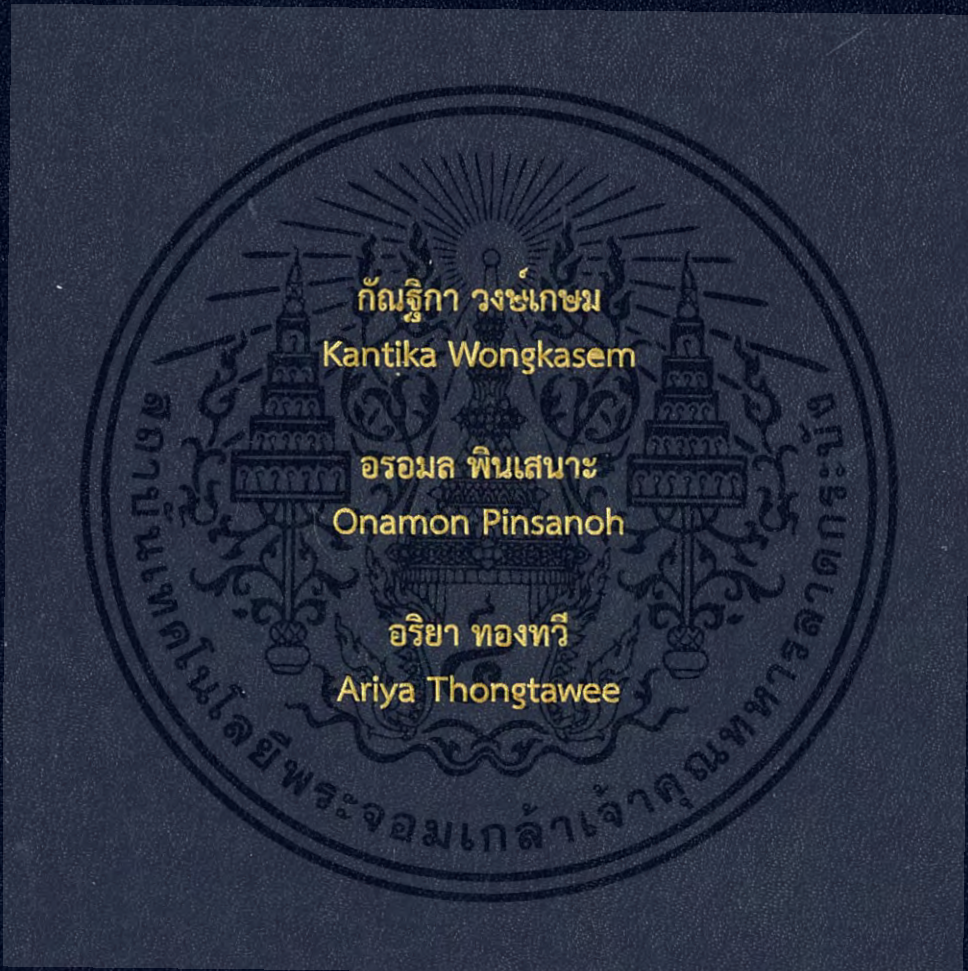


ระบบตรวจจับภาษามือ
Sign Language Detection



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบตรวจจับภาษามือ
Sign Language Detection

| | | |
|-------------------|--------------|----------|
| กัญฐิกา วงษ์เกษม | รหัสประจำตัว | 57010057 |
| Kantika Wongkasem | Student ID | 57010057 |
| อรอมล พินเสนาะ | รหัสประจำตัว | 57011504 |
| Onamon Pinsanoh | Student ID | 57011504 |
| อริยา ทองทวี | รหัสประจำตัว | 57011508 |
| Ariya Thongtawee | Student ID | 57011508 |

อาจารย์ที่ปรึกษา

ผศ.ดร.ยุทธนา คัดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2560

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับภาษามือ

Sign Language Detection

ผู้จัดทำ นางสาวกัญฉิกา วงษ์เกษม รหัสนักศึกษา 57010057

Miss Kantika Wongkasem Student ID 57010057

นางสาวอรอมล พินเสนาะ รหัสประจำตัว 57011504

Miss Onamon Pinsanoh Student ID 57011504

นางสาวอริยา ทองทวี รหัสประจำตัว 57011508

Miss Ariya Thongtawee Student ID 57011508

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

อาจารย์ที่ปรึกษา

(ผศ.ดร.ยุทธนา คิดใจเดี่ยว)

_____/_____/_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|------------------|--------------------------|--------------|----------|
| หัวข้อโครงการ | ระบบตรวจจับภาษามือ | | |
| นักศึกษา | นางสาวกัญฐิกา วงษ์เกษม | รหัสนักศึกษา | 57010057 |
| | นางสาวอรอมล พินเสนาะ | รหัสนักศึกษา | 57011504 |
| | นางสาวอริยา ทองทวี | รหัสนักศึกษา | 57011508 |
| ปริญญา | วิศวกรรมศาสตรบัณฑิต | | |
| สาขาวิชา | วิศวกรรมอิเล็กทรอนิกส์ | | |
| ปีการศึกษา | 2560 | | |
| อาจารย์ที่ปรึกษา | ผศ. ดร.ยุทธนา คิดใจเดียว | | |

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อวิเคราะห์จำแนกตัวอักษรภาษาอังกฤษของภาษามือโดยใช้วิธีทางเทคโนโลยีการประมวลผลภาพ การวิเคราะห์เพื่อจำแนกภาษามือนั้นมีความสำคัญเนื่องจากภาษามือเป็นส่วนหนึ่งของการสื่อสารและใช้ในการถ่ายทอดความหมายเมื่อเทียบกับการพูด ซึ่งจะทำให้คนปกติสามารถสื่อสารกับผู้ที่มีความผิดปกติทางการได้ยิน เครื่องมือที่ใช้วิเคราะห์จำแนกตัวอักษรภาษามือคือ โปรแกรม MATLAB (Matrix Laboratory) โดยใช้วิธีการทั้งสิ้น 4 วิธี ได้แก่ พื้นที่ของขอบภาพ 3 ด้าน, ความยาวนิ้วมือจากจุดเซ็นทรอยด์, มุมระหว่างนิ้วมือ และค่าเคลต้าของมุมระหว่างนิ้วมือของภาพเฟรมแรกและเฟรมสุดท้ายที่รับเข้ามาเพื่อวิเคราะห์ว่าเป็นตัวอักษรแบบสแตติกหรือไดนามิกจากนั้นใช้โครงข่ายประสาทเทียมในการจำแนกท่าทางของมือออกเป็นตัวอักษรต่างๆ โดยใช้เวลาในการประมวลผลเฉลี่ย 2 ตัวอักษรต่อ 1 วินาที

| | | |
|---------------|-----------------------------------|----------|
| Project Title | Sign Language Detection | |
| Student | Miss Kantika Wongkasem | 57010057 |
| | Miss Onamon Pinsanoh | 57011504 |
| | Miss Ariya Thongtawee | 57011508 |
| Degree | Bachelor of Engineering | |
| Program | Electronics Engineering | |
| Year | 2017 | |
| Advisor | Asst. Prof. Dr.Yuttana Kitjaidure | |

Abstract

This project aims to analyze the classification of the English alphabets of sign language by using image processing technology. The sign language analyze is important because it is a part of deaf communication and is used to convey meanings as voice from speech which made hearing impaired people to understand. The hardware used to classify is MATLAB (Matrix Laboratory) program by using 4 methods, which are number of white pixels at edge image, finger length from centroid point, angles between fingers and delta values of angle between fingers of the first and last frame to identify characteristics of each gesture then use the Artificial neural network to classify the gesture into 26 groups. According to the result, the average of processing time is 2 alphabets per second.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับการช่วยเหลือจากหลายฝ่าย ขอขอบพระคุณอาจารย์ที่ปรึกษาผู้ช่วยศาสตราจารย์ยุทธนา คิดใจเดียว ที่คอยให้คำแนะนำตลอดการทำงาน รวมถึงเอื้อเฟื้ออุปกรณ์ต่างๆ และสถานที่ที่ใช้ในการทำปริญญานิพนธ์ นอกจากนี้ขอขอบพระคุณอาจารย์ทุกท่านที่มอบเนื้อหาความรู้เพื่อให้สามารถนำมาประยุกต์ในการทำโครงงานชิ้นนี้ ขอขอบคุณพี่ๆ และเพื่อนๆ ที่คอยให้คำแนะนำและการช่วยเหลือตลอดการทำงาน

สุดท้ายนี้ผู้วิจัยขอกล่าวคำขอบคุณอย่างยิ่งที่ได้รับความช่วยเหลือจากทุกท่านไว้ ณ ที่นี้ด้วย



กัญฐิกา วงษ์เกษม
อรอมล พินเสนาะ
อริยา ทองทวี

สารบัญ

| | หน้า |
|---|------|
| บทคัดย่อ..... | I |
| Abstract..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญรูปภาพ..... | VII |
| สารบัญตาราง..... | IX |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ที่มาและความสำคัญ..... | 1 |
| 1.2 วัตถุประสงค์..... | 1 |
| 1.3 ขอบข่ายโครงการ..... | 2 |
| 1.4 ผลที่คาดว่าจะได้รับ..... | 2 |
| บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง..... | 3 |
| 2.1 การแบ่งข้อมูลภาพ..... | 3 |
| (Image Segmentation) | |
| 2.1.1 Amplitude segmentation methods..... | 3 |
| 2.1.2 Region segmentation methods..... | 5 |
| 2.2 กระบวนการที่เกี่ยวข้องอื่นๆ..... | 7 |
| 2.2.1 การเปลี่ยนแปลงภาพสีเป็นภาพระดับสีเทา..... | 7 |
| (Gray-Scale Image Transform) | |
| 2.2.2 การเปลี่ยนแปลงลักษณะของวัตถุ..... | 8 |
| (Morphological Operation) | |
| 2.2.3 ระยะทางแบบยุคลิด..... | 10 |
| (Euclidean distance) | |
| 2.2.4 ภาษามืออเมริกัน..... | 10 |
| (American Sign Language) | |
| 2.3 โครงข่ายประสาทเทียม..... | 10 |
| (Artificial neural network) | |

| | |
|--|----|
| 2.3.1 องค์ประกอบและโครงสร้างการทำงาน..... | 11 |
| 2.3.2 หลักการทำงาน..... | 12 |
| 2.3.3 การเรียนรู้สำหรับโครงข่ายประสาทเทียม..... | 14 |
| 2.3.4 ประโยชน์ของโครงข่ายประสาทเสมือน..... | 15 |
| บทที่ 3 วิธีการดำเนินงาน..... | 17 |
| 3.1 การรับภาพ..... | 17 |
| (Input image) | |
| 3.2 การปรับปรุงคุณภาพของภาพ..... | 17 |
| 3.2.1 ภาพระดับสีเทา..... | 17 |
| (Grayscale image) | |
| 3.2.2 ภาพขาว-ดำ..... | 18 |
| (Binary image) | |
| 3.2.3 การลบพื้นที่ขนาดเล็กกว่าที่กำหนด..... | 18 |
| (Remove small object) | |
| 3.2.4 การเติมพื้นที่หรือรูของภาพ..... | 19 |
| (Fill image regions and holes) | |
| 3.3 การระบุตำแหน่งของมือ..... | 19 |
| 3.3.1 การหาตำแหน่งของขอบมือ | 19 |
| 3.3.2 การหาวงกลมบนฝ่ามือ..... | 20 |
| 3.4 การวิเคราะห์คุณลักษณะของภาพ..... | 20 |
| 3.4.1 พื้นที่ของขอบภาพ 3 ด้าน..... | 20 |
| 3.4.2 ความยาวนิ้วมือจากจุดเซ็นทรอยด์..... | 22 |
| 3.4.3 มุมระหว่างนิ้วมือ..... | 24 |
| 3.5 การจำแนกตัวอักษรภาษามือ..... | 25 |
| บทที่ 4 ผลการทดลอง..... | 27 |
| 4.1 ผลการทดลองจากการเลือกใช้กล้อง WEBCAM..... | 27 |
| 4.2 ผลการทดลองจากการเพิ่มและลดนิ้วมือ..... | 27 |
| 4.3 ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม..... | 28 |

| | |
|--|----|
| 4.4.1 การเลือกใช้ค่า Hiddenlayer..... | 28 |
| 4.4.2 การเลือกใช้ค่า Testing และ Validation..... | 29 |
| 4.4.3 ผลการทดลองจากการเลือกค่าพารามิเตอร์และคุณลักษณะที่เหมาะสม..... | 30 |
| บทที่ 5 สรุปและวิจารณ์..... | 33 |
| 5.1 ปัญหาที่เกิดขึ้นกับโครงการ..... | 33 |
| 5.1.1 ด้านฮาร์ดแวร์..... | 33 |
| (Hardware) | |
| 5.1.2 ด้านซอฟต์แวร์..... | 33 |
| (Software) | |
| 5.2 สรุปผลการทดลอง..... | 33 |
| บรรณานุกรม..... | 34 |
| ภาคผนวก..... | 35 |



