

การสกัดคุณลักษณะแบบใหม่เพื่อใช้รู้จำภาษามือแบบ 2 มิติ โดยใช้โครงข่าย
ประสาทเทียม

A NOVEL FEATURE EXTRACTION FOR 2-D AMERICAN SIGN LANGUAGE
RECOGNITION USING ARTIFICIAL NEURAL NETWORKS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2562

KMITL-2019-EN-M-040-108

การสกัดคุณลักษณะแบบใหม่เพื่อใช้รู้จำภาษามือแบบ 2 มิติ โดยใช้โครงข่าย
ประสาทเทียม

A NOVEL FEATURE EXTRACTION FOR 2-D AMERICAN SIGN LANGUAGE
RECOGNITION USING ARTIFICIAL NEURAL NETWORKS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2562

KMITL-2019-EN-M-040-108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A NOVEL FEATURE EXTRACTION FOR 2-D AMERICAN SIGN LANGUAGE
RECOGNITION USING ARTIFICIAL NEURAL NETWORKS



ARIYA THONGTAWEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2019
KMITL-2019-EN-M-040-108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2019

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การสกัดคุณลักษณะแบบใหม่เพื่อใช้รู้จำภาษามือแบบ 2 มิติ โดยใช้ โครงข่ายประสาทเทียม
นักศึกษา	นางสาวอริยา ทองทวี
รหัสประจำตัว	61601011
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
พ.ศ	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. ดร.ยุทธนา คิดใจเดียว

บทคัดย่อ

ภาษามือถือเป็นการสื่อสารทางกายภาพแบบหนึ่งซึ่งช่วยถ่ายทอดความหมายแทนการใช้เสียงพูด เพื่อแสดงความคิดเห็นของผู้สื่อสาร วิทยานิพนธ์ฉบับนี้จึงนำเสนออัลกอริทึมที่เข้าใจได้ง่ายและมีประสิทธิภาพซึ่งใช้สกัดคุณลักษณะของมือเพื่อให้โปรแกรมสามารถรู้จำตัวอักษรและคำในภาษาอังกฤษ จากท่าทางซึ่งมีทั้งแบบหยุดนิ่งและเคลื่อนไหวได้ โดยทำการแบ่งการทดลองออกเป็น 2 ส่วนได้แก่ การสกัดคุณลักษณะเพื่อใช้รู้จำภาษามือที่เป็นตัวอักษรภาษาอังกฤษซึ่งประกอบไปด้วยคุณลักษณะที่แตกต่างกัน 4 คุณลักษณะเพื่อจำแนกตัวอักษรภาษาอังกฤษ 26 ตัว และส่วนที่สองคือการสกัดคุณลักษณะเพื่อใช้รู้จำภาษามือที่เป็นคำในภาษาอังกฤษประกอบไปด้วยคุณลักษณะที่แตกต่างกัน 11 คุณลักษณะเพื่อจำแนกคำที่ใช้ในชีวิตประจำวันจำนวน 27 คำ หลังจากสกัดคุณลักษณะแล้วจะข้อมูลจะถูกนำเข้าไปประมวลผลในระบบโครงข่ายประสาทเทียมเพื่อวิเคราะห์และจำแนกท่าทางที่รับเข้ามา ผลจากการทดลองพบว่าทั้งสองการทดลองมีค่าความถูกต้องสูงถึงประมาณร้อยละ 94

Thesis	A Novel Feature Extraction For 2-D American Sign Language Recognition using Artificial Neural Network
Student	Miss Ariya Thongtawee
Student ID.	61601011
Degree	Master of Engineering
Program	Electronics Engineering
Year	2019
Thesis Advisor	Asst. Prof. Dr. Yuttana Kitjaidure

Abstract

Sign language is physical communication for contributing the meaning instead of using voice to demonstrate communicator's opinion. This paper introduces simple and efficient algorithms for feature extraction to recognize American Sign Language alphabets and words which have both static and dynamic gestures letting the program to classify the signs into the specific alphabets and words. The experiments are divided into 2 sections: first is feature extraction for American Sign Language alphabets which consists of 4 different techniques to classify the sign into 26 groups (A-Z) and the second is feature extraction for American Sign Language words which consists of 11 different techniques to classify gestures into 27 groups (words used in daily life). After extracting features from input videos, an Artificial Neural Network (ANN) is used to classify the signs. The results of these experiments are achieved up to around 94%

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.ยุทธนา คิดใจเดียว ที่คอยให้ความช่วยเหลือและแนะนำแนวทางในการแก้ปัญหาที่เกิดขึ้นในการทำวิจัย อีกทั้งยังช่วยตรวจสอบแก้ไขความถูกต้องของผลงานวิชาการและรูปเล่มวิทยานิพนธ์จนเสร็จสมบูรณ์ นอกจากนี้ยังคอยปลุกฝังให้ฝึกฝนทักษะการเรียนรู้ด้วยตนเองและพร้อมที่จะหาข้อมูลมาคอยแนะนำอยู่เสมอ

ขอขอบคุณนางสาวอรอมล พิมเสนาะ เพื่อนคู่ใจปรเจศสมัยปริญญาตรีที่คอยให้คำแนะนำด้านการเขียนโปรแกรมแมทแลป และสอนเทคนิคการเรียบเรียงและจัดเก็บข้อมูลซึ่งทำให้ง่ายต่อการวิเคราะห์ผล อีกทั้งยังเป็นบุคคลผู้ซึ่งเป็นหนึ่งในฐานข้อมูลเพื่อใช้ในการทำวิทยานิพนธ์ฉบับนี้

ขอขอบคุณนางสาวพิมพ์พร เหมยน้อย ที่คอยให้คำแนะนำเกี่ยวกับความรู้เรื่อง Pattern recognition และความรู้ในเชิงวิชาการต่างๆซึ่งทำให้ตัวข้าพเจ้าสามารถเข้าใจหลักการทำงานของสก็ดคุณลักษณะได้ดียิ่งขึ้น

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้มารดาและครอบครัวเป็นที่รักและเคารพยิ่ง ตลอดจนครูบาอาจารย์และผู้มีพระคุณที่เคารพทุกท่าน

อริยา ทองทวี

สารบัญ

หน้า	
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	IX
สารบัญตาราง.....	XIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนของการศึกษา.....	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ภาษามือ (Sign Language).....	4
2.1.1 ตัวอักษรภาษามือ (Sign Language Alphabets).....	4
2.1.2 คำในภาษามือ (Sign Language Words)	6
2.2 การระบุเฟรมเริ่มต้นและเฟรมสิ้นสุดของท่าทาง	8
2.2.1 ตรวจจับสีผิวภายในเฟรม (Skin detection)	9
2.2.2 การคำนวณจำนวนวัตถุที่เป็นสีผิว (Counting Objects)	10
2.3 การตรวจจับมือ (Hand Detection).....	11
2.3.1 กระบวนการเปลี่ยนแปลงภาพสีเป็นภาพระดับสีเทา	11
(Gray-Scale Image Transform)	
2.3.2 การเปรียบเทียบจุดสีเพื่อหาตำแหน่งมือ (Background Subtraction).....	11
2.3.3 การตรวจจับใบหน้าด้วยวิธี Viola-Jones (Viola-Jones Algorithm).....	12

สารบัญ (ต่อ)

2.4 การปรับปรุงภาพ (Enhancement).....	16
2.4.1 กระบวนการแบ่งส่วน (Segmentation Techniques).....	16
2.4.1.1 กระบวนการแบ่งโดยใช้พิกเซล (Pixel Based Segmentation)	16
2.4.1.2 กระบวนการแบ่งโดยขอบ (Boundary based Segmentation)	18
2.4.1.3 กระบวนการแบ่งโดยพื้นที่ (Region based)	21
2.4.1.4 กระบวนการแบ่งโดยโมเดล (Model based)	22
2.4.2 กระบวนการกำจัดสิ่งรบกวน (Noise Reduction)	24
2.4.2.1 Closing Operation.....	24
2.4.2.1 Fill Holes Operation	24
2.4.2.3 Region Selection.....	25
2.5 การสกัดคุณลักษณะ (Feature Extraction).....	25
2.5.1 การตรวจจับลำดับการเคลื่อนไหว (Motion Sequences detection).....	26
2.5.1.1 การประมาณค่าในขงด้วยเส้นโค้ง (Cubic Spline).....	26
2.5.1.2 เอนโทรปี (Information Entropy).....	29
2.5.1.3 การตรวจจับเส้นทางของมือ (Hand trajectory detection).....	30
2.5.2 ชนิดของคุณลักษณะ (Feature-based).....	31
2.5.2.1 Intra-Feature.....	31
2.5.2.2 Inter-Feature.....	32
2.6 การวิเคราะห์องค์ประกอบหลัก (Principal Components Analysis).....	32
2.6.1 ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation).....	33
2.6.2 ค่าความแปรปรวนของข้อมูล (Variance).....	34
2.6.3 โควาเรียนซ์ (Covariance).....	34
2.6.4 โควาเรียนซ์เมทริกซ์ (Covariance Matrix).....	35
2.6.5 เวกเตอร์ลักษณะเฉพาะ (Eigen Vectors).....	35
2.6.6 ค่าลักษณะเฉพาะ (Eigen Values).....	35
2.7 โครงข่ายประสาทเทียม (Artificial neural network).....	40
2.7.1 ประเภทของการเรียนรู้โครงข่ายประสาทเทียม.....	41
2.7.1.1 Supervised Learning.....	41

สารบัญ (ต่อ)

2.7.1.2 Unsupervised Learning.....	41
2.7.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม.....	42
2.7.2.1 โครงข่ายการส่งข้อมูลแบบไม่ย้อนกลับ (Feedforward Network).....	42
2.7.2.2 โครงข่ายการส่งข้อมูลแบบย้อนกลับ (Feedback Network).....	42
2.7.2.3 อัลกอริทึมแบ็กพรอพากาชัน (Back propagation Algorithm).....	43
2.7.3 ฟังก์ชันถ่ายโอน (Transfer Function).....	44
2.7.4 ประโยชน์ของโครงข่ายประสาทเทียม.....	46
บทที่ 3 วิธีการดำเนินงาน.....	47
3.1 ระบบรู้จำตัวอักษรภาษาอังกฤษ	47
3.1.1 การรับภาพ (Input image).....	48
3.1.2 การปรับปรุงคุณภาพของภาพ (Enhancement).....	48
3.1.2.1 ภาพระดับสีเทา (Grayscale image).....	48
3.1.2.2 ภาพขาว-ดำ (Binary image).....	48
3.1.2.3 การลบพื้นที่ขนาดเล็กกว่าที่กำหนด (Remove small object).....	49
3.1.2.4 การเติมพื้นที่หรือรูของภาพ (Fill image regions and holes).....	49
3.1.3 การระบุตำแหน่งของมือ (Identify Hand Position).....	50
3.1.3.1 การหาตำแหน่งของข้อมือ.....	50
3.1.3.2 การหาวงกลมบนฝ่ามือ.....	50
3.1.3.2 การหาวงกลมบนฝ่ามือ.....	50
3.1.4 การวิเคราะห์คุณลักษณะของภาพ (Feature Extraction).....	51
3.1.4.1 พื้นที่ของขอบภาพ 3 ด้าน (NWE).....	51
3.1.4.2 ความยาวนิ้วมือจากจุดเซ็นทรอยด์ (Fcen)	53
3.1.4.3 มุมระหว่างนิ้วมือ (AngF)	55
3.1.4.4 ค่าเดลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสุดท้าย.....	55
(delAng)	
3.1.5 การจำแนกตัวอักษรภาษามือ (Classification).....	57
3.2 ระบบรู้จำคำภาษาอังกฤษ.....	58

สารบัญ (ต่อ)

3.2.1 การหาช่วงของท่าทาง (Identifying Started and Stopped Gesture).....	59
3.2.2 การคัดเลือกเฟรมหลัก (Motion Sequences Detection).....	59
3.2.3 การระบุตำแหน่งของมือ (Identify Hand Position).....	60
3.2.3.1 การหาตำแหน่งของใบหน้า (Face Position).....	61
3.2.3.2 การตรวจจับสีผิว (Skin Detection).....	61
3.2.3.3 กำจัดใบหน้า (Face elimination)	61
3.2.3.4 การกำจัดสัญญาณรบกวน (Noises Removing).....	62
3.2.4 การวิเคราะห์คุณลักษณะของท่าทาง (Feature Extraction).....	62
3.2.4.1 คุณลักษณะภายในกลุ่ม (Intra-Feature).....	64
3.2.4.2 คุณลักษณะระหว่างกลุ่ม (Inter-Feature).....	67
3.2.5 การปรับขนาดของข้อมูล (Normalized data).....	68
3.2.6 การจำแนกคำศัพท์ภาษามือ (Classification).....	69
3.2.6.1 ระบบรู้จำคำโดยใช้เฟรมทั้งหมด.....	69
3.2.6.2 ระบบรู้จำคำโดยใช้เฟรมหลัก.....	69
บทที่ 4 ผลการทดลอง.....	70
4.1 ระบบรู้จำตัวอักษรภาษาอังกฤษ.....	70
4.1.1 ผลการทดลองจากการเพิ่มและลดนิ้วมือ.....	70
4.1.2 ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม.....	71
4.1.2.1 การเลือกใช้ค่า Hidden layer.....	71
4.1.2.2 การเลือกใช้ค่า Train : Test.....	71
4.1.2.3 ผลค่าความถูกต้องที่ได้จากการทดลอง.....	72
4.1.3 ผลทดลองเปรียบเทียบกับงานวิจัยอื่นที่เกี่ยวข้อง.....	74
4.2 ระบบรู้จำคำในชีวิตประจำวัน.....	75
4.2.1 ผลการทดลองคำนวณคุณลักษณะของท่าทางโดยใช้เฟรมทั้งหมด.....	75
4.2.2 ผลการทดลองคำนวณคุณลักษณะของท่าทางโดยใช้เฟรมหลัก.....	78
บทที่ 5 สรุปและวิจารณ์ผลงานวิจัย.....	86

สารบัญ (ต่อ)

5.1 สรุปและวิจารณ์ระบบรู้จำตัวอักษรภาษาอังกฤษ.....	86
5.1.1 วิจารณ์ผลงานวิจัย.....	86
5.1.2 สรุปผลการทดลองงานวิจัย.....	87
5.2 สรุปและวิจารณ์ระบบรู้จำคำในชีวิตประจำวัน.....	87
5.2.1 วิจารณ์ผลงานวิจัย.....	87
5.2.2 สรุปผลการทดลองงานวิจัย.....	88
เอกสารอ้างอิง.....	90
ภาคผนวก.....	93
ประวัติผู้วิจัย.....	112



สารบัญรูปภาพ

รูปที่ 2.1 ตัวอย่างท่าทางตัวอักษรภาษามือแบบอเมริกัน.....	5
รูปที่ 2.2 ท่าทางตัวอักษรภาษามือแบบอเมริกัน.....	7
รูปที่ 2.3 ขั้นตอนการทำงานโดยรวมของระบบรู้จำภาษามือแบบอเมริกัน.....	8
รูปที่ 2.4 Block diagram ของขั้นตอนการระบุเฟรมเริ่มต้นและสิ้นสุด.....	9
รูปที่ 2.5 แสดงการเปรียบเทียบสีผิวโดยใช้ช่วงดังกล่าว.....	10
รูปที่ 2.6 การคำนวณจำนวนวัตถุภายในภาพ.....	10
รูปที่ 2.7 แสดงผลภาพที่เกิดจากการเปรียบเทียบจุดเพื่อลบพื้นหลัง.....	12
รูปที่ 2.8 Input image และ Integral image.....	13
รูปที่ 2.9 กลุ่มรูปแบบของ Haar-like feature.....	14
รูปที่ 2.10 ขั้นตอนวิธีการเรียนรู้แบบ AdaBoost.....	15
รูปที่ 2.11 ตัวจำแนกประเภทแบบ cascaded.....	16
รูปที่ 2.12 ตัวอย่างภาพเพื่อใช้แบ่งวัตถุด้วยพิกเซล.....	17
รูปที่ 2.13 ภาพซึ่งถูกกำจัดพิกเซลสีฟ้าโดยค่าเทรชโฮล.....	17
รูปที่ 2.14 ภาพวัตถุสุดท้ายหลังจากปรับปรุงคุณภาพของภาพ.....	18
รูปที่ 2.15 A.) รูปต้นฉบับ B.) Edge strength and Edge orientation C.) Canny Operator.....	18
รูปที่ 2.16 Gaussian Filtered.....	19
รูปที่ 2.17 กระบวนการหา Edge strength.....	19
รูปที่ 2.18 ช่วงการเปลี่ยนค่าเพื่อใช้ใน Edge orientation	20
รูปที่ 2.19 วิธีการ Non-maximum Suppression.....	20
รูปที่ 2.20 กระบวนการทั้ง 5 ขั้นตอนของ Canny Method.....	21
รูปที่ 2.21 การประยุกต์ใช้วิธี Region Growing.....	22
รูปที่ 2.22 ตัวอย่างการแบ่งส่วนโดยใช้ Gaussian model.....	23
รูปที่ 2.23 แสดง 3x3 pixels Closing Operation.....	24
รูปที่ 2.24 Fill Holes 4-connectivity Operation.....	24
รูปที่ 2.25 Fill Holes 8-connectivity Operation.....	25
รูปที่ 2.26 การเลือกกลุ่มของข้อมูลที่มากกว่า 20 พิกเซล.....	25
รูปที่ 2.27 แสดงภาพรวมของหัวข้อการสกัดคุณลักษณะ.....	26

สารบัญรูปภาพ (ต่อ)

รูปที่ 2.28 กราฟของ Linear Splines.....	27
รูปที่ 2.29 กราฟของ Quadratic Splines.....	28
รูปที่ 2.30 กราฟของ Cubic Splines.....	29
รูปที่ 2.31 แสดงกราฟของค่า entropy.....	30
รูปที่ 2.32 แสดงคลื่นทางของมือของท่าทางSorry (ซ้าย) Happy (ขวา).....	31
โดยพลอตจากค่าของเซ็นเซอร์ของมือ	
รูปที่ 2.33 แสดงคุณลักษณะภายใน (Intra-feature).....	32
รูปที่ 2.34 แสดงคุณลักษณะระหว่างคลาส (Inter-feature).....	32
รูปที่ 2.35 การกระจายของข้อมูล 2 ข้อมูล.....	33
รูปที่ 2.36 ค่าความแปรปรวนของข้อมูลระดับต่างๆ.....	34
รูปที่ 2.37 กราฟ scatter plot ของตารางที่ 2.1.....	36
รูปที่ 2.38 กราฟ scatter plot ของตารางที่ 2.2.....	37
รูปที่ 2.39 กราฟแสดงการเลือก Feature vector ทุกตัว.....	38
รูปที่ 2.40 กราฟแสดงการเลือก Feature vector เพียงตัวเดียว.....	39
รูปที่ 2.41 กราฟ reconstructedDataAdjust.....	39
รูปที่ 2.42 กราฟแสดงข้อมูลทั้งหมดของตัวอย่างโจทท์.....	40
รูปที่ 2.43 แสดง Model ของ Neuron ในคอมพิวเตอร์.....	40
รูปที่ 2.44 แสดงการเรียนรู้แบบ supervised learning.....	41
รูปที่ 2.45 แสดงการเรียนรู้แบบ unsupervised learning.....	41
รูปที่ 2.46 โครงสร้างโครงข่ายประสาทเทียมแบบไม่ย้อนกลับ.....	42
รูปที่ 2.47 โครงสร้างโครงข่ายประสาทเทียมแบบย้อนกลับ.....	43
รูปที่ 3.1 แสดง Block diagram ของระบบรู้จำตัวอักษร.....	47
รูปที่ 3.2 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ.....	48
รูปที่ 3.3 แสดงภาพ Grayscale ของตัวอักษร l.....	48
รูปที่ 3.4 แสดงภาพ Binary ของตัวอักษร l.....	49
รูปที่ 3.5 แสดงภาพการเติมสีขาวของพื้นที่ขนาดเล็กกว่าพิกเซลที่กำหนดของตัวอักษร l.....	49
รูปที่ 3.6 แสดงภาพพื้นที่ที่ถูกเติมพื้นที่สีขาวให้เต็มแล้วของตัวอักษร l.....	50
รูปที่ 3.7 แสดงเส้นคอนทัวร์ที่หาได้จากวิธี Canny.....	50

สารบัญรูปภาพ (ต่อ)

รูปที่ 3.8 แสดงการวาดวงกลมบนฝ่ามือ.....	51
รูปที่ 3.9 แสดงหมายเลขด้านทั้งสาม.....	51
รูปที่ 3.10 แสดงภาพที่พบฟิกเซลสีชาวด้านที่ 1.....	52
รูปที่ 3.11 แสดงภาพที่พบฟิกเซลสีชาวด้านที่ 2.....	52
รูปที่ 3.12 แสดงภาพที่พบฟิกเซลสีชาวด้านที่ 3.....	52
รูปที่ 3.13 แสดงการแบ่งฝ่ามือออกเป็น 2 ส่วน.....	53
รูปที่ 3.14 (ซ้าย)แสดงฟิคที่จับได้ไม่ครบ (ขวา)แสดงการเติมฟิคที่ขาด (จุดสีเขียว).....	54
รูปที่ 3.15 (ซ้าย)แสดงฟิคที่จับได้เกิน 5 ฟิค (ขวา)แสดงการตัดฟิคที่เกินทิ้ง.....	54
รูปที่ 3.16 แสดงเส้นระยะทางจากจุดเซ็นทรอยด์ถึงจุดฟิค.....	55
รูปที่ 3.17 แสดงภาพมุมระหว่างฟิคที่พบ.....	55
รูปที่ 3.18 โคจรขยายประสาทเทียมที่ใช้ในการจำแนกตัวอักษรภาษามือภาษาอังกฤษ.....	57
รูปที่ 3.19 แสดง Block diagram ของระบบรู้จำคำภาษาอังกฤษ.....	58
รูปที่ 3.20 ภาพเฟรมเริ่มต้น (ซ้าย) และเฟรมสิ้นสุด (ขวา) ของท่าทาง 'YES'.....	59
รูปที่ 3.21 Block diagram ของขั้นตอนการคัดเลือกเฟรมหลัก.....	59
รูปที่ 3.22 แสดงเฟรมหลักของการแบ่งท่าทางออกเป็น 4 ช่วง.....	60
รูปที่ 3.23 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ.....	60
รูปที่ 3.24 ตำแหน่งของฟิกเซลที่มีสีผิว.....	61
รูปที่ 3.25 ภาพก่อนลบตำแหน่งหน้าออก (ซ้าย) ภาพภายหลังลบตำแหน่งหน้า (ขวา).....	61
รูปที่ 3.26 ภายหลังจากกำจัดสิ่งรบกวนแล้ว.....	62
รูปที่ 3.27 การตัวแปรที่ใช้หาความกลมของวัตถุ.....	64
รูปที่ 3.28 ภาพจำลองหาความเอียงของวัตถุ.....	64
รูปที่ 3.29 เปรียบเทียบตำแหน่งมือและใบหน้าของท่าทางที่ต่างกัน.....	65
รูปที่ 3.30 เปรียบเทียบตำแหน่งมือทั้งสองข้างภายในเฟรมของท่าทางที่ต่างกัน.....	65
รูปที่ 3.31 อัตราส่วนระหว่างความกว้างและความยาวของ Blob ภายในเฟรม.....	66
รูปที่ 3.32 จำนวนวัตถุภายในเฟรม.....	66
รูปที่ 3.33 เวกเตอร์ตำแหน่งระหว่างหน้าและมือภายในเฟรม.....	66

สารบัญรูปภาพ (ต่อ)

รูปที่ 3.34	เปรียบเทียบตำแหน่งมือทั้งสองข้างของเฟรมที่อยู่ติดกัน.....	67
รูปที่ 3.35	ตำแหน่ง Blob ของเฟรมที่อยู่ติดกัน.....	67
รูปที่ 3.36	เวกเตอร์ตำแหน่งของมือระหว่างเฟรม.....	68
รูปที่ 4.1	ผลจากการเพิ่มและลดฟิคเทียมของมือ.....	70
รูปที่ 4.2	แสดงภาพเฟรมแรกของตัว I(ซ้าย) และ J(ขวา).....	73
รูปที่ 4.3	แสดงภาพเฟรมแรกของตัวอักษร M (ซ้าย) และ N (ขวา).....	73
รูปที่ 4.4	แสดงภาพเฟรมแรกของตัวอักษร P (ซ้าย) และ Q (ขวา).....	73
รูปที่ 4.5	แสดงการเปรียบเทียบกราฟ ROC ของการทดลองทั้งหมด.....	85



สารบัญตาราง

ตารางที่ 2.1 ตัวอย่างโจทย์.....	36
ตารางที่ 2.2 แสดงการนำข้อมูลดิบกลับกับค่าเฉลี่ย.....	37
ตารางที่ 2.3 ฟังก์ชันการแปลงแบบต่างๆ.....	45
ตารางที่ 3.1 แสดงคุณลักษณะทั้งหมดในการจำแนกตัวอักษรภาษามือภาษาอังกฤษ.....	56
ตารางที่ 3.2 แสดงคำศัพท์ทั้ง 4 หมวดหมู่.....	62
ตารางที่ 3.3 คุณลักษณะทั้งหมดที่ใช้จำแนกคำในภาษามือ.....	63
ตารางที่ 4.1 แสดงค่าเปอร์เซ็นต์ความถูกต้องของการใช้ Hidden layer ที่แตกต่างกัน.....	71
ตารางที่ 4.2 แสดงการเลือกใช้ค่า Testing และ Training.....	72
ตารางที่ 4.3 แสดงเปอร์เซ็นต์ความถูกต้องของตัวอักษร A-Z.....	72
ตารางที่ 4.4 เปรียบเทียบผลของการใช้อัลกอริทึมที่แตกต่างกัน.....	74
ตารางที่ 4.5 ผลการทดลองคำนวณคุณลักษณะของท่าทางโดยการกำหนดช่วงของเฟรม.....	76
โดยอ้างอิงจากค่าเฉลี่ยของเฟรมท่าทางทั้งหมด	
ตารางที่ 4.6 ตาราง Confusion Matrix ของระบบรู้จำตัวอักษรภาษามือภาษาอังกฤษ.....	77
ตารางที่ 4.7 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 4 ช่วง.....	79
ตารางที่ 4.8 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 5 ช่วง.....	80
ตารางที่ 4.9 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 6 ช่วง.....	81
ตารางที่ 4.10 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 7 ช่วง.....	82
ตารางที่ 4.11 ผลการทดลองเปรียบเทียบลำดับของท่าทางสำคัญของจำนวนช่วงที่ต่างกัน.....	83
ตารางที่ 4.12 เปรียบเทียบค่าความถูกต้องและเวลาในการประมวลผลของวิธีทั้งหมด.....	84
โดยใช้เพียง 27 คำ	

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

ภาษามือเป็นอวัจนภาษาอย่างหนึ่งซึ่งถูกใช้โดยผู้ที่เป็นใบ้เป็นหลักซึ่งประกอบด้วย การสื่อสารด้วยมือ, การสื่อสารด้วยร่างกายและการใช้ริมฝีปากในการสื่อความหมายแทนการใช้เสียงพูด การสื่อสารจะใช้ลักษณะการเคลื่อนไหวของมือ, แขนและร่างกายเป็นสัญลักษณ์เพื่อช่วยในการสื่อสาร อีกทั้งยังมีการแสดงความรู้สึกทางใบหน้าเพื่อช่วยในการสื่อสารความคิดของผู้สื่อ

ในปัจจุบันมีการค้นคว้าหาวิธีการเพื่อให้การสื่อสารทางภาษามือเป็นไปได้ง่ายขึ้น โดยมีจุดประสงค์เพื่อให้คนที่มีความบกพร่องทางการได้ยินและบุคคลปกติที่ไม่รู้ภาษามือให้สามารถสื่อสารกันได้ เนื่องจากมีคนมากกว่าครึ่งล้านคนบนโลกใบนี้ที่ใช้ภาษามือเป็นภาษาหลัก งานวิจัยชิ้นนี้จึงถือเป็นจุดเริ่มต้นซึ่งทำให้ผู้ที่มีความบกพร่องทางการได้ยินสามารถใช้ชีวิตอย่างปกติได้ดียิ่งขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ในการประมวลผลภาพภาษามือนั้นมีการคิดค้นและสร้างเครื่องมือขึ้นมามากมายแต่โดยส่วนใหญ่จะเป็นเทคโนโลยีขั้นสูง กล่าวคือใช้กล้องที่มีคุณภาพสูงและมีราคาแพง จากปัญหาดังกล่าวงานวิจัยนี้จึงมีจุดมุ่งหมายในการประมวลผลภาพโดยใช้กล้อง Webcam ปกติซึ่งมีราคาถูกและหาซื้อได้ทั่วไป ผู้ทำการทดลองใช้กล้อง Logitech C310 ซึ่งสามารถลดความละเอียดของการถ่ายวิดีโอได้ต่ำสุดคือ 320x240 และพยายามค้นหาอัลกอริทึมที่เหมาะสมเพื่อนำมาวิเคราะห์ภาษามือของตัวอักษรภาษาอังกฤษ 26 ตัว และคำที่ใช้ในชีวิตประจำวัน 27 คำแล้วจึงให้โปรแกรมแสดงผลลัพธ์ที่ประมวลได้ออกมา

