D(1,1) = (1)(4) + (2)(-2) + (6)(-3) + (-5)(1) = -23$$

$$D(1,2) = (1)(6) + (2)(4) + (3)(-2) + (6)(5) + (-5)(-3) + (4)(1) = 57$$

$$D(1,3) = (2)(6) + (3)(4) + (-1)(-2) + (-5)(5) + (4)(-3) + (3)(1) = -8$$

$$D(1,4) = (3)(6) + (-1)(4) + (4)(5) + (3)(-3) = 25$$

$$D(2,1) = (1)(8) + (2)(7) + (6)(4) + (-5)(-2) + (-2)(-3) + (4)(1) = 66$$

$$D(2,2) = (1)(-9) + (2)(8) + (3)(7) + (6)(6) + (-5)(4) + (4)(-2) + (-2)(5) + (4)(-3) + (-6)(1) = 8$$

$$D(2,3) = (2)(-9) + (3)(8) + (-1)(7) + (-5)(6) + (4)(4) + (3)(-2) + (4)(5) + (-6)(-3) + (7)(1) = 24$$

$$D(2,4) = (3)(-9) + (-1)(8) + (4)(6) + (3)(4) + (-6)(5) + (7)(-3) = -50$$

$$D(3,1) = (6)(8) + (-5)(7) + (-2)(4) + (4)(-2) + (5)(-3) + (0)(1) = -18$$

$$D(3,2) = (6)(-9) + (-5)(8) + (4)(7) + (2)(6) + (4)(4) + (-6)(-2) + (5)(5) + (0)(-3) + (1)(1) = -24$$

$$D(3,3) = (-5)(-9) + (4)(8) + (3)(7) + (4)(6) + (-6)(4) + (7)(-2) + (0)(5) + (1)(-3) + (-3)(1) = 78$$

$$D(3,4) = (4)(-9) + (3)(8) + (-6)(6) + (7)(4) + (1)(5) + (-3)(-3) = -6$$

$$D(4,1) = (-2)(8) + (4)(7) + (5)(4) + (0)(2) = 32$$

$$D(4,2) = (-2)(-9) + (4)(8) + (-6)(7) + (5)(6) + (0)(4) + (1)(-2) = 36$$

$$D(4,3) = (4)(-9) + (-6)(8) + (7)(7) + (0)(6) + (1)(4) + (-3)(-2) = -25$$

$$D(4,4) = (-6)(-9) + (7)(8) + (1)(6) + (-3)(4) = 104$$

2.4 การคืนสภาพของภาพ (Image Restoration)

ในหัวข้อนี้จะกล่าวถึงการที่ใช้การประมวลผลภาพดิจิทัลในการคืนสภาพของภาพ หรือการปรับปรุงภาพในบางหน้าที่ถูกกำหนดไว้แล้ว โดยการคืนสภาพของภาพเป็นความพยายามที่จะกู้รูปภาพให้กลับคืนสภาพเดิมหลังจากที่ถูกทำให้เสื่อมสภาพ โดยใช้ความรู้เบื้องต้นของปรากฏการณ์การเสื่อม ซึ่งความเสื่อมที่จะกล่าวถึงในบทนี้คือ ความเสื่อมที่เกิดจากฟังก์ชันความเสื่อม สัญญาณรบกวน และความบิดเบี้ยวของภาพ ตัวกรองที่กล่าวถึงในหัวข้อนี้คือการใช้ตัวกรองพหุขัณฑ์ -

(Spatial filter) ในหลายรูปแบบ และในท้ายที่สุดจะเป็นการกล่าวถึงการใช้กระบวนการเรขาคณิตในการคืนสภาพของภาพที่ถูกบิดเบี้ยวในกระบวนการการจัดเก็บภาพ

2.4.1 ตัวกรองเชิงพื้นที่

ตัวกรองเชิงพื้นที่ (Spatial Filtering) โดยปกติรูปที่มีสัญญาณรบกวนจะเป็นในลักษณะที่ค่าสีเทาในแต่ละตำแหน่งถูกเพิ่มสัญญาณรบกวนเข้าไป นั่นคือสมการที่ 2.13

$$f(x, y) = f_{original}(x, y) + n(x, y) \dots (2.13)$$

โดยที่ $f_{original}$ เป็นภาพตั้งต้นที่ปราศจากสัญญาณรบกวน ในขณะที่ f เป็นภาพที่มีสัญญาณรบกวน n เป็นรูปของสัญญาณรบกวน ซึ่งโดยปกติแล้วตำแหน่งที่มีสัญญาณรบกวนเพิ่มเข้าไปจะเป็นตำแหน่งที่มีการเปลี่ยนแปลงค่าสีเทาสูง ซึ่งเทียบเท่ากับว่าสัญญาณรบกวนเหล่านี้จะอยู่ในช่วงความถี่สูงนั่นเอง ดังนั้นในหัวข้อนี้จะกล่าวถึง การจัดการกับสัญญาณรบกวน n โดยใช้ตัวกรองเชิงพื้นที่ เช่น ตัวกรองเฉลี่ยได้เอาตัวกรองแบบเกาส์เซียนมาใช้ ตัวกรองสถิติเชิงอันดับ ได้เอาตัวกรองค่ามัธยฐานมาใช้ ซึ่งตัวกรองเหล่านี้เทียบเท่ากับตัวกรองความถี่ต่ำในโดเมนความถี่นั่นเอง ดังนั้นโดยปกติผลลัพธ์ที่ได้จากตัวกรองเชิงพื้นที่นี้ นอกจากเป็นการลดสัญญาณรบกวนแล้ว ภาพที่ได้ยังมีลักษณะเป็นภาพเบลออีกด้วย

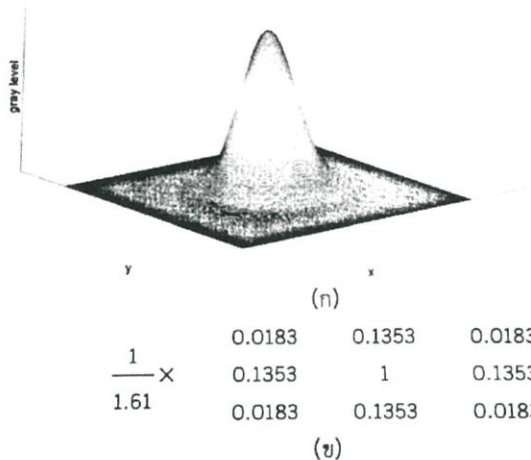
2.4.1.1 ตัวกรองค่าเฉลี่ย

ในการคำนวณหาภาพผลลัพธ์ g จากภาพอินพุต f กำหนดให้ S_{xy} เป็นเซตของพิกัดที่อยู่ในกรอบสี่เหลี่ยมของรูปย่อยที่มีจุดศูนย์กลางอยู่ที่ (x, y) และมีขนาดเป็น $m \times n$ และตัวกรองค่าเฉลี่ยมีหลายรูปแบบแต่จะขอพูดถึงแบบที่จะใช้ทำโครงการคือตัวกรองแบบบถ่วงน้ำหนัก คือ

1) ตัวกรองความถี่เกาส์เซียน เป็นตัวกรองที่มีค่าสูงสุดอยู่ที่จุดศูนย์กลางของฟังก์ชันและมีค่าลดลงทั้ง 2 แกน ดังแสดงในรูป 2.16 (ก) ซึ่งปกติสมการที่ใช้ในการคำนวณหาเกาส์เซียนมาสก์ (Gaussian mask) จะเป็นไปตามสมการ 2.13

$$m(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \dots (2.13)$$

ในการหาผลลัพธ์ของตัวกรองถ่วงน้ำหนัก เป็นการทำคอนโวลูชันระหว่างภาพอินพุต และมาสก์ของแต่ละแบบนั่นเอง



รูปที่ 2.16 (ก) ตัวกรองความถี่ต่ำเกาส์เซียน

(ข) ม้า็กสำหรับตัวกรองความถี่ต่ำเกาส์เซียน

ขนาด 3×3

2.4.1.2 ตัวกรองสถิติเชิงอันดับ

ตัวกรองสถิติเชิงอันดับ (Order Statistic Filter) เป็นตัวกรองเชิงพื้นที่ที่ไม่เป็นเชิงเส้น ซึ่งการตอบสนองของตัวกรองชนิดนี้ จะขึ้นอยู่กับค่าเรียงอันดับของค่าระดับสีเทาของพิกเซลที่อยู่ตรงกับม้า็กของตัวกรอง โดยที่ค่าระดับสีเทาที่เอาต์พุตพิกเซลที่อยู่ตรงกับจุดกึ่งกลางของม้า็ก จะเป็นค่าที่ได้จากผลการคำนวณหลังจากเรียงลำดับแล้วนั่นเอง ตัวอย่างของตัวกรองชนิดนี้มีดังนี้

1) **ตัวกรองมัชฌมาน** เป็นตัวกรองที่ต้องนำค่าระดับสีเทาของพิกเซลมาเรียงและหาค่ามัชฌมาน เพื่อเป็นค่าระดับสีเทาของเอาต์พุตพิกเซลที่อยู่ตรงกับตำแหน่งกึ่งกลางของม้า็ก ดังสมการที่ 2.14 ให้ภาพเอาต์พุตเป็น g ภาพอินพุตเป็น f และ S_{xy} เป็นเซตของพิกัดที่อยู่ในกรอบสี่เหลี่ยมของรูปย่อยหรือม้า็กที่มีจุดศูนย์กลางอยู่ที่ (x,y) ตัวแปรเหล่านี้นำมาใช้กับทุกสมการในหัวข้อนี้ด้วย เอาต์พุตจากตัวกรองมัชฌมานคือ

$$g(x,y) = \text{median}_{(s,t) \in S_{xy}} \{f(s,t)\} \quad \dots (2.14)$$

โดยปกติตัวกรองชนิดนี้ถูกนำไปใช้การทำให้ภาพมีความเรียบขึ้น แต่ว่าจะมีความเบลอน้อยกว่าตัวกรองที่เป็นเชิงเส้น ที่มีขนาดของม้า็กเท่ากัน

2.5 ฮิสโทแกรมของภาพ

ฮิสโทแกรมคือกราฟเส้นที่แสดงจำนวนพิกเซลทั้งหมดของข้อมูลภาพ ตามค่าระดับความเข้มเทาที่ปรากฏหรือแสดงอยู่บนภาพดิจิทัลใดๆ โดยค่าการกระจายของข้อมูลภาพจะเรียงจากความเข้มเทาต่ำสุดไปจนถึงระดับความเข้มเทาสูงสุด ตามตัวอย่างดังรูปที่ 2.17



รูปที่ 2.17 ภาพและฮิสโทแกรมของภาพ

2.6 การแบ่งส่วนของภาพ

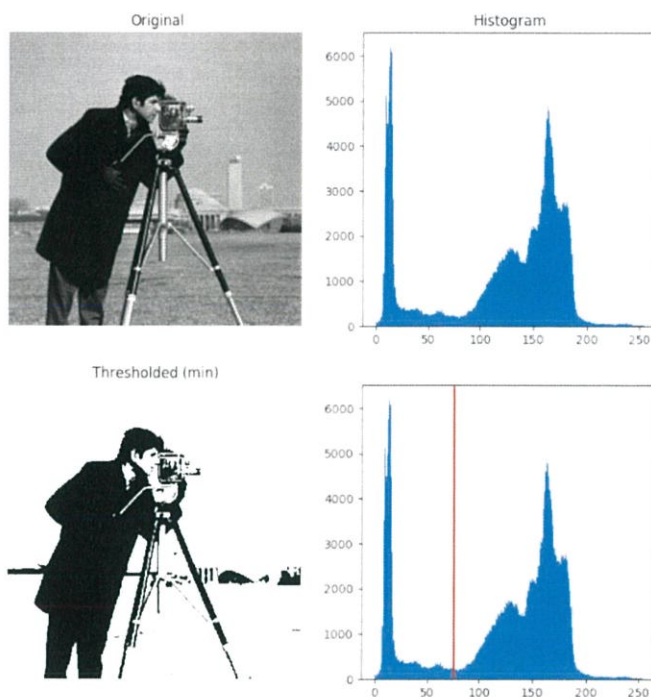
การแบ่งส่วนของภาพ (image segmentation) เป็นหนึ่งในขั้นตอนที่สำคัญสำหรับการวิเคราะห์ภาพ ซึ่งมีเป้าหมายหลักคือการแบ่งส่วนของภาพออกเป็นส่วนๆ โดยที่ในแต่ละส่วนประกอบด้วยวัตถุหรือพื้นที่ที่มีความสัมพันธ์กันอย่างชัดเจน โดยปกติในการแบ่งส่วนภาพมีด้วยกันหลายวิธี รวมถึง การแบ่งส่วนโดยใช้หลักการของการขีดแบ่ง (thresholding) ใช้หลักการของพื้นที่ (region-based) ใช้หลักการของเส้นขอบ (edge-based) หรือใช้หลักการของมอร์โฟโลยี (morphology) ซึ่งกระบวนการที่กล่าวมานี้จะกล่าวถึงในบทนี้