สารบัญรูปภาพ

| | |
|--|----|
| รูปที่ 2.1 แสดง Block diagram..... | 3 |
| รูปที่ 2.2 Bimodal image histogram..... | 4 |
| รูปที่ 2.3 A.) รูปต้นฉบับ B.) Edge strength and Edge orientation C.) Canny Operator..... | 5 |
| รูปที่ 2.4 Gaussian Filtered..... | 5 |
| รูปที่ 2.5 กระบวนการหา Edge strength..... | 5 |
| รูปที่ 2.6 ช่วงการเปลี่ยนค่าเพื่อใช้ใน Edge orientation..... | 6 |
| รูปที่ 2.7 วิธีการ Non-maximum Suppression..... | 6 |
| รูปที่ 2.8 กระบวนการทั้ง 6 ขั้นตอนของ Canny Method..... | 7 |
| รูปที่ 2.9 3x3 pixels Closing Operation..... | 8 |
| รูปที่ 2.10 Fill Holes 4-connectivity Operation..... | 9 |
| รูปที่ 2.11 Fill Holes 8-connectivity Operation..... | 9 |
| รูปที่ 2.12 การเลือกกลุ่มของข้อมูลที่มากกว่า 20 พิกเซล..... | 9 |
| รูปที่ 2.13 ท่าทางภาษามือแบบอเมริกัน..... | 10 |
| รูปที่ 2.14 แสดง Model ของ Neuron ในคอมพิวเตอร์..... | 12 |
| รูปที่ 2.15 แสดงโครงสร้างวงจร Neural Network..... | 13 |
| รูปที่ 2.16 แสดงรูปแบบ back propagation neural network..... | 13 |
| รูปที่ 2.17 แสดงการเรียนรู้แบบ supervised learning..... | 15 |
| รูปที่ 2.18 แสดงการเรียนรู้แบบ unsupervised learning..... | 15 |
| รูปที่ 3.1 แสดง Block diagram..... | 17 |
| รูปที่ 3.2 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ..... | 17 |
| รูปที่ 3.3 แสดงภาพ Grayscale ของตัวอักษร I..... | 18 |
| รูปที่ 3.4 แสดงภาพ Binary ของตัวอักษร I..... | 18 |
| รูปที่ 3.5 แสดงภาพการเติมสีขาวของพื้นที่ขนาดเล็กกว่าพิกเซลที่กำหนดของตัวอักษร I..... | 19 |
| รูปที่ 3.6 แสดงภาพพื้นที่ที่ถูกเติมให้เต็มแล้วของตัวอักษร I..... | 19 |
| รูปที่ 3.7 แสดงเส้นคอนทัวร์ที่หาได้จากวิธีแคนนี่..... | 20 |
| รูปที่ 3.8 แสดงการวาดวงกลมบนฝ่ามือ..... | 20 |
| รูปที่ 3.9 แสดงหมายเลขด้านทั้ง 3..... | 21 |
| รูปที่ 3.10 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 1..... | 21 |

| | |
|--|----|
| รูปที่ 3.11 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 2..... | 21 |
| รูปที่ 3.12 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 3..... | 22 |
| รูปที่ 3.13 แสดงการแบ่งฝ่ามือออกเป็น 2 ส่วน..... | 22 |
| รูปที่ 3.14 ภาพฝั่งซ้ายแสดงพิกเซลที่จับได้ไม่ครบ ภาพฝั่งขวาแสดงการเติมพิกเซลที่ขาด (จุดสีเขียว)..... | 23 |
| รูปที่ 3.15 ภาพฝั่งซ้ายแสดงพิกเซลที่จับได้เกิน 5 พิกเซล ภาพฝั่งขวาแสดงการตัดพิกเซลที่เกินทิ้ง..... | 24 |
| รูปที่ 3.16 แสดงเส้นระยะทางจากจุดเซ็นทรอยด์ถึงจุดพิกเซล..... | 24 |
| รูปที่ 3.17 แสดงภาพมุมระหว่างพิกเซลที่พบ..... | 25 |
| รูปที่ 4.1 แสดงภาพมือที่ได้จากการถ่ายในสภาพแวดล้อมต่างๆ..... | 27 |
| รูปที่ 4.2 แสดงภาพการเพิ่มและลดนิ้วมือ C (ซ้าย) J (กลาง) O (ขวา)..... | 28 |
| รูปที่ 4.3 แสดงภาพการเพิ่มและลดนิ้วมือ P (ซ้าย) Q (กลาง) X (ขวา)..... | 28 |
| รูปที่ 4.4 แสดงภาพเฟรมแรกของตัว I(ซ้าย) และ J(ขวา)..... | 31 |
| รูปที่ 4.5 แสดงภาพเฟรมแรกของตัวอักษร M (ซ้าย) และ N (ขวา)..... | 31 |
| รูปที่ 4.6 แสดงภาพเฟรมแรกของตัวอักษร P (ซ้าย) และ Q (ขวา)..... | 32 |



VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

| | |
|---|----|
| ตารางที่ 4.1 แสดงค่าเปอร์เซ็นต์ความถูกต้องของการใช้ Hidden layer ที่แตกต่างกัน..... | 29 |
| ตารางที่ 4.2 แสดงการเลือกใช้ค่า Testing และ Training..... | 29 |
| ตารางที่ 4.3 แสดงเปอร์เซ็นต์ความถูกต้องของตัวอักษร A-Z..... | 30 |



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ภาษามือเป็นอวัจนภาษาอย่างหนึ่ง ที่ประกอบด้วย การสื่อสารด้วยมือ, การสื่อสารด้วยร่างกาย และการใช้ริมฝีปากในการสื่อความหมายแทนการใช้เสียงพูด การสื่อสารจะใช้ลักษณะการเคลื่อนไหวของมือเป็นสัญลักษณ์และใช้การเคลื่อนไหวมือ, แขนและร่างกายเพื่อช่วยในการสื่อสาร อีกทั้งยังมีการแสดงความรู้สึกทางใบหน้าเพื่อช่วยในการสื่อสารความคิดของผู้สื่อ ซึ่งในปัจจุบันมีการค้นคว้าหาวิธีการเพื่อให้การสื่อสารทางภาษามือเป็นไปได้ง่ายขึ้น โดยมีวัตถุประสงค์เพื่อให้คนที่มีความบกพร่องทางการได้ยินและบุคคลปกติที่ไม่รู้ภาษามือสามารถสื่อสารกันได้ ซึ่งจะเป็นการลดช่องว่างความแตกต่างของคนปกติกับผู้ที่มีความบกพร่องลงได้ทำให้ผู้ที่มีความบกพร่องทางการได้ยินสามารถใช้ชีวิตอย่างปกติได้ดียิ่งขึ้น

ในการประมวลผลภาพภาษามือนั้นมีการคิดค้นและสร้างเครื่องมือขึ้นมามากมายแต่โดยส่วนใหญ่จะเป็นเทคโนโลยีขั้นสูง กล่าวคือใช้กล้องที่มีคุณภาพสูงและมีราคาแพง จากปัญหาดังกล่าวงานวิจัยนี้จึงมีจุดมุ่งหมายในการประมวลผลภาพโดยใช้กล้อง Webcam ปกติซึ่งมีราคาถูกและหาซื้อได้ทั่วไปผู้ทำการทดลองใช้กล้อง Logitech C310 ซึ่งสามารถลดความละเอียดของการถ่ายวิดีโอได้ต่ำสุดคือ 320x240 โดยพยายามค้นหาอัลกอริทึมที่เหมาะสมเพื่อนำมาวิเคราะห์ภาษามือของตัวอักษรภาษาอังกฤษทั้ง 26 ตัวแล้วให้แสดงผลของสิ่งที่ประมวลได้ออกมา

1.2 วัตถุประสงค์

- เพื่อศึกษาการวิเคราะห์ภาพโดยวิธีการประมวลผลภาพ
- เพื่อศึกษาค้นคว้าความหมายของอัลกอริทึมต่างๆ
- เพื่อช่วยเหลือผู้บกพร่องทางการได้ยินให้สื่อสารได้ง่ายยิ่งขึ้น
- เพื่อลดต้นทุนของการใช้อุปกรณ์ในการประมวลผลภาพที่มีราคาแพง

1.3 ขอบข่ายโครงการ

- ปรับปรุงคุณภาพของภาพเพื่อนำมาเข้าประมวลผลทาง Image processing
- ออกแบบขั้นตอนการประมวลผลภาพของท่าทางการทำตัวอักษรภาษาอังกฤษทั้ง 26 ตัว
- เขียนโปรแกรมเพื่อประมวลผลข้อมูลตามขั้นตอนที่ได้ออกแบบไว้
- สร้างระบบตรวจจับตัวอักษรภาษามือและวิเคราะห์แยกแยะออกมาเป็นตัวอักษรต่างๆ

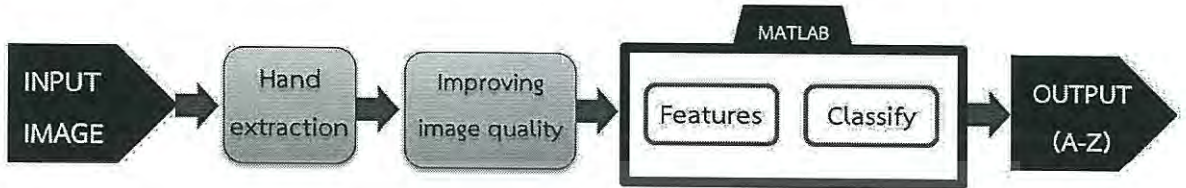
1.4 ผลที่คาดว่าจะได้รับ

- ได้รับความรู้เกี่ยวกับการเขียนโปรแกรม MATLAB
- สามารถเข้าใจการวิเคราะห์ภาพทาง Image processing , เข้าใจอัลกอริทึมต่างๆที่นำมาใช้งาน เข้าใจการออกแบบขั้นตอนการทำงาน และวิธีการคำนวณทางสถิติที่ใช้ในการประมวลผลได้
- ได้รับความรู้เพิ่มเติมนอกเหนือจากเนื้อหาที่เรียน



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง



รูปที่ 2.1 แสดง Block diagram

จากรูปที่ 2.1 แสดงการทำงานของระบบตรวจจับตัวอักษรภาษามือ โดยเริ่มจากการรับข้อมูลภาพมือด้วยกล้อง Webcam จากนั้นส่งข้อมูลสัญญาณเสียงที่ได้เข้าไปวิเคราะห์ด้วยโปรแกรม MATLAB โดยภาพจะผ่านคุณลักษณะต่างๆที่ถูกเขียนไว้ในโปรแกรมเพื่อสกัดคุณลักษณะท่าทางของมือนั้นๆ แล้วนำมาวิเคราะห์เพื่อจำแนกว่าควรอยู่ในกลุ่มมือนิตใดจากทั้งหมด 26 กลุ่ม (A-Z) และผลของการตัดสินใจการจำแนกกลุ่มข้อมูลหรือเอาท์พุทที่ได้จากการวิเคราะห์แสดงผลด้วยการแสดงผลบนหน้าจอ

2.1 การแบ่งข้อมูลภาพ (Image Segmentation)

เป็นการแบ่งข้อมูลภาพออกเป็นส่วนย่อยๆ ที่แต่ละส่วนจะมีพื้นที่ต่อกันไป แต่แต่ละส่วนจะเป็นพื้นที่ซึ่งอาจจะแทนวัตถุที่อยู่ในภาพ เช็กเม้นเตชันจะเสร็จเมื่อวัตถุที่ต้องการ ถูกแบ่งออกได้สมบูรณ์ ผลลัพธ์ที่ได้จากการเช็กเม้นต์จะเป็นตัวชี้วัดความสำเร็จในขั้นตอนการวิเคราะห์ภาพด้วยคอมพิวเตอร์ โดยการทำให้เช็กเม้นเตชันจะมีหลายวิธีด้วยกันได้แก่

2.1.1 Amplitude segmentation methods

สำหรับการ Segmentation ในหัวข้อนี้จะเป็นการพิจารณาความเข้มของจุดต่าง ๆ ภายในภาพ (pixel) ซึ่งผลของการ segment จะขึ้นอยู่กับวิธีการ Threshold ของส่วนประกอบที่เป็นความเข้มหรือสีของภาพ

1.) Bilevel Luminance Thresholding

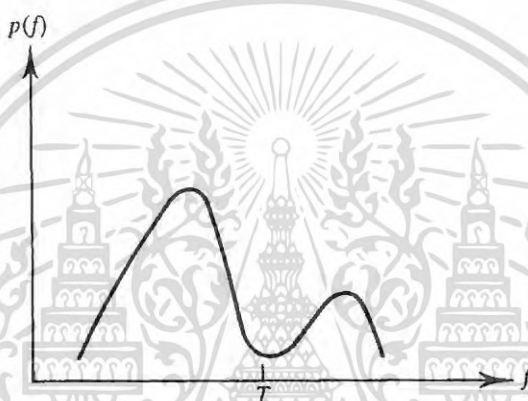
สำหรับภาพบางชนิดจะมีลักษณะวัตถุที่เราสนใจซึ่งมีความเข้มที่คงที่เมื่อเทียบกับพื้นหลัง ตัวอย่างได้แก่ ภาพของตัวอักษร (Text) เป็นต้น ซึ่งภาพเหล่านี้จะมีความเข้มของวัตถุที่เราสามารถแยกออกพื้นหลังได้อย่างชัดเจน (มีความเข้มชั้นสองระดับได้แก่ความเข้มของวัตถุและความเข้มของพื้นหลัง)

การทำการ Segmentation สามารถทำได้โดยการกำหนดค่า Threshold ซึ่งเป็นค่าความเข้มให้มีค่าที่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้ตัวอย่างอย่างเช่น ภาพของ

ตัวอักษรที่มีความความเข้มของตัวอักษรเป็น 0 (สีดำ) และมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่า Threshold จึงควรมีค่าเท่ากับ 128 เพื่อให้จะสามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่า Threshold จะขึ้นอยู่กับ Histogram ของภาพ ตามรูปที่ 2.2 แสดงการหาค่า Threshold โดยค่า Threshold ควรที่จะเลือกค่า histogram ที่อยู่ที่จุดต่ำสุดที่อยู่ระหว่างจุดสูงสุด (peaks)

$$g(x, y) = \begin{cases} 1 & \text{if } (x, y) > T \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