1.3 สมมติฐานของการศึกษา

คาดว่าจะสามารถคิดค้นอัลกอริทึมและจัดการขนาดของข้อมูลได้เหมาะสมเพื่อให้โปรแกรมสามารถจำแนกตัวอักษรและคำภาษามือในภาษาอังกฤษได้ โดยใช้วิธีการสกัดคุณลักษณะของมือออกมาจากขอบของภาพใบนารีซึ่งถูกบันทึกจากกล้องเว็บแคมเพื่อนำมาแก้ไขปัญหาอุปสรรคที่มีราคาแพงและลดความซับซ้อนในการประมวลผลจากกล้องสามมิติ อีกทั้งสามารถคิดค้นวิธีการลำดับการขยับท่าทางของมือเพื่อสามารถลดมิติของข้อมูลอินพุตที่จะถูกนำเข้าไปสอนในโครงข่ายประสาทเทียมเพื่อระยะเวลาในการสร้างโมเดลรวมถึงสามารถระบุช่วงเวลาของการเริ่มทำท่าทางและหยุดทำท่าทางได้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

งานวิจัยในการสกัดคุณลักษณะข้อมูลเพื่อนำมาใช้จำแนกตัวอักษรและคำภาษามือในภาษาอังกฤษนั้นโดยส่วนใหญ่แล้วจะกระทำโดยการใส่กล้องสามมิติหรืออุปกรณ์เสริมเช่นถุงมือ (Artificial gloves) โดยเฉพาะกับการจำแนกคำซึ่งส่วนใหญ่เป็นท่าทางเคลื่อนไหวและมีสีหน้ามาเกี่ยวข้อง กรณีการจำแนกตัวอักษรภาษามือหลายๆครั้งจะใช้สีของผิวในการแยกส่วนของมือออกมาก่อนจากนั้นจึงนำมาวิเคราะห์ด้วยการใช้ Canny method ซึ่งผู้ทำวิจัยพบว่าข้อมูลที่ได้นั้นมีขนาดใหญ่เกินไปและจำเป็นต้องนำไปลดมิติของข้อมูลก่อนซึ่งเป็นการเสียเวลา หรือแม้กระทั่งการสกัดข้อมูลจากมือทางตรงโดยการมองเป็นก้อนวัตถุชิ้นเดียวซึ่งทำให้สามารถแยกตัวอักษรได้เฉพาะท่าทางกางนิ้วมือนั้น ผู้ทำการทดลองจึงคิดค้นอัลกอริทึมที่สามารถจำลองนิ้วมือจากขอบภาพได้ซึ่งด้วยวิธีนี้เองจะทำให้สามารถจำแนกตัวอักษรทุกตัวออกจากกันได้นั่นเอง ส่วนการสกัดคุณลักษณะที่ใช้รู้จำคำภาษาอังกฤษนั้นผู้ทำการวิจัยเห็นว่าการสื่อสารจริงนั้นนอกจากใช้ตัวอักษรแล้วยังมีการทำท่าทางเป็นคำมาใช้อธิบายบริบทของประโยคด้วย จึงได้ทดลองนำกล้องเว็บแคมสองมิติตัวเดิมมาใช้หาอัลกอริทึมเพื่อแยกคำที่ใช้ในชีวิตประจำวันออกจากกันโดยจะเน้นไปที่การหาความแตกต่างระหว่างเฟรมเพื่อหาทิศทางและการเคลื่อนที่ของมือและรูปร่างของมือที่เปลี่ยนแปลงไป โดยอาศัยการวิเคราะห์ลำดับการขยับของท่าทางมือมาช่วยในการเลือกเฟรมที่จะนำมาวิเคราะห์จากนั้นจึงนำข้อมูลที่ได้ออกไปวิเคราะห์ผลโดยใช้โครงข่ายประสาทเทียม

1.5 ขอบเขตการวิจัย

คิดค้นวิธีการสกัดคุณลักษณะของท่าทางมือโดยใช้กล้องเว็บแคมปกติและวิเคราะห์หาคุณลักษณะของท่าทางมือจากเฟรมของภาพซึ่งได้มาจากการบันทึกวิดีโอ โดยการวิเคราะห์เฉพาะขอบของภาพไบนารีโดยแบ่งการทำวิจัยออกเป็นสองส่วนคือการสกัดคุณลักษณะเพื่อใช้รู้จำภาษามือที่เป็นตัวอักษรภาษาอังกฤษ 26 ตัว และการสกัดคุณลักษณะเพื่อใช้รู้จำภาษามือที่เป็นคำภาษาอังกฤษ 25 คำ จากนั้นใช้โครงข่ายประสาทเทียมเพื่อทำการเรียนรู้และจำแนกกลุ่มของข้อมูลที่กำหนดไว้

1.6 ขั้นตอนของการศึกษา

เริ่มต้นด้วยการรับวิดีโอจากกล้องเว็บแคมเข้าจากนั้นนำแต่ละเฟรมไปทำเป็นภาพไบนารีเพื่อใช้เพื่อวิเคราะห์เฟรมที่เป็นจุดเริ่มต้นและจุดสุดท้ายของท่าทางนั้นๆ จากนั้นทำการสกัดคุณลักษณะจากขอบภาพของแต่ละเฟรมเพื่อนำเข้าไปเข้าเรียนรู้ในโครงข่ายประสาทเทียม

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ภาษามือ (Sign Language)

ภาษามือเป็นส่วนหนึ่งของการสื่อสารและใช้ในการถ่ายทอดความหมายเมื่อเทียบกับการพูดคุย ภาษามือถูกคิดค้นโดยแนวความคิดที่ว่าหากคนเรานั้นไม่มีเสียงหรือไม่มีลิ้นจะสื่อสารกันได้อย่างไร และการใช้ภาษามือเพื่อเป็นการสื่อสารนี้เองได้เกิดขึ้นมานานแล้ว จนกระทั่งปี พ.ศ. 2223 มีการตีพิมพ์หนังสือ โดย John Bulwer ชื่อว่า ‘Didascalocophus’ ที่เสนอแนวความคิดของตนเองหลังจากที่ได้ศึกษาโดยการพูดคุยกับคนหูหนวกคนหนึ่ง จากนั้น Arthrological และ Charles de La Fin ได้ตีพิมพ์หนังสือซึ่งสอนการใช้ภาษามือร่วมกับร่างกายในการสื่อสาร โดยภาษามือนั้นจะมีความแตกต่างกันในแต่ละภูมิภาค เนื่องจากมีการพัฒนาหรือเปลี่ยนไปตามภาษาท้องถิ่นของตนเอง

ASL เป็นชื่อย่อของ American Sign Language หรือภาษามืออเมริกัน เป็นภาษาทางกายภาพที่ทำหน้าที่ในการสื่อสารระหว่างผู้ที่มีความบกพร่องทางการได้ยินทั้งหูหนวกและหูตึงใช้ติดต่อสื่อสารกับคนทั่วไปในสังคมของชาวอเมริกันและบางส่วนของแคนาดา นอกจากนี้แล้วภาษาถิ่นของ ASL ยังถูกใช้ในอีกหลายๆประเทศทั่วโลก รวมถึงแอฟริกันตะวันตกและเอเชียตะวันออกเฉียงใต้ ภาษามือนั้นถูกนำไปใช้อย่างกว้างขวางจนเปรียบเสมือนภาษาที่สองซึ่งทำหน้าที่เป็นภาษากลางในการสื่อสาร ซึ่ง ASL นั้นมีความใกล้เคียงกับภาษามือของฝรั่งเศสมากที่สุดเนื่องจากชาวฝรั่งเศสเป็นผู้ริเริ่มบุกเบิกการเรียนการสอนและการใช้ภาษามือในหมู่คนหูหนวกชาวอเมริกัน โดยภาษามือที่ใช้ในการสื่อสารนั้นแบ่งออกเป็น 2 แบบได้แก่ ภาษามือที่เป็นคำศัพท์ซึ่งใช้ในการสื่อสารโดยตรงและภาษามือเป็นตัวอักษรซึ่งใช้ในการสะกดคำ (Finger spelling)

2.1.1 ตัวอักษรภาษามือ (Sign Language Alphabets)

‘ตัวอักษร’ ภาษามือมักจะใช้กระบวนการสะกดคำ (Finger spelling) ซึ่งเป็นกระบวนการในแสดงท่าทางมือของตัวอักษรซึ่งสอดคล้องกับตัวอักษรในคำที่ต้องการสื่อและมักใช้ในงานที่เกี่ยวข้องกับการอ่านและเขียน โดยมักนำไปใช้ในการระบุชื่อสถานที่ซึ่งไม่ได้โด่งดังหรือสำคัญจนมีท่าทางเฉพาะ เช่น ชื่อประเทศ, ชื่อบุคคล, ชื่อบริษัท รวมไปถึงยี่ห้อของผลิตภัณฑ์ต่างๆ

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการจดจำท่าทางตัวอักษรภาษามือแบบอเมริกันพบว่า งานวิจัยส่วนใหญ่เน้นไปที่การจดจำตัวอักษรที่เป็นท่าทางไม่เคลื่อนไหวหรือเห็นนิ้วมือได้ชัดเจน เช่นตัว I, L, U และ V [1] เป็นต้น โดยงานวิจัยนี้ใช้วิธีการแกะรอยทิศทางพิกเซลขอบของภาพมือเพื่อทำการหาปลายนิ้วที่แท้จริงซึ่งมีประสิทธิภาพต่อท่าทางมือที่สามารถเห็นนิ้วมือชัดเจนเท่านั้น อีกทั้งยังพบงานวิจัยที่สามารถจำแนกตัวอักษรทุกตัว (A-Z) [2] ซึ่งเน้นไปที่การใช้วิธีของ Morphology เพื่อให้ได้ภาพไบนารีที่ต้องการก่อนจะใช้อัลกอริธึมแคนนี่เพื่อตรวจจับขอบภาพและจดจำตำแหน่งของพิกเซลเพื่อทำการจำแนกข้อมูล ซึ่งให้ผลความถูกต้องเพียง 60% และยังมีงานวิจัยที่มีความพยายามที่จะหานิ้วมือโดยใช้วิธีการลดขนาดของเส้นขอบ (Thinning method) [3] แล้วจึงนำไปหามุมระหว่างเส้นและระยะทางของเส้นจากจุดปลาย แต่วิธีการนี้ต้องผ่านขั้นตอนการปรับปรุงคุณภาพของภาพหลายขั้นตอนซึ่งหมายถึงระยะเวลาในการประมวลผลที่มากขึ้นซึ่งไม่เหมาะสมในการนำไปใช้งานจริง

นอกจากนี้งานวิจัย [4] ซึ่งใช้วิธีวิเคราะห์ภาพโดยรวม (Global feature) โดยการคำนวณพื้นที่ภาพไบนารีของมือแล้วทำการหาจุดเซ็นทรอยด์เพื่อนำไปหาระยะทางจากจุดบนซ้ายของ Blob ซึ่งให้อัตราความถูกต้องเพียง 80% อีกทั้งยังพบงานวิจัยที่พยายามจะรักษารายละเอียดของมือโดยใช้กล่อง 3 มิติ [5] ซึ่งมีราคาแพงกว่ากล่องทั่วไปซึ่งมีประโยชน์ในแง่ของการเปลี่ยนแปลงของการทำท่าทางและมุมการหมุนของมือ อย่างไรก็ตามงานวิจัยนี้ไม่สามารถจดจำตัวอักษรที่มีการเคลื่อนไหวได้ (J และ Z) รวมทั้งยังมีงานวิจัยที่ใช้อุปกรณ์เสริมอื่นๆ เช่น ถุงมือประดิษฐ์ เพื่อใช้ในการวัดตำแหน่งและรอยต่อของมือ [6] อย่างไรก็ตามอุปกรณ์ดังกล่าวต้องใช้สายเคเบิลจำนวนมากและไม่สะดวกไม่การนำไปใช้งานจริง

ดังนั้นวิทยานิพนธ์ฉบับนี้จึงให้ความสำคัญกับการใช้อุปกรณ์จำนวนน้อยชิ้น, ราคาถูก และไม่จำเป็นต้องมีการเชื่อมต่อซึ่งเป็นอุปสรรคในการใช้งานจริง และสามารถวิเคราะห์และจำแนกตัวอักษรภาษาอังกฤษทั้งแบบเคลื่อนไหวและแบบหยุดนิ่งได้อย่างมีประสิทธิภาพ



รูปที่ 2.1 ตัวอย่างท่าทางตัวอักษรภาษามือแบบอเมริกัน

2.1.2 คำในภาษามือ (Sign Language Words)

ภาษามือที่เป็น ‘คำ’ จะใช้แสดงท่าทางอารมณ์ซึ่งบ่งบอกถึงประธาน, กริยาและกรรมของรูปประโยคซึ่งแต่ละภาษาก็จะมีไวยากรณ์ที่แตกต่างกันตามแต่ละภาษาท้องถิ่นของตน โดยภาษามือที่เป็นคำจะมีการใช้งานที่แตกต่างกับตัวอักษรภาษามือ เนื่องจากภาษามือนั้นถูกสร้างขึ้นเพื่อจุดประสงค์หลักที่ใช้ในการสื่อสารระหว่างบุคคลเช่นเดียวกับภาษาอื่น ๆ ทั่วโลก ดังนั้นนอกเหนือจากการมีตัวอักษรซึ่งใช้ในงานอ่านและเขียนแล้วจึงจำเป็นต้องมีคำเพื่อที่จะให้การสื่อสารนั้นเป็นไปได้ง่ายขึ้นในการใช้งานจริง โดยกรณีของคนที่มีความบกพร่องทางการได้ยินนั้นก็ใช้คำในการจดจำลักษณะการสื่อสารแทนเสียงที่คนทั่วไปได้ยินนั่นเอง

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการวิเคราะห์และจำแนกท่าทางเคลื่อนไหวนั้นแตกต่างกับการวิเคราะห์จำแนกท่าทางที่ไม่เคลื่อนไหวคือนอกจากจะต้องมีการสกัดวัตถุออกมาจากภาพแล้ว ยังต้องมีการติดตามการเคลื่อนไหวของท่าทางนั้นๆ อีกด้วย โดยวิธีในการสกัดมือออกมาจากรูปภาพเคลื่อนไหวก็มีการใช้วิธีแบบตัวอักษรเช่นเดียวกันคือมีทั้งการใช้กล้อง 3 มิติเพื่อเก็บรายละเอียดโครงสร้างในการขยับมือ (Skeleton data) [7] ซึ่งหมายถึงการเก็บโครงสร้างของข้อต่อตั้งแต่ปลายนิ้วจนถึงข้อมือซึ่งงานวิจัยนี้ได้ทำการเก็บตำแหน่งของข้อต่อเหล่านี้แล้วนำมาเชื่อมต่อกันเพื่อให้รู้ถึงเส้นทางการเคลื่อนที่ของมือ (trajectory) โดยใช้คุณลักษณะที่เรียกว่า Handwriting-Inspired Features (HIF3D) [8] ซึ่งอ้างอิงจากหนึ่งในชุดคุณลักษณะของการเขียนที่มีประสิทธิภาพและใหม่ที่สุดซึ่งถูกนำเสนอโดย Delaye และ Anquetil [9]

รวมถึงมีงานวิจัยซึ่งจำเป็นต้องใช้ถุงมือประดิษฐ์ (textile-based sensing glove) [10] ซึ่งมีจุดมุ่งหมายเพื่อรู้จำคำสั่งที่ได้มาจากถุงมือประดิษฐ์จากมือของศัลยแพทย์ภายในช่องท้องของผู้ป่วยเพื่อเป็นแนวทางในการทำงานร่วมกันของหุ่นยนต์ โดยทำการวิเคราะห์รูปแบบการงอนิ้ว, ความเร็วในการขยับนิ้ว และระยะเวลาในการทำท่าทางดังกล่าว รวมถึงการใช้กล้องธรรมดาโดยการตรวจจับสีผิวเพื่อสกัดคุณลักษณะจากขอบของภาพอีกด้วย [11]

นอกจากนี้แสงจากสภาพแวดล้อมและอัตราเร็วที่ใช้ในการบันทึกวิดีโอมือก็มีผลมากเช่นเดียวกัน งานวิจัย [12] ซึ่งใช้กล้อง ToF ย่อมาจาก Time-of-Flight ซึ่งทำหน้าที่สำหรับตรวจจับความลึกของวัตถุ งานวิจัยนี้กล่าวว่าแสงมีผลกระทบต่อเซ็นเซอร์ซึ่งใช้ตรวจจับระยะทางของวัตถุ โดยพบว่าหากบันทึกภาพในบริเวณที่มีแสงสว่างมากจะทำให้ได้ภาพมือที่มีสัญญาณรบกวนเป็นจำนวนมากซึ่งยากแก่การนำไปปรับปรุงโดยปัญหาเรื่องแสงดังกล่าวของกล้อง ToF ก็ส่งผลเช่นเดียวกับกล้อง 2 มิติ เนื่องจากหากภาพที่มีแสงสว่างมากเกินไปจะทำให้ขั้นตอนการแบ่งส่วน (Segmentation) ทำได้ยากเช่นเดียวกันอัน

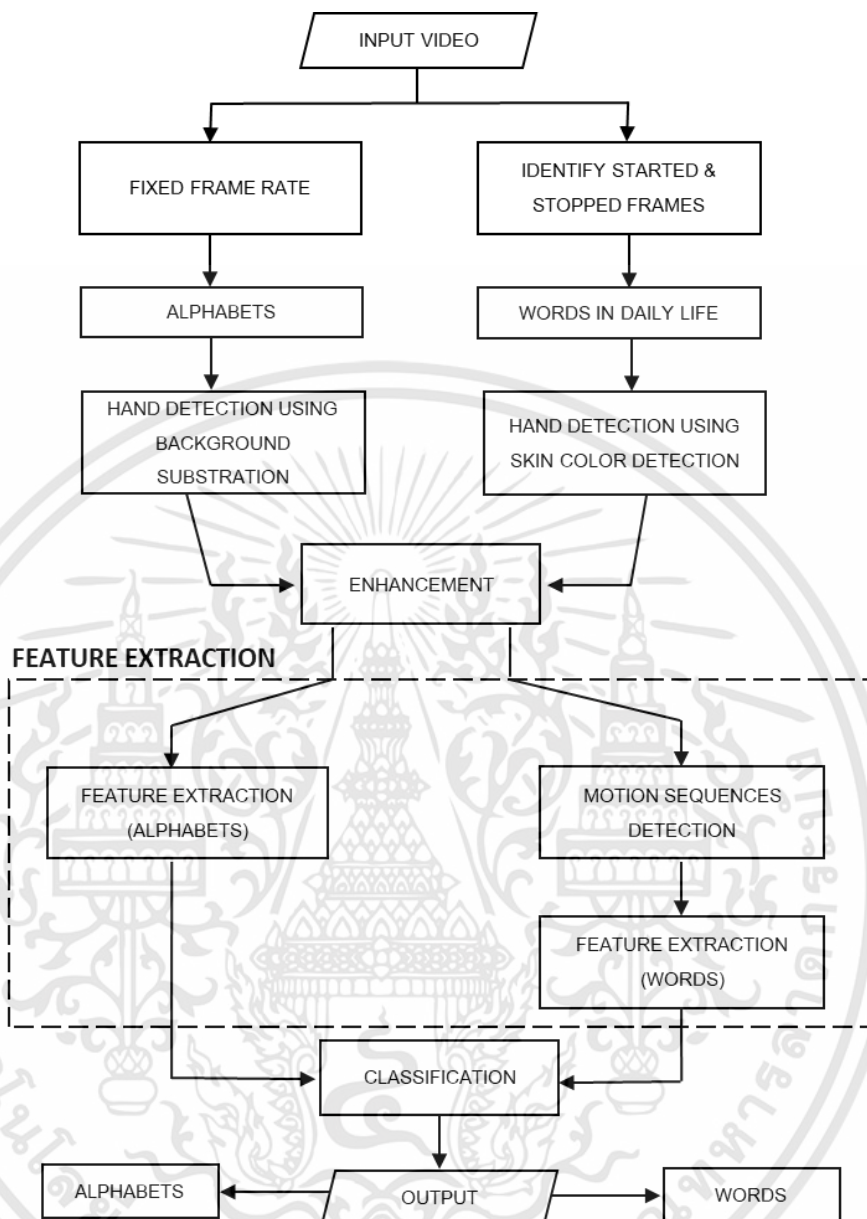
เนื่องจากการเกิดเงาและแสงสะท้อนบนสีผิวของผู้กระทำท่าทาง นอกจากนี้อัตราเร็วที่ใช้บันทึกวิดีโอที่ส่งผลกระทบต่อสัญญาณรบกวนเช่นเดียวกันโดยงานวิจัยกล่าวว่าช่วงของเฟรมเรทที่เหมาะสมซึ่งทำให้เกิดสัญญาณรบกวนน้อยคือ 17-54 เฟรม/วินาที

วิทยานิพนธ์ฉบับนี้จะเน้นไปที่การวิเคราะห์และจำแนกคำศัพท์ที่ใช้ในชีวิตประจำวันโดยการบันทึกวิดีโอจากกล้องเว็บแคมทั่วไปและใช้การตรวจจับสีผิวเพื่อสกัดวัตถุมือในสถานะแสงที่เหมาะสมก่อนจะนำชุดของภาพซึ่งได้มาจากวิดีโอดังกล่าวไปผ่านคุณลักษณะเพื่อใช้ในการติดตามการเคลื่อนไหวของมือเพื่อนำไปจำแนกชนิดของคำออกจากกัน



รูปที่ 2.2 ท่าทางตัวอักษรภาษามือแบบอเมริกัน

รูปที่ 2.3 แสดงขั้นตอนการทำงานโดยรวมของระบบรู้จำภาษามือแบบอเมริกัน โดยมีการแบ่งการวิจัยเป็น 2 การทดลองคือการรู้จำตัวอักษรและระบบรู้จำคำในชีวิตประจำวัน ทั้งสองการทดลองนี้เริ่มต้นการทำงานแบบเดียวกันนั่นคือรับวิดีโอท่าทางเข้ามาจากกล้องเว็บแคมแล้วจึงทำการระบุเฟรมเริ่มต้นและเฟรมสิ้นสุดของท่าทาง จากนั้นทำการระบุเฉพาะส่วนของมือออกมาจากภาพโดยขั้นตอนนี้ทั้งสองการวิจัยจะใช้วิธีที่ต่างกัน จากนั้นนำภาพที่เหลือแต่วัตถุมือจากวิธีดังกล่าวไปปรับปรุงคุณภาพเพื่อลดสัญญาณรบกวน แล้วจึงนำไปสกัดคุณลักษณะของท่าทางออกมาเพื่อให้เหมาะสมกับกลุ่มคำตอบปลายทาง จากรูปที่ 2.3 จะเห็นได้ว่าหากเป็นระบบรู้จำคำในชีวิตประจำวันนั้นจำเป็นต้องทำการหาลำดับการเกิดของท่าทางเสียก่อน จากนั้นจึงจะสามารถนำเข้าไปเรียนรู้ในโครงข่ายประสาทเทียมเพื่อวิเคราะห์ผลของคำตอบได้

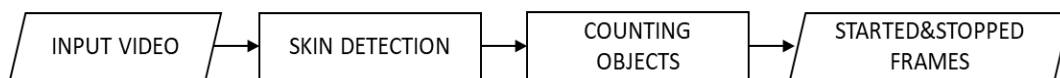


รูปที่ 2.3 ขั้นตอนการทำงานโดยรวมของระบบรู้จำภาษามือแบบอเมริกัน

2.2 การระบุเฟรมเริ่มต้นและเฟรมสิ้นสุดของท่าทาง

เนื่องจากคำในภาษามือนั้นเป็นท่าทางที่มีการเคลื่อนไหว (Gesture) ดังนั้นสิ่งสำคัญประการแรกก็คือการกำหนดช่วงของท่าทางโดยมีวิธีการคือ เมื่อรับวิดีโอขาของคำภาษามือเข้ามาแล้วเราจำเป็นที่จะต้องตรวจสอบว่าผู้ทำท่าทางนั้นเริ่มต้นท่าทางและสิ้นสุดท่าทางที่เฟรมใดเพื่อที่จะได้นำเฉพาะเฟรมช่วง

ของท่าทางนั้นไปวิเคราะห์ผลเพื่อลดหน่วยความจำของการวิเคราะห์ภาพทั้งวิดีโอ วิธีการระบุเฟรมเริ่มต้นและเฟรมสิ้นสุดดังนี้



รูปที่ 2.4 Block diagram ของขั้นตอนการระบุเฟรมเริ่มต้นและสิ้นสุด

2.2.1 ตรวจสอบสีผิวภายในเฟรม (Skin detection)

เนื่องจากเราต้องการทราบว่าในเฟรมดังกล่าวมีใบหน้าหรือมืออยู่ในเฟรมหรือไม่ เพื่อจะได้ไม่เสียเวลาในการประมวลผลเฟรมที่ไม่มีมืออยู่ โดยจะใช้ปริภูมิ YCbCr เนื่องจากปริภูมิตั้งนี้มีประสิทธิภาพการบีบอัดที่ต่ำกว่าและแยกสีผิวออกจากกันได้ดีเมื่อเทียบกับปริภูมิ HSV ซึ่งใช้เวลาในการประมวลผลนานกว่าและใช้งานได้ดีเฉพาะกับภาพที่มีสีพื้นหลังสม่ำเสมอทั้งภาพเนื่องจากหากมีการผันผวนของค่า hue และ saturation มาก พิกเซลที่มีความเข้มน้อยและมากจะไม่ถูกนำมาพิจารณา[13] ดังนั้นผู้วิจัยจึงเลือกใช้ปริภูมิ YCbCr ในการระบุสีผิว โดยถือเป็นปริภูมิจิตลและมีกำหนดเป็นมาตรฐานโดย BT.601 ซึ่งได้จากการแปลงองค์ประกอบ YUV อะนาล็อก โดยจะแบ่งข้อมูลสีในลักษณะที่สอดคล้องกับการรับรู้ของสายตามนุษย์โดยมีค่า Y เป็นองค์ประกอบด้านความสว่างซึ่งมีค่าตั้งแต่ 16 – 235 และมีองค์ประกอบ Cb และ Cr เป็นองค์ประกอบของสี โดยที่ค่า Cb คือค่าความแตกต่างระหว่างสีน้ำเงินกับความสว่าง (B-Y) และมีค่า Cr เป็นค่าความแตกต่างระหว่างสีแดงกับความสว่าง (R-Y) ซึ่งทั้งสององค์ประกอบต่างก็มีค่าตั้งแต่ 16 -240 การแปลงค่าจากสี RGB ไปเป็น YCbCr ทำได้โดยใช้สมการที่ 2.1

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

สี YCbCr นั้นแตกต่างกับ RGB คือสีจะไม่เปลี่ยนตามความสว่าง ซึ่งทำให้ผลลัพธ์ของสี YCbCr มีประสิทธิภาพมากกว่าสี RGB โดยช่วงของสีที่สอดคล้องกับสีผิวมนุษย์คือ

$$Y > 80, 85 < Cb < 135, 135 < Cr < 180 \text{ โดยที่ค่า } Y, Cb \text{ และ } Cr \text{ อยู่ในช่วง } [0 \ 255]$$

อย่างไรก็ตาม Chai และ Ngan [14] ได้พัฒนาอัลกอริทึมซึ่งใช้ประโยชน์จากปริภูมิภูมิทัศน์ (spatial characteristics) ของสีผิวมนุษย์ โดยค้นพบว่าช่วงของสีผิวมนุษย์นั้นควรจะต้องกำหนดด้วย