2.6.1 การแบ่งส่วนภาพโดยการขีดแบ่ง

การแบ่งส่วนภาพด้วยการขีดแบ่ง (thresholding) นั้นจะใช้ค่าขีดแบ่ง ค่าเดียวจะถูกใช้สำหรับพิจารณาทุกพิกเซล ซึ่งอาจจะมีฮิสโทแกรมดังรูปที่ 2.18 และภาพผลลัพธ์ (g) ที่ได้หลังจากการขีดแบ่งสามารถหาได้จากสมการ 2.15

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T_1 \\ 0 & \text{if } f(x, y) \leq T_1 \end{cases} \dots (2.15)$$

จากสมการ $T_1 = \text{Threshold value}$



รูปที่ 2.18 ตัวอย่างการแบ่งด้วยการขีดแบ่งด้วยค่า Threshold (ค่าขีดแบ่ง) = 78

2.6.2 การประมวลผลภาพด้วยมอร์โฟโลยีพื้นฐาน

การแบ่งส่วนของภาพนั้น ในบางครั้งจะถูกนำไปใช้ในการเป็นตัวแทน และรายละเอียดประกอบของรูปร่างพื้นที่ ซึ่งการแบ่งส่วนของภาพนี้รวมถึงการหาขอบ โครงร่าง และเปลือกหุ้มคอนเวกซ์ และกระบวนการหนึ่งที่เป็นส่วนหนึ่งของการแบ่งส่วนภาพเหล่านี้คือ มอร์โฟโลยีคณิตศาสตร์นี้มีทั้งที่เป็น ไบนารีและระดับเทา แต่ในหัวข้อนี้จะขอกกล่าวเฉพาะไบนารีเท่านั้น

ไบนารีมอร์โฟโลยีจะเป็นกระบวนการที่เกี่ยวข้องกับเซตที่สมาชิกของเซตแต่ละตัวเป็นเวกเตอร์ 2 มิติ และกระบวนการหลักที่เป็นพื้นฐานคือ การขยายขนาด การกร่อน การเปิดและปิด กระบวนการเหล่านี้จะถูกนำไปใช้ในการแปลงฮิตหรือมิส การสกัดขอบ และการเติมช่องว่าง จะขอกกล่าวถึงที่ใช้ในโครงงานนี้ คือ การกร่อน

2.6.2.1 การกร่อน

$$A - B = \{x \in E^2 | x + b \in A \text{ for every } b \in B\} \quad \dots (2.16)$$

ดังตัวอย่างการกร่อนของรูปที่ 2.19

| | | | | |
|-----|---|---|---|---|
| ● 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

(ก)

| | |
|-----|---|
| ● 1 | 1 |
|-----|---|

(ข)

| | | | | |
|-----|---|---|---|---|
| ● 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

(ค)

รูปที่ 2.19 (ก) A (ข) B (ค) A-B

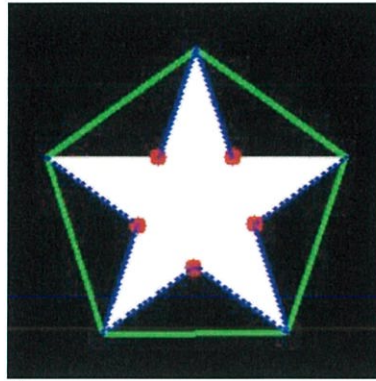
$A = \{(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\}$ B เป็นมาสก์ จะได้ว่าผลลัพธ์คือ $A-B = \{(1,1), (1,2), (2,2)\}$

2.7 ฟังก์ชันเพิ่มเติมใน opencv และ numpy

2.7.1 Contours และอัลกอริทึมที่ใช้

จะหาคอนทัวร์หรือเส้นขอบแต่ภาพต้องเป็นภาพไบนารีหรือขาวดำ ฟังก์ชันนี้ในไพทอนจะส่งค่ากลับมา 3 ค่า แต่ที่ใช้ในโครงการนี้จะพูดถึงแค่เส้นขอบหรือคอนทัวร์ ผลลัพธ์คอนทัวร์ในฟังก์ชันที่ได้จะเป็นลิสต์ของอาร์เรย์สองมิติของจุดโค-ออดิเนต (x,y) ของเส้นขอบที่เรียงต่อกันเป็นเส้น เมื่อเราเอามาแสดงจะได้ ดังรูปที่ 2.20 โดยมีจุดสีน้ำเงินรอบๆ ดาว

เนื่องจากคอนทัวร์เป็นลิสต์ของอาร์เรย์สองมิติ ดังนั้น ในฟังก์ชันที่ 2.7.2 อาร์กิวเมนต์ตัวแรก คือ contours[0] ก็คือเส้นขอบ (จุดสีน้ำเงินที่เรียงต่อกัน) ดังนั้นจะต้องมีหลายจุด



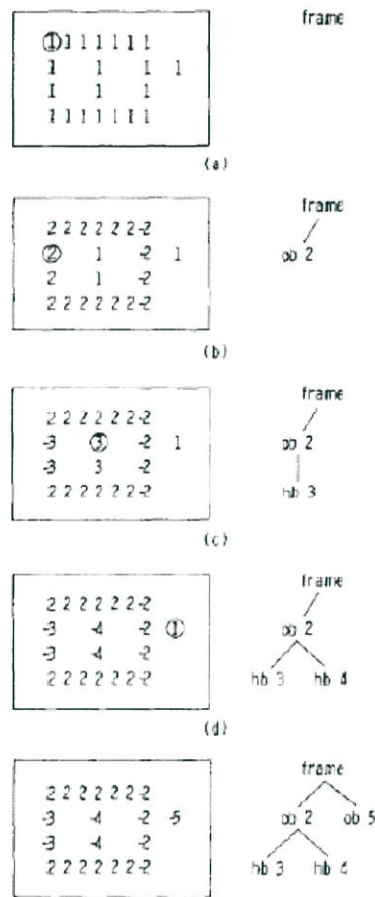
รูปที่ 2.20 ตัวอย่างการใช้ findContours

2.7.1.1 การวิเคราะห์โครงสร้างเชิงโครงสร้างของภาพไบนารีโดยใช้เส้นขอบอัลกอริทึม

ฟังก์ชันในไลบรารี opencv ใช้อัลกอริทึม Border following algorithm for topological analysis อัลกอริทึมนี้จะแยกเส้นขอบที่มีความสัมพันธ์กัน อย่างในไฟซอน ยกตัวอย่างเป็นแบบ Tree แบบ External ไปในภาพไบนารี(ขาว-ดำ) อัลกอริทึมมีดังนี้

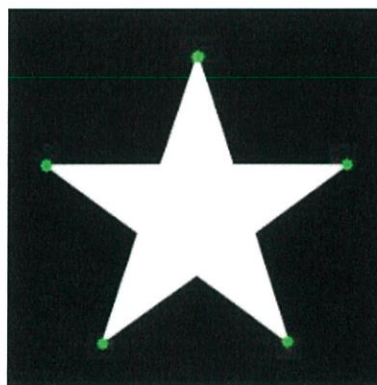
อัลกอริทึม Border following algorithm for Topological analysis

สแกนอินพุตทีละบรรทัดในรูป 2.21(a) และหยุดเมื่อเข้าเงื่อนไขคือพิกเซล (i,j) ถูกค้นพบว่าจุดเริ่มต้นเป็นได้ทั้งเส้นขอบนอกและโฮล เราสามารถระบุได้ว่าจุดนี้เป็นจุดเริ่มต้นของเส้นขอบ เราจะกำหนดให้เป็นเลข 2 เพราะ 1 ถูกกันเอาไว้ให้เส้นขอบนอกของรูปนั้น ต่อไปสแกนอีกรอบ คราวนี้จะถูกเปลี่ยนจาก 2 หรือ -2 ค่าที่เป็นบวกจะอยู่ทางด้านซ้าย และค่าที่เป็นลบจะอยู่ทางด้านขวาตามรูป 2.21(b) ต่อไปสแกนต่อ จุดวงกลมคือจุดที่เริ่มสแกนต่อ เราจะเห็นว่าเงื่อนไขโฮลได้ปรากฏ โฮลนั้นมีค่าติดลบ แต่ว่าจุดสแกนต่ออันนี้มีค่าติดลบไม่สามารถเอามาคิดได้ ดังนั้นเส้นที่เป็น parent จึงเอามาคิดก็คือเส้นตรงกลาง เรากำหนดให้เป็นเลข 3 (ก่อนหน้านั้นคือ 2) จะเห็นว่าจะมี 2 พิกเซลด้านซ้ายและตรงที่สแกนอยู่ ด้านซ้ายจะถูกเปลี่ยนจาก 2 เป็น -3 และที่สแกนจะถูกเปลี่ยนจาก 1 เป็น 3 จะทำขั้นตอนแบบนี้ไปเรื่อยๆจนจบที่รูปภาพ 2.21(c)



รูปที่ 2.21 ขั้นตอนของ Border following algorithm for topological analysis

2.7.2 Convex Hull และอัลกอริทึมที่ใช้

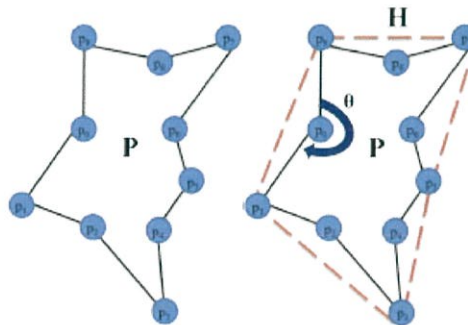


รูปที่ 2.22 ตัวอย่างฟังก์ชัน convexhull เมื่อ returnPoints เป็น True

2.7.2.2 อัลกอริทึม Sklansky's

พื้นที่หลายเหลี่ยมที่เล็กที่สุด P และจุดมุมแหลมของพื้นที่หลายเหลี่ยมนั้นถูกนิยามให้เป็น $N = \{p_0, p_1, p_2, \dots, p_n\}$ แต่ละจุดจะเรียกว่า hull

รูปภาพ 2.31 convex hull ถูกสร้างโดยการนำมุมภายใน(ข้างใน) θ ของจุดที่ติดกัน อย่างเช่น (p_1, p_0, p_9) ถ้า $\theta > \pi$ แล้ว p_0 จะถูกจัดให้เป็น reflex point หรือไม่เอาจุด p_0 เป็นจุด convex hull จะทำอย่างนี้ไปเรื่อยๆ ดังนั้นผลลัพธ์สุดท้ายที่จะได้เป็น convex hull คือ $H = \{p_1, p_9, p_7, p_5, p_3\}$



รูปที่ 2.23 Sklansky's square

The Sklansky อัลกอริทึม เป็นอัลกอริทึมลำดับเวลาเชิงเส้นใช้ในการคำนวณ convex hull สำหรับเซตของจุดมุมแหลมของรูปหลายเหลี่ยม

Sklansky อัลกอริทึมจะหาจุดที่เว้ามากที่สุด (p) เพื่อลบออกไปในทิศทางทวนเข็มนาฬิกา

Sklansky ได้กำหนดค่า P ไว้ว่า

- เป็นลำดับของจุดมุมแหลมของรูปหลายเหลี่ยม หรือ
- เป็นขอบเขตของรูปหลายเหลี่ยมปิด

อัลกอริทึม Sklansky'72

1. ให้ $P = \{p_0, p_1, p_2, \dots, p_N\}$ เป็นจุดตามทิศทางเข็มนาฬิกาของรูปหลายเหลี่ยม N
2. กำหนด p_L, p_R, p_T และ p_B เป็นจุดที่อยู่ซ้ายสุด ขวาสุด บนสุด และล่างสุดตามลำดับ
ถ้า p_L ถูกใช้ และสมมติให้ $p_L = p'_k$
(ก) ให้ $p_1 = p'_k$ และ
(ข) $p_2 = p'_{k+1}, \dots, p_N = p'_{m+k-1 \text{ mod } m}$
3. ให้ (x_i, y_i) ถูกกำหนดให้เป็น cartesian coordinate ของ p_i
4. คำนวณ $S_i \triangleq (x_{i+1} - x_{i-1})(y_{i-1} - y_i) + (y_{i+1} - y_{i-1})(x_i - x_{i-1})$
(ก) p_i เป็นจุดมุมแหลมที่ $S_i < 0$

- (จ) ถ้า $S_i \geq 0$ แล้วให้นำ p_i ออกจากเซต P
5. ขั้นตอนจะหยุดก็ต่อเมื่อ p_i ถูกนำกลับมาอีกครั้ง
6. ผลลัพธ์สุดท้ายของการขั้นตอนการนำออกจากเซต H คือเซต M

แต่ถูกปรับปรุงใหม่ เนื่องจากพอเจอรูปที่ซับซ้อน ผลที่ได้คือไม่สำเร็จ ก็เลยถูกปรับปรุงใหม่เป็น อัลกอริทึม Sklansky'82