เมื่อ เป็นข้อมูลภาพ ณ ตำแหน่งที่ x, y
เป็นค่า Threshold



รูปที่ 2.2 Bimodal image histogram

2.) Multilevel Luminance Thresholding

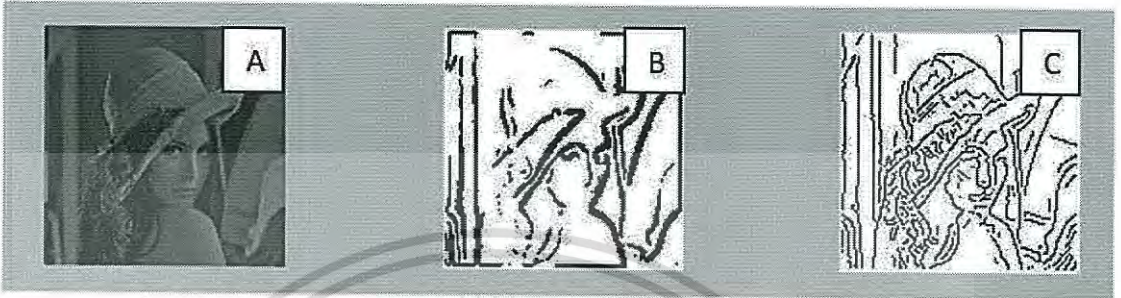
สำหรับภาพที่จะประกอบด้วยหลาย ๆ วัตถุสามารถทำการ Segmentation ได้โดยการใช้ค่า Threshold หลาย ๆ ค่า สำหรับภาพที่มี N วัตถุโดยที่แต่ละวัตถุจะมีช่วงกว้างของความเข้มเท่ากับ (กำหนดได้ด้วยค่า Threshold 2 ค่าคือ) สามารถทำการ Segment ได้ดังนี้

$$g(x, y) = R_i \quad \text{if } (T_{i-1} \leq f(x, y) \leq T_i), \quad i = 1, \dots, N \quad (2.2)$$

ค่า Threshold สามารถหาได้จาก histogram ของภาพ แต่ในหลาย ๆ กรณีที่การเปลี่ยนแปลงของ histogram ไม่สามารถบอกการเปลี่ยนแปลงระหว่างวัตถุได้อย่างชัดเจน วิธีการที่ง่ายที่สุดที่จะทำให้ histogram สามารถหาค่า Threshold ได้ง่ายขึ้นก็คือการใช้วิธี Edge Detection เพื่อพิจารณาพิกเซลต่าง ๆ ของภาพให้ว่าเป็นขอบของวัตถุ

2.1.2 CANNY OPERATOR

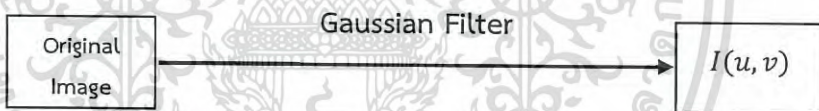
Canny operator เป็นหนึ่งในวิธีการหาขอบรูปภาพ โดยประกอบด้วยขั้นตอนหลายขั้น โดยมีวัตถุประสงค์เพื่อเพิ่มความสามารถในการลดขนาดของขอบภาพลงและทำให้สามารถเลือกช่วงความเข้มของขอบภาพที่ต้องการได้ ตัวอย่างดังรูป



รูปที่ 2.3 A.) รูปต้นฉบับ B.) Edge strength and Edge orientation C.) Canny Operator

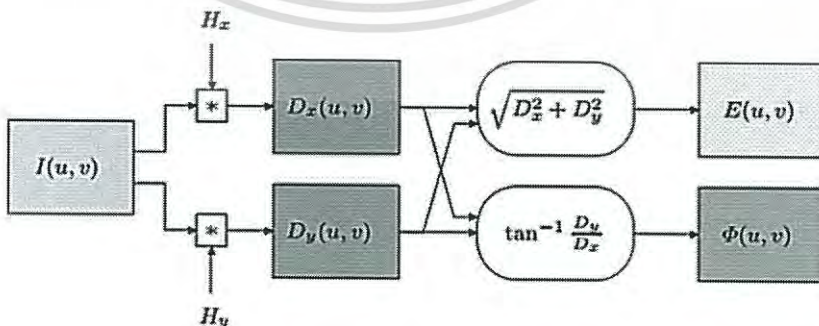
โดยกระบวนการจะประกอบไปด้วยทั้งหมด 6 ขั้นตอน

- 1.) Gaussian Filtered โดยมีวัตถุประสงค์เพื่อลด (กำจัด) noise ออกไปจากรูปภาพ โดยอาศัยคุณสมบัติการกำจัด noise จากภาพต้นแบบ ด้วย Gaussian Filtered matrix เนื่องจากจะทำให้ภาพ smooth ขึ้น ทำให้ไม่เกิดขอบภาพที่ไม่ต้องการ



รูปที่ 2.4 Gaussian Filtered

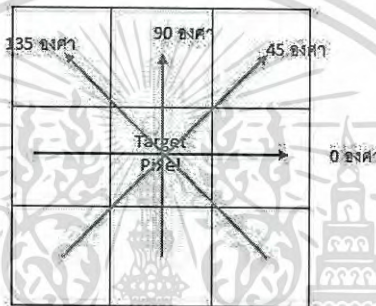
- 2.) Edge strength and Edge orientation หา Edge strength และ Edge orientation ของรูปที่ ภาพที่ผ่าน Gaussian Filtered มาแล้ว โดยอาจใช้กระบวนการแบบ Prewitt หรือ Sobel ก็ได้



รูปที่ 2.5 กระบวนการหา Edge strength

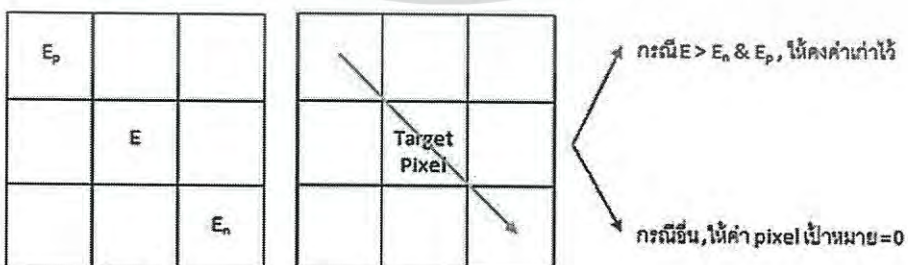
3.) Edge orientation Substituted เพื่อการนำไป Edge orientation ไปประยุกต์ใช้ในกระบวนการ Non-maximum Suppression เราจำเป็นต้องเปลี่ยนค่าให้อยู่ในช่วงที่สามารถระบุทิศทางเป็นตำแหน่งของ Pixel รอบๆ ได้ กล่าวคือค่าที่เป็นได้ประกอบด้วย 0,45,90,135 เท่านั้น ทั้งนี้ เราจะใช้วิธีการเปลี่ยนค่า Edge orientation ใหม่จากตามช่วงค่าดังนี้

- 0 องศา สำหรับค่า 0-22.5 & 157.5 – 180 องศา
- 45 องศา สำหรับค่า 22.5-67.5 องศา
- 90 องศา สำหรับค่า 67.5 – 112.5 องศา
- 135 องศา สำหรับค่า 112.5 – 157.5 องศา



รูปที่ 2.6 ช่วงการเปลี่ยนค่าเพื่อใช้ใน Edge orientation

4.) Non-maximum Suppression เมื่อเราทำการแปลงค่า Edge orientation ให้เป็นค่าที่สามารถระบุเป็นตำแหน่ง pixel โดยรอบได้แล้วนั้น ในกระบวนการนี้ เราจะใช้ค่า Edge orientation เพื่อเลือกคู่ Pixel โดยรอบที่เรียงตัวเป็นเป็นช่วงองศานั้น (positive/negative -arrangement pixel) มาทำการเปรียบเทียบค่า Edge strength กับค่า Edge strength ของ pixel นั้นๆ หากค่าของ Pixel กลางนั้นมีค่ามากกว่า Pixel ข้างเคียงที่เลือกมา ให้เก็บค่า Pixel นั้นๆไว้ สำหรับกรณีอื่นให้กำจัดค่าของ pixel นั้นทิ้ง (ให้เท่ากับ 0)



รูปที่ 2.7 วิธีการ Non-maximum Suppression

5.) Hysteresis (Double Threshold) หลังจากที่เรารับผ่านกระบวนการทั้งหมดมาแล้วนั้น ภาพที่ได้ยังมีความเป็นไปไม่ได้ที่จะยังหลงเหลือผลกระทบจาก noise ที่ไม่เกี่ยวข้องอยู่จำนวนหนึ่ง เพื่อตัดสิ่งรบกวนเหล่านี้ออกไปจึงมีการนำ Hysteresis เข้ามาช่วยโดยทำให้เราสามารถเลือกช่วงค่า Edge strength ที่ต้องการแสดงได้และค่าที่ต่ำกว่าในขอบเขตที่ระบุให้กำจัดค่าของ pixel นั้นทิ้ง(ให้เท่ากับ 0) ส่วนค่าที่มากกว่าให้เป็นขอบภาพความเข้มสูง(strong edge) และส่วนที่อยู่ในช่วงให้เป็นขอบความเข้มจาง(weak edge)ซึ่งจะนำมาพิจารณาในขั้นตอนต่อไป

6.) Hysteresis ในกระบวนการสุดท้ายเราจะทำการแยกขอบออกเป็นส่วนๆโดยแบ่งตามตำแหน่งที่เชื่อมต่อกันและความเข้ม โดยขอบส่วนที่มีความเข้มจาง (weak edge) ที่ไม่เชื่อมต่อกับขอบส่วนที่มีความเข้มสูง (strong edge) จะถูกกำจัดทิ้ง เนื่องจากเพราะโดยทั่วไปแล้วสันนิฐานว่าส่วนที่เป็นขอบจริงจะต้องมีส่วนที่เชื่อมต่อกับขอบที่มีความเข้มสูงเท่านั้นเสมอ



รูปที่ 2.8 กระบวนการทั้ง 6 ขั้นตอนของ Canny Method

2.2 กระบวนการที่เกี่ยวข้องอื่นๆ

2.2.1 การเปลี่ยนแปลงภาพสีเป็นภาพระดับสีเทา (Gray-Scale Image Transform)

เป็นการแปลงค่าข้อมูลภาพสีในระบบ RGB ให้แสดงถึงค่าความสว่างของภาพเพียงอย่างเดียว โดยปราศจากค่าข้อมูลของสีภาพ โดยทั่วไปภาพระดับสีเทาจะประกอบด้วยค่าความสว่างที่

แตกต่างกัน 256 ระดับ มีค่าตั้งแต่ 0 ถึง 255 นั่นคือไล่ระดับความสว่างจากมืดไปจนขาว และ ความสัมพันธ์การแปลงภาพสีเป็นภาพระดับสีเทา ดังสมการ

$$\text{Gray} = 0.299(\text{Red}) + 0.587(\text{Green}) + 0.114 (\text{Blue}) \quad (2.3)$$

โดยที่ Red คือ ค่าของสีแดงในระบบสี RGB

Green คือ ค่าของสีเขียวในระบบสี RGB

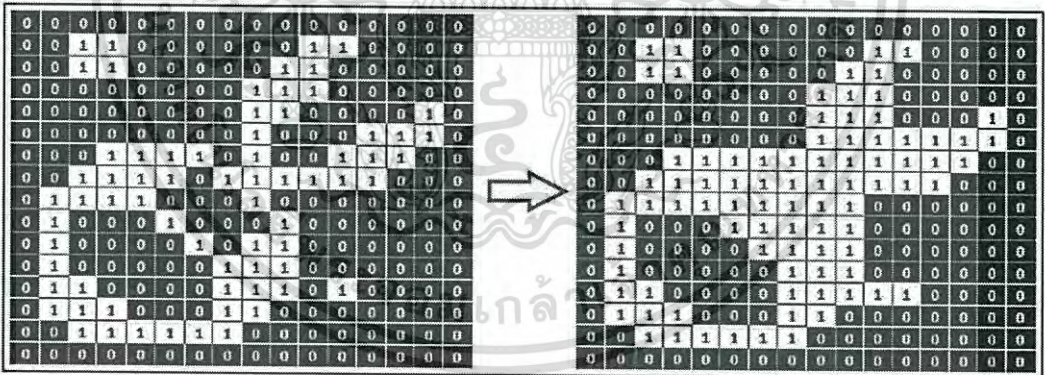
Blue คือ ค่าของสีฟ้าในระบบสี RGB

2.2.2 การเปลี่ยนแปลงลักษณะของวัตถุ (Morphological Operation)

เป็นโอเปอเรชันในข้อมูลภาพแบบไบนารี เพื่อตกแต่งให้ลักษณะของวัตถุให้มีความสมบูรณ์มากขึ้น

1.) Closing Operation เป็นการนำ Erosion Operation ตามด้วย Dilation Operation เพื่อให้ส่วนแคบๆของ Region ต่อกัน สามารถอธิบายดังสมการ

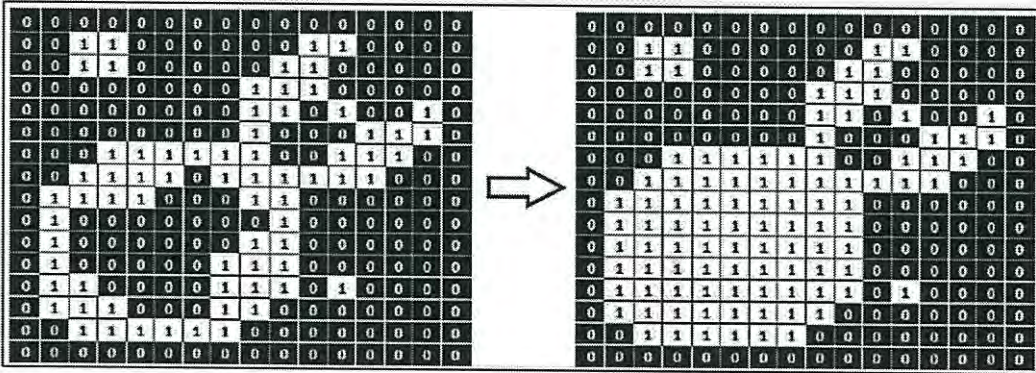
$$A \cdot B = (A \oplus B) \ominus B \quad (2.4)$$