องค์ประกอบความเข้มของสี(chrominance) และกำหนดช่วงของ Cb และ Cr ซึ่งใช้เป็นตัวแทนส่วนใหญ่ของสีผิวมนุษย์เป็นดังนี้

$$77 \leq Cb \leq 127 \text{ และ } 133 \leq Cr \leq 173$$

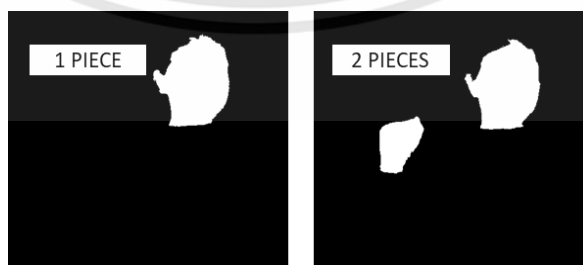
ซึ่งจะได้ผลลัพธ์ของการจับสีผิวของมนุษย์เป็นไปดังรูปที่ 2.5



รูปที่ 2.5 แสดงการเปรียบเทียบสีผิวโดยใช้ช่วงดังกล่าว

2.2.2 การคำนวณจำนวนวัตถุที่เป็นสีผิว (Counting Objects)

ภายหลังจากที่ตรวจจับสีผิวได้แล้วเราจำเป็นต้องตรวจสอบว่าภายในเฟรมดังกล่าวมีมืออยู่หรือไม่ เนื่องจากเรากำหนดเงื่อนไขขึ้นว่าในช่วงของการทำท่าทางภาษามือนั้นจำเป็นต้องมีมือเป็นองค์ประกอบภายในเสมอ ดังนั้นเฟรมใดที่ปรากฏเฉพาะใบหน้าหรือไม่มีสีผิวอยู่จะไม่ถือว่าเป็นช่วงของการทำท่าทาง ซึ่งเงื่อนไขนี้เองทำให้เราสามารถหาเฟรมเริ่มต้นและเฟรมสิ้นสุดของวิดีโอได้ โดยเราจะทำการคำนวณจำนวนวัตถุซึ่งเป็นสีผิวที่ได้จากหัวข้อที่ 2.2.1 ว่ามีมากกว่า 1 วัตถุหรือไม่ เนื่องจากการจับสีผิวนั้นจะได้หน้าเป็นวัตถุร่วมด้วยเสมอ ดังนั้นการนับวัตถุได้มากกว่า 1 ขึ้นจึงเป็นการตีความว่ามีมืออยู่ในเฟรมนั่นเอง จากรูปที่ 2.6 จะเห็นได้ว่าเมื่อวัตถุมีเพียง 1 ชิ้น เราจะพบเพียงใบหน้าของผู้ทำท่าทางเท่านั้น แต่เมื่อใดก็ตามที่วัตถุมีมากกว่า 1 ชิ้น จะปรากฏเป็นมือของผู้ทำท่าทางร่วมด้วย ดังนั้นเงื่อนไขของการหาเฟรมเริ่มต้นคือเมื่อเฟรมดังกล่าวเป็นเฟรมแรกที่มีวัตถุมากกว่า 1 ชิ้น และเงื่อนไขของเฟรมสิ้นสุดก็คือเมื่อเฟรมดังกล่าวเป็นเฟรมสุดท้ายก่อนที่พบวัตถุเท่ากับ 1 ชิ้นนั่นเอง



รูปที่ 2.6 การคำนวณจำนวนวัตถุภายในภาพ

2.3 การตรวจจับมือ (Hand Detection)

2.3.1 กระบวนการเปลี่ยนแปลงภาพสีเป็นภาพระดับสีเทา (Gray-Scale Image Transform)

การตรวจจับมือเริ่มต้นด้วยการแปลงค่าข้อมูลภาพสีในระบบ RGB ของเฟรมภาพทำทางเพื่อให้ได้เฉพาะค่าความสว่างของภาพเพียงอย่างเดียว โดยปราศจากค่าข้อมูลของสีภาพ โดยทั่วไปภาพระดับสีเทาจะประกอบด้วยค่าความสว่างที่แตกต่างกัน 256 ระดับ มีค่าตั้งแต่ 0 ถึง 255 นั่นคือไล่ระดับความสว่างจากมืดไปจนขาว โดยเราใช้การแปลงนี้เพื่อเป็นตัวกำหนดค่าเทรชโฮล(Threshold) ในการสร้างภาพไบนารีและเพื่อให้ง่ายต่อการหาขอบของภาพเพื่อนำไปสร้างรูปร่างของวัตถุ โดยความสัมพันธ์การแปลงภาพสีเป็นภาพระดับสีเทา ดังสมการ

$$\text{Gray} = 0.299(\text{Red}) + 0.587(\text{Green}) + 0.114 (\text{Blue}) \quad (2.2)$$

โดยที่ Red คือ ค่าของสีแดงในระบบสี RGB

Green คือ ค่าของสีเขียวในระบบสี RGB

Blue คือ ค่าของสีฟ้าในระบบสี RGB

เมื่อได้ภาพที่เป็นภาพระดับสีเทามาแล้วให้ทำการกำหนด Gray-level เพื่อเปลี่ยนภาพให้กลายเป็นไบนารีซึ่งมีเพียงค่า '0' และ '1' โดยเราจะกำหนดให้ค่าพิกเซลของสีที่ต่ำกว่าค่า Gray-level มีค่าเป็น 0 (สีดำ) และให้กำหนดให้ค่าพิกเซลที่มากกว่า Gray-level มีค่าเป็น 1 (สีขาว)

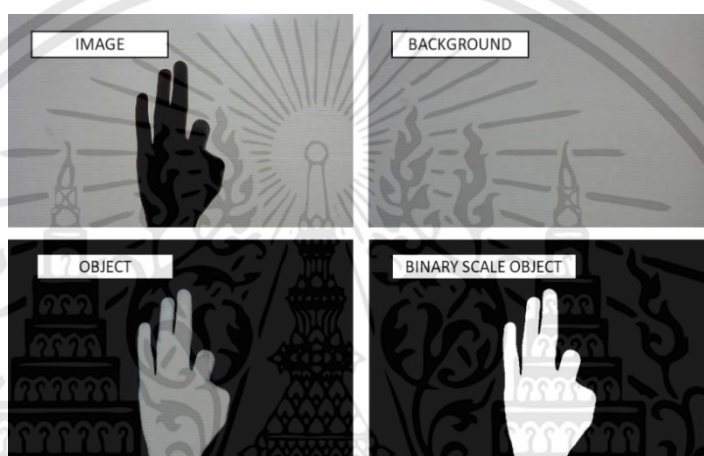
2.3.2 การเปรียบเทียบจุดสีเพื่อหาตำแหน่งมือ (Background Subtraction)

ภายหลังจากการเปลี่ยนเฟรมภาพให้อยู่ในรูปของไบนารีได้แล้ว เราจะสามารถทำการเปรียบเทียบจุดเพื่อหาตำแหน่งของมือได้ เนื่องจากภาพบิตแมพ(bitmap) จะประกอบไปด้วยจุดเล็กๆที่เรียกว่า pixel เราสามารถหาค่าสีแต่ละจุดของภาพ ได้ เราสามารถนำมาเปรียบเทียบกับจุดสีอื่นๆได้โดยวิธีการวนลูปแกน x และ แกน y ของภาพ bitmap นั้นๆได้ เช่นเดียวกันนั้นเราจะใช้วิธีนี้เองในการสกัดเอาวัตถุในภาพที่เราต้องการออกมาซึ่งเป็นภาพไบนารีที่มีเพียงค่า 0 และ 1 ซึ่งจะมีประสิทธิภาพสูงเมื่อภาพวัตถุนั้นไม่ได้เคลื่อนไหวหรือเคลื่อนที่อย่างช้าๆ และพื้นหลังของภาพนั้นเป็นสีเดียวกันโดยใช้สมการ

$$O = \text{IMG} - \text{BG} \quad (2.3)$$

- เมื่อ O คือ Object หรือวัตถุ
 IMG คือ Image หรือ รูปภาพไบนารีที่เราสนใจ
 BG คือ Background หรือ รูปภาพไบนารีพื้นหลัง

เหตุผลที่ต้องทำให้ภาพเป็น Binary Scale (ขาวดำ) เพื่อลดหน่วยความจำที่ไซ่เนื่องจาก Binary Scale 1 จุดใช้หน่วยความจำเพียง 1 บิต ในขณะที่ Gray Scale 1 จุดใช้ความจำถึง 8 bit อีกทั้งภาพขาวดำนั้นง่ายต่อการกำจัดสัญญาณรบกวนซึ่งส่งผลเสียต่อการวิเคราะห์คุณลักษณะของขอบมืออีกด้วย



รูปที่ 2.7 แสดงผลภาพที่เกิดจากการเปรียบเทียบจุดเพื่อลบพื้นหลัง

2.3.3 การตรวจจับใบหน้าด้วยวิธี Viola-Jones (Viola-Jones Algorithm)

เนื่องจากการตรวจจับวัตถุหรือมือจากวิดีโอโดยที่วัตถุมีการเคลื่อนไหวมากและมีใบหน้ามาเกี่ยวข้องนั้นการใช้วิธี Background Subtraction เพียงอย่างเดียวจะไม่สามารถประมวลผลได้อย่างมีประสิทธิภาพ ดังนั้นเราจึงจำเป็นต้องใช้การตรวจจับใบหน้าและสีผิวเข้าร่วมด้วยเพื่อแยกใบหน้าออกจากมือ โดยเราจะทำการตรวจหาตำแหน่งใบหน้าภายในเฟรมจากนั้นจึงนำตำแหน่งดังกล่าวไปลบออกจากภาพต้นฉบับเพื่อให้เหลือเฉพาะส่วนของมือซึ่งจะนำไปวิเคราะห์คุณลักษณะนั่นเอง เราจึงเลือกใช้วิธี Viola-Jones ซึ่งถูกนำเสนอโดย Paul viola และ Michael J. Jones ที่มีความเร็วและมีความถูกต้องในการตรวจจับสูงในปี 2001 โดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอวิธีการแทนรูปภาพที่ เรียกว่า “ภาพอินทิกรัล” (Integral Image) หรือที่รู้จักในชื่อตารางผลรวมของพื้นที่ดังแสดงในรูปที่ 2.8 ซึ่งช่วยให้การคำนวณพีเจอร์ทำได้รวดเร็วขึ้นและได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost (ซึ่งเลือกเอาเฉพาะ Critical features ที่ให้ classifiers ที่มีประสิทธิภาพสูงสุด) นอกจากนี้ยังได้อธิบายถึงการรวม classifiers แบบ cascade ซึ่งช่วย

ให้ส่วนพื้นหลังของภาพถูกปฏิเสธได้เร็วและเน้นการคำนวณไปที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น [15] ภาพอินทิกรัลสามารถสร้างได้ตามสมการที่ 2.4

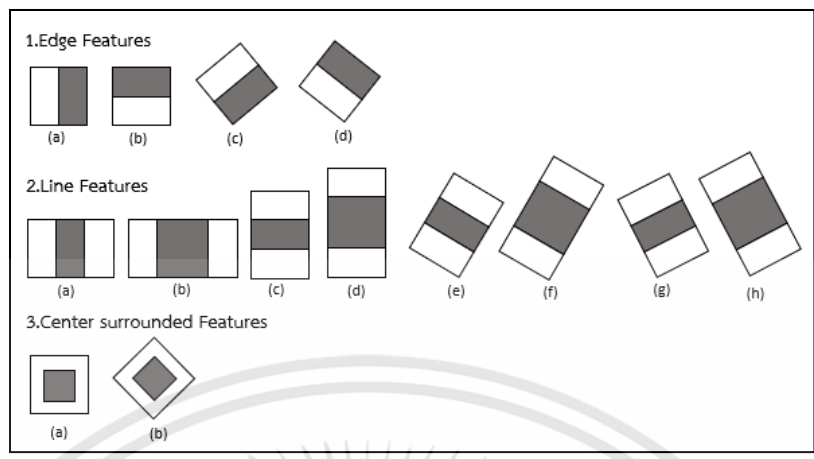
$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2.4)$$

เมื่อ $ii(x,y)$ คือ ภาพอินทิกรัลและ $i(x',y')$ คือภาพนำเขาจากโฟลวิดีโอที่จะนำมาทำการค้นหาวัตถุที่ต้องการ ซึ่งจะนำไปใช้ในขั้นตอนของ Cascade classifiers

1	1	1	1	2	3
1	1	1	2	4	6
1	1	1	3	6	9
Input image			Integral image		

รูปที่ 2.8 Input image และ Integral image

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการสแกน sub-window เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต การประมวลผลภาพแบบทั่วไปจะใช้การปรับขนาดภาพขาเข้าแตกต่างกันหลายๆขนาด และใช้ตัวตรวจหา (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ ซึ่งวิธีนี้ใช้เวลาในการคำนวณมาก เนื่องมาจากการคำนวณบนรูปภาพที่มีขนาดแตกต่างกัน ดังนั้น Viola-Jones จึงได้เสนอวิธีใหม่โดยการปรับขนาดตัวตรวจหาแทนที่จะปรับขนาดภาพขาเข้า และใช้ตัวตรวจหาค้นหาวัตถุหลายๆรอบ (แต่ละรอบใช้ขนาดแตกต่างกัน) ซึ่งทั้งสองวิธีนี้จะใช้เวลาในการคำนวณไม่ต่างกันมากนัก แต่ Viola-Jones ได้คิดค้นตัวตรวจหาที่ใช้จำนวนครั้งในการคำนวณคงที่แม้จะมีขนาดของภาพแตกต่างกัน โดยตัวตรวจหาดังกล่าวนี้สร้างขึ้นโดยใช้ features ของ Haar wavelets ดังแสดงในรูปที่ 2.9 การหารูปร่างของคุณลักษณะ (Feature) ในภาพจะใช้ตัวกรอง (Filter) ตามลักษณะพื้นฐานของ Haar wavelet จึงเรียกว่า Haar-like feature โดยตัวกรองจะมีลักษณะเป็นสี่เหลี่ยมและแบ่งเป็น 2 ส่วนคือสีขาวและสีดำ ในการทำงาน ภาพจะถูกกรองด้วยตัวกรองที่มีขนาดต่างๆกันและรูปร่างของคุณลักษณะที่ได้จากตัวกรองนี้จะนำไปใช้ในการกำหนดลักษณะของตัวจำแนกอย่างอ่อน (Weak classifier) ดังในสมการที่(2.5) และ (2.6) ต่อไป



รูปที่ 2.9 กลุ่มรูปแบบของ Haar-like feature

$$h_j(x) = \begin{cases} -1 & \text{if } f_j(x) \leq 0 \\ 1 & \text{otherwise} \end{cases} \quad (2.5)$$

เมื่อ $f(x)$ คือค่าความแตกต่างของผลรวมพิกเซลในพื้นที่ขาวและดำของ Haar-like feature ซึ่งมีค่าตามสมการที่ 2.6

$$f(x) = \text{Sum}_{(\text{Black rectangle})} - \text{Sum}_{(\text{White rectangle})} \quad (2.6)$$

โดยที่ x คือ ภาพตัวอย่างและ Sum คือผลรวมของภาพอินทิกรัลในบริเวณพื้นที่สีขาวและดำของ x

AdaBoost learning algorithm เป็นวิธีหาค่าตัวจำแนกอย่างอ่อน (Weak classifier) ที่มีความผิดพลาดของน้ำหนักน้อยที่สุด h_t เพื่อนำไปปรับน้ำหนักในรอบถัดไป ($t+1$) โดยเลือกส่งเสริม (boosting) น้ำหนักตัวที่ไม่ผ่านการจำแนก แต่ลดน้ำหนักตัวที่ผ่านการจำแนกตามขั้นตอนในรูปที่ 2.10

- Given $(x_1, y_1), \dots, (x_m, y_m) ; x_i \in X, y_i \in \{-1, +1\}$
- Initialize weights $D_1(i) = \frac{1}{m}$
- For $t = 1 \dots T$
 1. Find $h_t = \arg \min_{h_j \in H} \varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)]$
 2. If $\varepsilon_t \geq 1/2$ then stop
 3. Set $\alpha_t = \frac{1}{2} \log \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$
 4. Update

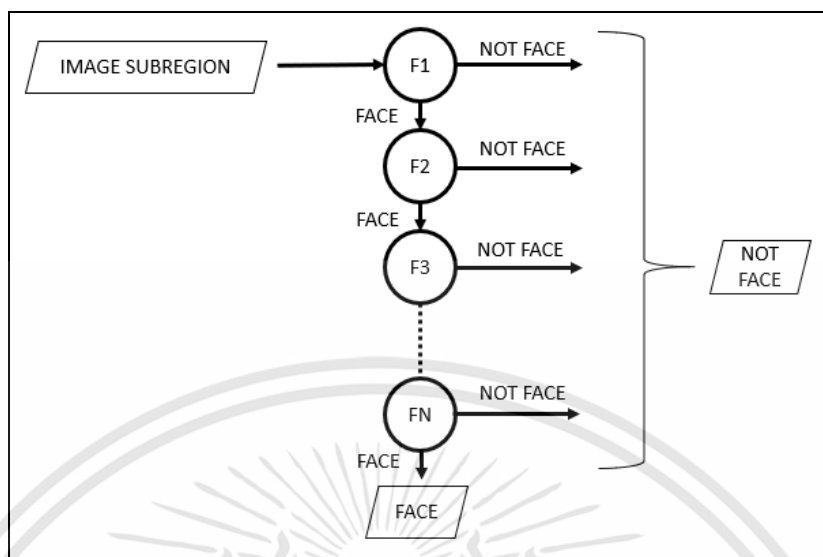
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$
 where Z_t is normalization factor

Output the final classifier :

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

รูปที่ 2.10 ขั้นตอนวิธีการเรียนรู้แบบ AdaBoost

หลักการของอัลกอริทึมค้นหาหน้าของ Viola-Jones คือการใช้ตัวตรวจหา สแกนหลายๆ ครั้งบนภาพเดิม แต่ด้วยขนาดที่แตกต่างกัน ถึงแม้ว่าจะมีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ได้โดยใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจว่าบริเวณใดๆ ไม่ใช่ใบหน้า นั้น ทำได้เร็วกว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ Cascaded (Cascaded classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.11 คือเมื่อ sub-window ถูกจัดประเภทเป็น ‘ไม่ใช่ใบหน้า’ (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็น ‘มีโอกาสเป็นใบหน้า’ (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้นของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยิ่งมีมากขึ้น



รูปที่ 2.11 ตัวจำแนกประเภทแบบ cascaded

2.4 การปรับปรุงภาพ (Enhancement)

หลังจากสามารถระบุช่วงของเฟรมภาพที่ต้องการและตำแหน่งของมือได้แล้วขั้นตอนต่อมาคือการนำเฟรมภาพในช่วงดังกล่าวมาผ่านกระบวนการแบ่งส่วนหรือ Segmentation เสียก่อนเพื่อให้เราได้เพียงวัตถุที่ต้องการ (เฉพาะส่วนของมือ) จากนั้นจึงจะนำภาพวัตถุที่ผ่านกระบวนการแบ่งส่วนแล้วไปปรับปรุงคุณภาพของภาพเพื่อกำจัดสิ่งรบกวนออกจากวัตถุที่เราต้องการหรือทำการเติมเต็มวัตถุนั้นกรณีเกิดความเสียหายจากการตรวจจับสีผิวหรือการเปรียบเทียบจุดสีก่อนเข้าสู่กระบวนการต่อไป

2.4.1 กระบวนการแบ่งส่วน (Segmentation Techniques)

เมื่อผ่านขั้นตอนการกำหนดเฟรมที่สนใจได้แล้วขั้นตอนต่อมาคือการสกัดเอาวัตถุที่สนใจหรือมือของผู้ทำท่าทางออกจากภาพโดยใช้เทคนิคการแบ่งส่วนทั่วไป 4 เทคนิค [16] ได้แก่ กระบวนการแบ่งโดยใช้พิกเซล (Pixel based), กระบวนการแบ่งโดยขอบ (Boundary based), กระบวนการแบ่งโดยพื้นที่ (Region based) และกระบวนการแบ่งโดยโมเดล (Model based) โดยการใช้งานแต่ละเทคนิคนั้นขึ้นอยู่กับจุดประสงค์และสภาพแวดล้อมของรูปภาพดังกล่าว

2.4.1.1 กระบวนการแบ่งโดยใช้พิกเซล (Pixel Based Segmentation) การแบ่งโดยใช้ค่าพิกเซลหรือจุดของภาพนั้นอาจเรียกอีกอย่างว่าการปรับโดยใช้ค่าเทรชโฮล(Threshold) ซึ่งเป็น

วิธีที่ง่ายที่สุดซึ่งขึ้นอยู่กับความต้องการค่าระดับสีของวัตถุในภาพที่เราต้องการ ยกตัวอย่างเช่นรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างภาพเพื่อใช้แบ่งวัตถุด้วยพิกเซล

หากเราต้องการเพียงภาพของผู้ชายที่กำลังกระโดด (ซึ่งถือเป็นวัตถุที่เราสนใจ) วิธีที่ง่ายที่สุดคือการแบ่งภาพโดยกำหนดจากค่าพิกเซล โดยจะสังเกตว่าตัวของผู้ชายในภาพมีสีที่แตกต่างจากพื้น หลังเราจึงสามารถแบ่งภาพได้ดังนี้

ขั้นที่ 1 กำหนดพิกเซลสีฟ้าซึ่งเป็นพื้นที่โดยส่วนใหญ่ของภาพโดยการตั้งค่าเทรชโฮลค่าหนึ่ง โดยสมมติว่าค่าพิกเซลที่ต่ำกว่าค่าเทรชโฮลจะเป็นค่าของวัตถุ

ขั้นที่ 2 ในตัวอย่างนี้กำหนดให้ Threshold (ของสีฟ้า) < 45 แล้วนำไปวิเคราะห์ที่ภาพไบนารีดังรูปที่ 2.13 พบว่ายังจะเหลือร่องรอยของวัตถุอื่นอยู่



รูปที่ 2.13 ภาพซึ่งถูกกำจัดพิกเซลสีฟ้าโดยค่าเทรชโฮล

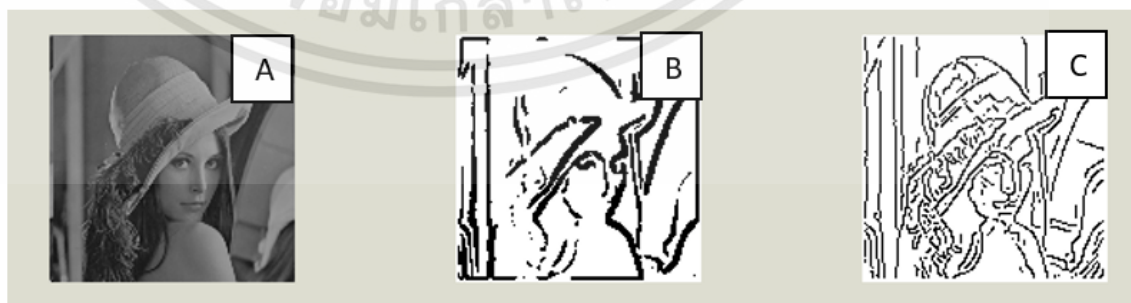
ขั้นที่ 3 ทำการปรับปรุงภาพโดยการเลือกใช้วิธีในหัวข้อ 2.4.2 เพื่อให้ได้แต่ภาพของผู้ชายดังกล่าว จากนั้นจึงทำการ Invert พิกเซลเพื่อให้วัตถุกลายเป็นสีดำ



รูปที่ 2.14 ภาพวัตถุสุดท้ายหลังจากปรับปรุงคุณภาพของภาพ

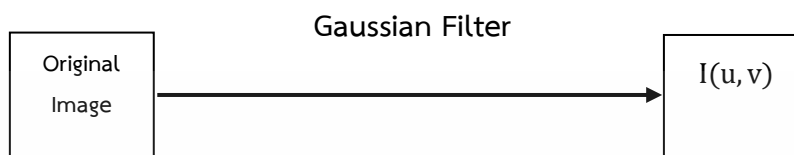
2.4.1.2 กระบวนการแบ่งโดยขอบ (Boundary based Segmentation) การแบ่งโดยใช้ขอบของภาพจะเป็นการแบ่งส่วนของภาพตามขอบ (Edge) โดยจะทำการหาพิกเซลที่เป็นขอบและเชื่อมต่อพิกเซลเหล่านั้นเพื่อสร้างรูปทรงของวัตถุที่สนใจ (Boundary) โดยปกติแล้วสามารถทำได้สองวิธีคือ วิธีทำด้วยมือโดยการลากเมาส์ตามเส้นขอบซึ่งจะใช้สร้างรูปทรงของวัตถุ หรืออีกวิธีคือ การทำแบบอัตโนมัติด้วยการใช้ตัวกรอง (Filter) ตรวจสอบขอบ ซึ่งพิกเซลของภาพจะถูกแบ่งเป็นส่วนของพิกเซลที่เป็นขอบและไม่ใช่ออกตามผลลัพธ์ของตัวกรอง ตัวอย่างของวิธีการแบ่งแบบนี้คือ Canny Edge Detector ซึ่งเป็นอัลกอริธึมสกัดขอบภาพเพื่อสร้างรูปทรงได้อย่างถูกต้องและมีประสิทธิภาพมากที่สุดวิธีหนึ่ง

Canny Edge Detector เป็นหนึ่งในวิธีการหาขอบรูปภาพ โดยประกอบด้วยขั้นตอนหลายขั้น โดยมีวัตถุประสงค์เพื่อเพิ่มความสามารถในการลดขนาดของขอบภาพลงและทำให้สามารถเลือกช่วงความเข้มของขอบภาพที่ต้องการได้ ตัวอย่างดังรูปที่ 2.15



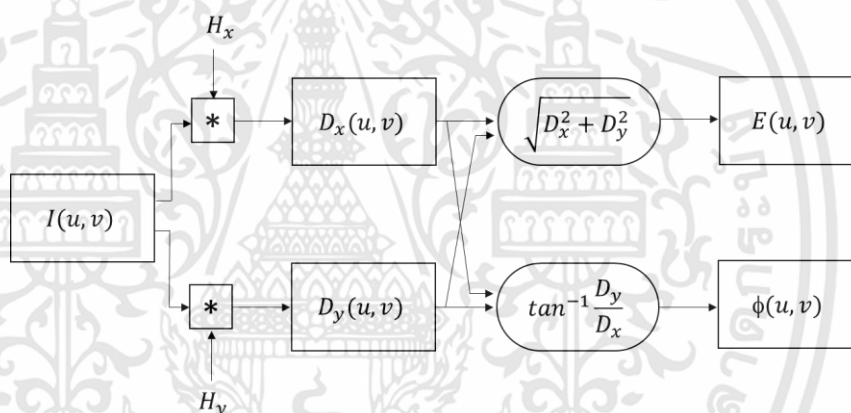
รูปที่ 2.15 A.) รูปต้นฉบับ B.) Edge strength and Edge orientation C.) Canny Operator โดยกระบวนการจะประกอบไปด้วยทั้งหมด 5 ขั้นตอน

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



รูปที่ 2.16 Gaussian Filtered

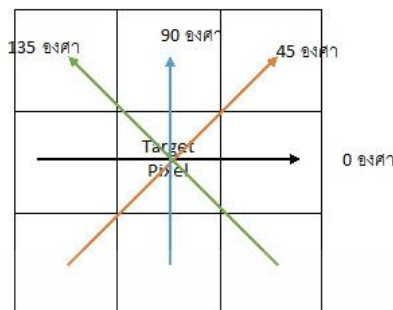
2. Edge strength และ Edge orientation หา Edge strength และ Edge orientation ของรูปที่ 2.15 ภาพที่ผ่าน Gaussian Filtered มาแล้ว โดยอาจใช้กระบวนการแบบ Prewitt หรือ Sobel ก็ได้



รูปที่ 2.17 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 2.18 ช่วงการเปลี่ยนค่าเพื่อใช้ใน Edge orientation