อัลกอริทึม Sklansky'82

1. ให้ $P = \{p_0, p_1, p_2, \dots, p_N\}$ เป็นจุดตามทิศทางวนเข็มนาฬิกาของรูปหลายเหลี่ยม N
2. หา p_L, p_R, p_T และ p_B เป็นจุดที่อยู่ซ้ายสุด ขวาสุด บนสุด และล่างสุดตามลำดับแล้ว จัดลำดับ ให้จุดอยู่ในทิศทางวนเข็มนาฬิกา
3. สร้างสี่เหลี่ยมให้จุดที่กล่าวมาอยู่ในสี่เหลี่ยมนั้น เสมือนแบ่งพื้นที่สามเหลี่ยม R_1, R_2, R_3, R_4 และพื้นที่สี่เหลี่ยมตรงกลาง Q ส่วนด้านตรงข้ามมุมฉาก คือ $\overline{p_L p_B} \overline{p_B p_R} \overline{p_R p_T} \overline{p_T p_L}$
4. ให้ p_1 ถูกกำหนดโดย p_L และ ให้ p_2, p_3, \dots, p_N ถูกกำหนดโดยจุดยอดที่เหลือในเซต P จัดลำดับโดยทิศทางวนเข็มนาฬิกา
5. ให้ H เป็น convex hull ของ P และ ให้ (x_i, y_i) เป็นจุด cartesian coordinate
6. รูปที่ 2.6
 - (ก) เริ่มด้วย subspace R_1 จะเก็บ p_1 เมื่อ p_1 อยู่ใน P เงื่อนไขคือ $p_1 \in H$
 - (ข) ถ้า $p_2 \notin R_1$ เอา p_2 ออกจากเซต P
 - (ค) ถ้า $p_3 \notin R_1$ เอา p_3 ออกจากเซต P
 - (ง) ถ้า $p_3 \in H$ ในรูป 2.30
 - กำหนดหามุม tangent ของเส้นตรงที่เชื่อมจุด p_1 กับจุด p_2
 - กำหนดหามุม tangent ของเส้นตรงที่เชื่อมจุด p_1 กับจุด p_3
 - ถ้า $x_3 - x_2 \geq 0$ และ $y_3 - y_2 \leq 0$ แล้ว จุด p_3 จะอยู่ในเซต P มิฉะนั้นให้นำจุด p_3 ออก
7. สมมติ p_i และ p_{i+1} ให้อยู่ในเซต P
 - (ก) ตรวจสอบว่า $p_{i+2} \notin R_1$ ถ้าเป็นจริงให้นำ p_{i+2} ออกจากเซต P
 - (ข) แต่ถ้าข้อ ก) ไม่เป็นจริง ให้กำหนดมุม tangent ของเส้นตรงที่เชื่อมระหว่างจุด p_i กับจุด p_{i+1} และมุม tangent ของเส้นตรงที่เชื่อมระหว่างจุด p_i กับจุด p_{i+2}
 - (ค) ถ้า $x_{i+2} - x_{i+1} \geq 0$ และ $y_{i+2} - y_{i+1} < 0$ แล้วจุด p_{i+1} จะอยู่ในเซต P

มีฉะนั้น

(i) ถ้า $\tan_{i,i+1} < \tan_{i,i+2}$ หรือถ้า $\tan_{i,i+1} > \tan_{i,i+2}$, $y_{i+2} > y_{i+1}$
และจุดมุมที่เราสนใจถูกนำออกไปแล้วก่อนหน้านี้แล้วให้ลบ p_{i+2}
ทิ้งได้เลย

8. ทำซ้ำข้อ 6-7 จนกว่า p_B จะเข้าเงื่อนไข ข้อ6-7 นั้นจะนำจุด reflex ออก ทำอีกครั้ง-
ในพื้นที่ R_2, R_3, R_4
9. นำไปใช้กับอัลกอริทึมก่อนหน้านี้อีกครั้ง
10. ผลลัพธ์สุดท้ายเซต P ก็คือ Convex hull

2.7.3 Background Subtraction

วิธี Background Subtraction เป็นวิธีสำหรับแยกพื้นที่ foreground ออกจากพื้นที่ background จะบอกได้ว่าพื้นที่ foreground กับพื้นที่ background นั้นจะแยกออกโดยจับการเคลื่อนไหวในภาพ ถ้าภาพอยู่นิ่งอย่างน้อย 3 วินาทีขึ้นไปจะบ่งบอกว่าเป็นพื้นที่ background ส่วนค่าเริ่มต้นนั้น วัตถุที่กำลังเคลื่อนที่คือสิ่งที่เราสนใจ (พื้นที่ foreground)

วิดีโอของภาพที่อยู่นิ่งๆจะถูกแยกออกมาจากพื้นที่ foreground โดย background subtraction โดยจะมีสมการดังนี้

$$p_{ab} = |p_r - p_c| \quad (2.17)$$

$$p_{ab} = |p_r - p_c| \leftrightarrow p_{ab}(x, y) = \begin{cases} 0 & p_{ab}(x, y) < T \\ 255 & p_{ab}(x, y) \geq T \end{cases} \quad (2.18)$$

สมการได้ให้ p_r เป็น background ที่อ้างถึง p_c เป็นภาพปัจจุบันที่มีสิ่งที่เราสนใจ สองภาพนี้ได้คำนวณค่าสัมบูรณ์ของความแตกต่าง $|P_{ab}|$ ตามสมการที่ 2.17 ต่อไปคือนำค่าสัมบูรณ์ที่ได้นั้นมาทำการเทรซโฮลโดยฟังก์ชันของพื้นที่ background เราจะกำหนดให้มันเป็น 0 หรือสีดำ ส่วนพื้นที่สิ่งของที่เราสนใจ เราจะกำหนดให้เท่ากับ 255 หรือสีขาว ตามสมการที่ 2.18

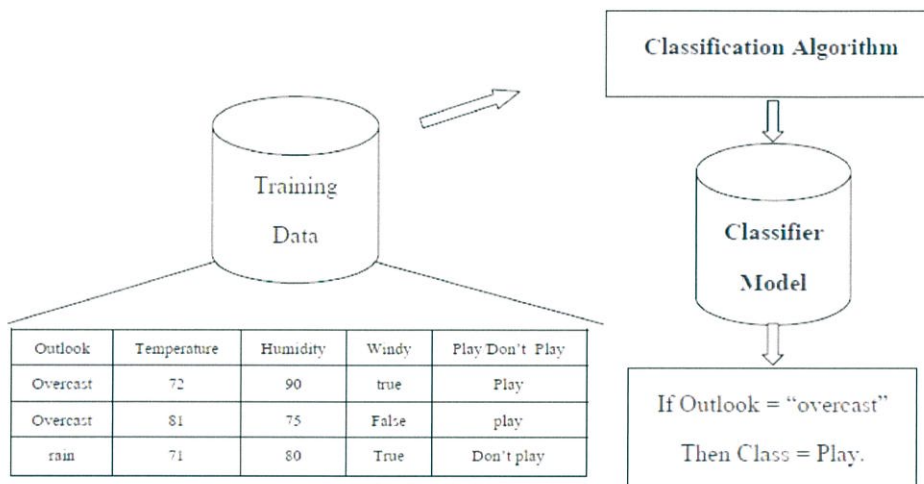
2.8 กฎการจำแนก (Classification Rules)

กฎการจำแนก เป็นกระบวนการในการจัดแบ่งข้อมูล ขึ้นอยู่กับลักษณะของวัตถุประสงค์นั้น ๆ ส่วนการจำแนกเป็นการวิเคราะห์เซตของกลุ่มข้อมูล (Data object) ที่ยังไม่จัดแบ่งประเภทเพื่อสร้างโมเดล (หรือฟังก์ชัน) ออกเป็นชุดข้อมูล (Class) ซึ่งลักษณะของคลาสถูกอธิบายโดยกลุ่มของ

คุณสมบัติ (แอตทริบิวต์) และกลุ่มของข้อมูล (Training data set) ที่ใช้ในการสร้าง โมเดลการจำแนก

ขั้นตอนการจำแนกข้อมูล มี 2 ขั้นตอนคือ

ขั้นตอนที่ 1. การสร้าง โมเดลต้นแบบ (Classifier Model) เป็นการนำชุดข้อมูล (Training Data) ผ่านกระบวนการของอัลกอริทึมการจำแนก (Classification Algorithm) ซึ่งผลลัพธ์ที่ได้จะอยู่ในรูปของโมเดลการจำแนก เช่น ต้นไม้ตัดสินใจ เป็นต้น จากนั้นสามารถสร้างเป็นกฎได้ดังภาพที่ 2.24



รูปที่ 2.24 ขั้นตอนการสร้างโมเดลการจำแนก (Classifier Model)

จากภาพที่ 2.24 เป็นตัวอย่างข้อมูลสำหรับการตัดสินใจเล่นกอล์ฟ ว่าสามารถเล่นได้ (Play) หรือเล่นไม่ได้ (Don't Play) มีรายละเอียดของข้อมูลคือ อากาศ (Outlook), อุณหภูมิ (Temperature), ความชื้น (Humidity) และ ลม (Windy) ซึ่งเมื่อผ่านอัลกอริทึมการจำแนกผลลัพธ์

ขั้นตอนที่ 2 การใช้โมเดลเพื่อการทำนาย (Prediction) ซึ่งจุดมุ่งหมายสูงสุดในการแก้ไขปัญหาคือการสร้างโมเดล เมื่อมีข้อมูลใหม่จะสามารถทำนายได้โดยการนำข้อมูลที่ได้รับทำการเปรียบเทียบกับโมเดลการจำแนก และวิเคราะห์เพื่อตัดสินใจความเป็นไปได้ของข้อมูลนั้น ๆ

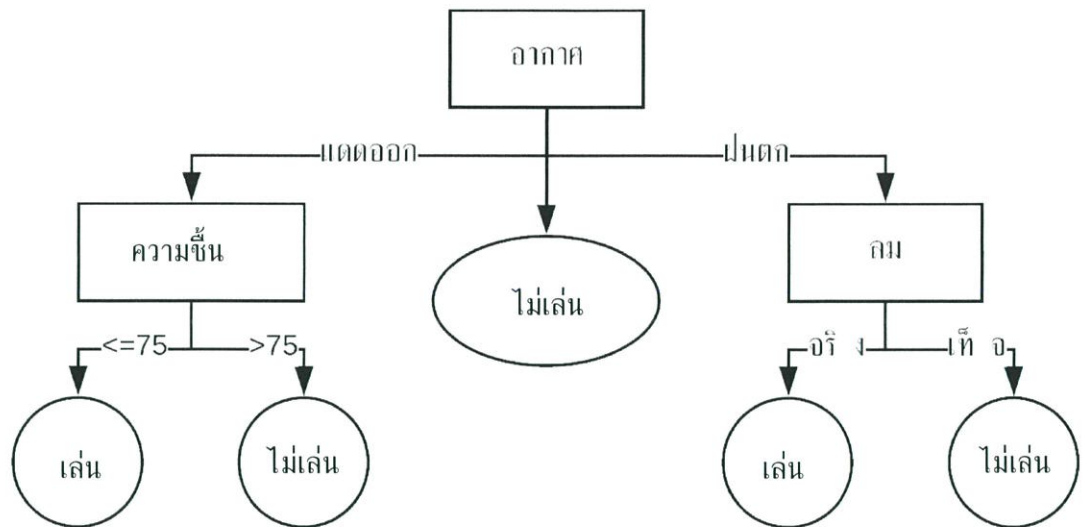
2.8.1 การสร้างต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ เป็นวิธีหนึ่งที่สำคัญในการจำแนกกฎ โดยจะมีลักษณะเป็นการทำงานเหมือนโครงสร้างต้นไม้ ที่แต่ละ โหนด (Node) แสดงคุณลักษณะ (แอตทริบิวต์) ที่ใช้ทดสอบข้อมูลแต่ละกิ่งแสดงผลในการทดสอบและลีฟ โหนด (Leaf Node) แสดงกลุ่มหรือคลาส (Class) ที่กำหนดไว้ ซึ่งต้นไม้ตัดสินใจนี้ง่ายต่อการเข้าใจและการปรับเปลี่ยนเป็นกฎการจำแนก (Classification Rules)

อัลกอริทึมพื้นฐานของการสร้างต้นไม้ตัดสินใจ คือ อัลกอริทึมละโมภ (Greedy-Algorithm) โดยจะสร้างต้นไม้จากบนลงล่างแบบวนซ้ำ (Recursive) ด้วยวิธีการแบ่งปัญหาใหญ่เป็นปัญหาย่อย (Divide-and-Conquer) ซึ่งรูปแบบของต้นไม้จะประกอบด้วยโหนดแรกสุดที่เรียกว่า Root Node จาก Root Node ก็จะแตกออกเป็น โหนดลูก และที่โหนดลูกก็จะมีลูกของตัวเองซึ่งโหนดในระดับสุดท้ายจะเรียกว่า Leaf Node