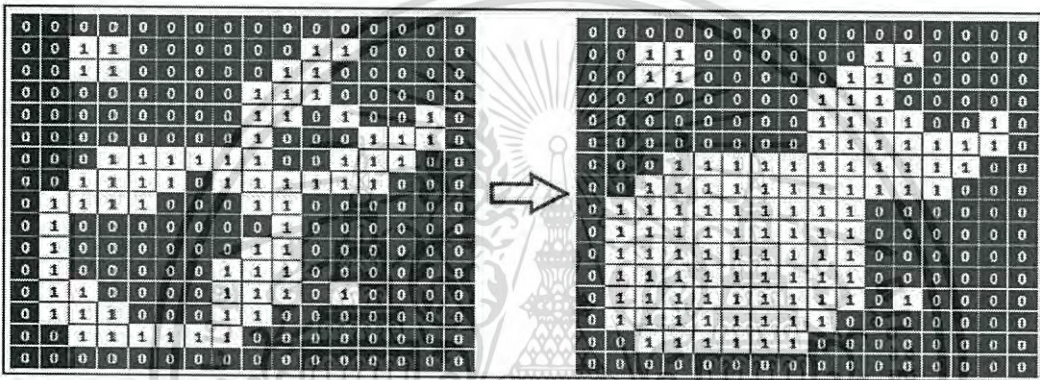
รูปที่ 2.9 3x3 pixels Closing Operation

2.) Fill Holes Operation เป็นการเติมส่วนที่เป็นช่องว่างที่เกิดขึ้นในวัตถุจากการหา

พิกเซลที่ถูกล้อมรอบ โดยอาศัยการหาจุดเชื่อมต่อระหว่างขอบของวัตถุที่เกิดขึ้นกับขอบของวัตถุข้างเคียง ซึ่งสามารถแบ่งจุดเชื่อมต่อได้ 2 ลักษณะ คือ 4-connectivity และ 8-connectivity

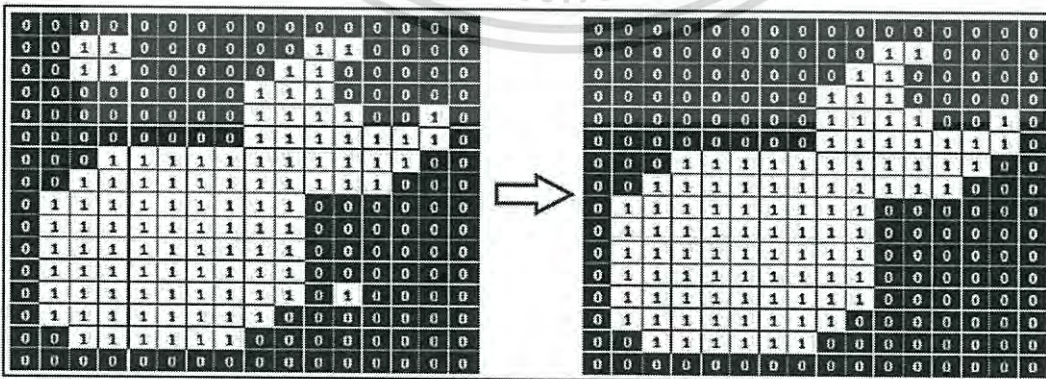


รูปที่ 2.10 Fill Holes 4-connectivity Operation



รูปที่ 2.11 Fill Holes 8-connectivity Operation

3.) Region Selection ใช้ในการเลือกกลุ่มของข้อมูลที่ต้องการ โดยอาศัยการหาจุดเชื่อมต่อระหว่างขอบของวัตถุที่เกิดขึ้นกับขอบของวัตถุข้างเคียง ซึ่งสามารถแบ่งจุดเชื่อมต่อได้ 2 ลักษณะ คือ 4-connectivity และ 8-connectivity เพื่อใช้ในการตรวจสอบขนาดของกลุ่มข้อมูลให้ได้ตามต้องการ



รูปที่ 2.12 การเลือกกลุ่มของข้อมูลที่มากกว่า 20 พิกเซล

2.2.3 Euclidean distance

Euclidean distance ในทางคณิตศาสตร์ Euclidean distance หรือ Euclidean metric คือ ระยะทาง ระหว่าง 2 จุด จุดที่จะวัดนั้นมีเงื่อนไขมีหลายค่าจากหลายมิติหรือขนาด ขึ้นกับรูปแบบ ซึ่งสามารถพิสูจน์หาค่าได้ด้วยทฤษฎีของ Pythagorean เมื่อมีการใช้สูตรเพื่อหา ระยะทาง ขนาดของ Euclidean จะเป็นเซตของ Vector

นิยาม

Euclidean $P = (p_1, p_2, \dots, p_n)$ ระยะทางระหว่างจุด และ $Q = (q_1, q_2, \dots, q_n)$ ใน Euclidean หลายขนาด, ระบุได้เป็น:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.5)$$

2.2.4 American Sign Language

ASL เป็นชื่อย่อของ American Sign Language หรือภาษามืออเมริกัน ภาษามืออเมริกันคือภาษาที่ผู้มีความบกพร่องทางการได้ยินทั้งหูหนวกและหูตึงใช้ติดต่อสื่อสารกับคนทั่วไปในสังคมชาวอเมริกัน



รูปที่ 2.13 ท่าทางภาษามือแบบอเมริกัน

2.3 โครงข่ายประสาทเทียม (Artificial neural network)

ระบบโครงข่ายประสาท (Neural Network) หรือ “โครงข่ายใยประสาทเสมือน (Artificial Neural Network: ANN)” หมายถึง คอมพิวเตอร์ที่สามารถเลียนแบบการทำงานของสมองมนุษย์ได้ ด้วยการประมวลผลข้อมูลสารสนเทศ และองค์ความรู้ได้ในคราวละหลายๆ

นอกจากนี้ ยังสามารถรับและจดจำสารสนเทศในรูปแบบที่เป็นประสบการณ์ได้ ทำให้สามารถเชื่อมโยงข้อเท็จจริงทั้งหลายเข้าด้วยกันเพื่อหาข้อสรุป และใช้ประสบการณ์ที่จัดเก็บไว้มาเรียนรู้และทำความเข้าใจว่า ข้อเท็จจริงใหม่ที่ได้รับเข้ามามีความเกี่ยวข้องกันอย่างไร เพื่อทำการปรับปรุงองค์ความรู้ให้มีความทันสมัยเพื่อประโยชน์ในอนาคต

2.3.1 องค์ประกอบและโครงสร้างการทำงาน

1.) ข้อมูลนำเข้า (Input) ข้อมูลนำเข้าจะถูกจำแนกตามคุณลักษณะ (Attribute) เช่น ถ้าปัญหาที่ระบบเฝ้าระวังประสาทเสมือนจะต้องตัดสินใจคือ การอนุมัติเงินกู้ว่าจะให้ผ่านหรือไม่ ข้อมูลนำเข้าก็就会被จำแนกเป็นคุณลักษณะ กล่าวคือ ระดับรายได้ และอายุ เป็นต้น ข้อมูลนำเข้านอกจากจะเป็นข้อความแล้ว ยังสามารถเป็นรูปภาพ หรือเสียงก็ได้ แต่อาจจะต้องผ่านการแปลงให้เป็นสัญลักษณ์หรือตัวเลขเพื่อให้เครื่องสามารถทำความเข้าใจได้ก่อน จากนั้นก็จะเข้าสู่การทำงานที่แท้จริงของระบบเฝ้าระวังประสาทเสมือนที่เริ่มต้นด้วยการนำข้อมูลเข้ามาน้ำหนัก (weight) ของความสัมพันธ์ระหว่างข้อมูลนำเข้าเหล่านั้นในเลเยอร์แรกภายใต้ขอบเขตของระบบ

2.) น้ำหนัก (Weight) เป็นส่วนประกอบที่สำคัญของระบบโครงข่ายเฝ้าระวังประสาท เนื่องจากเป็นส่วนที่ใช้หาน้ำหนักของความสัมพันธ์ระหว่างข้อมูลนำเข้า ว่าข้อมูลนำเข้าใดมีความสัมพันธ์กับข้อมูลนำเข้าอื่นในระดับใด ซึ่งจะทำให้สามารถเชื่อมโยงไปหาข้อสรุปได้ ด้วยการลองผิดลองถูกในความสัมพันธ์แต่ละแบบ และเก็บไว้เป็นแบบแผนหรือรูปแบบ (pattern) ของประสบการณ์เพื่อการเรียนรู้ของโครงข่าย

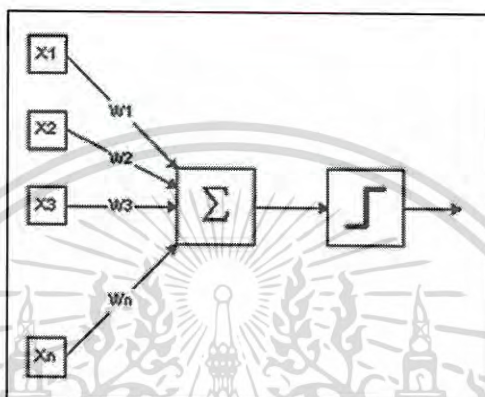
3.) ฟังก์ชันการรวม (Summation Function) เป็นโครงข่ายที่ทำหน้าที่ในการรวมค่าน้ำหนักที่ได้จากโครงข่ายในเลเยอร์ input เพื่อสรุปผลความสัมพันธ์ระหว่างข้อมูลนำเข้า รอการแปลงเป็นสารสนเทศที่มีความหมายในเลเยอร์ต่อไป

4.) ฟังก์ชันการแปลง (Transformation Function) เป็นโครงข่ายที่ทำหน้าที่ในการประสาน (integrate) สารสนเทศที่ผ่านการประมวลผลจากโครงข่ายในเลเยอร์ต่างๆแล้วทำการแปลง (Transform) ให้กลายเป็นสารสนเทศที่สื่อความหมาย และเป็นประโยชน์ต่อการนำไปใช้ได้เพื่อส่งออกไปเป็นผลลัพธ์ (Output)

5.) ผลลัพธ์ (Output) ผลลัพธ์ที่ได้จากโครงข่ายเฝ้าระวังประสาทเสมือน จะหมายถึงแนวทางในการแก้ไขปัญหา เช่น ปัญหาการอนุมัติเงินกู้ว่าผู้กู้จะผ่านการอนุมัติหรือไม่ “ผลลัพธ์” ที่ผู้ใช้

จะได้รับคือ "อนุมัติ" หรือ "ไม่อนุมัติ" ซึ่งโครงข่ายประสาทเสมือนจะใช้สัญลักษณ์แทนคำตอบทั้งหมด

ผลลัพธ์ที่ได้จากโครงข่ายหนึ่ง สามารถเป็นข้อมูลนำเข้า (input) ของอีกโครงข่ายหนึ่งได้ ทั้งนี้เพื่อเป็นข้อมูลนำเข้าของการตัดสินใจแก้ไขปัญหาอื่น เช่น ผลลัพธ์ที่ได้จากการอนุมัติเงินกู้ อาจจะไปใช้เป็นข้อมูลนำเข้าเพื่อการอนุมัติสินเชื่อที่อยู่อาศัยได้



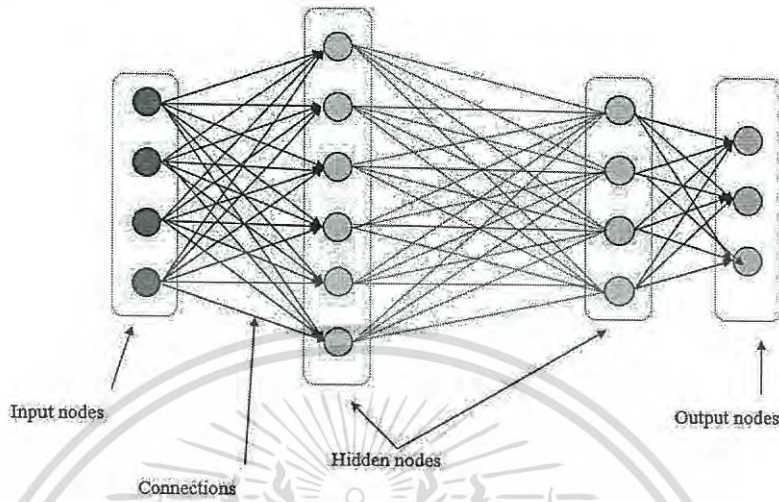
รูปที่ 2.14 แสดง Model ของ Neuron ในคอมพิวเตอร์

2.3.2 หลักการทำงาน

สำหรับในคอมพิวเตอร์ Neurons ประกอบด้วย input และ output เหมือนกัน โดยจำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะสามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทำงานร่วมกันการทำงานนี้ในทางตรรกแล้วก็จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง