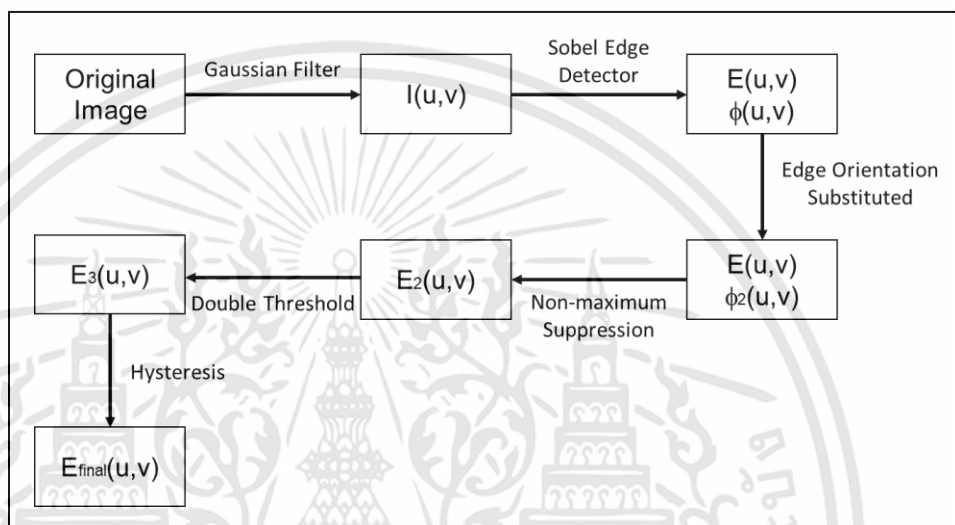
4. Non-maximum Suppression เมื่อเราทำการแปลงค่า Edge orientation ให้เป็นค่าที่สามารถระบุเป็นตำแหน่ง pixel โดยรอบได้แล้วนั้น ในกระบวนการนี้ เราจะใช้ค่า Edge orientation เพื่อเลือกคู่ Pixel โดยรอบที่เรียงตัวเป็นเป็นช่วงองศาขึ้น (positive/negative -arrangement pixel) มาทำการเปรียบเทียบค่า Edge strength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



รูปที่ 2.19 วิธีการ Non-maximum Suppression

5. Hysteresis หลังจากที่เราผ่านกระบวนการทั้งหมดมาแล้วนั้น ภาพที่ได้ยังมีความเป็นไปได้ที่จะยังหลงเหลือผลกระทบจาก noise ที่ไม่เกี่ยวข้องอยู่จำนวนหนึ่ง เพื่อตัดสิ่งรบกวนเหล่านี้ออกไปจึงมีการนำ Hysteresis เข้ามาช่วยโดยทำให้เราสามารถเลือกช่วงค่า Edge strength ที่ต้องการแสดงได้และค่าที่ต่ำกว่าในขอบเขตที่ระบุให้กำจัดค่าของ pixel นั้นทิ้ง (ให้เท่ากับ 0) ส่วนค่าที่มากกว่าให้เป็นขอบภาพความเข้มสูง (strong edge) และส่วนที่อยู่ในช่วงให้เป็นขอบภาพความเข้มจาง (weak edge) ซึ่งจะนำมาพิจารณาในขั้นตอนต่อไป

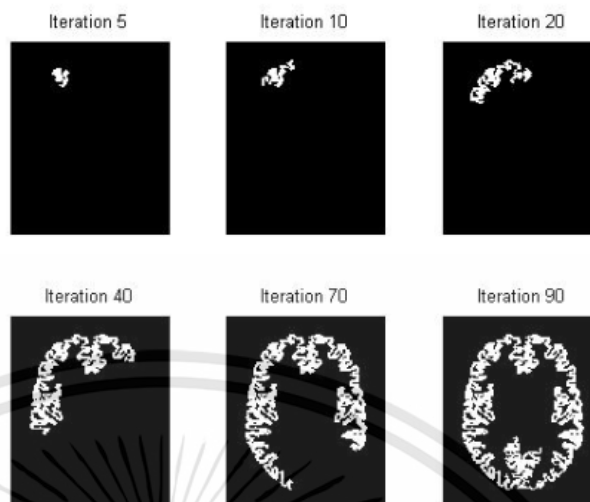
ในกระบวนการสุดท้ายเราจะทำการแยกขอบออกเป็นส่วนๆ โดยแบ่งตามตำแหน่งที่เชื่อมต่อกันและความเข้ม โดยขอบส่วนที่มีความเข้มจาง (weak edge) ที่ไม่เชื่อมต่อกับขอบส่วนที่มีความเข้มสูง (strong edge) จะถูกกำจัดทิ้ง เนื่องจาก เพราะโดยทั่วไปแล้วสันนิษฐานว่าส่วนที่เป็นขอบจริงจะต้องมีส่วนที่เชื่อมต่อกับขอบที่มีความเข้มสูงเท่านั้นเสมอ



รูปที่ 2.20 กระบวนการทั้ง 5 ขั้นตอนของ Canny Method

2.4.1.3 กระบวนการแบ่งโดยพื้นที่ (Region based) วิธีนี้เป็นการแบ่งส่วนของภาพออกไปเป็นกลุ่มต่าง ๆ หรือ Region โดยการค้นหาขอบของวัตถุของแต่ละกลุ่ม การหาขอบสามารถหาได้โดยการใช้ Edge detection ตามที่กล่าวมาแล้วหัวข้อที่ 2.4.1.2 แต่สำหรับในบางกรณีที่มีข้อมูลภาพมีสิ่งรบกวนหรือความแตกต่างของความเข้มระหว่างกลุ่มมีน้อยมากทำให้ไม่สามารถหาขอบของวัตถุได้ตั้งนั้นวิธีการหาขอบด้วยการเชื่อมขอบ (Edge linking techniques) จึงเป็นวิธีที่สมควรเลือกมาใช้มากกว่า ยกตัวอย่างเช่นวิธี Region growing

วิธีการ Region Growing จะนำพิกเซลข้างเคียงมาพิจารณาซึ่งจะทำการจัดกลุ่ม (Region) ของพิกเซลเหล่านี้เข้าไว้ด้วยกันโดยการพิจารณาถึงความเข้มของพิกเซล (ค่าของพิกเซลมีค่าใกล้เคียงกัน) ในการ Segmentation จะต้องมีการกำหนดกลุ่ม (Region) ต้องการจะแบ่ง โดยที่แต่ละกลุ่มจะต้องมีการกำหนดค่าความเข้มของพิกเซล ซึ่งมีไว้เพื่อใช้สำหรับการขยายตัวของกลุ่ม (growth) ในการขยายตัวของกลุ่มนี้จำเป็นจะต้องมีกฎเพื่อใช้เป็นวิธีการขยายตัวของกลุ่มรวมทั้งกฎของการตรวจสอบความเป็นอันหนึ่งอันเดียวกันของกลุ่ม (homogeneity) ของทุกระยะของการขยายตัว



รูปที่ 2.21 การประยุกต์ใช้วิธี Region Growing

2.4.1.4 กระบวนการแบ่งโดยโมเดล (Model based) เป็นกระบวนการแบ่งส่วนที่ต้องการข้อมูลของรูปภาพนั้นๆมากกว่าข้อมูลของพิกเซลที่อยู่รอบข้าง ตัวอย่างเช่น รูปทรงของวัตถุหรือรูปแบบทางเรขาคณิตที่ซ้ำกันของภายในพื้นที่ที่สนใจ จุดประสงค์ของวิธีการนี้คือความต้องการที่จะระบุข้อมูลของวัตถุที่จะนำไปฝึกสอน (train) เพื่อแสดงความน่าจะเป็นของข้อมูลตัวอย่างนั้นในเชิงสถิติ แล้วนำข้อมูลนั้นไปวิเคราะห์ความสัมพันธ์หรือความเกี่ยวข้องของข้อมูลวัตถุที่จะนำมาถูกทดสอบโดยใช้ข้อมูลทางสถิติดังกล่าว ตัวอย่างของกระบวนการแบ่งส่วนแบบนี้ก็คือ การประมาณรูปร่างผสมของเกาส์(Gaussian Mixture Model หรือ GMM)

วิธี GMM จะใช้การประมาณการค่ามากที่สุดด้วยเทคนิควิธีทางคณิตศาสตร์ที่เรียกว่า Expectation Maximization (EM) วิธีการดังกล่าวเป็นวิธีการทำซ้ำ (recursive) แบ่งเป็น 2 ขั้นตอนคือ 1. ขั้นตอนการประมาณการ (คาดคะเน) ซึ่งเป็นขั้นตอนของการคาดคะเนขั้นต้นก่อน และ 2. ขั้นตอนการปรับปรุงค่าแล้วส่งกลับไปให้ขั้นตอนที่หนึ่งทำงาน ซึ่งขั้นตอนทั้งหมดสามารถอธิบายได้จากสูตรทางคณิตศาสตร์ของฟังก์ชันความน่าจะเป็นของเกาส์ดังสมการที่ 2.7

$$g(\mathbf{x}; \mathbf{m}_k, \sigma_k) = \frac{1}{(\sigma\sqrt{2\pi})^D} e^{-\frac{1}{2}\left(\frac{\|\mathbf{x}-\mathbf{m}_k\|}{\sigma_k}\right)^2} \quad (2.7)$$

1. ขั้นตอนการประมาณการ (คาดคะเน)

$$p^{(i)}(\mathbf{k}|\mathbf{n}) = \frac{p_k^{(i)} g(\mathbf{x}_n; \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}{\sum_{k=1}^K p_k^{(i)} g(\mathbf{x}_n; \mathbf{m}_k^{(i)}, \sigma_k^{(i)})} \quad (2.8)$$

2. ขั้นตอนการปรับปรุงค่า

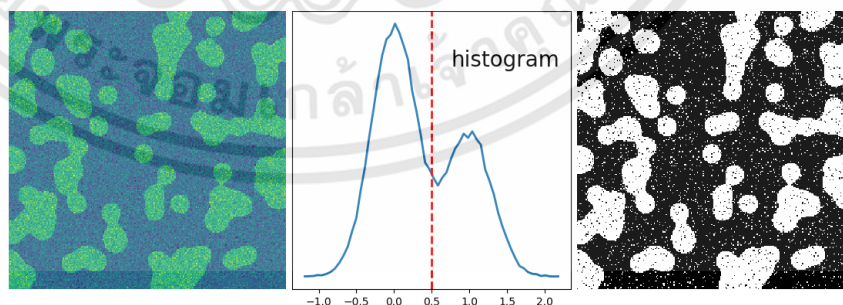
$$m_k^{(i+1)} = \frac{\sum_{n+1}^N p^{(i)}(k|n)x_n}{\sum_{n+1}^N p^{(i)}(k|n)} \quad (2.9)$$

$$\sigma_k^{(i+1)} = \sqrt{\frac{1}{D} \frac{\sum_{n+1}^N p^{(i)}(k|n) \|x_n - m_k^{(i+1)}\|^2}{\sum_{n+1}^N p^{(i)}(k|n)}} \quad (2.10)$$

$$p_k^{(i+1)} = \frac{1}{N} \sum_{n+1}^N p^{(i)}(k|n) \quad (2.11)$$

โดยที่ m_k	คือค่าเฉลี่ยของความเข้มของทุกๆจุดภาพ
σ_k	คือค่าเบี่ยงเบนมาตรฐาน
p_k	คือค่าถ่วงน้ำหนัก
$p(k n)$	คือค่าความน่าจะเป็นที่จุดภาพที่ n จะเป็นองค์ประกอบที่ k
N	คือจำนวนจุดภาพทั้งหมด
D	คือมิติของข้อมูล
K	คือจำนวนองค์ประกอบทั้งหมดของรูป
n	คือจุดภาพ
k	คือองค์ประกอบ (label)
l	คือครั้งที่ใช้ในการคำนวณ

โดยกระบวนการทั้งหมดจะหยุดทำต่อเมื่อขั้นตอนในรอบที่ผ่านมาเกือบรอบในปัจจุบันมีค่าใกล้เคียงกันมากๆ หรือก็คือค่าความน่าจะเป็นของความเข้มของแต่ละจุดภาพ



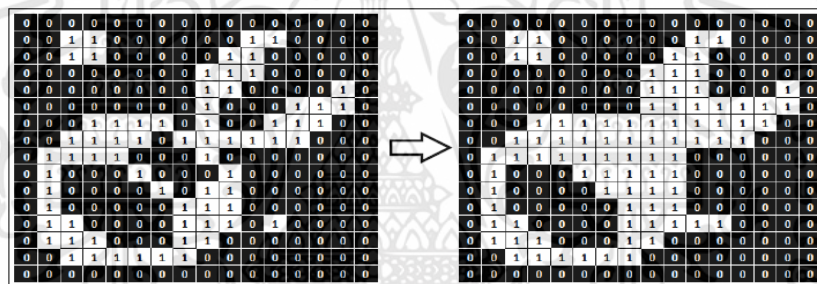
รูปที่ 2.22 ตัวอย่างการแบ่งส่วนโดยใช้ Gaussian model

2.4.2 กระบวนการกำจัดสิ่งรบกวน (Noise Reduction)

เมื่อผ่านกระบวนการข้างต้นมาจนได้เฉพาะวัตถุที่สนใจแล้ว ขั้นตอนต่อมาคือการปรับปรุงคุณภาพของภาพเพื่อกำจัดสิ่งรบกวนออกจากวัตถุที่เราต้องการหรือทำการเติมเต็มวัตถุนั้นกรณีเกิดความเสียหายจากการตรวจจับสีผิวหรือการเปรียบเทียบจุดสีเพื่อหาวัตถุที่สนใจก่อนนำเข้าสู่กระบวนการหาขอบของภาพต่อไป

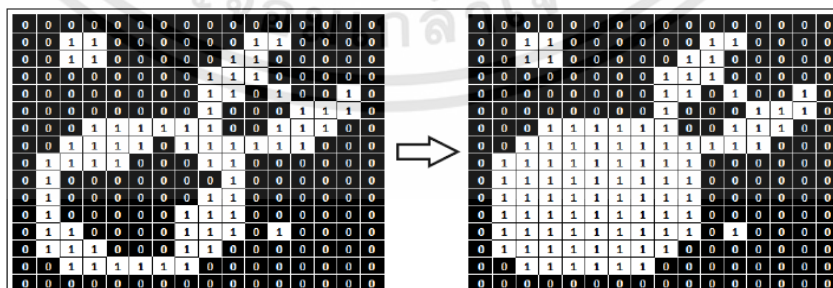
2.4.2.1 Closing Operation เป็นการนำเทคนิค Erosion และ Dilation มาใช้ร่วมกันโดยเริ่มจากการทำ Erosion ก่อนแล้วจึงตามด้วย Dialation เพื่อให้ส่วนแคบๆของพื้นที่ต่อเข้าด้วยกันสามารถอธิบายได้ดังสมการที่ 2.12

$$A \cdot B = (A \oplus B) \ominus B \quad (2.12)$$

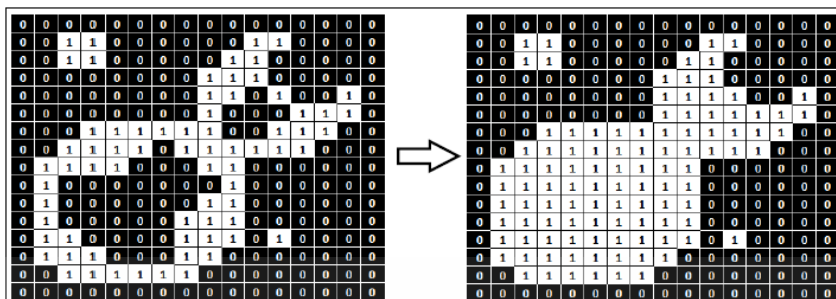


รูปที่ 2.23 แสดง 3x3 pixels Closing Operation

2.4.2.2 Fill Holes Operation เป็นการเติมส่วนที่เป็นช่องว่างที่เกิดขึ้นในวัตถุจากการหาพิกเซลที่ถูกล้อมรอบ โดยอาศัยการหาจุดเชื่อมต่อระหว่างขอบของวัตถุที่เกิดขึ้นกับขอบของวัตถุข้างเคียง ซึ่งสามารถแบ่งจุดเชื่อมต่อได้ 2 ลักษณะ คือ 4-connectivity และ 8-connectivity

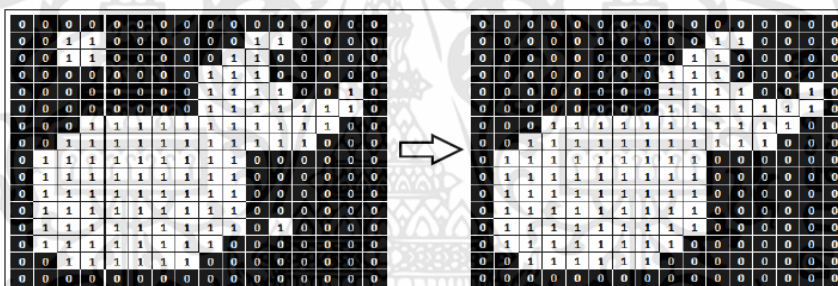


รูปที่ 2.24 Fill Holes 4-connectivity Operation



รูปที่ 2.25 Fill Holes 8-connectivity Operation

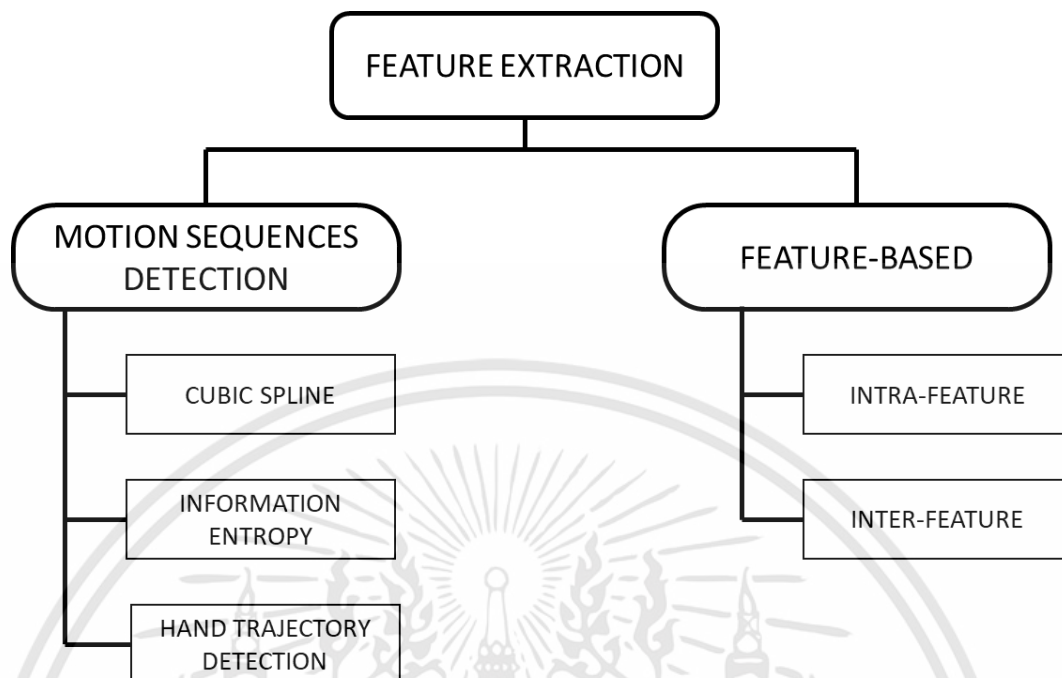
2.4.2.3 Region Selection ใช้ในการเลือกกลุ่มของข้อมูลที่ต้องการ โดยอาศัยการหาจุดเชื่อมต่อระหว่างขอบของวัตถุที่เกิดขึ้นกับขอบของวัตถุข้างเคียง ซึ่งสามารถแบ่งจุดเชื่อมต่อได้ 2 ลักษณะ คือ 4-connectivity และ 8-connectivity เพื่อใช้ในการตรวจสอบขนาดของกลุ่มข้อมูลให้ได้ตามต้องการ



รูปที่ 2.26 การเลือกกลุ่มของข้อมูลที่มากกว่า 20 พิกเซล

2.5 การสกัดคุณลักษณะ (Feature Extraction)

หัวข้อนี้จะกล่าวถึงทฤษฎีทั่วไปที่เกี่ยวข้องกับการสกัดคุณลักษณะ เมื่อได้มาซึ่งช่วงของเฟรมท่าทางในหัวข้อที่ 2.2 แล้ว (สำหรับท่าทางที่มีการเคลื่อนไหว หรือ Gesture) เราจำเป็นต้องนำช่วงของท่าทางที่ได้มาวิเคราะห์ผลว่าควรทำการทดลองอย่างไรต่อไปโดยจะแบ่งเป็น 2 หัวข้อได้แก่ การตรวจจับลำดับการเคลื่อนไหว (Motion Sequences detection) ซึ่งใช้วิเคราะห์เลือกเฟรมที่สำคัญเพื่อนำมาทำการทดลอง และชนิดของคุณลักษณะ (Feature-based) เพื่อใช้บ่งบอกความสัมพันธ์ของข้อมูล



รูปที่ 2.27 แสดงภาพรวมของหัวข้อการสกัดคุณลักษณะ

2.5.1 การตรวจจับลำดับการเคลื่อนไหว (Motion Sequences detection)

เนื่องจากคำภาษามือนั้นเป็นท่าทางที่มีการเคลื่อนไหวตลอดเวลา เราจึงไม่สามารถรู้ได้ว่าความสั้น-ยาว ของท่าทางนั้นมีมากน้อยเพียงใด แม้วิธีการก่อนหน้าจะทำให้เราทราบเฟรมเริ่มต้นและสิ้นสุดของท่าทางแล้ว แต่ในการนำข้อมูลเข้ามาวิเคราะห์ผลนั้นจำเป็นอย่างยิ่งที่เราต้องทำให้ข้อมูลขาเข้า(input) มีขนาดเท่ากันและสามารถเก็บรายละเอียดของท่าทางดังกล่าวไว้ได้มากที่สุด ดังนั้นเราจึงต้องใช้ทฤษฎีที่เกี่ยวข้อง 3 วิธีได้แก่ การประมาณค่าในช่วงด้วยเส้นโค้ง (Cubic Spline), เอนโทรปี (Information Entropy) และการตรวจจับเส้นทางของมือ(Hand trajectory) เพื่อนำมาวิเคราะห์จำแนกคำภาษามือภาษาอังกฤษซึ่งได้แก่

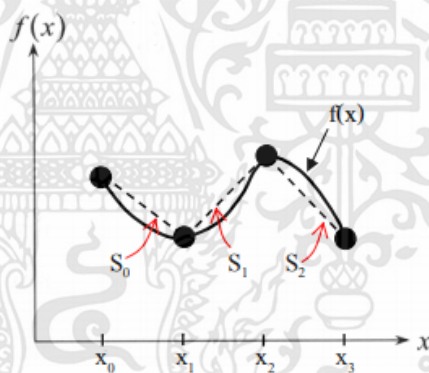
2.5.1.1 การประมาณค่าในช่วงด้วยเส้นโค้ง (Cubic Spline)

จะมีหลักการที่แตกต่างโดยสิ้นเชิงเมื่อเปรียบเทียบกับ การประมาณค่าในช่วงด้วยฟังก์ชันโพลีโนเมียล ทั้งแบบนิวตันและแบบลากรางจ์ซึ่งในวิธีแบบ Interpolating Polynomials เราจะได้ฟังก์ชัน (ต่อเนื่อง) ที่พยายามลากผ่านทุกๆจุดของข้อมูลถึงจำนวนจุดข้อมูลมากขึ้น ก็จะต้องใช้ฟังก์ชันพหุนาม ลำดับที่สูงขึ้น (higher order) ซึ่งทำให้ได้รูปกราฟที่เป็นเส้นโค้งกลับไปกลับมา ดังรูปที่ 2.28 โดยมีข้อสังเกตว่าจำนวนจุดวกกลับ = $n-1$ (เท่ากับจำนวนอันดับลบหนึ่ง เช่น $n=2$,

กราฟเป็นรูปพาราโบลา มีจุดวกกลับ = 2-1 เท่ากับ 1 จุด) ซึ่งวิธีการเช่นนี้จะก่อให้เกิดปัญหาเกี่ยวกับข้อมูลบางกรณี เช่น ในกรณีของ step function ซึ่งในกรณีเช่นนี้หลักการประมาณค่าในช่วงด้วย Splines จะใช้ผลได้ดี

หลักการของ Splines คือเราจะแทนค่าจุดข้อมูล $n+1$ จุดด้วย Splines จำนวน n เส้น โดยจะแบ่งเป็น 3 กรณีได้แก่ Linear Splines, Quadratic Splines และ Cubic Splines

1. Linear Splines สมมติว่าเรามีจุดข้อมูลอยู่ 4 จุดที่ตำแหน่ง x_0, x_1, x_2 และ x_3 โดยหลักการเราสามารถ fit 3 Linear Splines (คือ S_0, S_1 และ S_2) กับจุดข้อมูลเหล่านี้ ดังกราฟในรูปที่ 2.28 ในที่นี้ฟังก์ชัน $f(x)$ อาจเป็นฟังก์ชันจริงที่ลากผ่านข้อมูลทั้ง 4 จุด (หรือพูดอีกอย่างว่าข้อมูล 4 จุดเหล่านี้ถูกวัดค่าหรือ sampling หรือถูกเก็บข้อมูลออกมาเป็นฟังก์ชัน $f(x)$) ดังนั้น Splines, S_j (เป็น Linear function ชนิด piece-wise continuous function) จะเป็นการประมาณค่าฟังก์ชันจริง $f(x)$ ในแต่ละช่วงของค่า x ที่ Splines ตัวนั้นๆอยู่



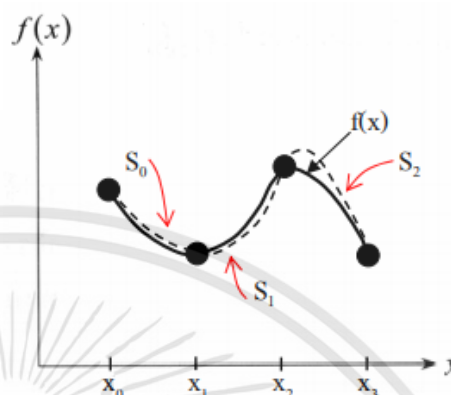
รูปที่ 2.28 กราฟของ Linear Splines

จุดต่อของ Linear Splines ที่เห็นได้ชัดคือ S_j (Spline ใดๆ) จะไม่ต่อเนื่องที่จุดข้อมูลหรือที่ node(s) โดยรูปแบบของ Linear splines สามารถเขียนได้ดังสมการที่ 2.13 หรือก็คือสมการเส้นตรงนั่นเอง

$$S_j = a_j + b_j(x - x_j) \quad (2.13)$$

2. Quadratic Splines จากกรณีของ Linear Splines เราจะสามารถแก้ปัญหาค่าไม่ต่อเนื่องที่ node โดยใช้ Splines ที่มี order สูงขึ้น ในที่นี้คือ Quadratic splines

(order=2) ซึ่งจะทำให้ Splines 2 อันต่อเนื่องที่ node แบบ first derivative พิจารณาได้จากรูปที่ 2.29

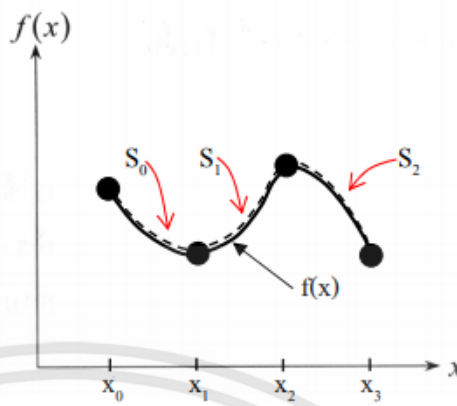


รูปที่ 2.29 กราฟของ Quadratic Splines

แต่ถึงแม้ Splines จะต่อเนื่องกัน (แบบ first derivative หรือแบบ Slope) ในบางกรณีข้อมูลที่เปลี่ยนแปลงกะทันหัน ในช่วงระยะ x ที่สั้นเกินไป Quadratic splines ก็จะไม่เพียงพอจากค่าที่ควรจะเป็น โดยรูปแบบสมการของ Quadratic splines เป็นดังนี้

$$S_j = a_j + b_j(x-x_j) + c_j(x-x_j)^2 \quad (2.14)$$

3. Cubic Splines ซึ่งเป็นวิธีการที่ใช้มากที่สุดในทางปฏิบัติ(โดยเฉพาะโปรแกรมกราฟิกต่างๆ) เหตุผลก็คือวิธี Cubic splines จะ fit curve ได้ smooth มาก ถึงแม้ว่าเส้นโค้งที่ได้จะประกอบจาก Splines หลายๆตัวต่อกันแต่เมื่อใช้สายตามนุษย์ในการมองก็จะเห็นเป็นเส้นเดียวลากยาวตลอด และเหตุผลอีกประการก็คือ นอกจากจะได้ smooth curve แล้วยังใช้ order ของ polynomial แค่เพียงอันดับสามเท่านั้น รูปที่ 2.30 แสดงการ fit curve ที่ผ่าน 4 จุดด้วย Cubic splines เมื่อเปรียบเทียบกับ Linear splines และ Quadratic splines แล้วจะพบว่ามีความใกล้เคียงฟังก์ชัน $f(x)$ จริงมากที่สุดและเส้นโค้ง S_j ที่ลากต่อกันมีความ smooth จนเหมือนเส้นเดียวกัน ที่เป็นเช่นนี้เพราะ Cubic splines มีความต่อเนื่องที่ nodes จนถึง 2nd Derivative นั่นเอง