ตารางที่ 2.3 ตัวอย่างข้อมูลการตัดสินใจเล่นกอล์ฟ

| อากาศ | อุณหภูมิ | ความชื้น | ลม (มิลม) | การตัดสินใจ เล่นกอล์ฟ |
|----------|----------|----------|--------------|--------------------------|
| แดดออก | 85 | 85 | เท็จ | ไม่เล่น |
| แดดออก | 80 | 90 | จริง | ไม่เล่น |
| เมฆครึ้ม | 83 | 78 | เท็จ | เล่น |
| ฝนตก | 70 | 96 | เท็จ | เล่น |
| ฝนตก | 68 | 80 | เท็จ | เล่น |
| ฝนตก | 65 | 70 | จริง | ไม่เล่น |
| เมฆครึ้ม | 64 | 65 | จริง | เล่น |
| แดดออก | 72 | 95 | เท็จ | ไม่เล่น |
| แดดออก | 69 | 70 | เท็จ | เล่น |
| ฝนตก | 75 | 80 | เท็จ | เล่น |
| แดดออก | 75 | 70 | จริง | เล่น |
| เมฆครึ้ม | 72 | 90 | จริง | เล่น |
| เมฆครึ้ม | 81 | 75 | เท็จ | เล่น |
| ฝนตก | 71 | 80 | จริง | ไม่เล่น |



รูปที่ 2.25 ตัวอย่างต้นไม้ตัดสินใจ (Decision Tree) จากชุดข้อมูลการตัดสินใจเล่นกอล์ฟ

ต้นไม้ตัดสินใจ (Decision Tree) เป็นเทคนิคที่ค่อนข้างแพร่หลาย เนื่องจากผู้ใช้สามารถทำความเข้าใจผลลัพธ์ได้ง่าย เทคนิคต้นไม้ตัดสินใจจะจำกัดข้อมูลที่เป็นตัวแปรตาม (Dependent Variable) 1 ตัวต่อ 1 แบบจำลอง ถ้าต้องการทำนายตัวแปรตามหลายๆ ตัว จะต้องสร้างแบบจำลองสำหรับตัวแปรตามแต่ละตัว อัลกอริทึมของเทคนิคแบบต้นไม้ตัดสินใจ ส่วนใหญ่ไม่รองรับแบบต่อเนื่อง (Continuous Data) จะต้องมีการแบ่งให้เป็นข้อมูลแบบไม่ต่อเนื่อง (Discrete Data) เสียก่อน ตัวอย่างอัลกอริทึมเช่น Classification and Regression Trees (CART), ID3, C4.5 และ C5.0 เป็นต้น

2.8.1.1 อัลกอริทึม ID3

เป็นอัลกอริทึมในการสร้างต้นไม้ตัดสินใจ ที่ใช้หลักการของทฤษฎีข่าวสาร ค่าที่วัดได้เรียกว่า ค่าสารสนเทศของข้อมูล (information measure) จะนำมาใช้ตัดสินใจว่าจะใช้ตัวแปรใดในการแบ่งข้อมูล โดยวิธีการกำหนดโครงสร้างต้นไม้ตัดสินใจจะเป็นการเลือกข้อมูลตามลำดับของตัวชี้วัด [5] หรือค่ามาตรฐานเกน (Gain) สูงที่สุด เป็นข้อมูลเริ่มต้นและข้อมูลถัดไปที่มีค่าลดหลั่นกันตามลำดับ ตัวอย่างเช่น การพิจารณาจากกลุ่มข้อมูล 2 คลาส คือ P N โดยจำนวนตัวอย่างในคลาส P คือ p ตัว และจำนวนตัวอย่างในคลาส N คือ n ตัว ส่วนค่าของกลุ่มข้อมูลคือ ค่าคาดคะเนที่กลุ่มตัวอย่างต้องใช้จำนวนบิตในการแยกคลาส P และ N โดยนิยามตามสมการที่ 2.19

$$I(p, n) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right) \dots (2.19)$$

ค่าคาดคะเนของข้อมูล (Entropy) ที่แยกโดยการใช้ลักษณะประจำ A ซึ่งกำหนดไว้ว่า A คือลักษณะประจำแบ่ง S ออกเป็น $\{S_1, S_2, \dots, S_V\}$ โดยให้ S_1 มีตัวอย่างจากคลาส P จำนวน p_1 และตัวอย่างจากคลาส N จำนวน n_1 ดังสมการที่ 2.20

$$E(A) = \sum_{i=1}^V \frac{p_i+n_i}{p+n} \times I(p_i, n_i) \quad \dots (2.20)$$

ดังนั้นค่าเกนข้อมูล (Data Gain) ที่ได้จากการแยกข้อมูลด้วยลักษณะประจำ A ดังสมการที่ 2.21

$$Gain(A) = I(p, n) - E(A) \quad \dots (2.21)$$

บทที่ 3

การออกแบบการทำงาน

ในบทนี้จะแสดงถึงการออกแบบ และรายละเอียดการทำงาน ซึ่งจะแสดงให้เห็นถึง ขั้นตอนการทำงานต่างๆทั้งหมด ได้แก่ ซอฟต์แวร์ที่ใช้ ภาพโดยรวมของระบบการทำงาน การออกแบบระบบ และแผนการดำเนินงาน

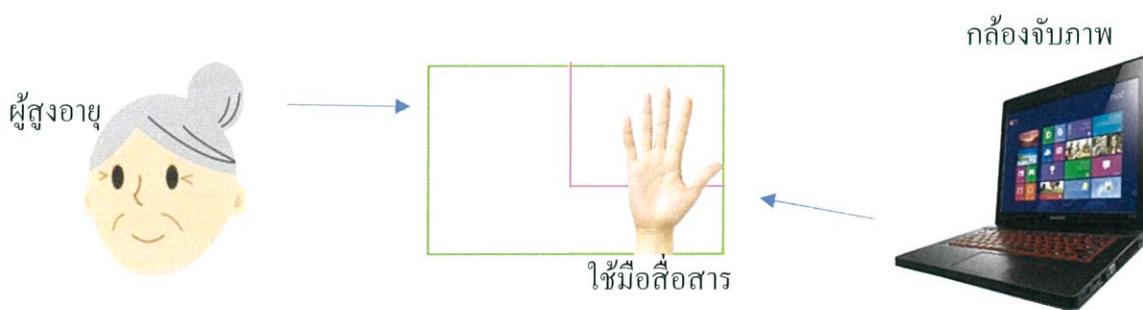
3.1 ซอฟต์แวร์ที่เกี่ยวข้องในการพัฒนา

1. python 3.5
2. opencv/numpy/matplotlib.pyplot library
3. windows 10

3.2 ฮาร์ดแวร์ที่เกี่ยวข้องในการพัฒนา

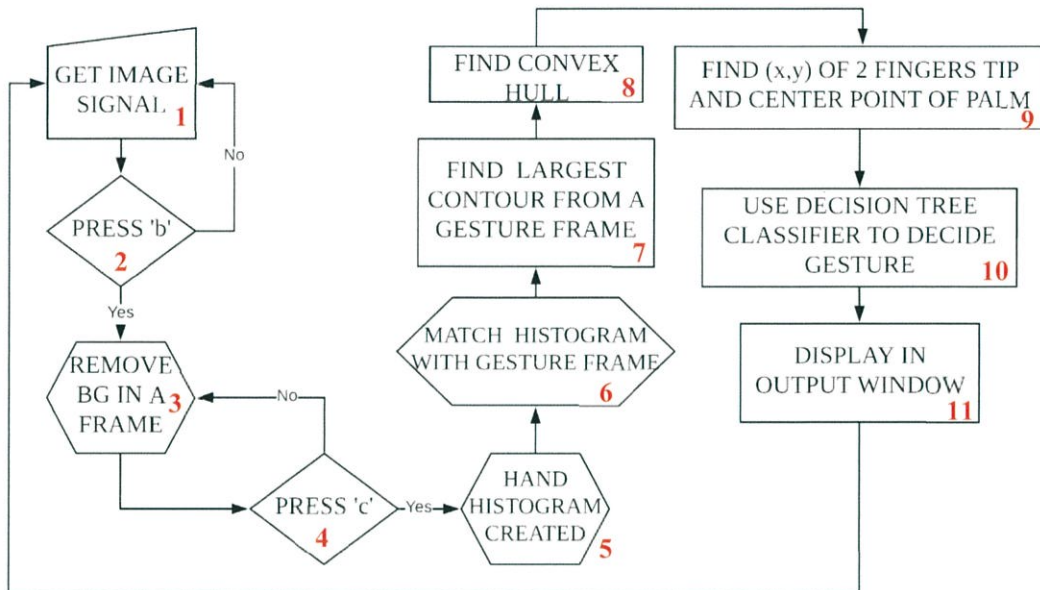
1. Notebook Lenovo Y410P NVIDIA GEFORCE RAM 8 GB
2. Notebook Asus On-board RAM 4GB

3.3 ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมของระบบ

3.4 ขั้นตอนการทำงาน



รูปที่ 3.2 ภาพของขั้นตอนการทำงาน

รับสัญญาณภาพ (1) → โปรแกรมให้ทำการกด b เพื่อบันทึก BG (2) (3) → และโปรแกรมก็จะให้เรากด c ก่อนกด c ให้เราชูมือไปตรงตำแหน่งที่เรากำหนดเพื่อทำการหาฮิสโทแกรมของสีฝ่ามือ (4) → กด c เพื่อบันทึกหาฮิสโทแกรมของฝ่ามือ (5) → Match ฮิสโทแกรม (6) → หาขอบมือ(พื้นที่ที่ใหญ่ที่สุดของเส้นขอบ) (7) → หา Convex hull (8) → หาจุดโคออดิเนต (x, y) ของจุดปลายนิ้วมือและของจุดศูนย์กลางฝ่ามือ (9) เพื่อเตรียมนำไปเตรียมเป็นแอตทริบิวต์ของ Decision Tree → เรียนรู้ Decision Tree และทดสอบเพื่อบอกท่าทาง (10) → แสดงออกมาทางหน้าจอ (11)

3.5 ออกแบบการดำเนินงาน

แอปพลิเคชันการสื่อสารด้วยท่าทางมือนี้อย่างแรกจะให้กด b เพื่อบันทึกภาพพื้นหลังหรือแบ็กกราวนด์ แล้วกด c เพื่อถ่ายรูปมือนับบันทึกข้อมูลเพื่อนำไปประมวลผล

3.5.1 ส่วนแบ็กกราวนด์

ก่อนจะทำขั้นตอนทั้งหลาย เราจะต้องใช้ backgroundsubtractor เพื่อหาจุดที่เคลื่อนไหว ดังนั้น if แรกก่อนที่เราจะเข้าไปทำอะไรต่อไป เราจะต้องใช้ backgroundsubtractor ทุกครั้ง วิธีนี้ทำให้เราสามารถลบสิ่งที่ไม่เคลื่อนไหวออกไปได้

3.5.2 ส่วนของสี่ขงมือ

กรอบชมพูเป็นเฟรมที่ต้องการ กรอบสี่เหลี่ยมเล็ก 9 ช่องเป็นที่ที่เก็บสี่ขงมือ ทำการคัดลอกภาพเอาไว้เพื่อทำส่วนขงสี่ขงมือ ดังรูป 3.3



รูปที่ 3.3 หน้าเริ่มต้นขงโปรแกรม

การเรียนรู้สี่ขงมือ และในส่วนนี้เป็นส่วนที่ไม่ต้องตัดภาพ จะเห็นว่ากรอบสี่เหลี่ยมเล็กๆ มีตัวอ่านค่าอยู่ ซึ่งไม่ต้องสนใจส่วนที่เหลือในภาพ ดังรูป 3.4



รูปที่ 3.4 แสดงตัวอย่างภาพ การเรียนรู้สี่ขงมือ

- เปลี่ยนเป็นสีภาพ เป็นสี HSV ดังรูป 3.5



รูปที่ 3.5 ตัวอย่างภาพที่ถูกเปลี่ยนเป็นสี HSV

สีของมือคือสีที่เราสนใจทั้ง 9 ช่องคือสี HSV ดังกล่าวเราจะนำไปหาฮิสโทแกรม เพื่อที่จะนำไปการ match ภาพมือ ส่วนสีของมือที่เป็นรูป HSV นี้ จะทำแค่ครั้งเดียวเท่านั้น ดังรูป 3.6



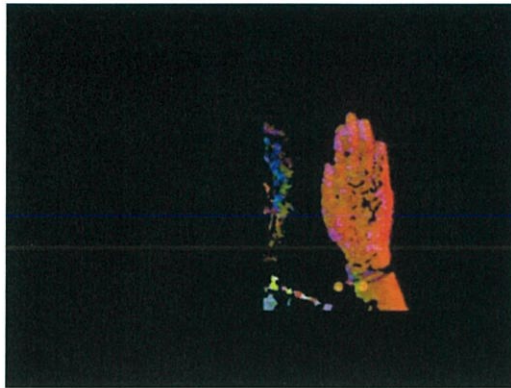
รูปที่ 3.6 ตัวอย่างอาร์เรย์ทั้ง 9 ช่องที่จะนำไปหา Histogram