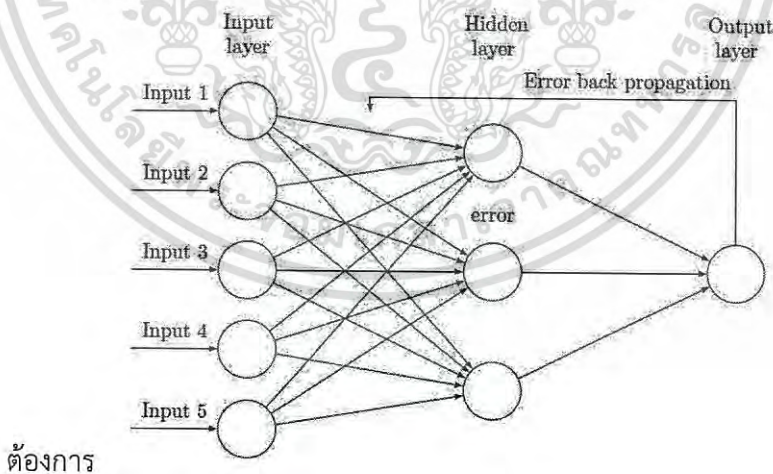
สิ่งสำคัญคือเราต้องทราบค่า weight และ threshold สำหรับสิ่งที่เราต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้ โดยการสอนให้มันรู้จัก pattern ของสิ่งที่เราต้องการให้มันรู้จำ เรียกว่า "back propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ในการฝึก feed-forward neural networks จะมีการใช้อัลกอริทึมแบบ back-propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของเครือข่าย (network weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหา

ค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนัก
 ค่ะแน่นอนต่อไป



รูปที่ 2.15 แสดงโครงสร้างวงจร Neural Network

- Back-propagation เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครือข่ายประสาทวิธีหนึ่ง
 ที่นิยมใช้ใน multilayer perceptron เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้
 เหมาะสม โดยการปรับค่านี้อาจจะขึ้นกับความแตกต่างของค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่



รูปที่ 2.16 แสดงรูปแบบ back propagation neural network

ขั้นตอนของ Back-propagation Algorithm มีดังนี้

1. กำหนดค่าอัตราเร็วในการเรียนรู้ (rate parameter: r)

2. สำหรับแต่ละตัวอย่างอินพุตให้ทำตามขั้นตอนต่อไปนี้จะกว่าได้ระดับ performance ที่ต้องการ

คำนวณหาค่าเอาต์พุตโดยใช้ค่าน้ำหนักเริ่มต้นซึ่งอาจได้จากการสุ่ม

- คำนวณหาค่า β : แทนประโยชน์ที่จะได้รับสำหรับการเปลี่ยนค่าเอาต์พุตของแต่ละโหนด
- ในชั้นเอาต์พุต (Output Layer)

$$\beta_z = d_z - o_z \quad (2.6)$$

เมื่อ d_z = ค่าเอาต์พุตที่ต้องการ

o_z = ค่าเอาต์พุตที่คำนวณได้

- ในชั้นซ่อน (Hidden Layer)

$$\beta_j = \sum w_{jk} o_k (1 - o_k) \beta_k \quad (2.7)$$

เมื่อ w_{jk} = น้ำหนักของเส้นเชื่อมระหว่างชั้นที่ j กับ k

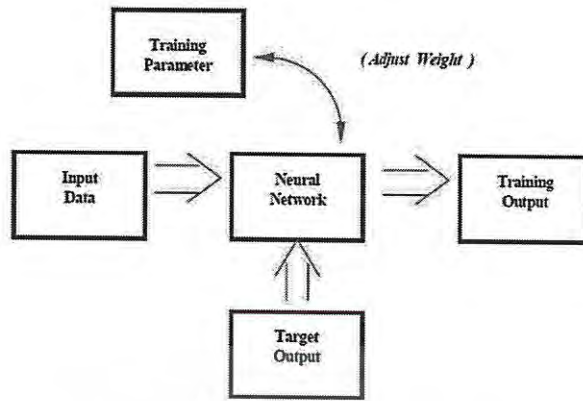
- คำนวณค่าน้ำหนักที่เปลี่ยนแปลงไปสำหรับในทุกน้ำหนัก ด้วยสมการต่อไปนี้

$$\Delta w_{ij} = r o_i o_j (1 - o_j) \beta_j \quad (2.8)$$

- เพิ่มค่าน้ำหนักที่เปลี่ยนแปลง สำหรับตัวอย่างอินพุตทั้งหมด และเปลี่ยนค่าน้ำหนัก

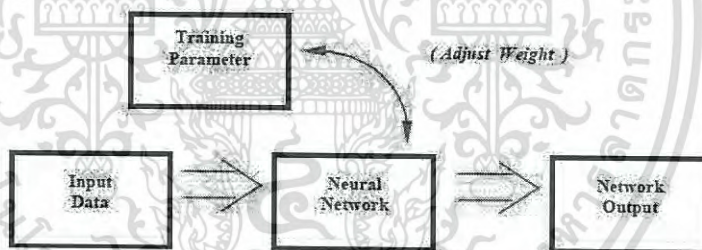
2.3.3 การเรียนรู้สำหรับโครงข่ายประสาทเทียม

1.) Supervised Learning การเรียนแบบมีการสอน เป็นการเรียนแบบที่มีการตรวจคำตอบเพื่อให้วงจรข่ายปรับตัว ชุดข้อมูลที่ใช้สอนวงจรข่ายจะมีคำตอบไว้คอยตรวจดูว่าวงจรข่ายให้คำตอบที่ถูกต้องหรือไม่ ถ้าตอบไม่ถูกต้อง วงจรข่ายก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น (เปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ)



รูปที่ 2.17 แสดงการเรียนรู้แบบ supervised learning

2.) Unsupervised Learning การเรียนแบบไม่มีการสอน เป็นการเรียนแบบไม่มีผู้แนะนำ ไม่มีการตรวจคำตอบว่าถูกหรือผิด วงจรข่ายจะจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล ผลลัพธ์ที่ได้ วงจรข่ายจะสามารถจัดหมวดหมู่ของข้อมูลได้ (เปรียบเทียบกับคน เช่น การที่เราสามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ตามลักษณะรูปร่างของมันได้เองโดยไม่มีใครสอน)



รูปที่ 2.18 แสดงการเรียนรู้แบบ unsupervised learning

2.3.4 ประโยชน์ของโครงข่ายประสาทเสมือน

1.) เกิดข้อผิดพลาดได้ยาก (Fault Tolerance) หากระบบโครงข่ายใยประสาทเสมือนประกอบไปด้วยโครงข่ายที่ใช้ในการประมวลผลมากมายหลายโครงข่าย ความผิดพลาดที่เกิดขึ้นจากเพียงหนึ่งหรือสองโครงข่ายจะไม่ทำให้ทั้งระบบเกิดข้อผิดพลาดได้

2.) ความสามารถในการหาเหตุผล (Generalization) เมื่อระบบโครงข่ายใยประสาทเสมือนได้รับข้อมูลนำเข้าที่ไม่ครบถ้วนหรือไม่เพียงพอต่อการหาข้อสรุป หรือได้รับข้อเท็จจริงที่

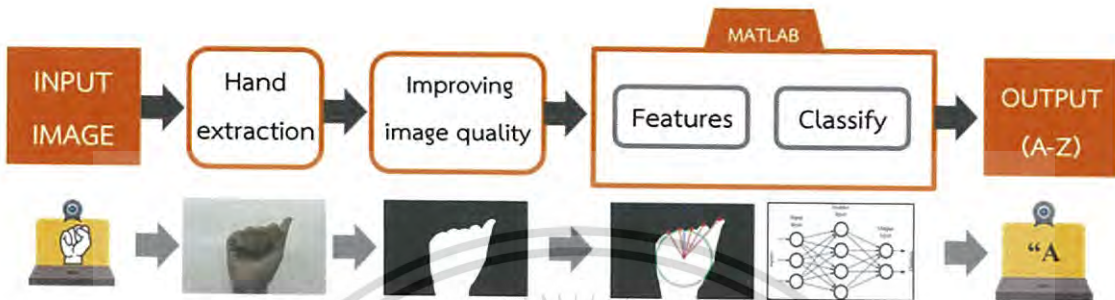
ไม่เคยได้รับมาก่อน ระบบจะสามารถลำดับการเชื่อมโยงข้อเท็จจริงจนสามารถให้ข้อสรุปและเหตุผลได้

3.) ความสามารถในการปรับเปลี่ยน (Adaptability) โครงข่ายใยประสาทเสมือนสามารถเรียนรู้สภาพแวดล้อมใหม่ได้ ดังนั้นเมื่อมีเหตุการณ์ใหม่ๆ เข้าสู่ระบบก็จะสามารถปรับเปลี่ยนหรือปรับปรุงองค์ความรู้ให้ทันสมัยตามเหตุการณ์ใหม่นั้น

4.) ความสามารถในการพยากรณ์ (forecasting Capability) โครงข่ายใยประสาทเสมือนสามารถนำข้อมูลทางสถิติเดิมที่มีอยู่ในระบบ มาใช้คาดการณ์หรือพยากรณ์ข้อมูลในอนาคตได้



บทที่ 3 วิธีการดำเนินงาน



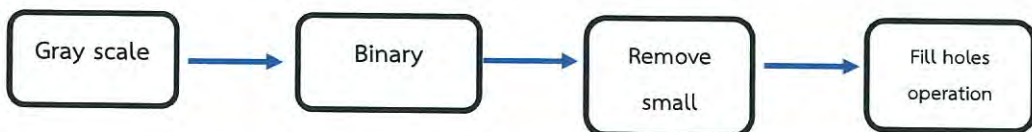
รูปที่ 3.1 แสดง Block diagram

จากรูปที่ 3.1 แสดงการทำงานของระบบตรวจจับตัวอักษรภาษามือ โดยเริ่มจากการรับข้อมูลภาพมือด้วยกล้อง Webcam จากนั้นส่งข้อมูลสัญญาณเสียงที่ได้เข้าไปวิเคราะห์ด้วยโปรแกรม MATLAB โดยภาพจะผ่านคุณลักษณะต่างๆที่ถูกเขียนไว้ในโปรแกรมเพื่อสกัดคุณลักษณะท่าทางของมือนั้นๆ แล้วนำมาวิเคราะห์เพื่อจำแนกว่าควรอยู่ในกลุ่มมือนิตใดจากทั้งหมด 26 กลุ่ม (A-Z) และผลของการตัดสินใจการจำแนกกลุ่มข้อมูลหรือเอ้าท์พุทที่ได้จากการวิเคราะห์แสดงผลด้วยการแสดงผลบนหน้าจอ

3.1 การรับภาพ (Input image)

การทดลองนี้ใช้กล้อง Logitech HD webcam C310 เพื่อรับภาพเข้ามาประมวลผลโดยผู้ทำการทดลองได้ทำการทดลองแบบ offline โดยทำการบันทึกวิดีโอแล้วนำมาตัดเป็นเฟรม โดยมีอัตราความเร็วของภาพ 30 เฟรมต่อวินาที จากนั้นนำภาพที่ได้มาเข้าสู่กระบวนการปรับปรุงคุณภาพ

3.2 การปรับปรุงคุณภาพของภาพ



รูปที่ 3.2 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ

3.2.1 ภาพระดับสีเทา (Grayscale image) โดยใช้คำสั่ง 'rgb2gray' เพื่อเปลี่ยนภาพ RGB ให้อยู่ในรูปของ Grayscale จากนั้นใช้คำสั่ง 'graythresh' เพื่อให้ได้ค่าระดับขีดแบ่ง (Threshold level)

ออกมาซึ่งเป็นค่าความเข้มปกติที่อยู่ในช่วง 0 ถึง 1 เพื่อนำไปเป็นเกณฑ์ในการเปลี่ยนเป็นภาพไบนารีต่อไป ด้วยคำสั่ง 'im2bw'



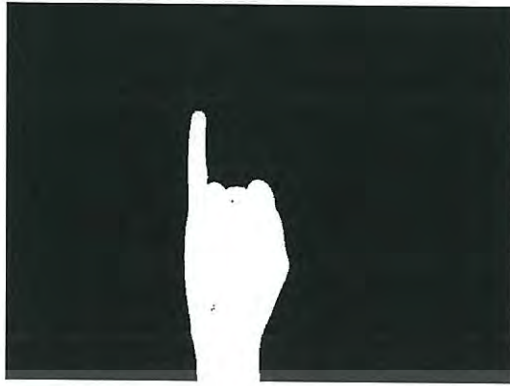
รูปที่ 3.3 แสดงภาพ Grayscale ของตัวอักษร I

3.2.2 ภาพขาว-ดำ (Binary image) การทำภาพให้เป็นไบนารีนั้นเพื่อต้องการให้ได้ขอบของภาพที่สมบูรณ์อีกทั้งยังเป็นการตัดเงาหรือส่วนที่ไม่จำเป็นออกจากวัตถุไปได้ส่วนหนึ่ง



รูปที่ 3.4 แสดงภาพ Binary ของตัวอักษร I

3.2.3 การลบพื้นที่ขนาดเล็กกว่าที่กำหนด (Remove small object) โดยการลดส่วนประกอบของภาพที่มีจำนวนพิกเซลน้อยกว่าที่เรากำหนดไว้ เพื่อเป็นการลดส่วนประกอบที่ไม่ใช่มือทิ้งไปโดยใช้คำสั่ง 'bwareaopen'



รูปที่ 3.5 แสดงภาพการเติมสีขาวของพื้นที่ขนาดเล็กกว่าพิกเซลที่กำหนดของตัวอักษร I

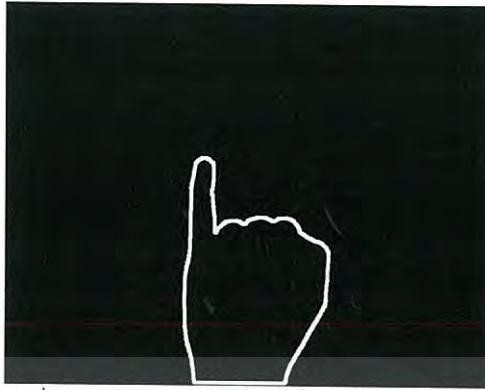
3.2.4 การเติมพื้นที่หรือรูของภาพ (Fill image regions and holes) กระบวนการนี้ใช้ในการเติมส่วนของพื้นที่ปิดหรือรูของภาพให้เต็มโดยการเติมส่วนที่เป็นช่องว่างในภาพที่มีพิกเซลสีขาว (มีค่าความเข้มเท่ากับ 1) ล้อมรอบเพื่อให้ได้ภาพมือที่เป็นสีขาวล้วนโดยที่ไม่มีรูอยู่ในภาพ