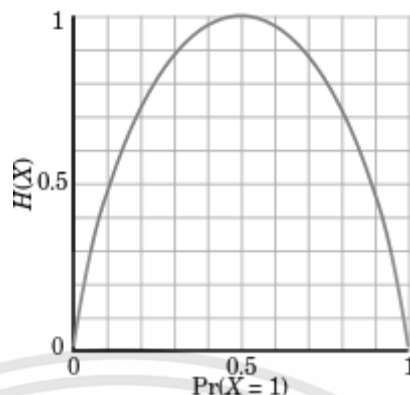


รูปที่ 2.30 กราฟของ Cubic Splines

2.5.1.2 เอนโทรปี (Information Entropy)

ถูกนำเสนอโดย Shannon เป็นนักฟิสิกส์, วิศวกรไฟฟ้าและนักคณิตศาสตร์ Entropy ถูกใช้ในการคำนวณหาความไม่บริสุทธิ์ (impurity) ของข้อมูลหมายความว่าข้อมูลที่เราสงสัยนั้น มีความแตกต่างกัน ยกตัวอย่างการส่งข้อมูล (transmission) ผ่าน channel หนึ่ง ข้อมูลของเรานั้นแรกเริ่มมีความบริสุทธิ์ (purity) แต่เมื่อส่งผ่าน channel ก็มักจะมี noise เข้ามารบกวนหรือ ข้อมูลอาจจะ lost ระหว่างส่งทำให้ข้อมูลเมื่อไปถึงฝั่งรับมีความไม่บริสุทธิ์ สมมุติว่าเราส่ง 11111111 ผ่าน channel แต่เมื่อถึงฝั่งรับกลับกลายเป็น 11110011 แสดงว่า impurity ของข้อมูลเราเพิ่มขึ้นเพราะมีส่วนที่แตกต่างไปจากเดิมคือมี 00 ปนเข้ามา อีกนัยหนึ่งคือให้ เนื้อหา (content) มากขึ้นเพราะจากเดิมที่มีแต่ 1 ก็มี 0 เพิ่มเข้ามา Entropy นั้นมีค่าอยู่ระหว่าง 0-1 และจะมีค่าสูงขึ้นเมื่อข้อมูลมีความแตกต่างกันมาก

สมมุติว่ามีถึงลูกบอลคละสีเป็น เขียว-แดงจำนวน 100 ลูก ค่าของ entropy จะมีค่ามากที่สุดเมื่อในถังมีลูกบอลสีเขียว 50 ลูกและมีลูกบอลสีแดง 50 ลูก (ต่างกันมากเพราะมีเท่ากัน) แต่ถ้ามีสีเขียวมากหรือสีแดงมากกว่า entropy จะเริ่มลดลงเพราะมีแนวโน้มที่ข้อมูลจะเหมือนกันมากขึ้นดังแสดงในรูปที่ 2.31 ให้อัตราส่วนระหว่างลูกบอล เขียว:แดง = 90:10 และ 70:30 ตามลำดับจะได้ว่าข้อมูลชุดที่ 1 จะมี entropy น้อยกว่าชุดที่ 2 คือถ้ามีอันใดอันหนึ่งมากเกินไป entropy จะลดลง



รูปที่ 2.31 แสดงกราฟของค่า entropy

Entropy สามารถคำนวณได้โดยใช้สูตรตามสมการที่ 2.15

$$\text{Entropy}(x_i) = -P(x_i) \log_2 P(x_i) \quad (2.15)$$

อธิบายได้ว่าหากต้องการทราบว่าแต่ละ class ให้ content มากน้อยขนาดไหน (ในที่นี้คือกี่ปิต) เราจะคำนวณได้จาก \log ฐาน 2 ของความน่าจะเป็นของ class นั้นๆ เช่นตัวอักษร A-Z มี 26 ตัว ความน่าจะเป็นของแต่ละตัวคือ $1/26$ จะได้ $-\log_2(1/26) = 4.700$ ถ้าตัวอักษรไทย ก-ฮ ก็ $1/44$ จะได้ $-\log_2(1/44) = 5.459$ เป็นต้น entropy หาได้จากความน่าจะเป็นของคลาสนั้นๆ คูณด้วย \log ฐาน 2 ของความน่าจะเป็นของคลาสนั้นเอง โดย x คือ class สามารถมีได้มากกว่า 1 class ฉะนั้น entropy ก็คือผลรวม probability ของแต่ละ class สามารถเขียนใน form ทั่วไปได้เป็นสมการที่ 2.16

$$H(x) = -\sum_{i=1}^n P(x_i) \log_2 P(x_i) \quad (2.16)$$

2.5.1.3 การตรวจจับเส้นทางของมือ (Hand trajectory detection)

โดยปกติแล้วภาพของมือที่ไม่มีการเคลื่อนไหวนั้นจะสามารถหาคุณลักษณะเด่นออกมาได้จากลักษณะของมือ (จำนวนนิ้ว, มุมของนิ้ว) และองศาของการหมุนของฝ่ามือได้เลย ซึ่งแตกต่างจากการวิเคราะห์วิดีโอของท่าทางมือซึ่งมีการเคลื่อนไหวตั้งแต่ต้นจนจบ ดังนั้นเราจึงประยุกต์ใช้การตรวจจับเส้นทางของมือร่วมกับองศาการหมุนของมือฝ่ามือและรูปร่างของมือเพื่อช่วยในการตรวจสอบและจำแนกว่าท่าทางเหล่านั้นเป็นท่าทางของคำว่าอะไร และเนื่องจากการทำท่าทางของคำภาษามือนั้นมีการเปลี่ยนแปลงของรายละเอียดรูปร่างมือและนิ้วค่อนข้างต่ำมาก

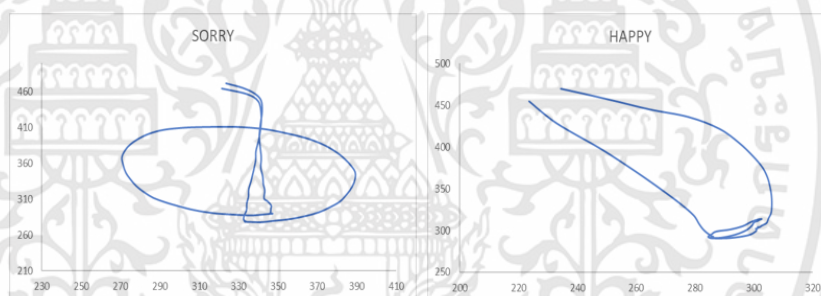
เราจึงต้องวิเคราะห์ลักษณะของมือโดยใช้ Global motion เป็นหลัก นั้นหมายถึงการคำนวณคุณสมบัติของมือโดยอ้างอิงจากข้อมูลโดยรวมของภาพ

ดังนั้นเนื่องจากการเปลี่ยนแปลงของ Local movement แอปไม่สามารถสังเกตได้เราจึงต้องกำหนดจุดอ้างอิงบนฝ่ามือขึ้นมาโดยกำหนดให้เป็น

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\} \quad (2.17)$$

โดยที่ x และ y เป็นตำแหน่งของพิกเซลบนภาพซึ่งใช้อ้างอิง
 i^{th} เป็นตำแหน่งจุดอ้างอิงของแต่ละเฟรม

ตัวอย่างการหาเส้นทางการขยับมือโดยใช้ Global movement คือค่า 'เซ็นทรอยด์' ของมือ ตั้งแต่ยกมือทำท่าทางจนกระทั่งเอามือลงปรากฏดังรูปที่ 2.32

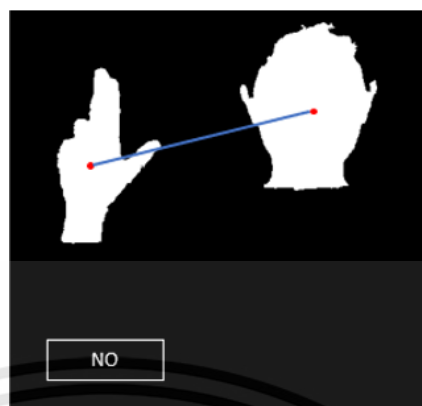


รูปที่ 2.32 แสดงเส้นทางของมือของท่าทางSorry (ซ้าย) Happy (ขวา)
 โดยพลอตจากค่าของเซ็นทรอยด์ของมือ

2.5.2 ชนิดของคุณลักษณะ(Feature-based)

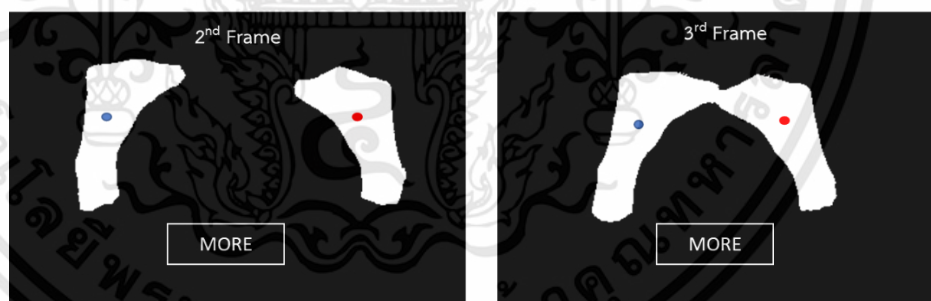
ในหัวข้อนี้เราต้องการจะแสดงความสัมพันธ์ของคุณลักษณะภายในเฟรมเดียวกัน (Intra-Feature) และคุณลักษณะระหว่างเฟรม(Inter-Feature)

2.5.2.1 Intra-Feature คำว่า Intra แปลว่า ภายใน ดังนั้น Intra-Feature คือคุณลักษณะที่แสดงความสัมพันธ์เฟรมเดียวกัน โดยเป็นค่าที่บ่งบอกเอกลักษณ์ของภาพในเฟรมนั้นๆ โดยไม่ได้มีการเปรียบเทียบกับเฟรมอื่นๆ ยกตัวอย่างเช่น คุณลักษณะ ระยะห่างระหว่างเซ็นทรอยด์ของหน้าและมือภายในเฟรมดังรูปที่ 2.33



รูปที่ 2.33 แสดงคุณลักษณะภายใน (Intra-feature)

2.5.2.2 Inter-Feature ตรงกันข้ามกับ Intra-Feature คำว่า Inter มีความหมายว่า ระหว่าง ดังนั้นคุณลักษณะประเภทนี้จะมีความสำคัญในการใช้เปรียบเทียบความแตกต่างระหว่างเฟรม เพื่อบ่งบอกถึงความเปลี่ยนแปลงของข้อมูลที่เราสสนใจ ยกตัวอย่างเช่น คุณลักษณะ ระยะห่างเซ็นทรอยด์ระหว่างมือของเฟรมถัดกัน พิจารณาจากรูปที่ 2.34 คุณลักษณะดังกล่าวจะใช้วัดระยะห่างของเซ็นทรอยด์ของมือซึ่งเป็นจุดสีเดียวกัน (ระยะห่างระหว่างจุดสีน้ำเงินจุดแรกด้านซ้ายและจุดที่สองด้านขวา) ซึ่งจะใช้บอกได้ว่ามือได้เปลี่ยนแปลงไปด้วยระยะเท่าไรนั่นเอง



รูปที่ 2.34 แสดงคุณลักษณะระหว่างคลาส (Inter-feature)

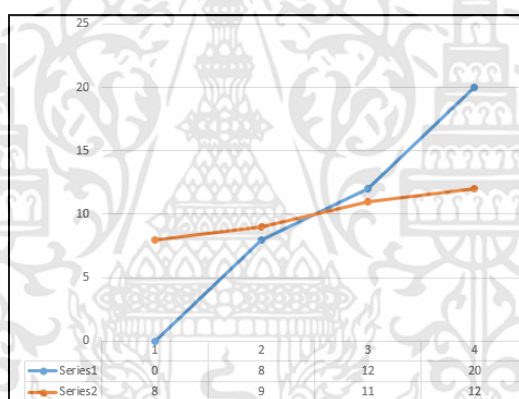
2.6 การวิเคราะห์องค์ประกอบหลัก (Principal Components Analysis)

Principal Components Analysis หรือ PCA คือการวิเคราะห์ข้อมูลโดยเฉพาะส่วนประกอบที่สำคัญเท่านั้น ซึ่งเป็นเทคนิคหนึ่งที่ใช้กระบวนการทางสถิติและเมทริกซ์ (Matrix) เข้ามาอธิบายข้อมูลให้เป็นที่เข้าใจได้ง่ายยิ่งขึ้น รูปแบบของข้อมูลที่ว่านี้มักอยู่ในรูปของตารางเพื่อลดความซับซ้อนลง โดยเราจะ

สร้างโมเดลขึ้นมาใหม่ ให้เป็นที่เข้าใจได้ง่าย เหมือนการมองชุดข้อมูลด้วยมุมมองใหม่ แต่ไม่ได้ทำการเปลี่ยนแปลงข้อมูลดิบเลยแม้แต่ตัวเดียว ตัวอย่างที่นำ PCA ไปใช้เช่น การตรวจจับใบหน้า การบีบอัดรูปภาพ และการหารูปแบบโครงสร้างของข้อมูลหลายมิติ

PCA เกิดจากผสมผสานเทคนิคต่างๆทางคณิตศาสตร์บางเรื่องเข้าด้วยกัน โดยเทคนิคที่ใช้ใน PCA ประกอบด้วย 6 องค์ประกอบต่อไปนี้ ได้แก่ ส่วนเบี่ยงเบนมาตรฐาน, ค่าความแปรปรวนของข้อมูล, โควาเรียนซ์, โควาเรียนซ์ เมทริกซ์, เวกเตอร์ลักษณะเฉพาะ (Eigen Vectors) และ ค่าลักษณะเฉพาะ (Eigenvalues)

2.6.1 ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) คือสถิติที่ใช้บอกว่าข้อมูลมีความกระจุกกระจายมากน้อยขนาดไหน พิจารณาจากกราฟข้างล่าง



รูปที่ 2.35 การกระจายของข้อมูล 2 ข้อมูล

จากรูปกราฟด้านบน ค่าเฉลี่ย(\bar{X}) ของแต่ละ Series มีค่าเท่ากันคือ 10

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.18)$$

แต่ 10 ไม่ได้บอกอะไรมากมายถึงความแตกต่าง จากรูปจะเห็นได้ว่ากราฟสีน้ำเงินมีการกระจายตัวมากกว่า แต่กราฟสีส้มมีข้อมูลที่ใกล้เคียงกัน การจะบอกถึงความแตกต่างนี้ ทางสถิติมีตัววัดตัวหนึ่งคือ Standard Deviation วิธีการหาค่าเฉลี่ยของระยะทางระหว่างจุดแต่ละจุด กับจุดกึ่งกลาง (จุดค่าเฉลี่ย) เขียนเป็นสมการได้ว่า

$$S = \sqrt{\frac{\sum_{i=1}^n (x-\bar{x})^2}{n-1}} \quad (2.19)$$

จากข้อมูลข้างต้น จะได้ว่า

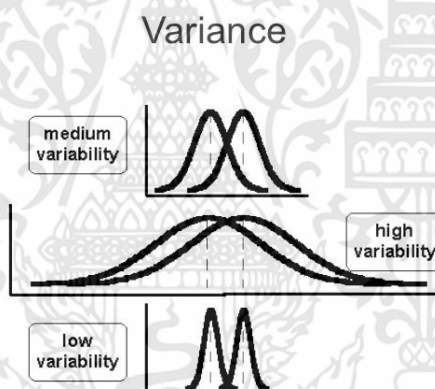
series1 มี Standard Deviation = 8.3266639978645

series2 มี Standard Deviation = 1.8257418583506

ค่า SD จึงใช้บอกความกระจายของข้อมูล และทำให้เห็นภาพได้ชัดเจนยิ่งขึ้นนั่นเอง

2.6.2 ค่าความแปรปรวนของข้อมูล (Variance) ใช้วัดค่าการกระจายตัวของข้อมูลเช่นเดียวกับส่วนเบี่ยงเบนมาตรฐาน โดยมีสูตรเป็น

$$S^2 = \frac{\sum_{i=1}^n (x-\bar{x})^2}{n-1} \quad (2.20)$$



รูปที่ 2.36 ค่าความแปรปรวนของข้อมูลระดับต่างๆ

2.6.3 โควาเรียนซ์ (Covariance) หรือการวิเคราะห์ความแปรปรวนร่วม คือการผสมผสานของการวิเคราะห์ระหว่างการถดถอยกับการวิเคราะห์ความแปรปรวน เราจะใช้โควาเรียนซ์เมื่อตัวแปรตาม Y ได้รับอิทธิพลจากทรีทเมนต์และยังมีความสัมพันธ์เชิงเส้นกับตัวแปร X ตัวอื่นๆ โดยใช้สัญลักษณ์ $COV(X,Y)$ ซึ่งคำนวณได้จากค่าเฉลี่ยของผลคูณระหว่างส่วนเบี่ยงเบนจากค่ากลาง(mean) ของลักษณะทั้งสอง

$$\text{cov}(X,Y) = \frac{\sum_{i=1}^n (X_i-\bar{X})(Y_i-\bar{Y})}{(n-1)} \quad (2.21)$$

เมื่อ \bar{x} คือ ค่าเฉลี่ยของข้อมูล X และ

\bar{y} คือ ค่าเฉลี่ยของข้อมูล y

n คือ จำนวนของข้อมูล

2.6.4 โควาเรียนซ์เมทริกซ์ (Covariance Matrix) คือ Matrix ของ Covariance ทุกคู่ระหว่างมิติต่างๆ ยกตัวอย่างเช่น ถ้าข้อมูล 3 มิติ x, y และ z ดังนั้นแล้ว Covariance Matrix จะเขียนได้ว่า

$$\begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{pmatrix}$$

จุดที่น่าสังเกตคือ แกนหลัก (main diagonal) ก็คือ Variance นั่นเอง นอกจากนี้แล้ว $\text{cov}(a,b) = \text{cov}(b,a)$ ดังนั้นเมทริกซ์นี้จึงสมมาตรกันในแกนหลักโดยสูตรของ Covariance คือ

$$C^{n \times n} = (c_{ij}, c_{ij} = \text{cov}(\text{dim}_i, \text{dim}_j)) \quad (2.22)$$

ซึ่ง $C^{n \times n}$ เป็นเมทริกซ์ n แถวและ n คอลัมน์ และ dim_x คือมิติที่ x ของข้อมูลนั้นๆ

2.6.5 เวกเตอร์ลักษณะเฉพาะ (Eigen Vectors) คือการคูณกันระหว่าง matrix กับ vector แต่ที่พิเศษมากไปกว่าการคูณเมทริกซ์ธรรมดา ก็คือ ผลคูณระหว่าง matrix กับ vector นี้ ก็คือค่า vector ที่เอาไปคูณนั่นเอง เพียงแต่อาจมีขนาดสั้นยาวของเวกเตอร์ผลลัพธ์ที่อาจแตกต่างไปจาก vector ตัวตั้งที่เอาไปคูณ

$$Av = \lambda v \quad (2.23)$$

2.6.6 ค่าลักษณะเฉพาะ (Eigen Values) คือค่าที่มากับ Eigen Vector หรือก็คือ λ นั่นเอง คือค่าที่บอกว่าหากมีการ transform Eigen Vector นั้นๆ แล้วและเมื่อมีการตั้งค่าตัวร่วมมันก็คือค่า Eigen value นั่นเอง

จากองค์ประกอบที่กล่าวมาข้างต้น เราสามารถนำมาจัดการกับข้อมูลเพื่อค้นหารูปแบบของข้อมูลนั้นๆ ซึ่งถือเป็นการมองข้อมูลในมุมใหม่ที่แสดงถึงความสัมพันธ์กันภายในข้อมูลชุดนั้นๆ สมมุติว่าเรามี

ข้อมูลหลายมิติ ยากที่จะทำความเข้าใจ หรือยากที่จะแสดงออกมาในมุมมองที่มีความหมาย PCA จึงเป็นตัวช่วยซึ่งใช้ลดความซับซ้อนข้อมูลตรงนี้ลงไป

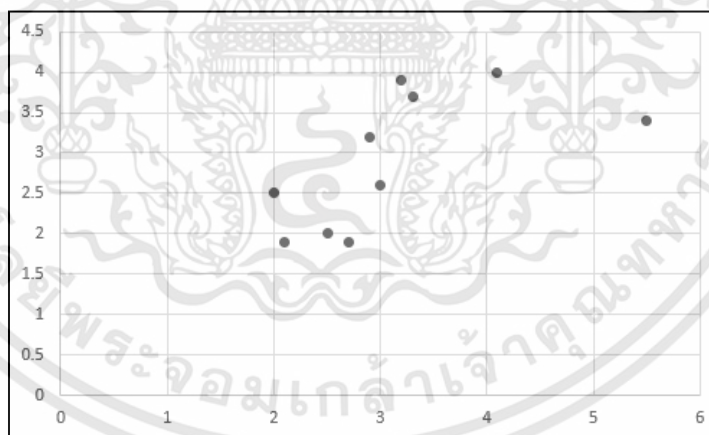
ข้อดีอีกอย่างหนึ่งก็คือหลังจากที่เราเจอแบบแผน (Pattern) ข้อมูลแล้ว เราสามารถลดขนาดของข้อมูลโดยการลดมิติที่ไม่ค่อยมีความสำคัญออกไปได้ โดยที่ไม่ได้สูญเสียข้อมูลไปมากมายนัก จุดนี้สามารถเอาไปใช้ประโยชน์ในงานอื่นๆได้เช่น การบีบอัดรูปภาพดิจิทัล เป็นต้น

โดยขั้นตอนการทำ PCA สามารถแบ่งย่อยได้เป็น 7 ขั้นตอน ได้แก่ เก็บข้อมูล, นำข้อมูลติดลบกับค่าเฉลี่ย, คำนวณค่า Covariance matrix, คำนวณค่า Eigen values และ Eigen vectors, เลือก Feature vector, สร้างข้อมูลชุดใหม่ในมิติของ Eigen vectors และ แสดงข้อมูลชุดใหม่ที่มีมิติน้อยลง

1. ข้อมูล ในตัวอย่างต่อไปนี้จะใช้ข้อมูล 2 มิติเพื่อให้ง่ายต่อการพลอตกราฟและทำให้เห็น ภาพเปรียบเทียบได้ชัดเจน ดังตัวอย่างโจทย์ตารางที่ 2.1 ด้านล่างนี้

x	5.5	2.7	3.2	2.9	4.1	3.3	3	2	2.5	2.1
y	3.4	1.9	3.9	3.2	4	3.7	2.6	2.5	2	1.9

ตารางที่ 2.1 ตัวอย่างโจทย์



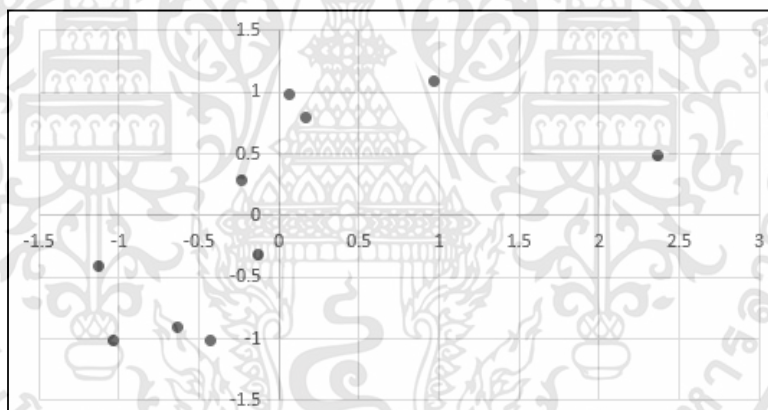
รูปที่ 2.37 กราฟ scatter plot ของตารางที่ 2.1

2. นำข้อมูลดิบกลับกับค่าเฉลี่ย จากโจทย์ตารางที่ 2.1 จะได้ค่าเฉลี่ย $\bar{X}=3.13$ และ $\bar{Y}=2$.

ดังตารางที่ 2.2

X	Y	$x-\bar{x}$	$y-\bar{y}$
5.5	3.4	2.37	0.49
2.7	1.9	-0.43	-1.01
3.2	3.9	0.07	0.99
2.9	3.2	-0.23	0.29
4.1	4	0.97	1.09
3.3	3.7	0.17	0.79
3	2.6	-0.13	-0.31
2	2.5	-1.13	-0.41
2.5	2	-0.63	-0.91
2.1	1.9	-1.03	-1.01

ตารางที่ 2.2 แสดงการนำข้อมูลดิบกลับกับค่าเฉลี่ย



รูปที่ 2.38 กราฟ scatter plot ของตารางที่ 2.2

3. คำนวณค่า Covariance Matrix

$$\text{cov} = \begin{pmatrix} 1.06455556 & 0.54522222 \\ 0.54522222 & 0.69433333 \end{pmatrix}$$

4. คำนวณ Eigen vectors และ Eigen values

$$\text{Eigen values} = \begin{pmatrix} 1.45523381 \\ 0.30365508 \end{pmatrix}$$

$$\text{Eigen vectors} = \begin{pmatrix} 0.81286254 & -0.58245557 \\ 0.58245557 & 0.81286254 \end{pmatrix}$$

จุดที่สำคัญตรงนี้คือ Eigen vectors ต้องเป็น unit eigenvectors หรือก็คือมีความยาวเท่ากับ 1 เช่น

$$\sqrt{0.81286254^2 + 0.58245557^2} = 1$$

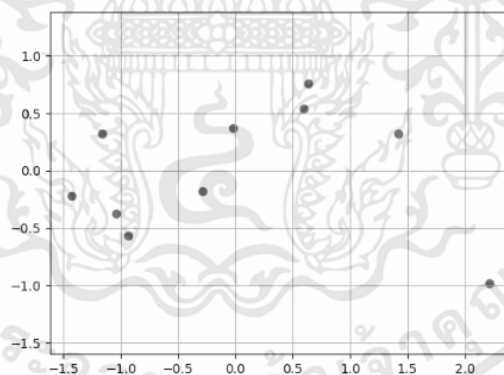
5. เลือก Feature vector จากหัวข้อ Eigen Vectors เราจะเห็นว่าค่า λ ที่มีค่ามากที่สุดคือองค์ประกอบที่สำคัญของข้อมูล (Principal component) ดังนั้นเราจะทำการเรียงลำดับ Eigen Vector จากมากไปหาน้อยได้

$$\text{Feature vector} = \begin{pmatrix} 0.81286254 \\ 0.58245557 \end{pmatrix}$$

ถ้าหากมีหลายมิติเราจะจัดเรียก Feature vector เป็นดังนี้

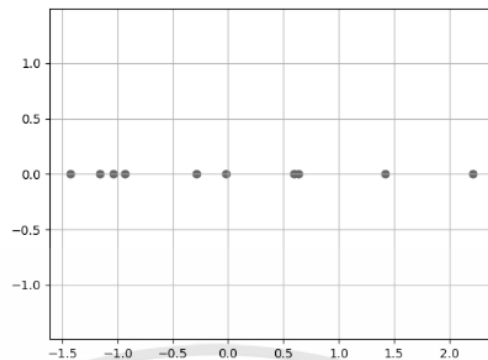
$$\text{Feature vector} = (\text{eig}_1 \text{eig}_2 \text{eig}_3 \dots \text{eig}_n) \quad (2.24)$$

แล้วจึงทำการเลือก Feature vector ที่ต้องการ หากสมมุติว่าถ้าเราเลือก Feature vector จาก Eigen vector ทุกตัว (2มิติ) จะได้กราฟเป็นดังตารางด้านล่างนี้



รูปที่ 2.39 กราฟแสดงการเลือก Feature vector ทุกตัว

แต่หากเราเลือก Feature vector เพียงตัวเดียว (1มิติ) จะได้กราฟเป็นดังด้านล่างนี้โดยจะเห็นได้ว่าเป็นการปรับมิติที่ 2 ให้เป็น 0 นั่นเอง