3.5.3 ส่วนของมือที่เหลือในการเตรียมหาทำทาง

ส่วนนี้นำผลลัพธ์จากส่วนแบ็กกราวนด์มาเป็นเฟรมที่ใช้ จากรูปจะให้ส่วนที่ไม่ต้องการทำส่วนนั้นให้เป็นสีดำ จะเหลือส่วนตรงกรอบชมพูเอาไว้ แล้วเปลี่ยนให้เป็นสี HSV เพื่อที่จะนำเข้าสู่ calcBackProject

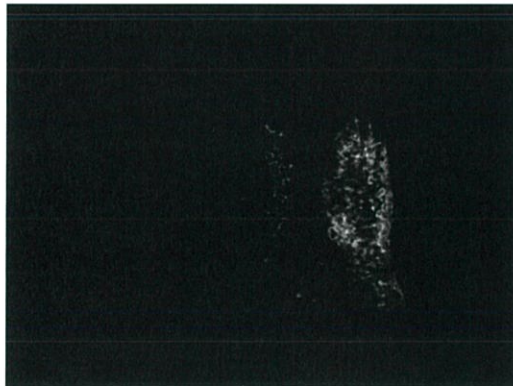


รูปที่ 3.7 ตัวอย่างผลลัพธ์จากส่วนแบ็กกราวนด์



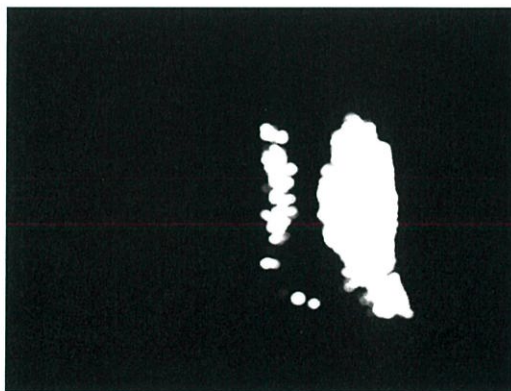
รูปที่ 3.8 ตัวอย่างการครอบตัด แล้วเปลี่ยนให้เป็นสี HSV

และใช้ฟังก์ชัน `calcBackProject` ซึ่งพารามิเตอร์ที่สำคัญคือฮิสโทแกรมของมือ



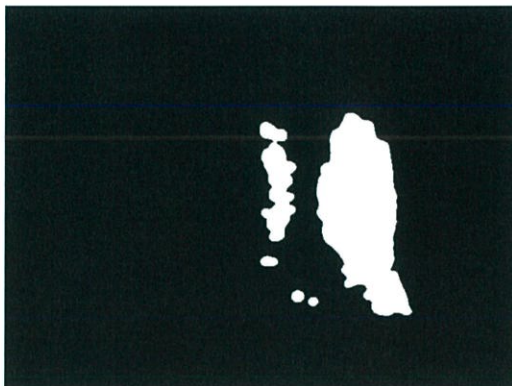
รูปที่ 3.9 ผลการจากการนำฮิสโทแกรมมา MATCH กับสีของมือที่เราต้องการ

ทำการคอนโวลูชันของส่วนบริเวณสีขาวที่ไม่ชัดเจน เนื่องจากอัลกอริทึมที่เราใช้ ยังมีส่วนที่ยังไม่ชัดเจนดังรูป 3.10



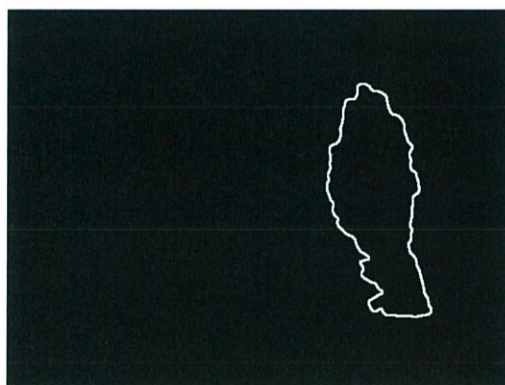
รูปที่ 3.10 ตัวอย่างการคอนโวลูชันด้วย Ellipse (13,13)

และเทรซโฮลไว้เพื่อหาเส้นขอบดังรูปที่ 3.11



รูปที่ 3.11 ตัวอย่างรูปที่เทรซโฮล

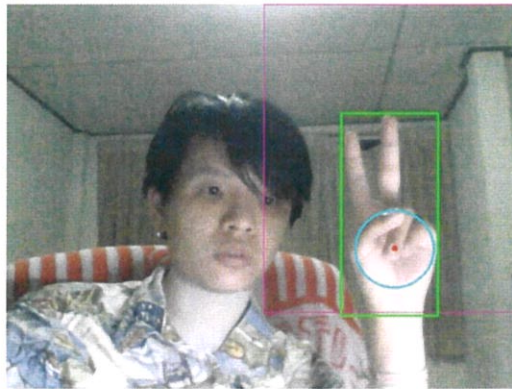
หาเส้นขอบเพื่อที่จะหา Convex Hull



รูปที่ 3.12 ตัวอย่างการหาเส้นขอบ

3.5.3.1 หาจุดศูนย์กลางมือ

หาจุดศูนย์กลางมือ โดยใช้ฟังก์ชัน `pointPolygonTest` จากนั้นหาจุดที่ห่างจากเส้นขอบมากที่สุด เพื่อประหยัดเวลาในการหา เราจะสแกนตรงกลางมือ 30%-60% เนื่องจากจุดศูนย์กลางมือไม่ได้อยู่บริเวณขอบดังรูปที่ 3.13



รูปที่ 3.13 แสดงตัวอย่างการหาจุดศูนย์กลางมือ

3.5.3.2 หา Convex Hull

หาปลายนิ้วมือโดยใช้อัลกอริทึม Convex Hull แล้วลากเส้นตรงให้ไปถึงจุดศูนย์กลางมือ ซึ่งอัลกอริทึม Convex Hull ใช้ในการหานิ้วมือนั้น ความเป็นไปได้ที่จะหาออกมา 5 นิ้วเลยแทบไม่มีดังนั้นจึงต้องเอาเทคนิคเหล่านี้มาช่วย

1. ตัดจุด Convex ที่มันใกล้กันหรือไกลกันเกินไป
2. เมื่อยังมีจำนวน Convex เกินจำนวนนิ้ว เหาระยะทางระหว่างจุด Convex กับจุดศูนย์กลางมือมา ถ้าน้อยกว่ารัศมีของฝ่ามือให้ตัดทิ้งไป

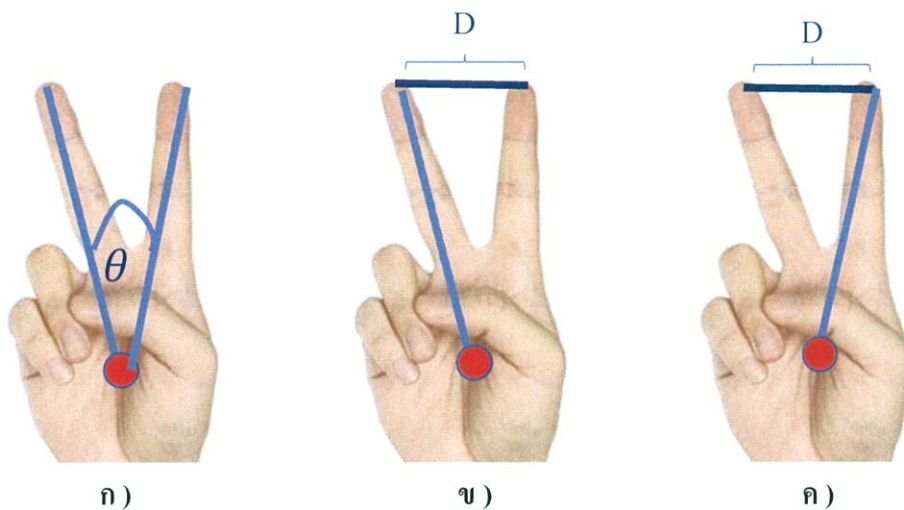


รูปที่ 3.14 แสดงตัวอย่าง Convex Hull

3.6 หาทำทางมือโดยใช้ Decision Tree Classifier

scikit-learn มี package สำเร็จรูปมาให้ใช้ Decision Tree Classifier แต่วิธีของ Decision Tree จะคัดเลือกจากแอตทริบิวต์ที่มีความสัมพันธ์กับคลาสมากที่สุดขึ้นมาเป็น โหนดบนสุดของ Tree หรือ รูด โหนด (Root node) หลังจากนั้นก็จะหาแอตทริบิวต์โดยวิธีเดิมเรื่อยๆ ในการหาความสัมพันธ์นี้ เราจะเลือกใช้ตัววัดที่เรียกว่า Information Gain ค่านี้จะคำนวณได้จาก โดยแอตทริบิวต์ที่จะใช้มี

1. มุมของนิ้ว 2. อัตราส่วนของความยาวนิ้วที่หนึ่งหารด้วยระยะ D และ 3. อัตราส่วนของความยาวนิ้วที่สองหารด้วยระยะ D



รูปที่ 3.15 แอตทริบิวต์ของ Tree

ก) มุมของนิ้วมือ

ข) อัตราส่วนระหว่างความยาวนิ้วที่หนึ่งกับระยะห่างระหว่างนิ้ว

ค) อัตราส่วนระหว่างความยาวนิ้วที่สองกับระยะห่างระหว่างนิ้ว

3.6.1 การสร้างต้นไม้ตัดสินใจด้วยอัลกอริทึม ID3

3.6.1.1 การหาค่าคาดคะเนของข้อมูล (Entropy)

การหาตัวแบ่งข้อมูลในการสร้างต้นไม้ตัดสินใจซึ่งมี 4 คุณลักษณะ (แอตทริบิวต์) นำมาพิจารณาคำตอบสำหรับการคัดเลือกทำทางในการสื่อสาร คือ มุม , อัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่หนึ่ง ($\frac{d}{l_1}$) และอัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่สอง ($\frac{d}{l_2}$) โดยมีข้อมูลทั้งหมด 80 แบบ มีท่า L ท่า V ท่าคาราบาวและท่าความรัก อย่างละ 20 แบบ รวมทั้งหมด 80 แบบ ดังนั้นค่าความสับสนเบื้องต้น คือ

$$\begin{aligned}
 I(p, n) &= I(20, 20, 20, 20) \\
 &= -\frac{20}{80} \log_2 \left(\frac{20}{80} \right) - \frac{20}{80} \log_2 \left(\frac{20}{80} \right) - \frac{20}{80} \log_2 \left(\frac{20}{80} \right) - \frac{20}{80} \log_2 \left(\frac{20}{80} \right) \\
 &= 2
 \end{aligned}$$

หาตัวแบ่งลำดับที่ 1 พิจารณาที่ละแอดทริบิวต์ ตามลำดับคือ

พิจารณา แอดทริบิวต์ “มุม” ซึ่งมีอยู่ด้วยกัน 5 กลุ่ม ดังมีข้อมูลต่อไปนี้

ตารางที่ 3.1 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “มุม”

| มุม | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|---------------------------|-------|-------|------------|------------|
| ≤ 58.668 | 0 | 20 | 0 | 17 |
| $58.668 < x \leq 70.612$ | 11 | 0 | 0 | 0 |
| $70.612 < x \leq 80.034$ | 6 | 0 | 0 | 2 |
| $80.034 < x \leq 89.623$ | 1 | 0 | 2 | 1 |
| $89.623 < x \leq 136.202$ | 0 | 0 | 18 | 0 |
| > 136.202 | 2 | 0 | 0 | 0 |

จากข้อมูลจะได้

$$\begin{aligned}
 E(\text{angle}) &= \frac{37}{80} \times I(0, 20, 0, 17) + \frac{11}{80} \times I(11, 0, 0, 0) + \frac{8}{80} \times I(6, 0, 0, 2) \\
 &\quad + \frac{4}{80} \times I(1, 0, 2, 1) + \frac{18}{80} \times I(0, 0, 18, 0) + \frac{2}{80} \times I(2, 0, 0, 0) \\
 &= 0.4603 + 0 + 0.0811 + 0.075 + 0 + 0 = 0.6164
 \end{aligned}$$

$$\text{ดังนั้น Gain (มุม)} = I(p, n) - E(\text{มุม})$$

$$= 2 - 0.6164$$

$$= 1.3836$$

ขั้นตอนที่ 2 พิจารณาแอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขานิ้วที่หนึ่ง”

ตาราง 3.2 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขานิ้วที่หนึ่ง”

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความขานิ้วที่หนึ่ง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|---|-------|-------|------------|------------|
| ≤ 0.785 | 3 | 0 | 18 | 1 |
| > 0.785 | 17 | 20 | 2 | 19 |

$$E\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความขานิ้วที่หนึ่ง}}\right) = \frac{22}{80} \times I(3,0,18,1) + \frac{58}{80} \times I(17,20,2,19)$$

$$= 0.228 + 1.264$$

$$= 1.492$$

$$\text{ดัชนี Gain}\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความขานิ้วที่หนึ่ง}}\right) = I(p,n) - E\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความขานิ้วที่หนึ่ง}}\right)$$

$$= 2 - 1.492$$

$$= 0.508$$

ขั้นตอนที่ 3 พิจารณาแอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขานิ้วที่สอง”