รูปที่ 3.6 แสดงภาพพื้นที่ที่ถูกเติมให้เต็มแล้วของตัวอักษร I

3.3 การระบุตำแหน่งของมือ

3.3.1 การหาตำแหน่งของขอบมือ ซึ่งหาได้จากรัศมีสูงสุดของฝ่ามือที่เป็นไปได้โดยอ้างอิงจากขอบคอนทัวร์ (contours) ของรูปมือที่ได้จากวิธีการ Canny ซึ่งเป็นการตรวจจับขอบภาพเป็นวิธีการระบุจุดบนภาพดิจิทัลที่ซึ่งความสว่างเปลี่ยนแปลงอย่างรวดเร็วหรือไม่ต่อเนื่องนั่นเอง จากนั้นจะเก็บค่าที่ได้ไปใช้ประมวลผลต่อไป



รูปที่ 3.7 แสดงเส้นคอนทัวร์ที่ทำได้จากวิธีแคนนี่

3.3.2 การหาวงกลมบนฝ่ามือ โดยนำค่าคอนทัวร์ที่ได้ไปใช้หารัศมีของวงกลมสูงสุดที่สามารถวาดบนฝ่ามือได้โดยใช้สมการ

$$x^2 + y^2 = r^2 \quad (3.1)$$

จากนั้นจะอ้างอิงจุดศูนย์กลางวงกลมเป็นจุดเซ็นทรอยด์ เพื่อแก้ปัญหาคำนวณตำแหน่งเซ็นทรอยด์ที่อาจผิดพลาดเมื่อภาพที่รับเข้ามามีข้อผิดพลาดด้วยนั่นเอง

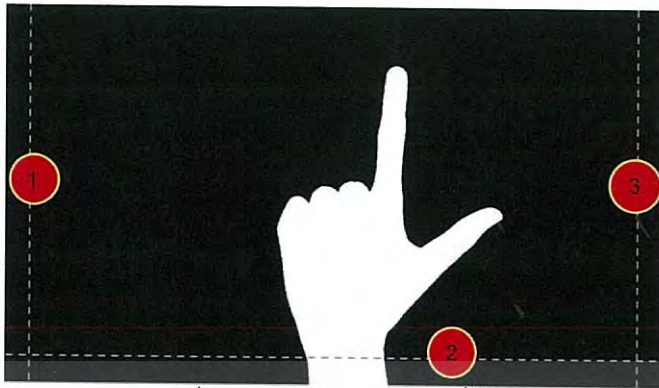


รูปที่ 3.8 แสดงการวาดวงกลมบนฝ่ามือ

3.4 การวิเคราะห์คุณลักษณะของภาพ

การวิเคราะห์คุณลักษณะของภาพผู้ทำการทดลองได้ทำการหาเวกเตอร์เพื่อใช้เป็นข้อมูลในการเข้าเรียนรู้ในโครงข่ายประสาทเทียม โดยวิเคราะห์จาก 3 คุณลักษณะได้แก่ พื้นที่ของขอบภาพ 3 ด้าน, ความยาวของนิ้วจากจุดเซ็นทรอยด์ และมุมระหว่างนิ้วมือ

3.4.1 พื้นที่ของขอบภาพ 3 ด้าน เพื่อใช้วิเคราะห์ว่าภาพมือที่รับเข้ามาเป็นแนวตั้งหรือแนวนอน โดยคำนวณจากจำนวนพิกเซลสีขาวที่พบในแต่ละด้าน



รูปที่ 3.9 แสดงหมายเลขด้านทั้ง 3

- หากพบว่าไม่มีพิกเซลสีขาวจากขอบภาพด้านที่ 1 เข้ามา 5 คอลัมน์ให้พิจารณาว่ารูปเป็นแนวอนฝั่งซ้ายและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



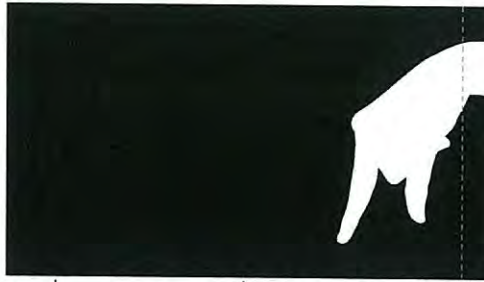
รูปที่ 3.10 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 1

- หากพบว่าไม่มีพิกเซลสีขาวจากขอบภาพด้านที่ 2 เข้ามา 5 แถวให้พิจารณาว่ารูปเป็นแนวตั้งและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



รูปที่ 3.11 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 2

- หากพบว่าไม่มีพิกเซลสีขาวจากขอบภาพด้านที่ 3 เข้ามา 5 คอลัมน์ให้พิจารณาว่ารูปเป็นแนวอนฝั่งขวาและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



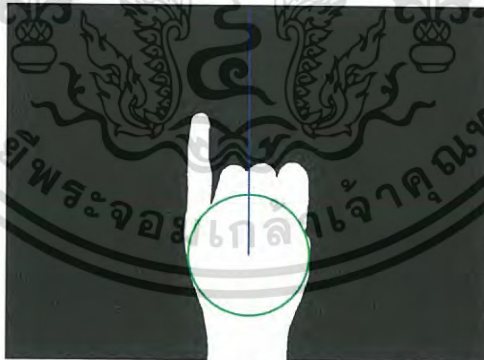
รูปที่ 3.12 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 3

3.4.2 ความยาวนิ้วมือจากจุดเซ็นทรอยด์ มีขั้นตอนดังนี้

3.4.2.1 ทำการหาปลายนิ้วมือของรูปที่รับเข้ามาโดยใช้วิธีการหาพีคของรูป จากนั้นเก็บค่าไว้วิเคราะห์ต่อไป

3.4.2.2 การหาพีคที่ถูกต้องที่สุด เนื่องจากการหาพีคจากปลายนิ้วของรูปภาพบางครั้งจะไม่ได้ปลายนิ้วที่ครบทั้ง 5 นิ้ว อันเนื่องจากการทำท่าทางของมือที่แตกต่างกันทำให้บางครั้งนิ้วทุกนิ้วไม่ปรากฏออกมาให้เห็น ดังนั้นเราจึงต้องทำการลดหรือเพิ่มพีคที่หาได้ โดยเริ่มจาก

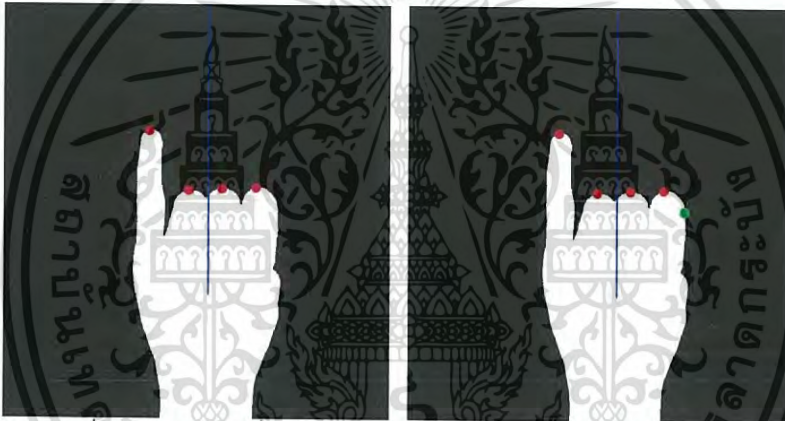
- 1) แบ่งฝ่ามือออกเป็นสองส่วน โดยใช้จุดเซ็นทรอยด์แบ่งเป็นด้านซ้ายและด้านขวาเพื่อ วิเคราะห์ว่านิ้วมือแต่ละนิ้วควรอยู่ฝั่งไหนเพื่อวิเคราะห์ต่อไปว่าควรจะมีหรือลดพีคที่พบ โดยเราพบว่าโดยปกติและฝั่งซ้ายจะพบนิ้วมือ 2 นิ้ว คือนิ้วก้อยและนิ้วนาง ส่วนด้านขวาจะพบนิ้วมือ 3 นิ้ว คือนิ้วโป้ง, นิ้วชี้ และนิ้วกลาง โดยเราจะใช้เส้นสรน้ำเงินแทนเส้นที่ใช้แบ่งฝ่ามือ



รูปที่ 3.13 แสดงการแบ่งฝ่ามือออกเป็น 2 ส่วน

- 2) การเติมพีคให้แก่ภาพที่จับพีคได้ไม่ครบ 5 นิ้ว โดยมีหลักการว่า

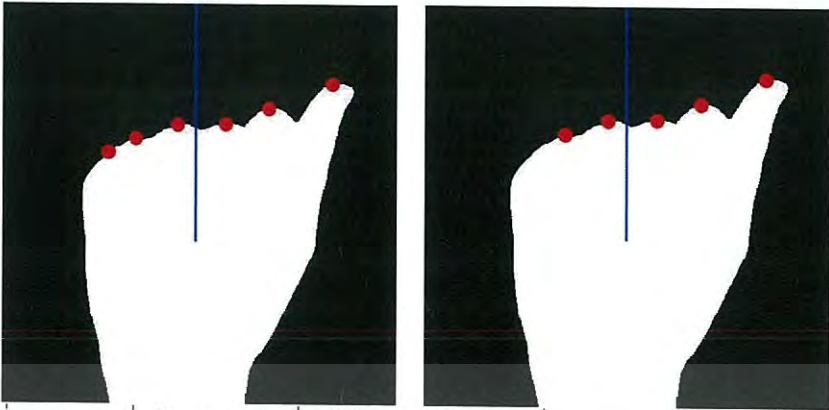
- หากพิคของฝั่งซ้ายไม่ครบ 2 พิคให้ทำการเติมนิ้วเข้าไปให้ครบโดยจะเติมเริ่มจากเส้นเดิมที่มีอยู่แล้วและขยับไปเป็นมุมเท่ากับค่ามุมเฉลี่ยที่หาได้จากนิ้วมือที่พบพิคโดยจะเติมขยับไปทางฝั่งซ้ายมือ
- เช่นเดียวกันกับฝั่งขวามือหากพบว่าพิคที่ได้ไม่ครบ 3 พิคก็ให้ทำการเติมเพิ่มเข้าไปตามค่ามุมเฉลี่ยที่หาได้จากพิคที่พบก่อนหน้านี้
- กรณีไม่พบพิคใดๆเลยบนฝ่ามือให้ทำการเติมมุมจากเส้นกลางไปทางฝั่งซ้าย 2 เส้น และไปทางฝั่งขวา 2 เส้น โดยอ้างอิงเส้นเริ่มต้นจากเส้นแบ่งฝ่ามือและขยับไปเป็นมุม 25° (แม้ว่ามุมระหว่างนิ้วปกติจะเป็น 15° หากแต่เราพบจากการทดลองว่าตัวอักษรที่จะไม่พบพิคเลยคือตัว O จึงต้องการทำให้มุมที่ได้แตกต่างไปจากทำทางตัวอักษรอื่นๆ)



รูปที่ 3.14 ภาพฝั่งซ้ายแสดงพิคที่จับได้ไม่ครบ ภาพฝั่งขวาแสดงการเติมพิคที่ขาด (จุดสีเขียว)

3) การลดพิคให้แก่ภาพที่จับพิคได้เกิน 5 นิ้ว โดยมีหลักการว่า

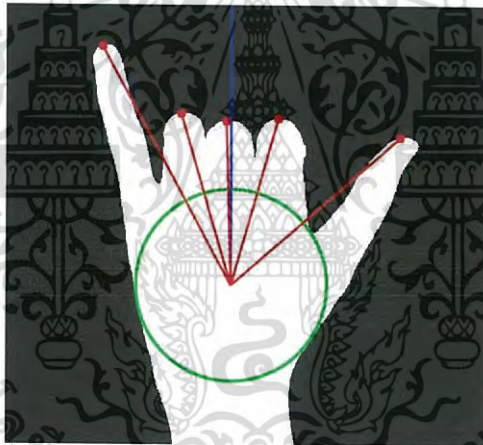
- หากพิคของฝั่งซ้ายเกิน 2 พิค ให้ทำการลบพิคที่เกินโดยอ้างอิงจากพิคที่ทำให้มุมระหว่างพิคมีค่าแตกต่างไปจากค่ามุมเฉลี่ยของพิคที่พบมากที่สุด
- เช่นเดียวกันกับฝั่งขวามือหากพบพิคเกินให้ทำการลบพิคที่เกินโดยอ้างอิงจากพิคที่ทำให้มุมระหว่างพิคมีค่าแตกต่างไปจากค่ามุมเฉลี่ยของพิคที่พบมากที่สุด



รูปที่ 3.15 ภาพฝั่งซ้ายแสดงฟีดที่จับได้เกิน 5 ฟีด ภาพฝั่งขวาแสดงการตัดฟีดที่เกินทิ้ง

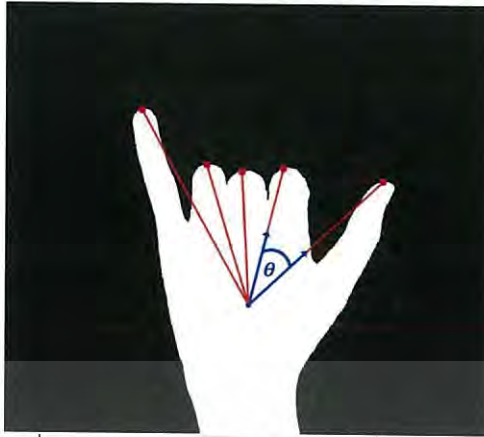
4) กรณีพบฟีดครบ 5 ฟีดแล้วให้คงค่าไว้เพื่อเข้าสู่ขั้นตอนถัดไป