รูปที่ 2.40 กราฟแสดงการเลือก Feature vector เพียงตัวเดียว

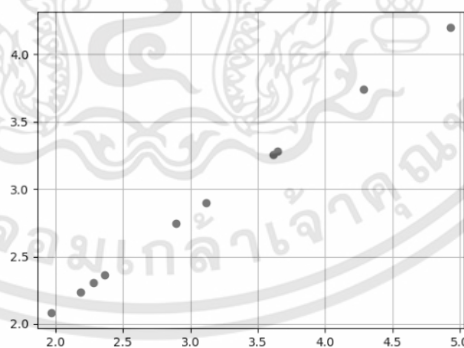
6. สร้างข้อมูลชุดใหม่ในมิติของ Eigen vectors คือขั้นตอนสุดท้ายของกระบวนการ PCA เราจะทำการแปลงค่าข้อมูลเดิม กับ Feature vector ที่เราเลือกไว้ แล้วเราก็จะได้ข้อมูลใหม่ สังเกตว่าตอนแรกเป็น 10×2 มิติ แต่เมื่อเปลี่ยนด้วยกระบวนการนี้โดยเลือก Feature vector เพียงตัวเดียว ก็จะเหลือแค่ 10×1 มิติ เท่านั้น

$$\text{FinalData} = \text{DataAdjust} \times \text{FeatureVector} \quad (2.25)$$

7. แสดงข้อมูลชุดใหม่ที่มีมิติน้อยลง

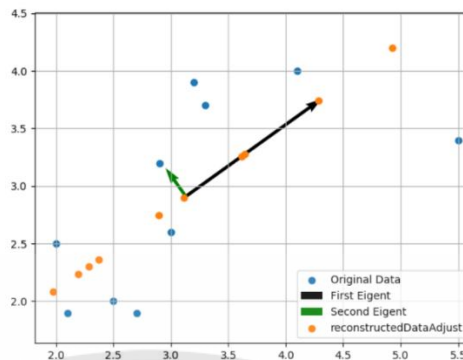
$$\text{reconstructedData} = \text{FinalData} \cdot \text{FeatureVector}^T \quad (2.26)$$

$$\text{reconstructedDataAdjust} = \text{reconstructedData} + \text{means} \quad (2.27)$$



รูปที่ 2.41 กราฟ reconstructedDataAdjust

และถ้าเราลองพลอตทั้งข้อมูลเดิม, ข้อมูลที่ทำการลดมิติและ Eigent Vector ทั้งหมดจะได้

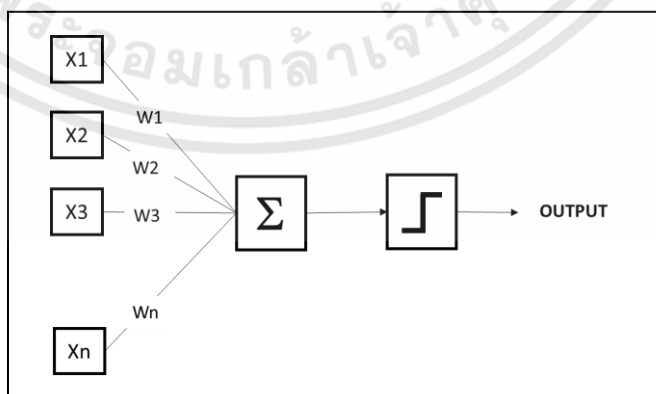


รูปที่ 2.42 กราฟแสดงข้อมูลทั้งหมดของตัวอย่างโจทย

2.7 โครงข่ายประสาทเทียม (Artificial neural network)

เป็นหนึ่งในเทคนิคของการทำเหมืองข้อมูล (Data Mining) คือโมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนกชันนิสต์ (Connectionist) เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์

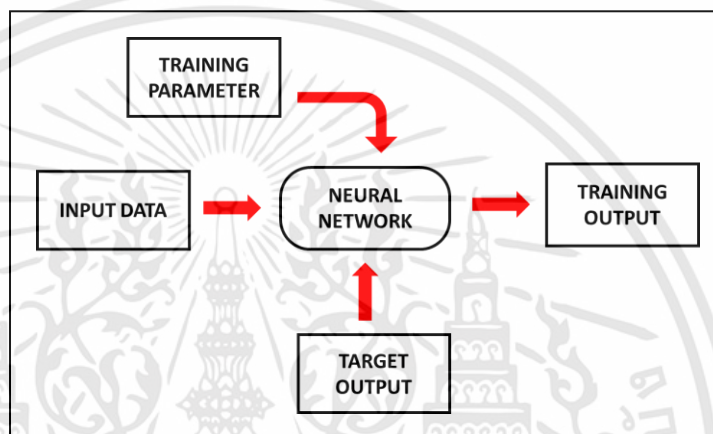
หลักการทำงานสำหรับคอมพิวเตอร์ Neurons ประกอบด้วย input และ output เหมือนกัน โดยจำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะสามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทำงานร่วมกันการทำงานนี้ในทางตรรกะแล้วก็จะเหมือนกับปฏิ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



รูปที่ 2.43 แสดง Model ของ Neuron ในคอมพิวเตอร์

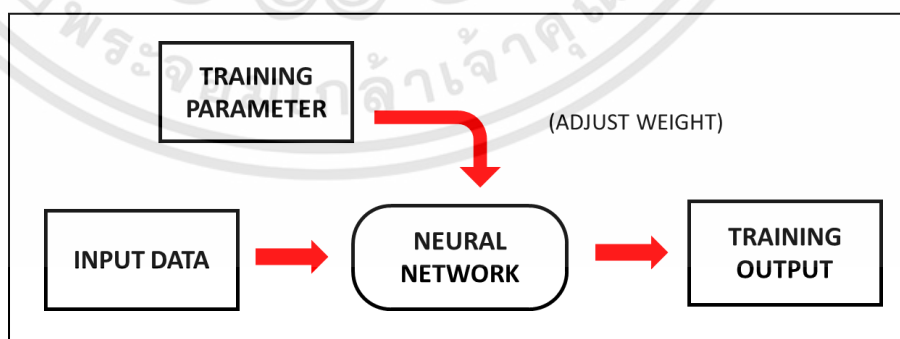
2.7.1 ประเภทของการเรียนรู้โครงข่ายประสาทเทียม

2.7.1.1 Supervised Learning การเรียนแบบมีการสอน เป็นการเรียนแบบที่มีการตรวจคำตอบเพื่อให้วงจรข่ายปรับตัว ชุดข้อมูลที่ใช้สอนวงจรข่ายจะมีคำตอบไว้คอยตรวจสอบว่าวงจรข่ายให้คำตอบที่ถูกหรือไม่ ถ้าตอบไม่ถูก วงจรข่ายก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น (เปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ)



รูปที่ 2.44 แสดงการเรียนรู้แบบ supervised learning

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



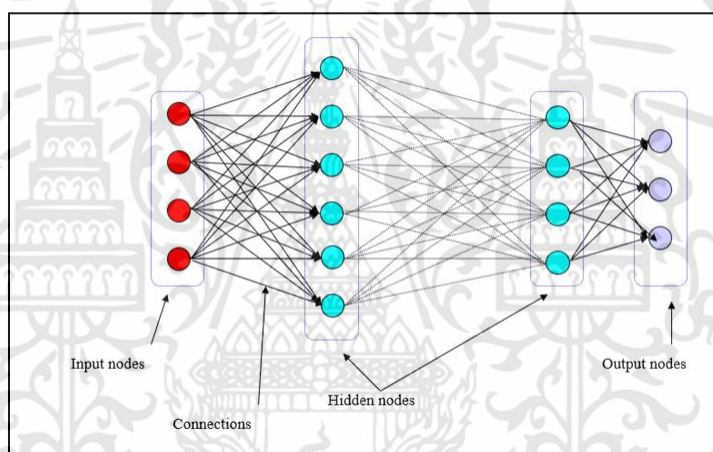
รูปที่ 2.45 แสดงการเรียนรู้แบบ unsupervised learning

2.7.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม

เป็นการแบ่งลักษณะตามโครงสร้างและวิธีการทำงานของโครงข่ายประสาทเทียมโดยแบ่งออกเป็น 2 แบบดังนี้

2.7.2.1 โครงข่ายการส่งข้อมูลแบบไม่ย้อนกลับ (Feedforward Network)

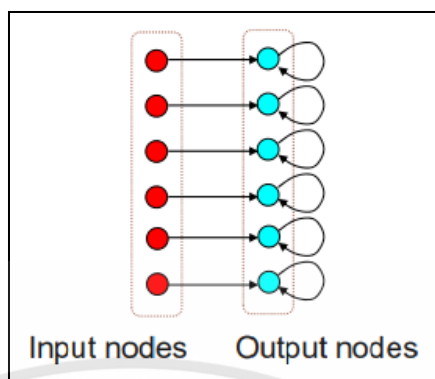
เป็นสถาปัตยกรรมที่กำหนดให้การส่งข้อมูลจากข้อมูลในชั้นข้อมูลขาเข้า (Input Layer) เข้ามาภายในชั้นซ่อน (Hidden Layer) และส่งไปยังข้อมูลขาออก (Output Layer) จะมีทิศทางของการไหลของข้อมูลไปในทิศทางเดียวกันจาก ข้อมูลขาเข้าส่งต่อไปเรื่อยๆจนถึงข้อมูลขาออก โดยไม่มีการย้อนกลับของข้อมูลหรือแม้แต่ nodes ใน layer เดียวกันก็ จะไม่มีการเชื่อมต่อกัน



รูปที่ 2.46 โครงสร้างโครงข่ายประสาทเทียมแบบไม่ย้อนกลับ

2.7.2.2 โครงข่ายการส่งข้อมูลแบบย้อนกลับ (Feedback Network)

เป็นสถาปัตยกรรมที่กำหนดให้การส่งข้อมูลจากข้อมูลในชั้นข้อมูลขาเข้า (Input Layer) เข้ามาภายในชั้นซ่อน (Hidden Layer) และส่งไปยังข้อมูลขาออก (Output Layer) จะมีทิศทางของการไหลของข้อมูลไปในทิศทางเดียวกัน จาก ข้อมูลขาเข้าส่งต่อไปเรื่อยๆจนถึงข้อมูลขาออก โดยมีการย้อนกลับของข้อมูลเพื่อให้การจำแนกมีประสิทธิภาพมากขึ้นนั่นเอง



รูปที่ 2.47 โครงสร้างโครงข่ายประสาทเทียมแบบย้อนกลับ

2.7.2.3 อัลกอริทึมแบ็กพรอพาเกชัน (Back propagation Algorithm)

อัลกอริทึมนี้เป็นการเรียนรู้ของโครงข่ายประสาทเทียมที่นิยมใช้ในโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network) โดยข้อมูลจากชั้นอินพุตจะถูกคำนวณและส่งผ่านฟังก์ชันจากชั้นแฝงไปยังชั้นเอาต์พุต ซึ่งหลักการสำคัญของการเรียนรู้คือ การเปลี่ยนแปลงค่าน้ำหนักของแต่ละเส้นเชื่อมระหว่างโหนด โดยในการปรับแก้ค่าน้ำหนักจะขึ้นกับค่าความแตกต่างระหว่างค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ สำหรับขั้นตอนการปรับแก้ค่าน้ำหนักมีขั้นตอนดังต่อไปนี้

1. กำหนดค่าอัตราเร็วในการเรียนรู้ (rate parameter: r)
 2. สำหรับแต่ละตัวอย่างอินพุตให้ทำตามขั้นตอนต่อไปนี้จนกว่าได้ระดับ performance ที่ต้องการ
 - I. คำนวณหาค่าเอาต์พุตโดยใช้ค่าน้ำหนักเริ่มต้นซึ่งอาจได้จากการสุ่ม
 - II. คำนวณหาค่า β ผลต่างที่ได้รับสำหรับการเปลี่ยนค่าเอาต์พุตของแต่ละโหนด
- ในชั้นเอาต์พุต (Output Layer)

$$\beta_z = d_z - o_z \quad (2.8)$$

เมื่อ d_z = ค่าเอาต์พุตที่ต้องการ
 o_z = ค่าเอาต์พุตที่คำนวณได้

- ในชั้นซ่อน (Hidden Layer)

$$\beta_z = \sum w_{jk} o_k (1 - o_k) \beta_k \quad (2.29)$$

เมื่อ w_{jk} = น้ำหนักของเส้นเชื่อมระหว่างชั้นที่ j กับ k


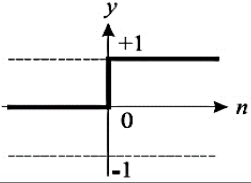

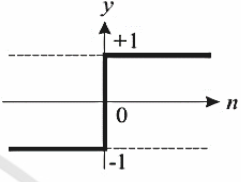

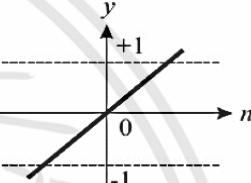

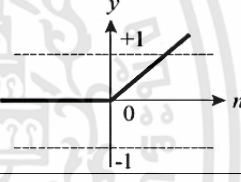

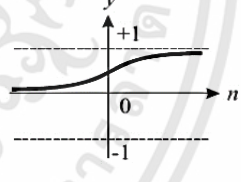

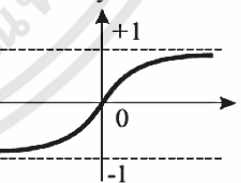
- คำนวณค่าน้ำหนักที่เปลี่ยนแปลงไปสำหรับในทุกน้ำหนัก ด้วยสมการต่อไปนี้

$$\Delta w_{ij} = r o_i o_j (1 - o_j) \beta_j \quad (2.30)$$

- เพิ่มค่าน้ำหนักที่เปลี่ยนแปลง สำหรับตัวอย่างอินพุตทั้งหมด และเปลี่ยนค่าน้ำหนัก

2.7.3 ฟังก์ชันถ่ายโอน (Transfer Function)

ฟังก์ชันการแปลง เป็นส่วนที่ทำหน้าที่รวมค่าเชิงตัวเลขจากเอาต์พุตของนิวรอล แล้วทำการตัดสินใจว่าจะส่งสัญญาณเอาต์พุตออกไปในรูปใด ฟังก์ชันการแปลงสามารถเป็นได้ทั้งแบบเชิงเส้นหรือไม่เป็นเชิงเส้น การเลือกใช้ฟังก์ชันการแปลงจะขึ้นอยู่กับลักษณะของระบบ ที่นำเอาโครงข่ายประสาทเทียมไปประยุกต์ใช้ ฟังก์ชันการแปลงมีอยู่หลายรูปแบบดังตารางที่ 2.3

ชื่อฟังก์ชัน	สมการ ความสัมพันธ์	MATLAB ฟังก์ชัน	สัญลักษณ์	กราฟความสัมพันธ์
ฮาร์ดลิมิต (hard limit)	$y = 0$ ถ้า $n < 0$ $y = 1$ ถ้า $n \geq 0$	hardlim		
ฮาร์ดลิมิตแบบ สมมาตร (symmetrical hard limit)	$y = -1$ ถ้า $n < 0$ $y = +1$ ถ้า $n \geq 0$	hardlims		
เส้นตรง (linear)	$y = n$	purelin		
เส้นตรงบวก (positive linear)	$y = 0$ ถ้า $n < 0$ $y = n$ ถ้า $n \geq 0$	poslin		
ซิกมอยแบบ ลอการิทึม (log- sigmoid)	$y = \frac{1}{1+e^{-n}}$	logsig		
ซิกมอยแบบเส้น สัมผัส ไฮเปอร์ โบลาร์ (hyperbolic tangent sigmoid)	$y = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	tansig		

ตารางที่ 2.3 ฟังก์ชันการแปลงแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4 ประโยชน์ของโครงข่ายประสาทเทียม

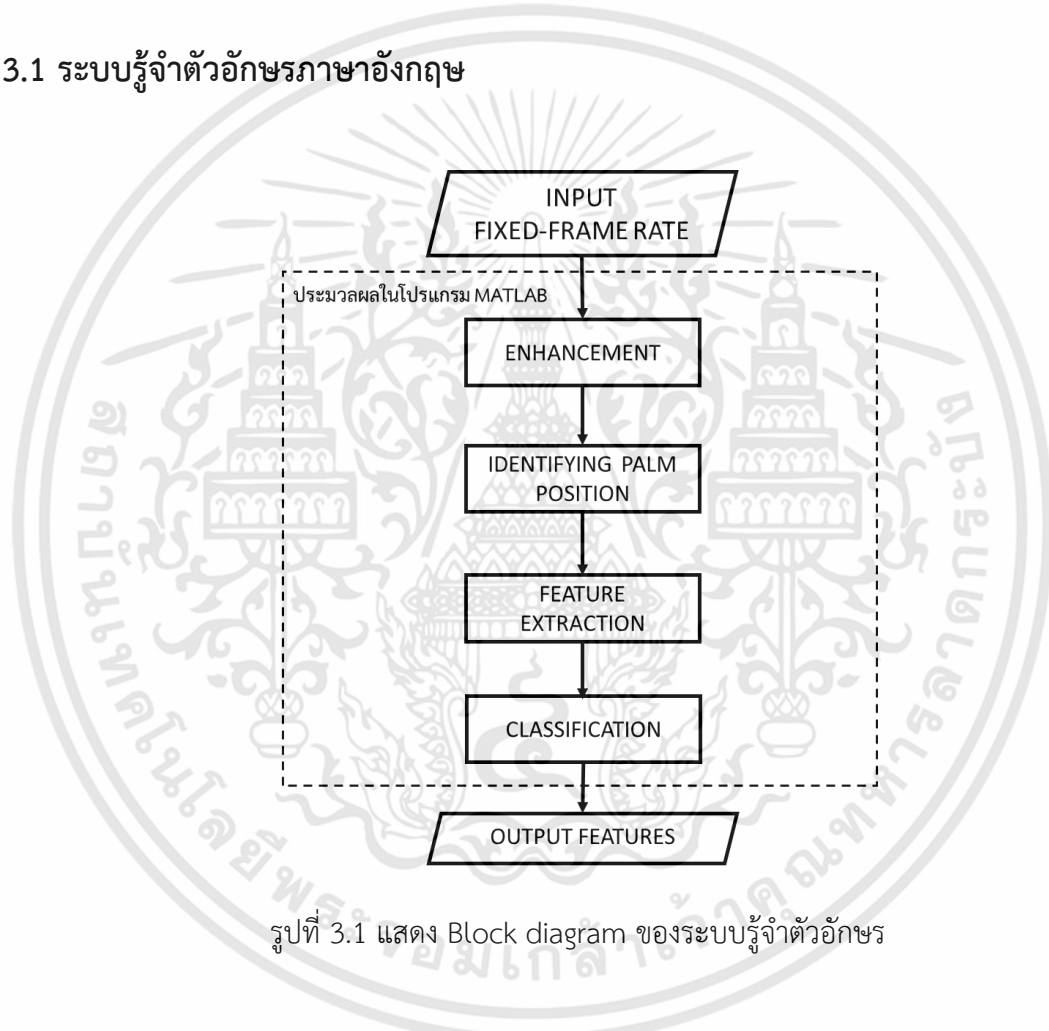
- เกิดข้อผิดพลาดได้ยาก (Fault Tolerance) หากระบบโครงข่ายประสาทเทียมประกอบไปด้วยโครงข่ายที่ใช้ในการประมวลผลมากมายหลายโครงข่าย ความผิดพลาดที่เกิดขึ้นจากเพียงหนึ่งหรือสองโครงข่ายจะไม่ทำให้ทั้งระบบเกิดข้อผิดพลาดได้
- ความสามารถในการหาเหตุผล (Generalization) เมื่อระบบโครงข่ายประสาทเทียมได้รับข้อมูลนำเข้าที่ไม่ครบถ้วนหรือไม่เพียงพอต่อการหาข้อสรุป หรือได้รับข้อเท็จจริงที่ไม่เคยได้รับมาก่อน ระบบจะสามารถลำดับการเชื่อมโยงข้อเท็จจริงจนสามารถให้ข้อสรุปและเหตุผลได้
- ความสามารถในการปรับเปลี่ยน (Adaptability) โครงข่ายประสาทเทียมสามารถเรียนรู้สภาพแวดล้อมใหม่ได้ ดังนั้นเมื่อมีเหตุการณ์ใหม่ๆ เข้าสู่ระบบก็จะสามารถปรับเปลี่ยนหรือปรับปรุงองค์ความรู้ให้ทันสมัยตามเหตุการณ์ใหม่นั้น
- ความสามารถในการพยากรณ์ (forecasting Capability) โครงข่ายประสาทเทียมสามารถนำข้อมูลทางสถิติเดิมที่มีอยู่ในระบบ มาใช้คาดการณ์หรือพยากรณ์ข้อมูลในอนาคตได้

บทที่ 3

วิธีการดำเนินงาน

ขั้นตอนการดำเนินงานจะแบ่งออกเป็น 2 การทดลองได้แก่ ระบบรู้จำตัวอักษรและระบบรู้จำคำในชีวิตประจำวัน จากนั้นทั้งสองการทดลองจะถูกนำเข้าสู่กระบวนการวิเคราะห์จำแนกกลุ่มโดยใช้โครงข่ายประสาทเทียม

3.1 ระบบรู้จำตัวอักษรภาษาอังกฤษ



รูปที่ 3.1 แสดง Block diagram ของระบบรู้จำตัวอักษร

จากรูปที่ 3.1 แสดงการทำงานของระบบรู้จำตัวอักษรภาษามือ โดยเริ่มจากการรับข้อมูลภาพมือด้วยกล้อง Webcam ซึ่งมีการกำหนดค่าเฟรมเรทเป็นค่าคงที่ค่าหนึ่ง จากนั้นข้อมูลสัญญาณภาพที่ได้จะถูกส่งเข้าไปประมวลผลด้วยโปรแกรม MATLAB เพื่อทำการปรับปรุงภาพแล้วจึงเข้าสู่กระบวนการระบุตำแหน่งของฝ่ามือ จากนั้นภาพจะผ่านคุณลักษณะต่างๆที่ถูกเขียนขึ้นในโปรแกรมเพื่อนำไปวิเคราะห์จำแนกชนิดของตัวอักษรว่าควรจัดอยู่ในกลุ่มใดจากทั้งหมด 26 กลุ่ม (A-Z)

3.1.1 การรับภาพ (Input image)

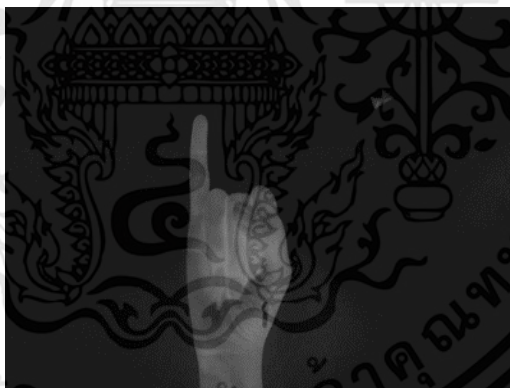
การทดลองนี้ใช้กล้อง Logitech HD webcam C310 เพื่อรับภาพเข้ามาประมวลผลโดยทำการทดลองแบบ offline ด้วยการบันทึกวีดิโอเป็นเวลา 2 วินาที/ท่าทาง โดยมีอัตราความเร็วของภาพ 30 เฟรมต่อวินาที จากนั้นเลือกเฉพาะเฟรมแรกและเฟรมสุดท้ายเพื่อไปเข้าสู่กระบวนการปรับปรุงคุณภาพ

3.1.2 การปรับปรุงคุณภาพของภาพ (Enhancement)



รูปที่ 3.2 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ

3.1.2.1 ภาพระดับสีเทา (Grayscale image) เริ่มต้นด้วยการเปลี่ยนภาพสี RGB ให้อยู่ในรูปของ Grayscale เพื่อให้ได้ค่าระดับขีดแบ่ง (Threshold level) โดยใช้สมการที่ 2.5 ซึ่งเป็นค่าความเข้มปกติที่อยู่ในช่วง 0 ถึง 1 เพื่อนำไปเป็นเกณฑ์ในการเปลี่ยนเป็นภาพไบนารีต่อไป



รูปที่ 3.3 แสดงภาพ Grayscale ของตัวอักษร I

3.1.2.2 ภาพขาว-ดำ (Binary image) การทำภาพให้เป็นไบนารีนั้นเพื่อต้องการให้ได้ขอบของภาพที่สมบูรณ์อีกทั้งยังเป็นการตัดเงาหรือส่วนที่ไม่จำเป็นออกจากวัตถุไปได้ส่วนหนึ่งซึ่งเราจะกำหนดค่า Threshold เป็นค่าคงที่ไว้ค่าหนึ่ง จากนั้นนำค่า gray level ที่ได้จากขั้นตอนที่ก่อนหน้ามาเปรียบเทียบกับ หาก gray level ของพิกเซลนั้นๆน้อยกว่า Threshold ที่ตั้งไว้จะให้มีค่าเป็น '0' และหากมากกว่าจะให้มีค่าเป็น '1'



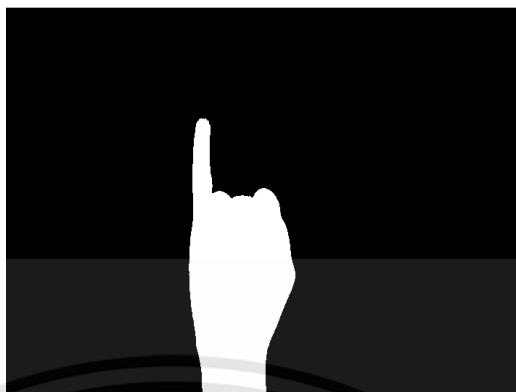
รูปที่ 3.4 แสดงภาพ Binary ของตัวอักษร I

3.1.2.3 การลบพื้นที่ขนาดเล็กกว่าที่กำหนด (Remove small object) โดยการลดส่วนประกอบของภาพที่มีจำนวนพิกเซลน้อยกว่าที่กำหนดไว้ เพื่อเป็นการลบส่วนประกอบที่ไม่ใช่มือทิ้งไปโดยใช้วิธีการกำหนด Threshold ของขนาดวัตถุสีขาวที่พบในภาพไบนารี หากวัตถุที่พบใดๆมีขนาดเล็กกว่า Threshold ดังกล่าวจะถูกลบทิ้ง



รูปที่ 3.5 แสดงภาพการเติมสีขาวของพื้นที่ขนาดเล็กกว่าพิกเซลที่กำหนดของตัวอักษร I

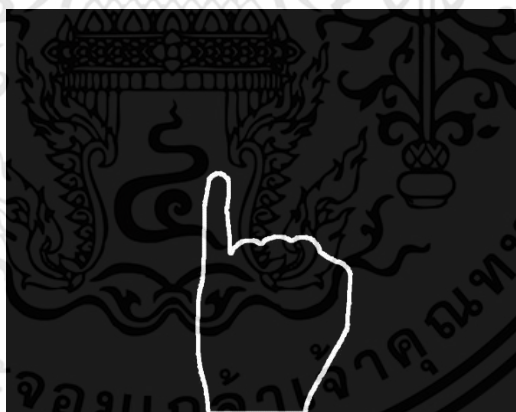
3.1.2.4 การเติมพื้นที่หรือรูของภาพ (Fill image regions and holes) กระบวนการนี้ใช้ในการเติมส่วนของพื้นที่ปิดหรือรูของภาพให้เต็มโดยการเติมส่วนที่เป็นช่องว่างในภาพที่มีพิกเซลสีขาว (มีค่าความเข้มเท่ากับ 1) ล้อมรอบเพื่อให้ได้ภาพมือที่เป็นสีขาวล้วนโดยที่ไม่มีรูอยู่ในภาพ



รูปที่ 3.6 แสดงภาพพื้นที่ที่ถูกเติมพื้นที่สีขาวให้เต็มแล้วของตัวอักษร I

3.1.3 การระบุตำแหน่งของมือ (Identify Hand Position)

3.1.3.1 การหาตำแหน่งของขอบมือ ซึ่งหาได้จากรัศมีสูงสุดของฝ่ามือที่เป็นไปได้โดยอ้างอิงจากขอบคอนทัวร์ (contours) ของรูปมือที่ได้จากวิธีการ Canny ซึ่งเป็นการตรวจจับขอบภาพเป็นวิธีการระบุจุดบนภาพดิจิทัลที่ซึ่งความสว่างเปลี่ยนแปลงอย่างรวดเร็วหรือไม่ต่อเนื่องนั่นเอง จากนั้นจะเก็บค่าที่ได้ไปใช้ประมวลผลต่อไป

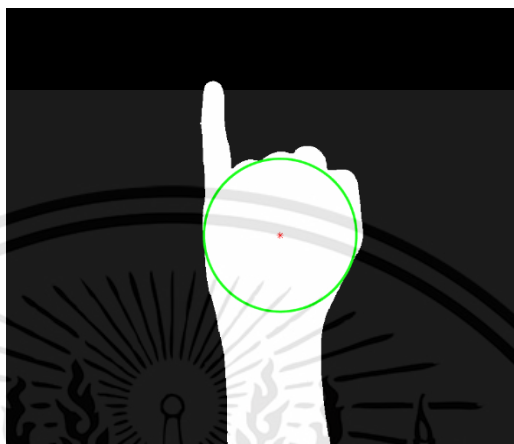


รูปที่ 3.7 แสดงเส้นคอนทัวร์ที่หาได้จากวิธี Canny

3.1.3.2 การหาวงกลมบนฝ่ามือ โดยนำค่าคอนทัวร์ที่ได้ไปใช้หารัศมีของวงกลมสูงสุดที่สามารถวาดบนฝ่ามือได้โดยใช้สมการ