ตารางที่ 3.3 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความขานิ้วที่สอง”

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความขานิ้วที่สอง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|---|-------|-------|------------|------------|
| > 1.396 | 0 | 20 | 0 | 0 |
| $0.777 < x \leq 1.396$ | 8 | 0 | 0 | 19 |
| ≤ 0.777 | 12 | 0 | 20 | 1 |

$$E\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}\right) = \frac{20}{80}I(0,20,0,0) + \frac{27}{80}I(8,0,0,19) + \frac{33}{80}I(12,0,20,1)$$

$$= 0 + 0.296 + 0.4626$$

$$= 0.7586$$

$$\text{Gain}\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}\right) = I(p,n) - E\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}\right)$$

$$= 2 - 0.7586$$

$$= 1.2414$$

เมื่อพิจารณาครบทุกแอดทริบิวต์จะได้ผลลัพธ์ ดังนี้

$$\text{Gain (มุม)} = 1.3836$$

$$\text{Gain}\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}\right) = 0.508$$

$$\text{Gain}\left(\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}\right) = 1.2414$$

จะเห็นได้ว่า Gain (มุม) มีค่ามากที่สุด ซึ่งเลือกเป็นแอดทริบิวต์ในการนำมาเป็นทางเลือกตัวแรก ซึ่งต่อไปจะเป็นการหาลำดับถัดไป

ขั้นตอนที่ 4 หาดัชนีลำดับที่ 2 จากการแบ่งมุม (มุม) มี 6 ทางเลือกดังต่อไปนี้

พิจารณา มุม $\leq 58.668 \rightarrow$ มุม

ตารางที่ 3.4 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ มุม กรณีมุม ≤ 58.668

| มุม | ท่า L | ท่า V | ท่าการาบาว | ท่าความรัก |
|---------------|-------|-------|------------|------------|
| ≤ 58.668 | 0 | 20 | 0 | 17 |

จากข้อมูลจะได้

$$I(\text{มุม} \rightarrow \text{มุม}) = \frac{37}{37}I(0,20,0,17) = 0.995$$

$$\text{ดังนั้น Gain (มุม} \rightarrow \text{มุม)} = I(0,20,0,17) - I(0,20,0,17) = 0$$

- พิจารณา $\text{มุมมอง} \leq 58.668 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$

ตารางที่ 3.5 ข้อมูลสำหรับพิจารณาที่แอดทริบิวต์ “อัตราส่วนระหว่างปลายนิ้วกับความยาวนิ้วที่หนึ่ง” กรณี $\text{มุมมอง} \leq 58.668$

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| ≤ 0.785 | 0 | 0 | 0 | 0 |
| > 0.785 | 0 | 20 | 0 | 17 |

จากข้อมูลจะได้

$$E(\text{มุมมอง}) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} = \frac{37}{37} I(0,20,0,17) = 0.995$$

$$\begin{aligned} \text{ดังนั้น Gain (มุมมอง)} &\rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} = I(p,n) - E(\text{มุมมอง}) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} \\ &= 0.995 - 0.995 \\ &= 0 \end{aligned}$$

- พิจารณา $\text{มุมมอง} \leq 58.668 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$

ตารางที่ 3.6 ข้อมูลสำหรับพิจารณาที่แอดทริบิวต์ “อัตราส่วนระหว่างความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่สอง” กรณี $\text{มุมมอง} \leq 58.668$

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| > 1.396 | 0 | 20 | 0 | 0 |
| $0.777 < x \leq 1.396$ | 0 | 0 | 0 | 17 |
| ≤ 0.777 | 0 | 0 | 0 | 0 |

$$\begin{aligned} E(\text{มุมมอง}) &\rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} = \frac{0}{37} I(0,0,0,0) + \frac{20}{37} I(0,20,0,0) + \frac{17}{37} I(0,0,0,17) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{Gain (มุมมอง)} &\rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} = I(p,n) - E(\text{มุมมอง}) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} \\ &= 0.995 - 0 \\ &= 0.995 \end{aligned}$$

จาก มุม ≤ 58.668 ได้ผลลัพธ์ดังนี้

$$\text{Gain (มุมมอง} \leq 58.668) \rightarrow \text{มุมมอง} = 0$$

$$\text{Gain (มุมมอง} \leq 58.668) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} = 0$$

$$\text{Gain (มุมมอง} \leq 58.668) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} = 0.995$$

ทำให้สรุปได้ว่า มุม ≤ 58.668 จะต้องใช้ $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$ เป็นตัวแบ่งต่อไป

ขั้นตอนที่ 5 พิจารณาแอดทริบิวต์ มุม $> 58.668 \rightarrow$ มุม เพื่อหาตัวแบ่งลำดับ 3

ตารางที่ 3.7 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์มุม ในกรณีที่มุมมีค่ามากกว่า 58.668

| มุมมอง | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|------------|-------|-------|------------|------------|
| > 58.668 | 20 | 0 | 20 | 3 |

$$\begin{aligned} I(\text{มุมมอง} \rightarrow \text{มุมมอง}) &= \frac{43}{43} I(20,0,20,3) \\ &= 1.295 \end{aligned}$$

▪ พิจารณาแอดทริบิวต์ มุม $> 58.668 \rightarrow$ มุม ≤ 80.034

ตารางที่ 3.8 ข้อมูลพิจารณาที่แอดทริบิวต์มุม ในกรณีที่มุมอยู่ระหว่าง 58.668 ถึง 80.034

| มุมมอง | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--------------------------|-------|-------|------------|------------|
| $58.668 < x \leq 80.034$ | 17 | 0 | 0 | 2 |

$$\begin{aligned} E(\text{มุมมอง} > 58.668 \rightarrow \text{มุมมอง} \leq 80.034) &= \frac{19}{19} I(17,0,0,2) \\ &= 0.485 \end{aligned}$$

$$\begin{aligned} \text{Gain (มุมมอง} > 58.668 \rightarrow \text{มุมมอง} \leq 80.034) &= 1.295 - 0.485 \\ &= 0.81 \end{aligned}$$

- พิจารณาแอดทริบิวต์ มุม $> 58.668 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$

ตารางที่ 3.9 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่หนึ่ง” ในกรณีที่มีมุมมีค่ามากกว่า 58.668

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| ≤ 0.785 | 3 | 0 | 18 | 1 |
| > 0.785 | 17 | 0 | 2 | 2 |

$$E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}) = \frac{22}{43} \times I(3,0,18,1) + \frac{21}{43} \times I(17,0,2,2)$$

$$= 0.7275$$

$$\text{Gain}(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}) = I(p, n) - E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}})$$

$$= 1.295 - 0.7275$$

$$= 0.5675$$

- พิจารณาแอดทริบิวต์ มุม $> 58.668 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$

ตารางที่ 3.10 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่สอง” ในกรณีที่มีมุม > 58.668

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| > 1.396 | 0 | 0 | 0 | 0 |
| $0.777 < x \leq 1.396$ | 8 | 0 | 0 | 2 |
| ≤ 0.777 | 12 | 0 | 20 | 1 |

$$E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) = \frac{10}{43} \times I(8,0,0,2) + \frac{33}{43} \times I(12,0,20,1)$$

$$= 0.16789 + 0.8606$$

$$= 1.028$$

$$\begin{aligned} \text{Gain (มุมมอง} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) &= I(p,n) - E(\text{มุมมอง} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) \\ &= 1.295 - 1.028 \\ &= 0.267 \end{aligned}$$

จากมุมมอง > 58.668 ได้ผลลัพธ์ดังนี้

$$\text{Gain (มุมมอง} > 58.668) \rightarrow \text{มุมมอง} \leq 80.034 = 0.81$$

$$\text{Gain (มุมมอง} > 58.668) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} = 0.5675$$

$$\text{Gain (มุมมอง} > 58.668) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} = 0.267$$

ทำให้สรุปได้ว่า มุมมอง > 58.668 จะต้องใช้ มุมมอง ≤ 80.034 เป็นตัวแบ่งต่อไป

ขั้นตอนที่ 6 พิจารณาแอดทริบิวต์ มุมมอง > 80.034

ตารางที่ 3.11 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ มุมมอง > 80.034

| มุมมอง | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|----------|-------|-------|------------|------------|
| > 80.034 | 3 | 0 | 20 | 1 |

จากข้อมูลจะได้

$$I(\text{มุมมอง} \rightarrow \text{มุมมอง}) = \frac{24}{24} \times I(3,0,20,1) = 0.785$$

- พิจารณาแอดทริบิวต์ มุมมอง > 80.034 → มุมมอง ≤ 136.202

ตารางที่ 3.12 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ 80.034 < มุมมอง < 136.202

| มุมมอง | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|----------------------|-------|-------|------------|------------|
| 80.034 < x ≤ 136.202 | 1 | 0 | 20 | 1 |

จากข้อมูลจะได้

$$E(\text{มุมมอง} > 80.034 \rightarrow \text{มุมมอง} \leq 136.202) = \frac{22}{22} I(1,0,20,1) = 0.53$$

$$\begin{aligned} \text{Gain (มุมมอง} > 80.034 \rightarrow \text{มุมมอง} \leq 136.202) &= 0.785 - 0.53 \\ &= 0.255 \end{aligned}$$

- พิจารณาแอดทริบิวต์ มุม $> 80.034 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$

ตารางที่ 3.13 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่หนึ่ง” ในกรณีที่มีมุมมีค่ามากกว่า 80.034

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| ≤ 0.785 | 2 | 0 | 18 | 1 |
| > 0.785 | 1 | 0 | 2 | 0 |

$$E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}) = \frac{21}{24} \times I(2,0,18,1) + \frac{3}{24} \times I(1,0,2,0)$$

$$= 0.6325 + 0.115$$

$$= 0.7475$$

$$\text{Gain}(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}}) = I(p,n) - E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}})$$

$$= 0.785 - 0.7475$$

$$= 0.0375$$

- พิจารณาแอดทริบิวต์ มุม $> 80.034 \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$

ตารางที่ 3.14 ข้อมูลสำหรับการพิจารณาที่แอดทริบิวต์ “อัตราส่วนความยาวระหว่างปลายนิ้วกับความยาวนิ้วที่สอง” ในกรณีที่มีมุม > 80.034

| $\frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$ | ท่า L | ท่า V | ท่าคาราบาว | ท่าความรัก |
|--|-------|-------|------------|------------|
| > 1.396 | 0 | 0 | 0 | 0 |
| $0.777 < x \leq 1.396$ | 0 | 0 | 0 | 0 |
| ≤ 0.777 | 3 | 0 | 20 | 1 |

$$E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) = \frac{24}{24} \times I(3,0,20,1) \\ = 0.785$$

$$\text{Gain}(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) = I(p,n) - E(\text{มุม} \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}) \\ = 0.785 - 0.785 \\ = 0$$

จากมุม > 80.034 ได้ผลลัพธ์ดังนี้

$$\text{Gain}(\text{มุม} > 80.034) \rightarrow \text{มุม} \leq 136.202 = 0.255$$

$$\text{Gain}(\text{มุม} > 80.034) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} = 0.0375$$