3.4.2.3 หาคความยาวของนิ้วมือจากจุดเซ็นทรอยด์ โดยการหาระยะทางแบบ Euclidean distance จากนั้นทำการนอร์มัลไลซ์ (Normalize) ข้อมูลโดยอ้างอิงจากระยะทางนิ้วมือที่มีค่ามากที่สุด เพื่อแก้ปัญหการรับรูปภาพที่มีขนาดต่างกันเข้ามาประมวลผล



รูปที่ 3.16 แสดงเส้นระยะทางจากจุดเซ็นทรอยด์ถึงจุดฟีด

3.4.3 มุมระหว่างนิ้วมือ เมื่อเราได้ฟีดที่สมบูรณ์ครบทั้ง 5 ฟีด ซึ่งเปรียบเสมือนปลายนิ้วทั้ง 5 แล้วให้ทำการวัดมุมระหว่างฟีดที่พบแล้วเก็บค่าเอาไว้วิเคราะห์ผลในขั้นตอนถัดไป



รูปที่ 3.17 แสดงภาพมุมระหว่างฟีดที่พบ

3.5 การจำแนกตัวอักษรภาษามือ

ผู้ทำการทดลองได้เลือกการใช้ระบบโครงข่ายประสาทเทียม (Artificial neural network) จากโปรแกรม MATLAB 2016 เพื่อใช้ในการจำแนกตัวอักษรภาษามือภาษาอังกฤษทั้ง 26 ตัวออกจากกัน โดยสถาปัตยกรรมของโครงข่ายที่ใช้จะเป็นแบบ Two-layer feed-forward โดยใช้ฟังก์ชัน sigmoid ที่ชั้นซ่อนตัว (hidden layer) และ Softmax ฟังก์ชันที่เอาต์พุทเลเยอร์ และจะถูกสอนโดยใช้วิธี conjugate gradient backpropagation จากนั้นทำการเลือกเปอร์เซ็นต์ของพารามิเตอร์ Testing, Training, Validation และ hidden layer จำนวนที่เหมาะสมเพื่อปรับปรุงโครงข่ายนิเวศให้มีประสิทธิภาพสูงสุด เนื่องจากภาพตัวอักษรภาษามือภาษาอังกฤษนั้นมีทั้งที่เป็นสถิติกและไดนามิก เราจึงเลือกใช้การวิเคราะห์คุณลักษณะเวกเตอร์ของเฟรมเริ่มต้นและเฟรมสุดท้ายสุดของภาพ โดยจะเก็บข้อมูลวิธีโอบการท่าทางแต่ละครั้งเป็นเวลา 2 วินาที และใช้เฟรมเรท 30 เฟรม/วินาที เพื่อที่จะแบ่งได้ว่าท่าทางแบบใดเป็นสถิติกและท่าทางใดเป็นไดนามิก จากนั้นจะนำคุณลักษณะที่วิเคราะห์ได้ด้านบนของเฟรมเริ่มต้นและเฟรมสุดท้ายมาต่อกัน จึงทำให้เราได้เวกเตอร์ที่ใช้เข้านิเวศทั้งหมด 28 เวกเตอร์ เพื่อทำการแยกตัวอักษรภาษาอังกฤษ A-Z ทั้งหมด 26 ตัวโดยคุณลักษณะที่นำมาให้นิเวศเรียนรู้มีด้วยกัน 4 วิธีคือ

1. พื้นที่ของขอบภาพ 3 ด้านเพื่อแยกกลุ่มตัวอักษรออกเป็น 3 กลุ่มคือตัวอักษรในแนวตั้ง, แนวนอน ผึ่งซ้ายและแนวนอนผึ่งขวา รวมทั้งหมด 6 เวกเตอร์ (เลขไปนารี)
2. ความยาวนิ้วมือจากจุดเซ็นทรอยด์ ใช้เพื่อให้สามารถแยกได้ว่ามีนิ้วใดบ้างที่ถูกยกขึ้นและนิ้วใดที่ยังคงกำอยู่เพื่อแยกท่าทางของการกำมือและแบมือออกจากกัน รวมทั้งหมด 10 เวกเตอร์
3. มุมระหว่างนิ้วมือของเฟรมแรกและเฟรมหลัง ใช้เพื่อแยกลักษณะท่าทางมือที่มีลักษณะใกล้เคียงกันมากเช่นตัวอักษร A และ S ออกจากกัน รวมทั้งหมด 8 เวกเตอร์
4. ค่าเดลต้าของมุมระหว่างนิ้วมือ ใช้แยกตัวอักษรที่เป็นไดนามิกออกจากสถิติกได้ เพื่อให้ทราบว่ามีมือมีการขยับเปลี่ยนท่าทางรวมทั้งหมด 4 เวกเตอร์

โดยในส่วนของผลการแสดงผลนั้น โปรแกรม MATLAB จะรับภาพมือเข้ามาประมวลผลตัวละ 20 เฟรมและแสดงข้อความแจ้งเตือนผลลัพธ์ของการวิเคราะห์คุณลักษณะของภาพมือที่รับเข้ามา โดยจะแสดงเป็นตัวอักษรของท่าทางมือนั้นๆ



บทที่ 4

ผลการทดลอง

บทที่ 4 แสดงผลการทดลองที่ได้จากการวิเคราะห์แยกคุณลักษณะของภาพตัวอักษรภาษามือภาษาอังกฤษ (A-Z) ของ การทดลองที่ได้กล่าวไว้ข้างต้นโดยผู้ทำการทดลองได้นำข้อมูลเข้าเรียนรู้ในโครงข่ายระบบประสาทเทียมและได้มีการปรับค่าพารามิเตอร์ต่างๆ โดยผลการทดลองกระบวนการมีดังนี้

- 1.ผลการทดลองจากการเลือกใช้กล้อง WEBCAM
- 2.ผลการทดลองจากการเพิ่มและลดนิ้วมือ
- 3.ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม

4.1 ผลการทดลองจากการเลือกใช้กล้อง WEBCAM

ผู้ทำการทดลองพบว่าการใช้กล้อง webcam ในการรับภาพนั้นมีผลกระทบหลายอย่างด้วยกันทั้งจากสีและความสว่างของภาพที่ได้จะเปลี่ยนแปลงอยู่เสมอในสภาพแวดล้อมที่ต่างกันทำให้ผู้ทำการทดลองไม่สามารถใช้วิธีค้นหาตำแหน่งมือจากการตรวจจับสีผิวได้ จึงจำเป็นต้องใช้การนำภาพมือที่ได้มาลบกับภาพพื้นหลังในตอนเริ่มต้นเพื่อที่จะได้ภาพมือที่สมบูรณ์ออกมาซึ่งทำให้ต้องมีการจัดฉากเพื่อถ่ายอยู่เสมอ อีกทั้งเฟรมเรทของภาพนั้นจะมีความไม่เท่ากันหากความสว่างของสภาพแวดล้อมที่ถ่ายไม่เพียงพอทำให้บางครั้งต้องมีการเพิ่มระยะเวลาในการถ่ายเพิ่มขึ้นเพื่อให้ได้ข้อมูล Database ที่ได้มีจำนวนเท่ากับสภาพแวดล้อมที่สว่างปกติ อีกทั้งข้อมูลที่จะสามารถนำมาวิเคราะห์ได้จะเป็นการใช้ข้อมูลของขอบภาพเท่านั้น

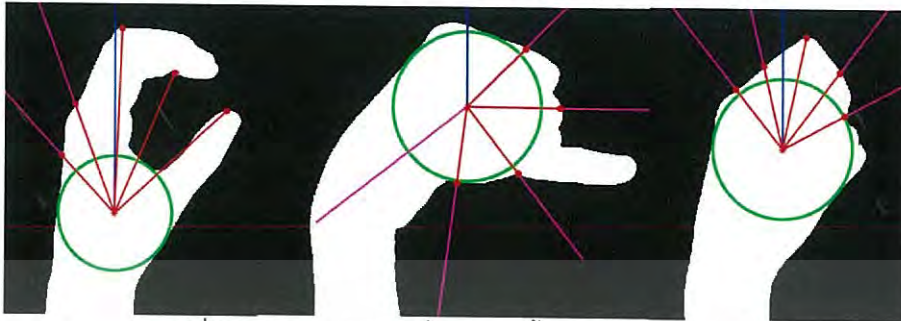


รูปที่ 4.1 แสดงภาพมือที่ได้จากการถ่ายในสภาพแวดล้อมต่างๆ

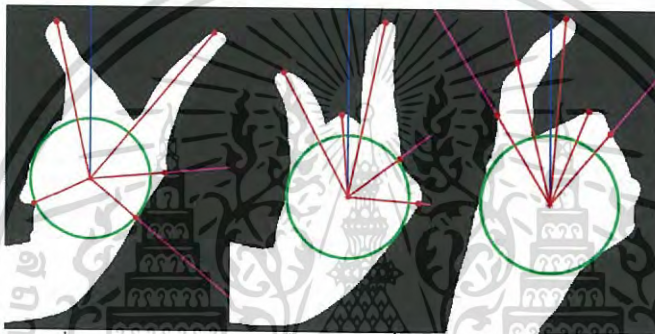
4.2 ผลการทดลองจากการเพิ่มและลดนิ้วมือ

เนื่องจากเหตุผลที่กล่าวไปแล้วข้างต้นว่าตัวอักษรบางไม่สามารถมองเห็นนิ้วมือที่ชัดเจนได้ซึ่งทำให้ยากต่อการวิเคราะห์ลักษณะของท่าทางมือ ผู้ทำการทดลองจึงได้มีการเพิ่มและลดฟิคของภาพที่จับได้เพื่อให้ได้นิ้วมือที่ถูกต้องครบทั้งห้านิ้ว โดยผู้ทำการทดลองพบว่าตัวอักษรที่มีปัญหาทำให้ผลการทดลองไม่

เป็นไปตามที่กำหนดคือตัวอักษรที่ไม่สามารถเห็นนิ้วมือได้อย่างชัดเจนทำให้การเติมและลดพิกมีความผิดพลาดเนื่องจากไม่พบนิ้วที่แท้จริง โดยตัวที่มีปัญหาทั้งหมดได้แก่ตัว C J O P Q X



รูปที่ 4.2 แสดงภาพการเติมและลดนิ้วมือ c (ซ้าย) j (กลาง) o (ขวา)



รูปที่ 4.3 แสดงภาพการเติมและลดนิ้วมือ P (ซ้าย) Q (กลาง) X (ขวา)

4.3 ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม

พบว่า การนำข้อมูลที่ได้จากการผ่านคุณลักษณะต่างๆ ในบทที่ 3 นั้นจะให้ผลลัพธ์ที่แตกต่างกันไป เมื่อมีการปรับค่าพารามิเตอร์ Testing, Validation และ Hidden layer โดยผู้ทำการทดลองได้คงค่า Training ไว้ที่ 70 % เนื่องจากเป็นค่าที่เหมาะสมมากที่สุด ซึ่งพบว่า

4.4.1 การเลือกใช้ค่า Hidden layer เมื่อกำหนดให้ค่า Testing และ Validation เท่ากันและกำหนด Hidden layer เป็น 100 และ 10 ตามลำดับ ผลจากการทดลอง 3 ครั้งเป็นไปตามตารางที่ 4.1 โดยค่าเฉลี่ยความถูกต้องของ Hidden layer = 100 และ Hidden layer = 10 มีค่าเป็น 89.97% และ 72.21% ตามลำดับ จะเห็นได้ว่าการค่าเปอร์เซ็นต์ความถูกต้องของการใช้ Hidden layer = 100 ของการทดลองทั้ง 3 ครั้งจะมากกว่าการใช้ Hidden layer = 10 ดังนั้นผู้ทำการทดลองจึงเลือกใช้ Hidden layer = 100 ในการเข้าเรียนรู้ในโครงข่ายประสาทเทียม

| | | Accuracy (%) | | | |
|-----------------------|-----|--------------|-------|-------|---------|
| | | 1 | 2 | 3 | Average |
| Hidden layer value | 100 | 86.77 | 86.85 | 90.31 | 87.97 |
| | 10 | 66.54 | 72.77 | 77.31 | 72.21 |

ตารางที่ 4.1 แสดงค่าเปอร์เซ็นต์ความถูกต้องของการใช้ Hidden layer ที่แตกต่างกัน

4.4.2 การเลือกใช้ค่า Testing และ Validation เนื่องจากการทดลองนำข้อมูลเข้าไปเรียนรู้ในโครงข่ายประสาทเทียมในแต่ละครั้งนั้นจะให้ค่าเปอร์เซ็นต์ความถูกต้องที่ต่างกัน ผู้ทำการทดลองจับคู่ใช้ค่า Testing และ Validation เท่ากับ 5 , 15 และ 25 โดยใช้ Hidden layer = 100 ได้ผลการทดลองดังตารางที่ 4.2 จะเห็นได้ว่าค่าความถูกต้องของการใช้ค่า Testing และ Validation ที่เท่ากันคือ 15% จะให้ความถูกต้องสูงที่สุดคือ 90.42% รองลงมาคือการใช้ Testing มากกว่า Validation โดยมีค่าความถูกต้อง 82.93% และที่น้อยที่สุดคือการใช้ค่า Testing น้อยกว่าค่า Validation โดยมีค่าความถูกต้อง 73.77% ผู้ทำการทดลองจึงเลือกใช้ค่านี้เพื่อทำการทดลองต่อไป

| Testing (%) | Validation (%) | Accuracy (%) |
|-------------|----------------|--------------|
| 5 | 25 | 73.77 |
| 15 | 15 | 90.42 |
| 25 | 5 | 82.93 |