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

จากนั้นจะอ้างอิงจุดศูนย์กลางวงกลมเป็นจุดเซ็นทรอยด์ เพื่อแก้ปัญหาคำนวณตำแหน่งเซ็นทรอยด์ที่อาจผิดพลาดเมื่อภาพที่รับเข้ามามีข้อผิดพลาดด้วยนั่นเอง

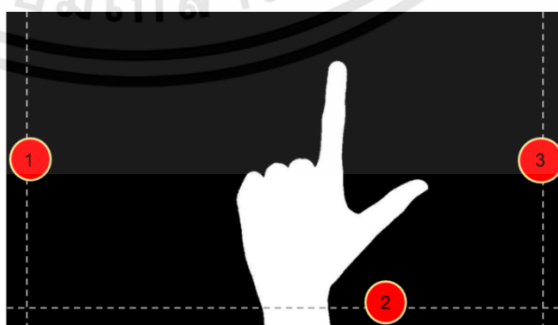


รูปที่ 3.8 แสดงการวาดวงกลมบนฝ่ามือ

3.1.4 การวิเคราะห์คุณลักษณะของภาพ (Feature Extraction)

การวิเคราะห์คุณลักษณะของภาพผู้ทำการทดลองได้ทำการหาเวกเตอร์เพื่อใช้เป็นข้อมูลในการเข้าเรียนรู้ในโครงข่ายประสาทเทียม โดยวิเคราะห์จาก 4 คุณลักษณะได้แก่ พื้นที่ของขอบภาพ 3 ด้าน, ความยาวของนิ้วจากจุดเซ็นทรอยด์, มุมระหว่างนิ้วมือ และค่าเคลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสิ้นสุด

3.1.4.1 พื้นที่ของขอบภาพ 3 ด้าน (NWE) เพื่อใช้วิเคราะห์ว่าภาพมือที่รับเข้ามาเป็นแนวตั้งหรือแนวนอน โดยคำนวณจากจำนวนพิกเซลสีขาวที่พบในแต่ละด้าน



รูปที่ 3.9 แสดงหมายเลขด้านทั้งสาม

- หากพบว่ามีพิกเซลสีขาวจากขอบภาพด้านที่ 1 เข้ามา 5 คอลัมน์ให้พิจารณาว่ารูปเป็นแนวนอนฝั่งซ้ายและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



รูปที่ 3.10 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 1

- หากพบว่ามีพิกเซลสีขาวจากขอบภาพด้านที่ 2 เข้ามา 5 แถวให้พิจารณาว่ารูปเป็นแนวตั้งและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



รูปที่ 3.11 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 2

- หากพบว่ามีพิกเซลสีขาวจากขอบภาพด้านที่ 3 เข้ามา 5 คอลัมน์ให้พิจารณาว่ารูปเป็นแนวนอนฝั่งขวาและให้ด้านที่พบมีค่าเป็น 1 ส่วนด้านอื่นให้มีค่าเป็น 0



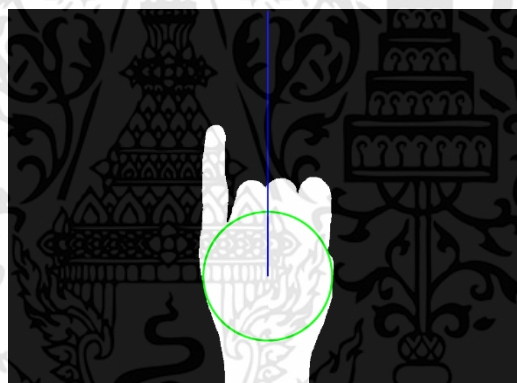
รูปที่ 3.12 แสดงภาพที่พบพิกเซลสีขาวด้านที่ 3

3.1.4.2 ความยาวนิ้วมือจากจุดเซ็นทรอยด์ (Fcen) มีขั้นตอนดังนี้

1. ทำการหาปลายนิ้วมือของรูปที่รับเข้ามาโดยใช้วิธีการหาพิกของคอนทัวร์ของภาพมือ จากนั้นเก็บค่าไว้วิเคราะห์ต่อไป

2. การหาพิกที่ถูกต้องที่สุด เนื่องจากการหาพิกจากปลายนิ้วของรูปภาพบางครั้งจะไม่ได้ ปลายนิ้วที่ครบทั้ง 5 นิ้ว อันเนื่องจากการทำท่าทางของมือที่แตกต่างทำให้บางครั้งนิ้วทุกนิ้วไม่ปรากฏออกมาให้เห็น ดังนั้นเราจึงต้องทำการลดหรือเพิ่มพิกที่หาได้ โดยเริ่มจาก

2.1) แบ่งฝ่ามือออกเป็นสองส่วน โดยใช้จุดเซ็นทรอยด์แบ่งเป็นด้านซ้ายและด้านขวาเพื่อ วิเคราะห์ว่านิ้วมือแต่ละนิ้วควรอยู่ฝั่งไหนเพื่อวิเคราะห์ต่อไปว่าควรจะมีหรือลดพิกที่พบ โดยเราพบว่าโดยปกติและฝั่งซ้ายจะพบนิ้วมือ 2 นิ้ว คือนิ้วก้อยและนิ้วนาง ส่วนด้านขวากจะพบนิ้วมือ 3 นิ้ว คือนิ้วโป้ง, นิ้วชี้ และนิ้วกลาง โดยเราจะใช้เส้นสีน้ำเงินแทนเส้นที่ใช้แบ่งฝ่ามือ

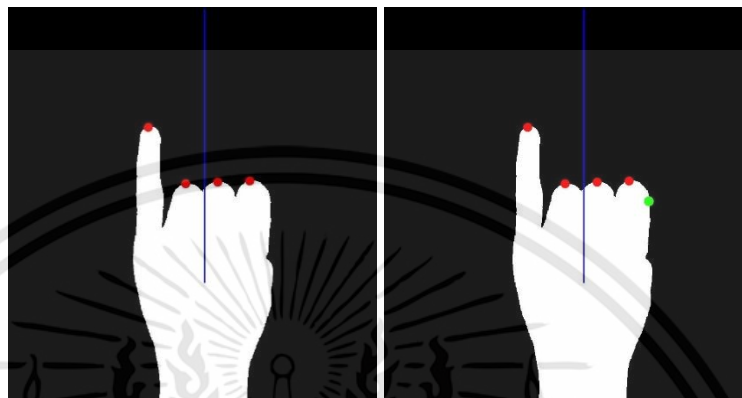


รูปที่ 3.13 แสดงการแบ่งฝ่ามือออกเป็น 2 ส่วน

2.2) การเติมพิกให้แก่ภาพที่จับพิกได้ไม่ครบ 5 นิ้ว โดยมีหลักการว่า

- หากพิกของฝั่งซ้ายไม่ครบ 2 พิกให้ทำการเติมนิ้วเข้าไปให้ครบ โดยจะเติมเริ่มจากเส้นเดิมที่มีอยู่แล้วและขยับไปเป็นมุมเท่ากับค่ามุมเฉลี่ยที่หาได้จากนิ้วมือที่พบพิกโดยจะเติมขยับไปทางฝั่งซ้ายมือ
- เช่นเดียวกันกับฝั่งขวามือหากพบว่าพิกที่ได้ไม่ครบ 3 พิกก็ให้ทำการเติมเพิ่มเข้าไปตามค่ามุมเฉลี่ยที่หาได้จากพิกที่พบก่อนหน้านี้
- กรณีไม่พบพิกใดๆเลยบนฝ่ามือให้ทำการเติมมุมจากเส้นกลางไปทางฝั่งซ้าย 2 เส้น และไปทางฝั่งขวา 2 เส้น โดยอ้างอิงเส้นเริ่มต้นจากเส้น

แบ่งฝ่ามือและขยับไปเป็นมุม 25° (แม้ว่ามุมระหว่างนิ้วปกติจะเป็น 15° หากแต่เราพบจากการทดลองว่าตัวอักษรที่จะไม่พบพิกเฉยคือตัว O จึงต้องการทำให้มุมที่ได้แตกต่างไปจากท่าทางตัวอักษรอื่นๆ)



รูปที่ 3.14 (ซ้าย)แสดงพิกที่จับได้ไม่ครบ (ขวา)แสดงการเติมพิกที่ขาด (จุดสีเขียว)

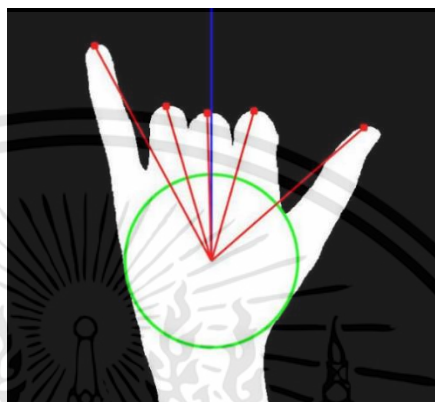
2.3) การลดพิกให้แก่ภาพที่จับพิกได้เกิน 5 นิ้ว โดยมีหลักการว่า

- หากพิกของฝั่งซ้ายเกิน 2 พิก ให้ทำการลบพิกที่เกินโดยอ้างอิงจากพิกที่ทำให้มุมระหว่างพิกมีค่าแตกต่างไปจากค่ามุมเฉลี่ยของพิกที่พบมากที่สุด
- เช่นเดียวกันกับฝั่งขวามือหากพบพิกเกินให้ทำการลบพิกที่เกินโดยอ้างอิงจากพิกที่ทำให้มุมระหว่างพิกมีค่าแตกต่างไปจากค่ามุมเฉลี่ยของพิกที่พบมากที่สุด



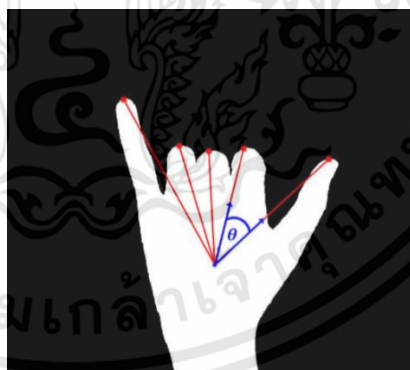
รูปที่ 3.15 (ซ้าย)แสดงพิกที่จับได้เกิน 5 พิก (ขวา)แสดงการตัดพิกที่เกินทิ้ง

2.4) กรณีพบพีคครบ 5 พีคแล้วให้คงค่าไว้เพื่อหาความยาวของนิ้วมือจากจุดเซ็นทรอยด์ โดยการหาระยะทางแบบ Euclidean distance จากนั้นทำการนอร์มัลไลซ์ (Normalize) ข้อมูลโดยอ้างอิงจากระยะทางนิ้วมือที่มีค่ามากที่สุด เพื่อแก้ปัญหารับรูปภาพที่มีขนาดต่างกันเข้ามาประมวลผล



รูปที่ 3.16 แสดงเส้นระยะทางจากจุดเซ็นทรอยด์ถึงจุดพีค

3.1.4.3 มุมระหว่างนิ้วมือ (AngF) เมื่อเราได้พีคที่สมบูรณ์ครบทั้ง 5 พีค ซึ่งเปรียบเสมือนปลายนิ้วทั้ง 5 แล้วให้ทำการวัดมุมระหว่างพีคที่พบแล้วเก็บค่าเอาไว้วิเคราะห์ผลในขั้นตอนถัดไป



รูปที่ 3.17 แสดงภาพมุมระหว่างพีคที่พบ

3.1.4.4 ค่าเตลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสิ้นสุด (delAng) เนื่องจากว่าตัวอักษรภาษามือแบบอเมริกันนั้นประกอบไปด้วยท่าทางที่มีการเคลื่อนไหวและท่าทางหยุดนิ่ง ดังนั้นเราจึงใช้เทคนิคนี้เพื่อแยกท่าทางเหล่านั้นออกจากกัน โดยใช้เงื่อนไขว่า เมื่อใดก็ตามที่ท่าทางที่พิจารณาไม่มีการขยับค่าเตลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสิ้นสุดจะต้องมีค่าเป็น

ศูนย์ และในทางตรงกันข้ามหากท่าทางที่พิจารณาเป็นท่าทางเคลื่อนไหวค่าเดลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสุดท้ายจะมีค่ามาก

โดยสรุปจากคุณลักษณะทั้ง 4 ประการข้างต้นเมื่อนำมาวิเคราะห์โดยนำเฟรมแรกและเฟรมสิ้นสุดมาต่อกันจะได้เวกเตอร์ทั้งหมด 25 ค่า/ 1 ท่าทาง ดังตารางที่ 3.1

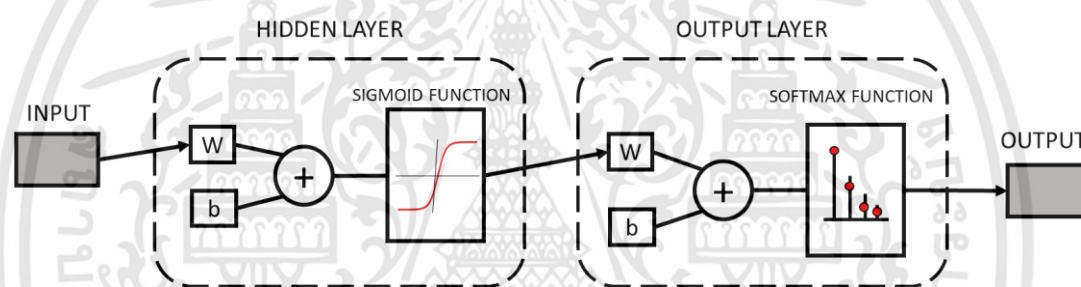
	FEATURES		
	TECHNIQUES	SUB -METHODS	VALUES
1	NUMBER OF WHITE PIXELS IN BINARY IMAGE ALONG THE EDGE	-	3
2	FINGER LENGTH FROM THE CENTROID POINT	- ADDING - DELETING	10
3	ANGLES BETWEEN FINGERS	-	8
4	DIFFERENCES OF ANGLES BETWEEN THE FIRST AND LAST FRAME	-	4
		TOTAL	25

ตารางที่ 3.1 แสดงคุณลักษณะทั้งหมดในการจำแนกตัวอักษรภาษามือภาษาอังกฤษ

1. พื้นที่ของขอบภาพ 3 ด้านเพื่อแยกกลุ่มตัวอักษรออกเป็น 3 กลุ่มคือตัวอักษรในแนวตั้ง, แนวนอนฝั่งซ้ายและแนวนอนฝั่งขวา รวมทั้งหมด 3 ค่า (เลขไบนารี)
2. ความยาวนิ้วมือจากจุดเซ็นทรอยด์ ใช้เพื่อให้สามารถแยกได้ว่ามีนิ้วใดบ้างที่ถูกยกขึ้นและนิ้วใดที่ยังคงกำอยู่เพื่อแยกท่าทางของการกำมือและแบมือออกจากกัน รวมทั้งหมด 10 ค่า
3. มุมระหว่างนิ้วมือของเฟรมแรกและเฟรมสิ้นสุด ใช้เพื่อแยกลักษณะท่าทางมือที่มีลักษณะใกล้เคียงกันมากเช่นตัวอักษร A และ S ออกจากกัน รวมทั้งหมด 8 ค่า
4. ค่าเดลต้าของมุมระหว่างนิ้วมือ ใช้แยกตัวอักษรที่หยุดนิ่งและเคลื่อนไหวออกจากกัน เพื่อให้ทราบว่ามือมีการขยับเปลี่ยนท่าทางรวมทั้งหมด 4 ค่า

3.1.5 การจำแนกตัวอักษรภาษามือ (Classification)

ผู้ทำการทดลองได้เลือกการใช้ระบบโครงข่ายประสาทเทียม (Artificial neural network) จากโปรแกรม MATLAB 2016 เพื่อใช้ในการจำแนกตัวอักษรภาษาอังกฤษทั้ง 26 ตัวออกจากกัน โดยสถาปัตยกรรมของโครงข่ายที่ใช้จะเป็นแบบ Two-layer feed-forward backpropagation โดยใช้ฟังก์ชัน sigmoid ที่ชั้นซ่อนตัว (hidden layer) และ ฟังก์ชัน Softmax ที่ชั้นเอาต์พุต และจะถูกสอนโดยใช้วิธี conjugate gradient backpropagation จากนั้นทำการเลือกเปอร์เซ็นต์ของพารามิเตอร์ Testing, Training และ hidden layer จำนวนที่เหมาะสมเพื่อปรับปรุงโครงข่ายประสาทเทียมให้มีประสิทธิภาพสูงสุด เนื่องจากภาพตัวอักษรภาษาอังกฤษนั้นมีทั้งที่หยุดนิ่งและเคลื่อนไหว เราจึงเลือกใช้การวิเคราะห์คุณลักษณะเวกเตอร์ของเฟรมเริ่มต้นและเฟรมท้ายสุดของภาพ

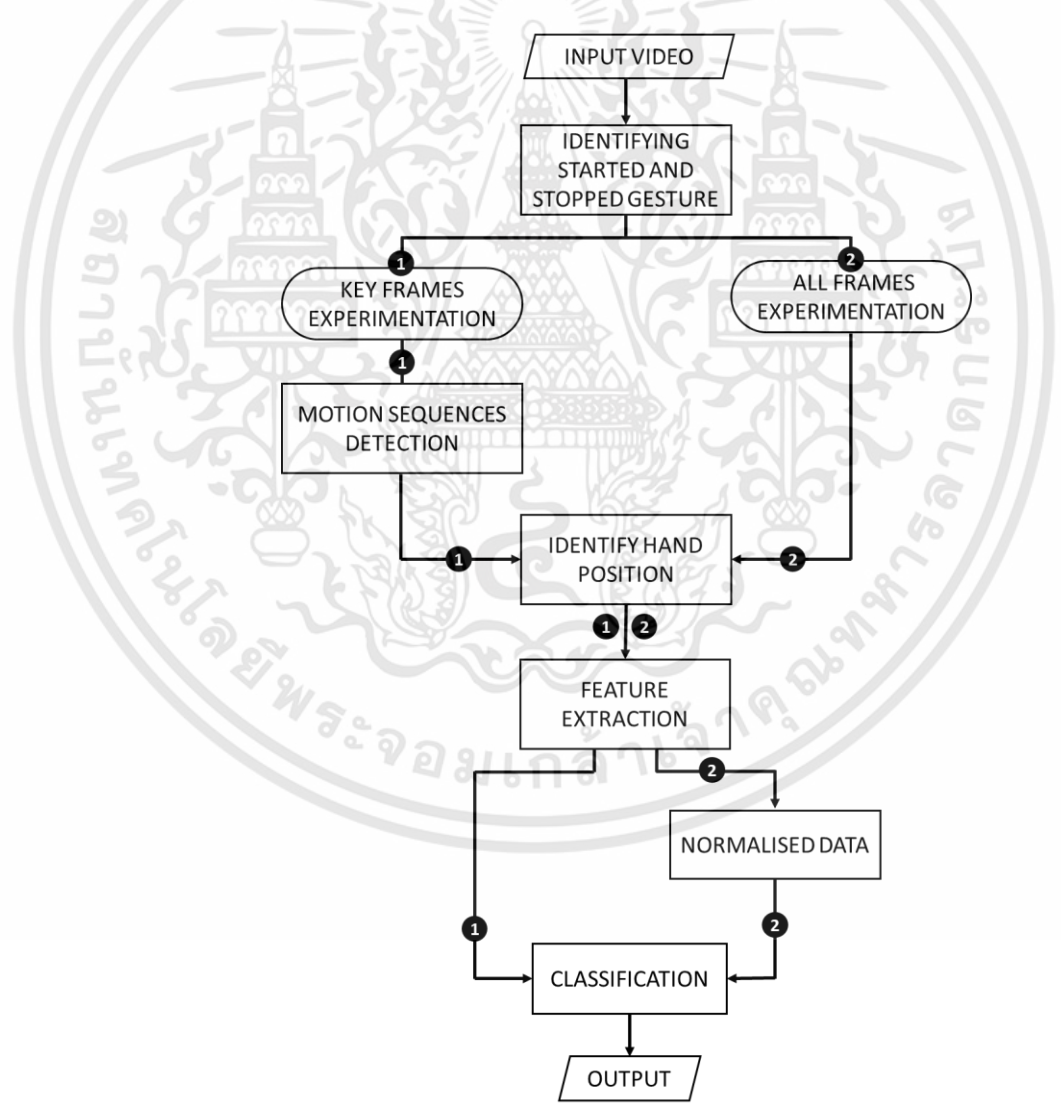


รูปที่ 3.18 โครงข่ายประสาทเทียมที่ใช้ในการจำแนกตัวอักษรภาษาอังกฤษ

วิธีการคือเก็บข้อมูลวิดีโอการทำท่าทางแต่ละครั้งเป็นเวลา 2 วินาทีซึ่งเป็นระยะเวลาเฉลี่ยในการทำท่าทางตัวอักษรภาษามือและใช้เฟรมแรก 30 เฟรม/วินาที รวมทั้งสิ้น 60 เฟรม/ท่าทาง จากนั้นทำการระบุเฟรมในการนำมาใช้หาคุณลักษณะ โดยการทดลองนี้กำหนดให้ใช้เฟรมแรกและเฟรมสุดท้ายของเฟรมทั้งหมด (เฟรมที่ 1 และ 60) เพื่อที่จะแบ่งได้ว่าท่าทางแบบใดหยุดนิ่งและท่าทางใดเคลื่อนไหว จากนั้นจะนำคุณลักษณะที่วิเคราะห์ได้ของเฟรมเริ่มต้นและเฟรมสุดท้ายมาต่อกัน จึงได้ค่าคุณลักษณะที่ใช้เข้าเรียนรู้ในโครงข่ายประสาทเทียมทั้งหมด 25 ค่า เพื่อทำการแยกตัวอักษรภาษาอังกฤษ A-Z ทั้งหมด 26 ตัวอักษร

3.2 ระบบรู้จำคำภาษาอังกฤษ

หัวข้อนี้จะอธิบายถึงขั้นตอนการทำงานของระบบรู้จำคำภาษาเมื่อภาษาอังกฤษ โดยเริ่มจากการรับข้อมูลภาพท่าทางด้วยกล้อง Webcam จากนั้นส่งสัญญาณภาพที่ได้ไปวิเคราะห์หาเฟรมเริ่มต้นและเฟรมสิ้นสุดของท่าทางก่อนแล้วจึงแบ่งการทดลองออกเป็น 2 เทคนิคได้แก่ เทคนิครู้จำคำโดยใช้เฟรมหลัก หรือ key frame (หมายเลข 1) และเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด (หมายเลข 2) ซึ่งเทคนิคการใช้เฟรมหลักนั้น ภาพจะถูกวิเคราะห์โดยใช้กระบวนการคัดเลือกเฟรมหลักที่เหมาะสมก่อนจึงจะสามารถนำเข้าสู่กระบวนการระบุตำแหน่งมือได้ หลังจากนั้นจะผ่านกระบวนการสกัดคุณลักษณะซึ่งเทคนิค ‘ระบบรู้จำคำโดยใช้เฟรมทั้งหมด’ จะต้องถูกนำไปลดมิติของข้อมูลก่อนที่จะนำเข้าสู่โครงข่ายประสาทเทียมเพื่อจำแนกคำแต่ละกลุ่มออกจากกันทั้งหมด 27 กลุ่ม (27 คำในชีวิตประจำวัน)



รูปที่ 3.19 แสดง Block diagram ของระบบรู้จำคำภาษาอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การหาช่วงของท่าทาง (Identifying Started and Stopped Gesture)

การทดลองนี้ใช้กล้อง Logitech HD webcam C310 เพื่อรับภาพเข้ามาประมวลผลโดยผู้ทำการทดลองได้ทำการทดลองแบบ offline โดยทำการบันทึกวิดีโอจากนั้นนำเฟรมที่ได้ไปทำการตรวจจับสีผิวเพื่อหาวัตถุที่เป็นมือของเฟรมนั้นๆ จากนั้นจึงนับจำนวนวัตถุที่พบในภาพโดยใช้วิธีของหัวข้อที่ 2.2.2 เพื่อวิเคราะห์หาเฟรมเริ่มต้นและเฟรมสิ้นสุด หลังจากนั้นจึงมีการแบ่งการทดลองออกเป็น 2 เทคนิคได้แก่ เทคนิครู้จำคำโดยใช้เฟรมหลัก (หมายเลข 1) และเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด (หมายเลข 2)



รูปที่ 3.20 ภาพเฟรมเริ่มต้น (ซ้าย) และเฟรมสิ้นสุด (ขวา) ของท่าทาง 'YES'

3.2.2 การคัดเลือกเฟรมหลัก (Motion Sequences Detection)

เมื่อได้ช่วงของท่าทางมาแล้ว เทคนิค 'รู้จำคำโดยใช้เฟรมหลัก' (หมายเลข 1) จำเป็นต้องผ่านขั้นตอนการเลือกเฟรมหลักเสียก่อน เนื่องจากการถ่ายวิดีโอท่าทางของแต่ละคำศัพท์นั้นไม่สามารถจำกัดเวลาหรือเฟรมเรทในการประมวลผลได้เช่นเดียวกับท่าทางของตัวอักษรเพราะทุกท่าทางมีการเคลื่อนไหวตลอดเวลา ดังนั้นเทคนิครู้จำคำโดยใช้เฟรมหลักจึงเป็นสมมติฐานในการแก้ปัญหานี้โดยคาดว่าจะเป็นตัวช่วยไม่ให้ข้อมูลอินพุตมีมากเกินไป กล่าวคือทำให้ไม่จำเป็นต้องคำนวณคุณลักษณะของทุกเฟรมภาพซึ่งเป็นการสิ้นเปลืองทั้งเวลาในการคำนวณคุณลักษณะ, เวลาการสร้างโมเดล และเวลาในการประมวลผลคำตอบ โดยมีขั้นตอนการคำนวณ 2 ขั้นตอนดังนี้



รูปที่ 3.21 Block diagram ของขั้นตอนการคัดเลือกเฟรมหลัก

1. การแบ่งเฟรมภาพออกเป็นช่วงต่างๆ จากการสังเกตการเปลี่ยนแปลงท่าทางของมือในแต่ละคานั้น ผู้ทำการทดลองพบว่าจะมีการเปลี่ยนแปลงท่าทางไม่เกิน 4 ครั้ง/ท่าทาง จึงเกิดแนวคิดที่จะเก็บข้อมูลหรือเฟรมภาพที่สำคัญของท่าทางในช่วงต่างๆ (Local information) ซึ่งใช้บ่งบอกการเปลี่ยนแปลงของท่าทางตั้งแต่ต้นจนจบท่าทาง จึงได้ทำการแบ่งสัดส่วนช่วงของท่าทางออกเป็น 4 การทดลอง ได้แก่ การแบ่งท่าทางออกเป็น 4, 5, 6 และ 7 ช่วงด้วยกัน เพื่อทดลองว่าขนาดของค่าการแบ่งช่วงใดที่ให้ค่าความถูกต้องในการเรียนรู้เหมาะสมที่สุด

2. วิเคราะห์ผลต่างระหว่างเฟรมปัจจุบันและเฟรมถัดไป ขั้นตอนนี้เพื่อให้ทราบว่ามีการขยับท่าทางซึ่งจะทำให้ค่าผลต่างของค่าพิกเซลของเฟรมปัจจุบันและเฟรมถัดไปมีค่ามากหรือไม่ โดยให้คำนวณผลต่างตั้งแต่เฟรมแรกจนถึงเฟรมสิ้นสุด ก่อนจะเลือกเฟรมที่มีค่าผลต่างระหว่างเฟรมต่ำที่สุดในแต่ละช่วงมาเป็น ‘เฟรมหลัก’ ซึ่งจะบ่งบอกว่าเฟรมภาพ ณ ตำแหน่งนั้นเป็นเฟรมภาพที่มีการขยับน้อยที่สุดหรือหยุดนิ่งมากที่สุด (Entropy ต่ำสุด) ซึ่งง่ายต่อการนำไปคำนวณคุณลักษณะต่อไป รูปที่ 3.22 แสดงเฟรมหลักทั้งหมดของการแบ่งท่าทางออกเป็น 4 ช่วง