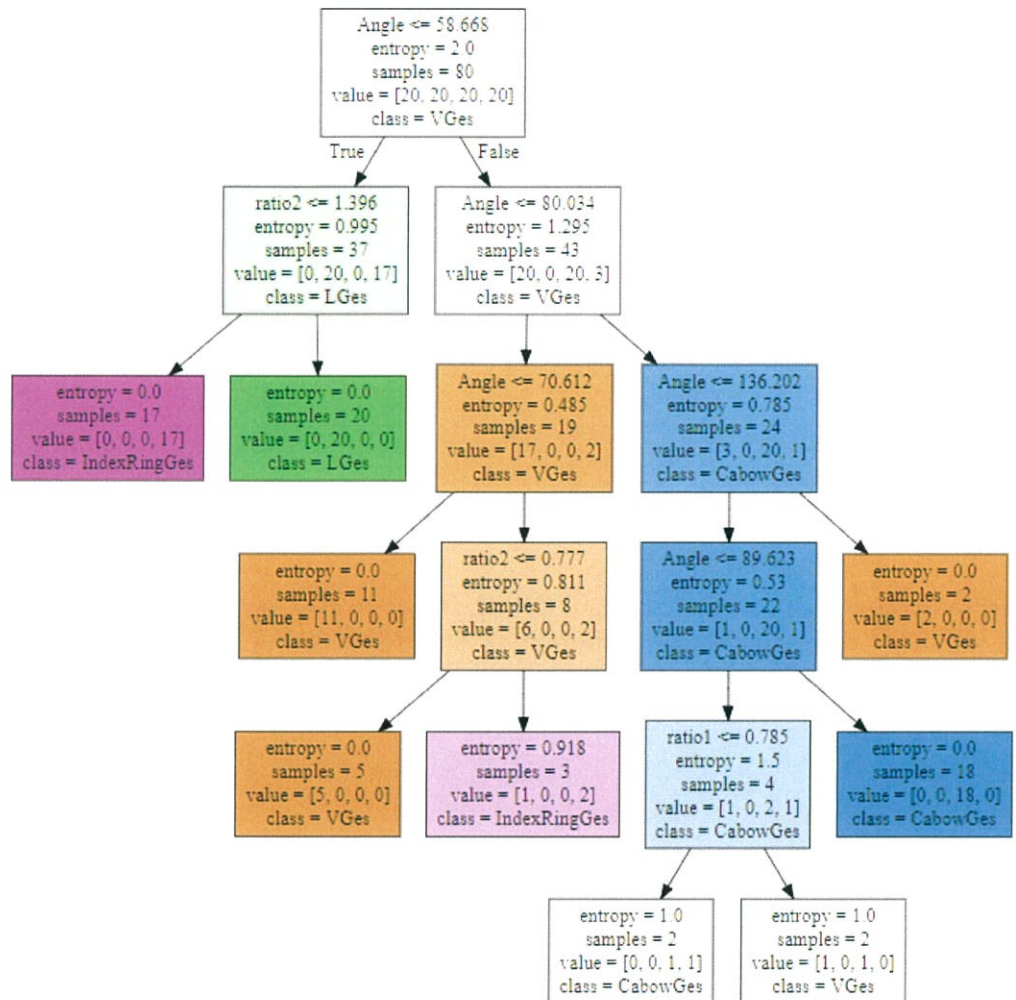
$$\text{Gain}(\text{มุม} > 80.034) \rightarrow \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}} = 0$$

ทำให้สรุปได้ว่า มุม > 80.034 จะต้องใช้ มุม ≤ 136.202 เป็นตัวแบ่งต่อไป

วิธีการนี้จะได้นั้นไม่ตัดสินใจตามรูปที่ 3.16

$$\text{angle คือมุม ratio1 คือ } \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่หนึ่ง}} \text{ และ ratio2 คือ } \frac{\text{ความยาวระหว่างปลายนิ้ว}}{\text{ความยาวนิ้วที่สอง}}$$

3.6.1.2 ต้นไม้ตัดสินใจด้วยอัลกอริทึม ID3



รูปที่ 3.16 แสดง Decision Tree

บทที่ 4

การทดลอง

4.1 การเรียนรู้

การเรียนรู้ครั้งนี้ ใช้หนึ่งคนในการเรียนรู้ ทำ 4 ท่า ท่าละ 20 ครั้ง โดยใช้ Decision Tree Classifier โดยมีแอตทริบิวต์สามแอตทริบิวต์ คือ 1.มุม 2.อัตราส่วนของความยาวกับระยะห่างนิ้วที่หนึ่ง 3. อัตราส่วนของความยาวกับระยะห่างนิ้วที่สอง ได้ผลดังนี้



รูปที่ 4.1 ตัวอย่างการเรียนรู้ ทั้ง 4 ท่า ท่า L ท่า V ท่าการขาว และท่าความรัก

144. 40745049, 0.56037807, 0.48963189, 1LGe: 20.94323171, 2.79290974, 2.66658764, 2VGes
 163. 12421292, 0.40857098, 0.59624301, 1LGe: 26.00847568, 2.27648508, 2.10511156, 2VGes
 71. 20936169, 0.7980114, 0.60681294, 1LGe: 21.45616383, 2.72946881, 2.48394847, 2VGes
 72. 69140341, 0.96091007, 0.68386288, 1LGe: 23.40332188, 2.32508959, 2.51733885, 2VGes
 79. 50794553, 0.86466612, 0.68390021, 1LGe: 25.70382484, 2.18285659, 2.28883748, 2VGes
 77. 01476327, 0.89888596, 0.68447299, 1LGe: 41.86863973, 1.39163942, 1.40686553, 2VGes
 64. 51983391, 0.88039687, 0.72923543, 1LGe: 26.16120117, 2.22408622, 2.19228638, 2VGes
 67. 3589766, 0.98507291, 0.79566111, 1LGe: 32.70613432, 1.84505946, 1.63067906, 2VGes
 68. 94979255, 0.97716196, 0.76127954, 1LGe: 40.19132458, 1.25299918, 1.5454991, 2VGes
 70. 02595821, 1.00217938, 0.67824413, 1LGe: 22.93115393, 2.54087782, 2.48122647, 2VGes
 88. 49762782, 0.81038936, 0.6075239, 1LGe: 29.79064915, 1.91921023, 1.96689169, 2VGes
 59. 05335052, 1.16245983, 0.67544673, 1LGe: 29.50757163, 2.01054026, 1.88892342, 2VGes
 64. 82083692, 0.83942689, 1.00744768, 1LGe: 30.95138431, 1.90152652, 1.83947785, 2VGes
 68. 07625096, 0.78652438, 0.97749346, 1LGe: 28.20031283, 2.04510086, 2.05930248, 2VGes
 66. 40235629, 0.78652372, 1.00804533, 1LGe: 36.77465908, 1.61748598, 1.54523774, 2VGes
 77. 22249626, 0.71805517, 0.87268381, 1LGe: 36.4964935, 1.54363384, 1.63723639, 2VGes
 63. 61427775, 0.80772706, 1.04920399, 1LGe: 30.46932524, 1.8923128, 1.91252995, 2VGes
 64. 34452858, 0.81427246, 1.03169966, 1LGe: 30.14224212, 1.88457178, 1.95293067, 2VGes
 63. 82865684, 0.79929661, 1.04924632, 1LGe: 29.71902969, 1.92799589, 1.96842329, 2VGes
 72. 08998031, 0.92325366, 0.7616395, 1LGe: 26.7613758, 2.18514952, 2.12970806, 2VGes

รูปที่ 4.2 แอดทริบิวต์ 3 แอดทริบิวต์ของท่า L และท่า V

80. 5593184, 0.84218171, 0.6947482, 3CabowGes 47. 58466018, 0.99132627, 1.35009264, 4IndexRing
 94. 60804024, 0.77255786, 0.57590431, 3CabowGes 47. 72631099, 1.02281662, 1.34164079, 4IndexRing
 95. 56021086, 0.74798441, 0.59518789, 3CabowGes 48. 55707033, 1.0004026, 1.32367193, 4IndexRing
 119. 30163852, 0.7095973, 0.43826006, 3CabowGes 46. 74632011, 1.04433782, 1.36480944, 4IndexRing
 111. 67714067, 0.65054299, 0.55628237, 3CabowGes 45. 42863873, 1.13323924, 1.38545416, 4IndexRing
 127. 92646233, 0.50763534, 0.60431261, 3CabowGes 58. 28239513, 0.85590952, 1.02245591, 4IndexRing
 127. 9962994, 0.44754492, 0.62858727, 3CabowGes 45. 87933631, 1.07922328, 1.38359526, 4IndexRing
 95. 7416009, 0.80069506, 0.52430051, 3CabowGes 71. 19850475, 0.86324468, 0.85459423, 4IndexRing
 124. 80978662, 0.49106973, 0.56919033, 3CabowGes 56. 28480275, 0.913515, 1.15713755, 4IndexRing
 95. 12585789, 0.6987088, 0.65569929, 3CabowGes 48. 92316897, 0.99356305, 1.31543717, 4IndexRing
 102. 64024128, 0.59329221, 0.68556017, 3CabowGes 48. 63994679, 1.09873962, 1.29162976, 4IndexRing
 88. 77044261, 0.7590264, 0.66755086, 3CabowGes 48. 26774787, 1.1125019, 1.29797384, 4IndexRing
 89. 7735592, 0.76862028, 0.64275019, 3CabowGes 54. 1957765, 1.08675717, 1.10817513, 4IndexRing
 90. 07813944, 0.75300664, 0.65698677, 3CabowGes 48. 52657268, 1.21274143, 1.22069744, 4IndexRing
 101. 31157358, 0.73635909, 0.54740271, 3CabowGes 56. 58305018, 0.91672975, 1.14868421, 4IndexRing
 116. 77505309, 0.6653274, 0.5047481, 3CabowGes 73. 74747235, 0.87085793, 0.7923719, 4IndexRing
 110. 36681214, 0.60692656, 0.61112066, 3CabowGes 89. 47226444, 0.72467163, 0.69580134, 4IndexRing
 111. 91761911, 0.66749051, 0.53604134, 3CabowGes 56. 42306289, 1.00468404, 1.10277867, 4IndexRing
 115. 82639003, 0.67109593, 0.50457447, 3CabowGes 57. 68094432, 1.02691603, 1.04588423, 4IndexRing
 116. 63992583, 0.62334629, 0.55089908, 3CabowGes 55. 42490968, 0.98999561, 1.14106443, 4IndexRing

รูปที่ 4.3 แอดทริบิวต์ 3 แอดทริบิวต์ของท่าการาบาว และท่าความรัก

4.2 วิเคราะห์การเรียนรู้

แอดทริบิวต์แรก มุม จะเห็นได้ชัดว่าเมื่อมุมเล็กๆ เช่น น้อยกว่า 40 จะตัดสินใจได้ทันทีว่าเป็นท่า V แต่ถ้านกที่ทำท่า V แต่มุมคั่นกว้างจะมีโอกาสเป็นท่าอื่นทันที และท่า V สังเกตได้ว่า แอดทริบิวต์ที่สอง กับแอดทริบิวต์ที่สาม อัตราส่วนความยาวกับระยะห่างจะเห็นว่ามีความมากที่สุด

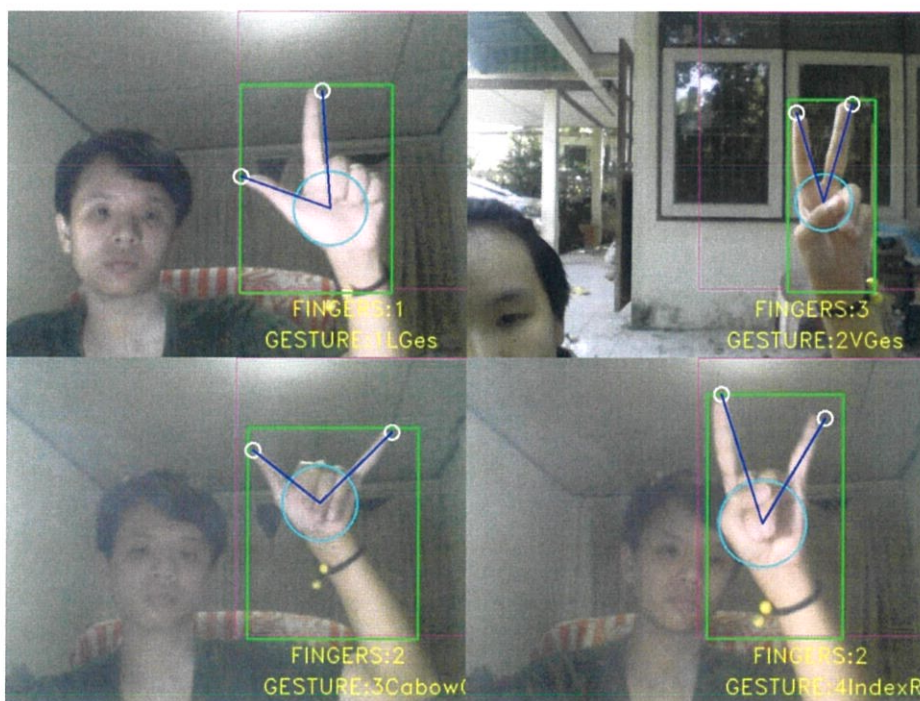
4.3 ทดสอบ

4.3.1 สมมติฐาน

ความเข้มแสงในขณะทำการทดลองอาจไม่เท่ากัน ดังนั้นจึงขอปรับขนาดของสีในขณะ Threshold ด้วยตนเอง