ตารางที่ 4.2 แสดงการเลือกใช้ค่า Testing และ Training

4.4.3 ผลการทดลองจากการเลือกค่าพารามิเตอร์และคุณลักษณะที่เหมาะสมเพื่อเข้าเรียนรู้ใน
โครงข่ายประสาทเทียมเป็นดังตารางที่ 4.3

| Alphabet | Accuracy (%) | | | |
|----------|--------------|-----|-----|---------|
| | 1 | 2 | 3 | Average |
| A | 99 | 100 | 99 | 99.3 |
| B | 100 | 100 | 100 | 100 |
| C | 97 | 100 | 97 | 98 |
| D | 98 | 98 | 95 | 97 |
| E | 87 | 98 | 95 | 93.3 |
| F | 100 | 100 | 100 | 100 |
| G | 100 | 100 | 100 | 100 |
| H | 100 | 100 | 100 | 100 |
| I | 100 | 100 | 100 | 100 |
| J | 90 | 95 | 85 | 90 |
| K | 97 | 95 | 100 | 97.3 |
| L | 80 | 95 | 100 | 92 |
| M | 70 | 65 | 72 | 69 |
| N | 92 | 80 | 75 | 82.3 |
| O | 95 | 98 | 92 | 95 |
| P | 85 | 82 | 70 | 79 |
| Q | 98 | 100 | 95 | 97.7 |
| R | 100 | 100 | 100 | 100 |
| S | 100 | 100 | 100 | 100 |
| T | 95 | 85 | 75 | 85 |
| U | 100 | 100 | 100 | 100 |
| V | 100 | 100 | 100 | 100 |
| W | 80 | 90 | 89 | 86.3 |
| X | 95 | 97 | 95 | 95.7 |
| Y | 100 | 100 | 100 | 100 |
| Z | 100 | 100 | 100 | 100 |

ตารางที่ 4.3 แสดงเปอร์เซ็นต์ความถูกต้องของตัวอักษร A-Z

จากตารางที่ 4.3 จะเห็นได้ว่าจากการทดลองทั้งสามครั้งได้เปอร์เซ็นต์ความถูกต้องใกล้เคียงกันคือประมาณ 95% ตัวอักษรที่มีเปอร์เซ็นต์ความถูกต้องน้อย ได้แก่ J, M, N และ ซึ่งมีเปอร์เซ็นต์ความถูกต้อง 64.67%, 90%, 80.67% และ 54% ตามลำดับ โดยตัวอักษร J ตอบผิดเป็น I , ตัวอักษร M และ N จะตอบผิดสลับกันและตัวอักษร P ตอบผิดเป็น Q จะเห็นได้ว่าเนื่องจากตัวอักษรดังกล่าวนี้มีความคล้ายคลึงกันมากตัวอย่างเช่น

- ตัวอักษร J และ I ซึ่งมีเฟรมแรกของภาพเหมือนกันทุกประการ



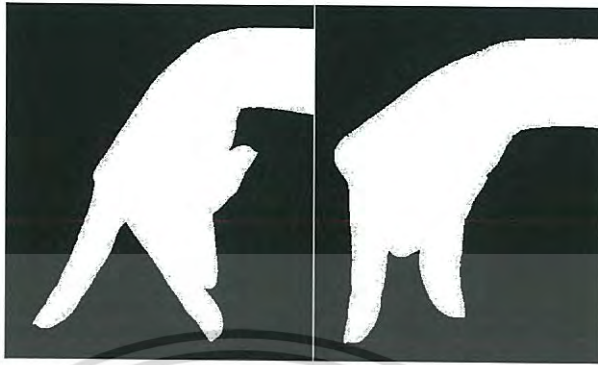
รูปที่ 4.4 แสดงภาพเฟรมแรกของตัว I (ซ้าย) และ J (ขวา)

- ตัวอักษร M และ N ซึ่งมีลักษณะคล้ายกันทั้งเฟรมแรกและเฟรมสุดท้าย



รูปที่ 4.5 แสดงภาพเฟรมแรกของตัวอักษร M (ซ้าย) และ N (ขวา)

- ตัวอักษร P และ Q ซึ่งมีลักษณะคล้ายกันทั้งเฟรมแรกและเฟรมสุดท้าย



รูปที่ 4.6 แสดงภาพเฟรมแรกของตัวอักษร P (ซ้าย) และ Q (ขวา)



บทที่ 5

สรุปและวิจารณ์

5.1 ปัญหาที่เกิดขึ้นกับโครงการ

5.1.1 ด้านฮาร์ดแวร์(Hardware)

ภาพที่จะนำมาวิเคราะห์นั้นต้องมีการจัดแสงและฉากให้เหมาะสมนั้นคือต้องทำให้เห็นแต่ภาพมือที่ชัดเจนไม่มีเงาเพื่อให้ภาพที่เก็บมาได้สามารถแยกมือออกจากพื้นหลังได้ดี มีเช่นนั้นการวิเคราะห์ผลจะผิดเพี้ยนไปซึ่งอ้างอิงจากรายที่ 4.3 จะเห็นว่ามิตัวอักษรบางตัวมีความถูกต้อง 100 เปอร์เซ็นต์เนื่องจากภาพมือที่รับเข้ามานั้นเป็นภาพมือที่ตรงทำให้สามารถประมวลผลได้ถูกต้องทั้งหมด ในขณะที่ตัวอักษรบางตัวมีความผิดพลาดมากเนื่องจากเป็นตัวอักษรกำมือซึ่งต้องทำมือให้ตรงมากจึงจะได้ค่าที่ถูกต้องออกมาเนื่องจากตัวอักษรกำมือส่วนใหญ่มีลักษณะที่คล้ายกัน

5.1.2 ด้านซอฟต์แวร์(Software)

เนื่องจากการคุณลักษณะที่สกัดจากมือนั้นต้องมีความไวต่อองศาการเปลี่ยนแปลงของมือมาก ดังนั้นหากภาพมือที่รับเข้ามาเอียงเพียงเล็กน้อยก็จะทำให้ผลของการจำแนกตัวอักษรผิดพลาดไปอย่างมาก

5.2 สรุปผลการทดลอง

การวิเคราะห์จำแนกตัวอักษรภาษาอังกฤษของภาษามือออกจากกันพบว่าวิธีที่นำมาใช้สกัดคุณลักษณะมีทั้งสิ้น 4 วิธีซึ่งประกอบด้วยพื้นที่ของขอบภาพ 3 ด้าน, ความยาวนิ้วมือจากจุดเซ็นทรอยด์, มุมระหว่างนิ้วมือของเฟรมแรกและเฟรมหลัง และค่าเดลต้าของมุมระหว่างนิ้วมือของเฟรมหน้าและเฟรมหลัง รวมทั้งหมด 28 คุณลักษณะ โดยข้อมูลทั้งหมดถูกแบ่งออกเป็น 2 กลุ่มคือชุดข้อมูลสำหรับการ Training เพื่อสร้างโมเดลและ ชุดข้อมูลสำหรับ Testing เพื่อทดสอบประสิทธิภาพของโมเดลและ Validation ในขั้นตอนของการสร้างโมเดลเราเลือกใช้วิธีโครงข่ายประสาทเทียมแบบ Two-layer feed-forward บนโปรแกรม MATLAB โดยสามารถแยกกลุ่มตัวอักษรออกจากกันได้อย่างมีประสิทธิภาพด้วยค่าความถูกต้องเท่ากับร้อยละ 95 จะเห็นได้ว่าความถูกต้องจากการทดลองมีประสิทธิภาพสูงแม้ว่าจะมีตัวอักษรที่มีความผิดพลาดอยู่บ้างแต่ก็อยู่ในเกณฑ์ที่ยอมรับได้ อีกทั้งการประมวลผลเป็นไปด้วยความถูกต้องและรวดเร็ว

บรรณานุกรม

- [1] Wing Kwong Chung, Xinyu Wu, Yangsheng Xu ,“ A Realtime Hand Gesture Recognition based on Haar Wavelet Representation” Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, pp.336-341 , 2008
- [2] Erdem Yörük, Ender Konukoğlu, Bülent Sankur, Senior Member, IEEE, Jérôme Darbon, “Shape-Based Hand Recognition”, IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 15, no. 7,pp.1803-1806, JULY 2006,
- [3] Rajeshree Rokade, Dharpal Doye, Manesh Kokare, “Hand Gesture Recognition by Thinning Method”, International Conference on Digital Image Processing ,SGGS Institute of Engg. andTech. Nanded (INDIA), pp.285-287
- [4] Wikipedia, the free encyclopedia. (2017, 9 November). Sign language. Retrieved from https://en.wikipedia.org/wiki/Sign_language
- [5] Su Dongcai. (2009, 16 November . Euclidean skeleton. Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/25865-euclidean-skeleton>



ภาคผนวก

```

function ASLDetection
cam=webcam('Logitech HD Webcam C310');
cam.Resolution = '640x480';

hFig = figure('Visible','on', 'Menu','none', 'Name','American Sign Language
Detection','NumberTitle','off','Resize','off', 'Position',[0 0 1240 650]);
movegui(hFig,'center')

    hBtnGrp = uibuttongroup('Position',[0.01 0.01 0.1 0.98],'Title','MENU',
'Units','Normalized');
% hBtnGrp = uibuttongroup('Position',[0 0 0.3 1], 'Units','Normalized');
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 550
100 40], 'String','START','Callback',@bt_start)
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 500
100 40], 'String','Show Camera','Callback',@bt_show)
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 450
100 40], 'String','Capture BG','Callback',@bt_bg)
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 400
100 40], 'String','EXIT','Callback',@bt_exit)

% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 550
100 40], 'String','START','Tag','start')
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 500
100 40], 'String','Show Camera','Tag','show')
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 450
100 40], 'String','Capture BG','Tag','capture')
% uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 400
100 40], 'String','EXIT','Tag','exit')

```

```

uicontrol('Style','Radio', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 550 100 40],
'String','START','Tag','start')
uicontrol('Style','Radio', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 500 100 40],
'String','Show Camera','Tag','show')
uicontrol('Style','Radio', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 450 100 40],
'String','Capture BG','Tag','capture')
uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 300 100
40], 'String','DONE','Callback',@bt_done)
uicontrol('Style','pushbutton', 'Parent',hBtnGrp, 'HandleVisibility','off', 'Position',[10 250 100
40], 'String','EXIT','Callback',@bt_exit)
set(hFig, 'Visible','on')

```

```

tend=0;
while 1
% IM=snapshot(cam);
% IM=imrotate(IM,-90);
if tend==1;
break
else
IM=snapshot(cam);
IM=imrotate(IM,-90);
uicontrol('Parent',hFig, 'HandleVisibility','off');
imshow(IM);title('Show Webcam');
line([240 240],[0 640],'LineWidth',1,'Color','r');
end
end

```

```

function bt_done(src,ev)
tend==1;
switch get(get(hBtnGrp,'SelectedObject'),'Tag')

```

```

case 'start',
    disp('start')
    disp('Start Record')
%% WEBCAM INPUT PROCESS
for i=1:20
    sn=snapshot(cam);
    Snap(:,:,i)=imrotate(sn,-90);
    uicontrol('Parent',hFig, 'HandleVisibility','off')
    imshow(Snap(:,:,i));title(i);
    line([240 240],[0 640],'LineWidth',1,'Color','r');
% hold on
    text(320,25,'Recording','Color','red','FontSize',20)
end

%% FEATURE EXTRACTION
BG=imread('C:\Project 4C\Final\BG.png');
IMG = Snap(:,:,2);
[out]=imgretouch(IMG,BG);
uicontrol('Parent',hFig, 'HandleVisibility','off')
imshow(out);title('IMG Start');
    out=imcomplement(out);
    Start=Handprocess(out);

    IMG = Snap(:,:,19);
    [out]=imgretouch(IMG,BG);
    uicontrol('Parent',hFig, 'HandleVisibility','off')
    imshow(out);title('IMG Stop');
        out=imcomplement(out);
        Stop=Handprocess(out);
        del=abs(Start(:,7:10) - Stop(:,7:10));

```

```
Vector=[transpose(Start); transpose(Stop(:,7:15)); transpose(del)];
```

```
%% Classification
```

```
[y1] = Mix2(Vector);
```

```
tend = abs(1-y1);
```

```
Answer = find(tend==min(tend));
```

```
Answer = char(Answer + 64);
```

```
pause(0.1)
```

```
text(600,250,Answer,'Color','red','FontSize',100)
```

```
uiwait(msgbox(Answer,'Success'));
```

```
% msg = msgbox(Answer,'Success');
```

```
% set(h, 'position', [450 200 100 80]); %makes box bigger
```

```
% ah = get( h, 'CurrentAxes' );
```

```
% ch = get( ah, 'Children' );
```

```
% set( ch, 'FontSize', 25); %makes text bigger
```

```
% tx=tx+1;
```

```
case 'show',
```

```
disp('Show Webcam')
```

```
while 1
```

```
IM=snapshot(cam);
```

```
IM=imrotate(IM,-90);
```

```
if tend==2;
```

```
break
```

```
end
```

```
uicontrol('Parent',hFig, 'HandleVisibility','off')
```

```
imshow(IM);title('Show Webcam');
```

```
line([240 240],[0 640],'LineWidth',1,'Color','r');
```

```
end
```

```

case 'capture',
    disp('Capture Background')
    IMBG=snapshot(cam);
    IMBG=imrotate(IMBG,-90);
    imwrite(IMBG,fullfile('C:\Project 4\Final\','BG.png'));
end
end

function bt_exit(src,ev)
    tend=2;
    close all
end
clear all
end

```