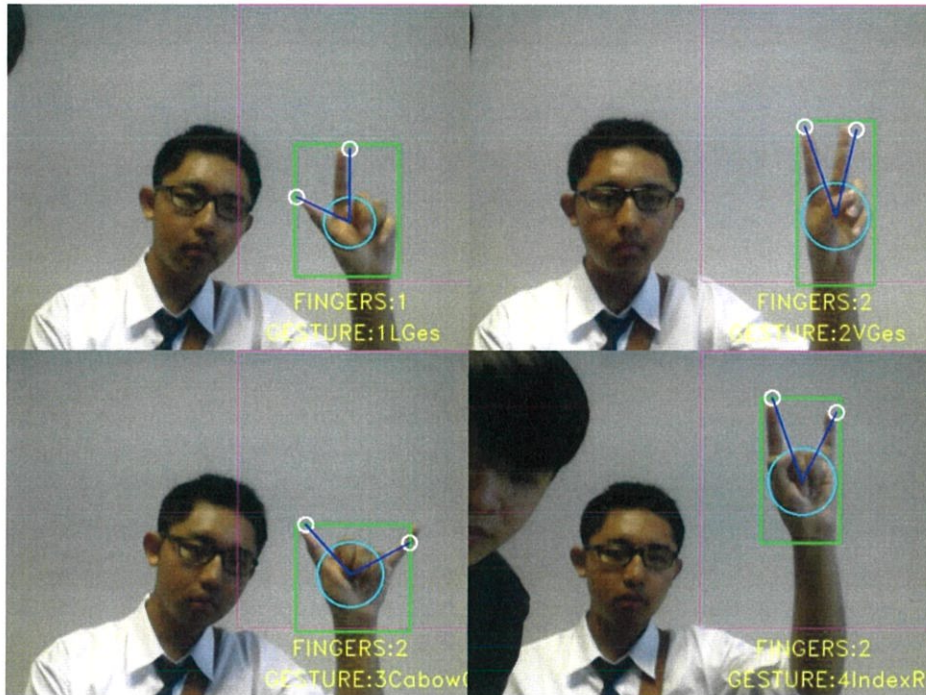
4.3.2 ผลการทดลอง

คนที่หนึ่งทดสอบ ทำละ 1 ครั้ง ถูกทุกท่า



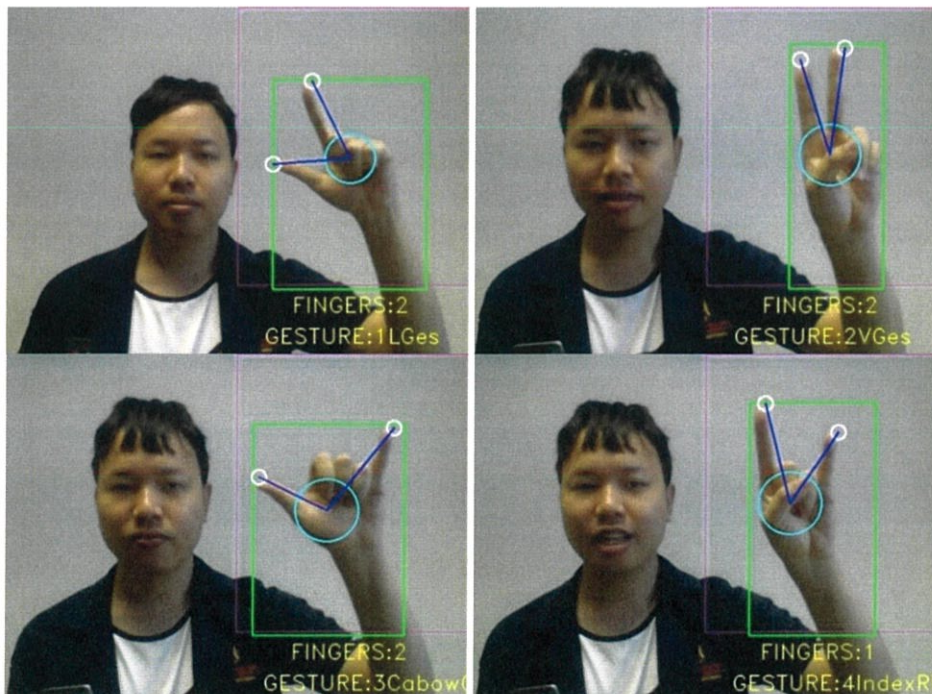
รูปที่ 4.4 ตัวอย่างการทดสอบกับคนที่หนึ่ง

คนที่สอง ทดสอบ ท่าละ 1 ครั้ง รวมทั้งหมด 4 ท่า ถูกทุกท่า



รูปที่ 4.5 ตัวอย่างการทดสอบกับคนที่สอง

คนที่สามทดสอบ ท่าละ 1 ครั้ง รวมทั้งหมด 4 ท่า ถูกทุกท่า



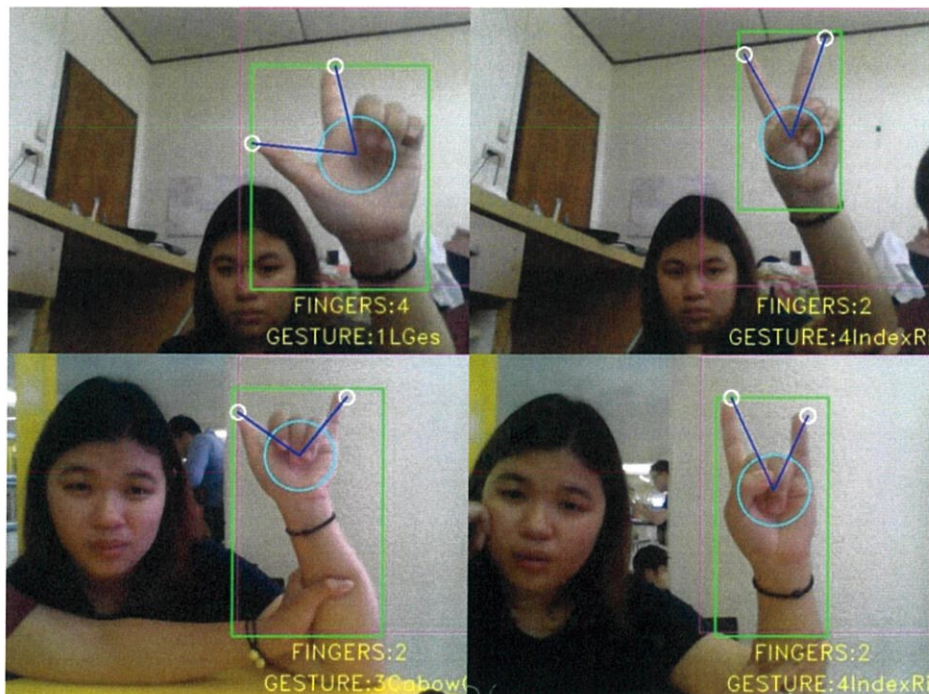
รูปที่ 4.6 ตัวอย่างการทดสอบกับคนที่สาม

คนที่สี่ทดสอบ ท่าละ 1 ครั้ง รวมทั้งหมด 4 ท่า ถูกทุกท่า



รูปที่ 4.7 ตัวอย่างการทดสอบกับคนที่สี่

คนที่ห้าทดสอบ ท่าละ 1 ครั้ง รวมทั้งหมด 4 ท่า ผิดท่าที่ 2 คือท่า V



รูปที่ 4.8 ตัวอย่างการทดสอบกับคนที่ห้า

วิเคราะห์ข้อผิดพลาดของคนที่ห้า

สังเกตว่า V ของผู้ทดลอง มุมจะกว้างเป็นพิเศษ รูปที่ 4.7 จะเห็นว่ามันใกล้เคียงกันมาก ซึ่งถ้ากลับไปดูรูปที่ 4.2 ทำ V นั้นจะเป็นมุมระหว่าง 20-40 ซึ่ง 40 ที่ การเรียนรู้ มีอยู่แค่กรณีเดียว ดังนั้นคิดว่าถ้าการที่เพิ่มคนในการเรียนรู้ นำมือหลายๆแบบมาเรียนรู้จะดีขึ้น

43,1.38096905127,1.34451781967
07_16_22_12Apr2018
VGes

49,1.04511945638,1.30581400459
09_57_16_12Apr2018
IndexRing

รูปที่ 4.9 ผลแอดทริบิวต์ทั้งสองกรณี

4.4 วิเคราะห์ผล

ปริศยานิพนธ์นี้ เบื้องหลังได้ใช้สี HSV ซึ่งความเข้มของแสงไม่มีผลต่อความแม่นยำของโปรแกรม แต่อย่างไรก็ตาม เมื่อทดลองจริงก็ได้ได้ทราบว่าความเข้มของแสงก็มีผลเล็กน้อยต่อความแม่นยำของการจับภาพ และสีบริเวณพื้นหลังมีผลอย่างมากต่อการใช้เคสเป็นปัจจัยหลักในการจับภาพมือ

4.5 สรุปผลการทดลอง

ตารางที่ 4.1 สรุปผลการทดลองการตรวจจับท่าทาง คนละ4ท่า ท่าละ 1 ครั้ง

| คนที่ | การเรียนรู้ (ครั้ง) (ท่าละ 20 ครั้ง) | ทดสอบ (ครั้ง) | L | V | CB | <3 | ความถูกต้อง (%) | หมายเหตุ |
|-------|---|------------------|---|---|----|----|-----------------|-----------------|
| 1 | 80 | 4 | √ | √ | √ | √ | 100%(4/4) | - |
| 2 | | 4 | √ | √ | √ | √ | 100%(4/4) | - |
| 3 | | 4 | √ | √ | √ | √ | 100%(4/4) | - |
| 4 | | 4 | √ | √ | √ | √ | 100%(4/4) | - |
| 5 | | 4 | √ | × | √ | √ | 75%(3/4) | นิ้วอ้วนและสั้น |

จากการ ทดสอบ ทั้ง 5 คนนั้นทำให้เห็นว่ารูปร่างมือแต่ละคนอาจจะทำให้ความแม่นยำของ โปรแกรมคลาดเคลื่อนได้ และยังรูปร่างมือไม่เป็นมาตรฐานทำให้โปรแกรมตรวจจับได้ยาก แนวทางการแก้ไข คือ เพิ่มคน การเรียนรู้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงการ

โครงการนี้เป็นโครงการที่พัฒนาใช้สำหรับให้คนสูงอายุใช้สื่อสาร ซึ่งพัฒนาโดยใช้ภาษา Python ร่วมกับไลบรารี Opencv ,numpy ซึ่งเป็นไลบรารีเกี่ยวกับการประมวลผลภาพ scikit-learn.tree ซึ่งเป็นไลบรารีสำเร็จรูปเกี่ยวกับการจำแนก (Classifier)

Opencv , numpy จะมีอัลกอริทึมหลายแบบให้เราเลือกใช้ พร้อมปรับปรุงคุณภาพด้วยเทคนิคที่กล่าวมาตอนบทต้นๆแล้ว แล้วภาพที่ เรียนรู้ ทั้งหมดก็จะไปเรียกใช้ไลบรารี scikit-learn ที่ใช้ Decision Tree Classifier มาเป็นตัวจำแนกให้เป็นแต่ละท่า

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

- 1) ในการทดสอบ บางครั้งความเข้มของแสงมีผลต่อความแม่นยำของโปรแกรม เราจึงต้องปรับแสงด้วยตนเองจะเห็นได้ว่าช้าพอควร นานกว่าจะได้แสงที่พอดี
- 2) ระบบไม่สามารถตรวจจับมือที่นิ่งๆได้ ต้องเคลื่อนไหวตลอดเวลา เพราะที่เราเลือกใช้ อัลกอริทึมที่เป็นวิดีโอ (Video Analysis)

5.3 แนวทางการพัฒนาต่อ

- 1) เพิ่มจำนวนผู้ที่เข้ามาเรียนรู้ระบบ เนื่องจากระบบของเราคน 1 คนเป็นคนเรียนรู้ ถ้ามือที่ให้จับภาพ ขนาดไม่เป็นมาตรฐานจะไม่สามารถตรวจจับท่าทางที่ต้องการถึงจะตรวจจับได้ว่าเป็นมือก็ตาม
- 2) เพิ่มระบบตรวจวัดแสงเพื่อที่เราจะไม่ต้องปรับค่าเอง
- 3) เพิ่มคอมพิวเตอร์ขนาดเล็กหลายๆตัวเพื่อที่จะได้เอาไว้ติดตามบ้านได้

บรรณานุกรม

- ผศ.ดร.สันสนีย์ เอื้อพันธ์วิริยะกุล.2555. **Digital Image Processing** .เชียงใหม่ : มหาวิทยาลัยเชียงใหม่
- รศ.ดร.กิตติ ไพฑูรย์วัฒนกิจ .2549. การประมวลผลภาพดิจิทัล **Digital Image Processing** . กรุงเทพฯ : สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- ผศ.ดร.สมเกียรติ อุดมหารธยากุล .2550. การประมวลผลภาพเบื้องต้น **Fundamental of Digital Image Processing** .กรุงเทพฯ : สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- OpenCV** . 2016 . [Online] . Available : <https://opencv.org/> .
- Python** . 2017 . [Online] . Available : <https://docs.python.org/> .
- ระบบสี **HSV** . [online] . Available : <http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm>
- Hand Gesture Recognition** . [online] . Available : <https://github.com/mahaveerverma/hand-gesture-recognition-opencv/blob/master/HandRecognition.py>
- Decision Tree** . 2014 . [Online] . Available : <http://dataminingtrend.com/2014/decision-tree-model/>
- BUILDING DECISION TREE ALGORITHM IN PYTHON WITH SCIKIT LEARN** . 2017 . [Online] . Available : <http://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/>
- COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING** . 1985 . [Online] . Available : <https://pdfs.semanticscholar.org/e9f2/1d440f9317fc8b6c294b9826b70ad5312d36.pdf>
- Menatoallah M. Youssef. 2011 . **In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical Engineering** . Dayton in Ohio : University of Dayton