รูปที่ 3.22 แสดงเฟรมหลักของการแบ่งท่าทางออกเป็น 4 ช่วง

3.2.3 การระบุตำแหน่งของมือ (Identify Hand Position)



รูปที่ 3.23 แสดง Block diagram ของขั้นตอนการปรับปรุงคุณภาพของภาพ

หลังจากสามารถระบุเฟรมที่จะนำไปใช้หาค่าคุณลักษณะได้แล้วทั้งสองเทคนิคจะต้องผ่านการระบุตำแหน่งมือภายในเฟรมนั้นๆ เพื่อกำจัดบริเวณอื่นๆที่ไม่ใช่มือออกไป โดยมีขั้นตอนตามรูปที่ 3.23

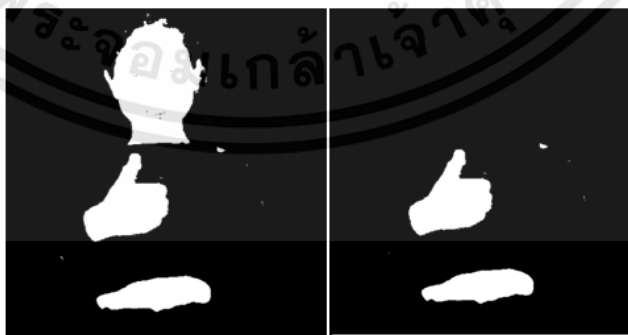
3.2.3.1 การหาตำแหน่งของใบหน้า (Face Position) โดยใช้อัลกอริทึม Viola-Jones ซึ่งอธิบายในหัวข้อที่ 2.3.2 ซึ่งมีความสามารถในการประมวลผลได้รวดเร็วและมีอัตราความถูกต้องในการตรวจหาสูง เพื่อใช้ระบุตำแหน่งของใบหน้าก่อนที่จะนำเข้าสู่กระบวนการถัดไปเพื่อทำการลบใบหน้าออกจากเฟรม ซึ่งจากการทดลองกับฐานข้อมูลของวิทยานิพนธ์เล่มนี้พบว่ามี ความถูกต้องในการตรวจจับหน้าถึงประมาณ 99%

3.2.3.2 การตรวจจับสีผิว (Skin Detection) โดยเริ่มจากการแปลงสีโหมด RGB ของรูปภาพไปเป็น สีโหมด YCbCr เนื่องจากสี YCbCr สีจะไม่เปลี่ยนตามความสว่างทำให้ ซึ่งทำให้ ผลลัพธ์ของสีมีประสิทธิภาพมากกว่าสี RGB อีกทั้งยังสามารถนำไปประยุกต์ใช้กับภาพที่มีความ สว่างไม่สม่ำเสมอได้ดีกว่าเมื่อเทียบกับปริภูมิ HSV จากนั้นทำการไล่หาพิกเซลเพื่อระบุ ตำแหน่งของทุกบริเวณที่พบสีผิว



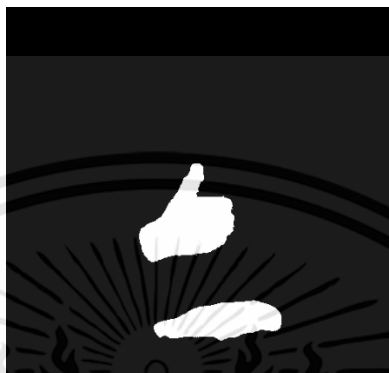
รูปที่ 3.24 ตำแหน่งของพิกเซลที่มีสีผิว

3.2.3.3 กำจัดใบหน้า (Face elimination) เมื่อได้ทุกๆตำแหน่งของบริเวณที่เป็นสีผิว แล้วให้นำไปลบกับตำแหน่งของใบหน้าซึ่งอธิบายในหัวข้อ 3.2.2.1 ผลที่ได้จะเป็นไปดังรูปที่ 3.24



รูปที่ 3.25 ภาพก่อนลบตำแหน่งหน้าออก (ซ้าย) ภาพภายหลังลบตำแหน่งหน้า (ขวา)

3.2.3.4 การกำจัดสัญญาณรบกวน (Noises Removing) หลังจากที่กำลังปรับใบหน้าออกไปแล้ว รูปภาพจะเข้าสู่กระบวนการกำจัดสิ่งรบกวนทิ้งทิ้งที่เป็นจุดเล็กๆ และการเติมเต็มหากวัตถุที่สนใจมีรูโหว่ดังอธิบายในหัวข้อที่ 2.4.2



รูปที่ 3.26 ภายหลังจากกำจัดสิ่งรบกวนแล้ว

3.2.4 การวิเคราะห์คุณลักษณะของท่าทาง (Feature Extraction)

หลังจากทั้งสองเทคนิคผ่านขั้นตอนการหาตำแหน่งมือได้แล้ว (ซึ่งเทคนิครู้จักค่าโดยใช้เฟรมหลักผ่านขั้นตอนวิเคราะห์เลือกเฟรมหลักที่เหมาะสมของแต่ละช่วงของท่าทางในหัวข้อ 3.2.2 แล้ว) ภาพเฟรมจะถูกนำเข้ามาคำนวณคุณลักษณะโดยใช้คำศัพท์ซึ่งแบ่งเป็น 4 หมวดหมู่ดังนี้

หมวดหมู่ของคำศัพท์ในชีวิตประจำวัน** (27 คำ)				
คำศัพท์ทั่วไป (11 คำ)		คำคุณศัพท์ (6 คำ)	อาหารและเครื่องดื่ม (5 คำ)	คำศัพท์เกี่ยวกับสุขภาพ (5 คำ)
ไป (Go)	เพิ่มขึ้น (More)	เหมือน (Same)	ชา (Tea)	ตั้งครรภ์ (Pregnant)
มา (Come)	ต้องการ (Need)	แตกต่าง (Different)	ไข่ (Egg)	กระเพาะ (Stomach)
ใช่ (Yes)	กรุณา (Please)	มีความสุข (Happy)	นม (Milk)	ยา (Medicine)
ไม่ (No)	ด้วย (With)	หิว (Hungry)	ฮอตดอก (Hot dog)	ฉีดยา (Injection)
เสียใจ (Sorry)	ช่วยเหลือ (Help)	หนาว (Cold)	ซอสมะเขือเทศ (Ketchup)	รักษา (Heal)
รัก (Love)		แข็งแรง (Strong)		

ตารางที่ 3.2 แสดงคำศัพท์ทั้ง 4 หมวดหมู่

โดยพื้นฐานของคำศัพท์ที่ผู้ทำการทดลองเลือกมาใช้เน้นไปที่ท่าทางที่มีมือของผู้กระทำจะไม่พาดผ่านใบหน้าหน้าเนื่องจากข้อจำกัดของการใช้กล้อง 2 มิติในการแยกวัตถุด้านและวัตถุด้านหลัง

โดยทั้งสองเทคนิคจะใช้คุณลักษณะที่เหมือนกันทุกประการซึ่งแบ่งเป็น 2 หมวดหมู่ได้แก่ คุณลักษณะภายในกลุ่ม (Intra-Feature) จำนวน 8 คุณลักษณะเพื่อบ่งบอกเอกลักษณ์ของภาพในเฟรมนั้นๆ และคุณลักษณะระหว่างกลุ่ม (Inter-Frame) จำนวน 3 คุณลักษณะเพื่อใช้เปรียบเทียบความแตกต่างระหว่างเฟรมซึ่งบ่งบอกถึงความเปลี่ยนแปลงของข้อมูลที่เราสนใจ รวมทั้งหมด 11 คุณลักษณะดังตารางที่ 3.3

ชนิดของคุณลักษณะ (11 คุณลักษณะ)	
คุณลักษณะภายในกลุ่ม (Intra-Feature)	คุณลักษณะระหว่างกลุ่ม (Inter-Feature)
1. ความกลมของวัตถุ (Circularity)	1. ระยะห่างเส้นทรวงคี่มีระหว่างเฟรมที่อยู่ติดกัน**
2. ทิศทางของวัตถุ (Orientation)	2. ระยะห่างระหว่างมุมของ Blob ฝั่งซ้ายบนของมีระหว่างเฟรมที่อยู่ติดกัน**
3. อัตราส่วนความยาวระหว่างแกนเอกและแกนโท (the ratio of Major axis and Minor axis)	3. เวกเตอร์เส้นทรวงคี่ของมีระหว่างเฟรมที่อยู่ติดกัน**
4. ระยะห่างระหว่างเส้นทรวงคี่ของหน้าและมือภายในเฟรม	
5. ระยะห่างเส้นทรวงคี่ระหว่างมือซ้ายและมือขวาภายในเฟรม	
6. อัตราส่วนระหว่างความกว้างและความยาวของ Blob ภายในเฟรม	
7. จำนวนวัตถุภายในเฟรม	
8. เวกเตอร์ตำแหน่งระหว่างหน้าและมือภายในเฟรม	

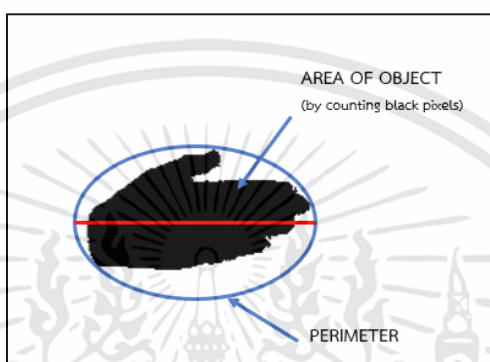
ตารางที่ 3.3 คุณลักษณะทั้งหมดที่ใช้จำแนกคำในภาษามือ

** เฟรมที่อยู่ติดกันกรณี ‘เทคนิครู้จำคำโดยใช้เฟรมหลัก’ จะหมายถึงเฟรมหลักในช่วงถัดไป ไม่ใช่เฟรมที่อยู่ติดกับเฟรมปัจจุบันเหมือนกับ ‘เทคนิครู้จำคำโดยใช้เฟรมทั้งหมด**

3.2.4.1 คุณลักษณะภายในกลุ่ม (Intra-Feature)

1. ความกลมของวัตถุ (Circularity) ใช้คำนวณความกลมของวัตถุของภาพเพื่อใช้แยกว่าท่าทางมีลักษณะการกำมือหรือมีการขึ้นนิ้วหรือไม่ โดยคำนวณได้จากสูตร

$$\text{Circularity} = 4(\text{Area})\pi/(\text{Perimeter})^2 \quad (3.2)$$



รูปที่ 3.27 การตัวแปรที่ใช้หาความกลมของวัตถุ

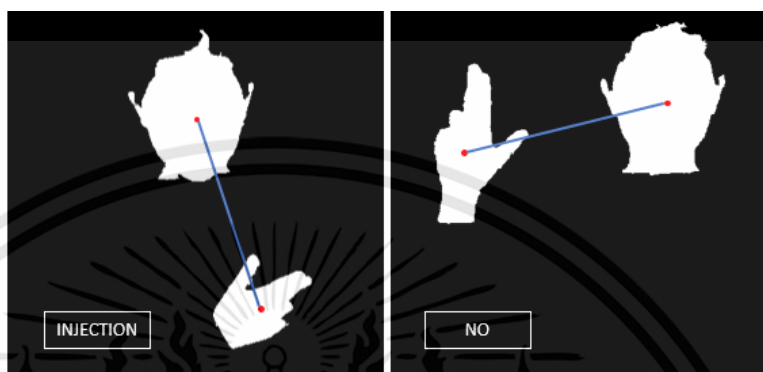
2. ทิศทางของวัตถุ (Orientation) ใช้วิเคราะห์ว่าการเปลี่ยนแปลงแต่ละเฟรมของท่าทางมีลักษณะเอียงไปจากแนวระนาบหรือไม่ โดยคำนวณจากมุมระหว่างแกน X และแกนเอกของวงรีที่สร้างขึ้นมาล้อมรอบวัตถุดังรูปที่ 3.28 เส้นประสีฟ้าแสดงแกนระนาบ จุดสีแดงคือจุดโฟกัสของแกนเอก



รูปที่ 3.28 ภาพจำลองหาความเอียงของวัตถุ

3. อัตราส่วนความยาวระหว่างแกนเอกและแกนโท (the ratio of Major axis and Minor axis) จากรูปที่ 3.28 ให้หาความยาวของแกนหลักและแกนโทของแต่ละลำดับเฟรมออกมาแล้วหาอัตราส่วน เพื่อวิเคราะห์อัตราส่วนความกว้างยาวของวัตถุที่แท้จริง

4. ระยะห่างระหว่างเซ็นเซอร์ของหน้าและมือภายในเฟรม เพื่อแยกท่าทางของมือ บริเวณที่ใกล้ใบหน้าและที่ไกลจากใบหน้า (บริเวณอกหรือหน้าท้อง) จากรูปที่ 3.29 ให้จุดสีแดง แสดงจุดเซ็นเซอร์ของใบหน้าและมือ



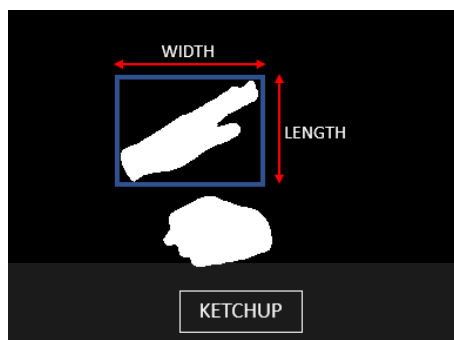
รูปที่ 3.29 เปรียบเทียบตำแหน่งมือและใบหน้าของท่าทางที่ต่างกัน

5. ระยะห่างเซ็นเซอร์ระหว่างมือซ้ายและมือขวาภายในเฟรม ใช้เพื่อวิเคราะห์ความห่างของมือทั้งสองข้างดังรูปที่ 3.30



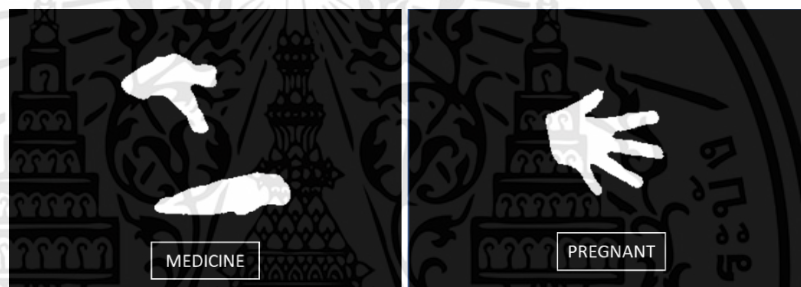
รูปที่ 3.30 เปรียบเทียบตำแหน่งมือทั้งสองข้างภายในเฟรมของท่าทางที่ต่างกัน

6. อัตราส่วนระหว่างความกว้างและความยาวของ Blob ภายในเฟรม เพื่อใช้เป็นค่าวิเคราะห์อัตราส่วนของท่าทางในแนวตั้งฉากของทั้งมือซ้ายและมือขวา ซึ่งแตกต่างจากอัตราส่วนระหว่างแกนเอกและแกนโทซึ่งเป็นอัตราส่วนจริงของวัตถุ



รูปที่ 3.31 อัตราส่วนระหว่างความกว้างและความยาวของ Blob ภายในเฟรม

7. จำนวนวัตถุภายในเฟรม เป็นการวิเคราะห์ว่าท่าทางนั้นๆ ใช้มือกี่ข้างและในแต่ละเฟรมท่าทางนั้นมีการเปลี่ยนแปลงจาก 2 มือไปเป็น 1 มือหรือไม่ดังรูปที่ 3.32



รูปที่ 3.32 จำนวนวัตถุภายในเฟรม

8. เวกเตอร์ตำแหน่งระหว่างหน้าและมือภายในเฟรม เพื่อวิเคราะห์ว่ามีมือที่ใช้นั้นอยู่ทางทิศใดของใบหน้าดังรูปที่ 3.33



รูปที่ 3.33 เวกเตอร์ตำแหน่งระหว่างหน้าและมือภายในเฟรม

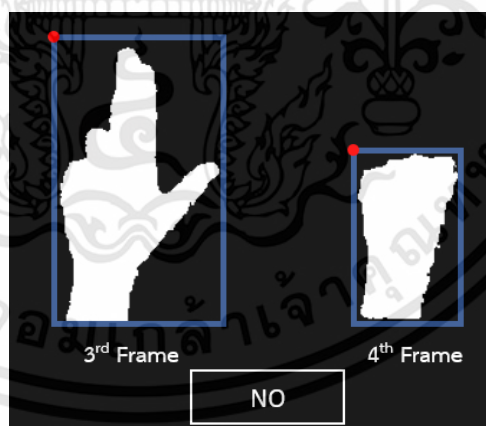
3.2.4.2 คุณลักษณะระหว่างกลุ่ม (Inter-Feature)

1. ระยะห่างเซ็นทรอยด์ระหว่างมือของเฟรมที่อยู่ติดกัน เพื่อใช้ระบุระยะห่างของมือที่เปลี่ยนแปลงไปในท่าทางนั้นๆ โดยจะหาระยะห่างระหว่างเซ็นทรอยด์ของมือซ้ายเฟรมปัจจุบันและมือซ้ายของเฟรมถัดไป เช่นเดียวกันกับมือขวา ดังรูปที่ 3.34



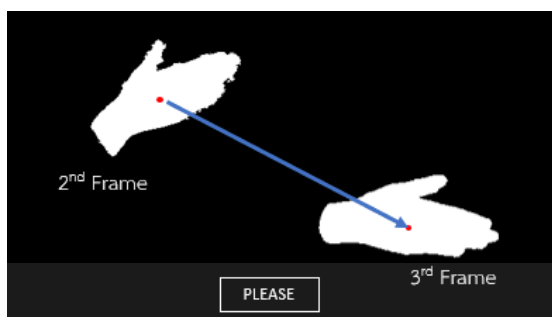
รูปที่ 3.34 เปรียบเทียบตำแหน่งมือทั้งสองข้างของเฟรมที่อยู่ติดกัน

2. ระยะห่างระหว่างมุมบนซ้ายบน Blob ของมือระหว่างเฟรมที่อยู่ติดกัน เนื่องจากการใช้จุดเซ็นทรอยด์จะเกิดปัญหาเมื่อท่าทางมือเปลี่ยนไปจากเดิมซึ่งหมายถึงพื้นที่ของพิกเซลสีขาวของภาพเปลี่ยนแปลงไป เราจึงนำตำแหน่งมุมบนซ้ายของ Blob (จุดสีแดง) มาเพื่อช่วยในการยืนยันตำแหน่งอีกครั้งดังรูปที่ 3.35



รูปที่ 3.35 ตำแหน่ง Blob ของเฟรมที่อยู่ติดกัน

3. เวกเตอร์เซ็นทรอยด์ของมือระหว่างเฟรมที่อยู่ติดกัน เพื่อวิเคราะห์ว่ามีมือในแต่ละเฟรมขยับไปทิศทางใดดังรูปที่ 3.36 แสดงตำแหน่งของเซ็นทรอยด์ที่แตกต่างกันระหว่างเฟรมหลักที่ 2 และเฟรมหลักที่ 3 ของท่าทาง 'Please'



รูปที่ 3.36 เวกเตอร์ตำแหน่งของมือระหว่างเฟรม

โดยหลักการในการเก็บค่าก็คือหาวัตถุที่พบในเฟรมมีจำนวน 2 ชั้นตั้งแต่เฟรมเริ่มต้นจนถึงเฟรมสิ้นสุดเราจะเก็บค่าคุณลักษณะของมือซ้ายกับมือขวาแยกกันโดยใช้แนวคิดที่ว่า มือซ้ายคือตำแหน่งวัตถุที่มีค่าตำแหน่ง X น้อยที่สุดเมื่อเปรียบเทียบกับวัตถุอีกชิ้น ไม่ว่าจะมือของท่าทางจริงจะสลับกัน (มือซ้ายจริงสลับไปอยู่ทางขวา) เราจะยังใช้แนวคิดนี้ในการระบุข้างของมือในการคำนวณเสมอ แต่เมื่อใดก็ตามที่จำนวนวัตถุที่เฟรมเริ่มต้นเปลี่ยนจากจำนวน 2 ชั้นเป็น 1 ชั้น เราจะถือว่ามือทั้งสองขยับมาอยู่ติดกันและจะกำหนดให้ค่าคุณลักษณะของมือซ้ายและมือขวาเป็นค่าเดียวกันตลอดเส้นทางการเปลี่ยนแปลงท่าทางมือในคำศัพท์นั้นๆ จนกระทั่งพบว่ามีมือทั้งสองแยกออกจากกันกลับมาเป็น 2 ชั้นอีกครั้งจึงกลับมาเก็บค่าคุณลักษณะโดยใช้แนวคิดข้างต้นดังเดิม ในขณะเดียวกันนั้นเมื่อใดก็ตามที่วัตถุมีเพียงชิ้นเดียวตั้งแต่เฟรมเริ่มต้นจนกระทั่งเฟรมสุดท้าย เราจะถือให้วัตถุนั้นเป็นมือขวาและให้ค่าคุณลักษณะของมือซ้ายเป็นศูนย์ทั้งหมด อ้างอิงจากการทำท่าภาษามือแบบมือเดียวซึ่งใช้มือขวาเสมอ

3.2.5 การปรับขนาดของข้อมูล (Normalized data)

ภายหลังจากนำเฟรมภาพไปคำนวณหาค่าคุณลักษณะแล้ว เทคนิคที่จำต้องใช้เฟรมภาพทั้งหมดจะต้องผ่านขั้นตอนการปรับขนาดของข้อมูลของค่าคุณลักษณะให้มีขนาดเท่ากันก่อน เนื่องจากการทดลองนี้ไม่มีการกำหนดเฟรมหลักของแต่ละท่าทาง ดังนั้นขนาดของข้อมูลจึงไม่เท่ากันแม้จะเป็นค่าเดียวกันก็ตาม วิธีการคือนำค่าคุณลักษณะแต่ละคุณลักษณะที่ผ่านการคำนวณมาแล้วมาทำการปรับขนาดให้มีขนาดเท่ากับ 75 ซึ่งเป็นค่าเฉลี่ยของจำนวนเฟรมทั้ง 27 ค่า โดยค่าคุณลักษณะใดที่มีขนาดมากกว่า 75 ให้ทำการ Down sampling ในทางตรงกันข้ามหากขนาดของข้อมูลคุณลักษณะน้อยกว่า 75 ก็ให้ทำการ Up sampling นั่นเอง

3.2.6 การจำแนกคำศัพท์ภาษามือ (Classification)

หัวข้อนี้จะกล่าวถึงกระบวนการในการจำแนกกลุ่มของท่าทางคำศัพท์ซึ่งผ่านกระบวนการสกัดคุณลักษณะแล้ว โดยหัวข้อนี้จะแบ่งเป็น 2 เทคนิคเช่นเดิมคือ เทคนิครู้จำคำโดยใช้เฟรมทั้งหมด และเทคนิครู้จำคำโดยใช้เฟรมหลัก

3.2.6.1 เทคนิครู้จำคำโดยใช้เฟรมทั้งหมด เนื่องจากเทคนิคนี้เป็นการนำข้อมูลเฟรมทั้งหมดไปผ่านการสกัดคุณลักษณะซึ่งจะทำให้ข้อมูลขาเข้ามีจำนวนมากและมีความซับซ้อนเกินกว่าที่จะนำไปเรียนรู้ในโครงข่ายประสาทเทียม ดังนั้นจึงต้องผ่านกระบวนการลดมิติของข้อมูลเสียก่อนโดยใช้วิธีวิเคราะห์องค์ประกอบหลัก (PCA) ซึ่งอธิบายไว้ในหัวข้อที่ 2.6 วิธีการคือเมื่อได้ข้อมูลของทุกเฟรมซึ่งผ่านการสกัดคุณลักษณะ 11 คุณลักษณะและทำการปรับขนาดของข้อมูลตามหัวข้อที่ 3.2.5 แล้ว ซึ่งจะได้ทั้งหมด 1783 ค่า / ท่าทาง ให้นำค่าเหล่านั้นเราไปลดมิติโดยการใช้ PCA และใช้ค่าความน่าเชื่อถือที่ร้อยละ 95 ซึ่งได้จากการวิเคราะห์ค่าสัมประสิทธิ์ของข้อมูลซึ่งจะได้ข้อมูลขาเข้าเท่ากับ 59 ค่า/ ท่าทาง จากนั้นนำข้อมูลไปเข้าสู่กระบวนการเรียนรู้ด้วยโครงข่ายประสาทเทียมเพื่อทำการจำแนกประเภทของคำออกเป็น 27 กลุ่ม โดยมีค่าใช้วิธีเช่นเดียวกับการจำแนกตัวอักษรภาษามือซึ่งอธิบายในหัวข้อที่ 3.1.5

3.2.6.2 ระบบรู้จำคำโดยใช้เฟรมหลัก เนื่องจากการทดลองนี้ถูกออกแบบมาเพื่อแก้ไขปัญหาข้อมูลขาเข้าที่มากเกินไปโดยทำการเลือกเฟรมหลักที่สำคัญในแต่ละช่วงของท่าทางไว้แล้วตั้งหัวข้อที่ 3.2.2 ดังนั้นหลังจากผ่านการคุณลักษณะทั้งหมด 11 คุณลักษณะซึ่งจะทำให้ได้ค่าข้อมูลขาเข้าทั้งหมด 24 ค่า/1 เฟรมหลัก ดังนั้นการเลือกจำนวนเฟรมหลักที่แตกต่างกันจึงทำให้ได้ค่าข้อมูลขาเข้าที่แตกต่างกันด้วย จากนั้นนำข้อมูลขาเข้าไปเรียนรู้ในโครงข่ายประสาทเทียมดังอธิบายในหัวข้อ 3.1.5 เพื่อทำการจำแนกประเภทของคำออกเป็น 27 กลุ่ม

บทที่ 4

ผลการทดลอง

ผลการทดลองในบทนี้จะแบ่งออกเป็น 2 ผลการทดลองได้แก่ ผลการทดลองระบบรู้จำตัวอักษร และผลการทดลองระบบรู้จำคำในชีวิตประจำวัน

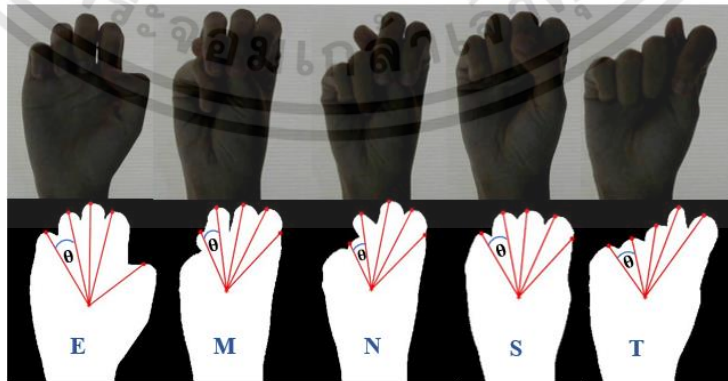
4.1 ระบบรู้จำตัวอักษรภาษาอังกฤษ

ผลการทดลองที่ได้จากการวิเคราะห์แยกคุณลักษณะของภาพตัวอักษรภาษาอังกฤษ (A-Z) ของ การทดลองระบบรู้จำตัวอักษรภาษาอังกฤษ โดยผู้ทำการทดลองได้นำของข้อมูลเข้าเรียนรู้ใน โครงข่ายระบบประสาทเทียมและได้มีการปรับค่าพารามิเตอร์ต่างๆ โดยผลการทดลองกระบวนการมีดังนี้

- ผลการทดลองจากการเติมและลดนิ้วมือ
- ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม
- ผลทดลองเปรียบเทียบกับงานวิจัยอื่นที่เกี่ยวข้อง

4.1.1 ผลการทดลองจากการเพิ่มและลดนิ้วมือ

เนื่องจากตัวอักษรบางตัวไม่สามารถมองเห็นนิ้วมือที่ชัดเจนได้ซึ่งทำให้ยากต่อการวิเคราะห์ ลักษณะของท่าทางมือ ผู้ทำการทดลองจึงได้มีการเพิ่มและลดนิ้วมือซึ่งอธิบายในหัวข้อที่ 3.1.4 เรื่องการ หาพิกที่ถูกต้องที่สุด เพื่อให้ได้นิ้วมือที่ถูกต้องครบทั้งห้านิ้ว โดยผู้ทำการทดลองพบว่าตัวอักษรที่มีปัญหา ทำให้ผลการทดลองไม่เป็นไปตามที่กำหนดคือตัวอักษรที่ไม่สามารถเห็นนิ้วมือได้อย่างชัดเจนทำให้การเติม และลดพิกมีความผิดพลาดบ้าง เช่นตัวอักษร E M N S และ T เนื่องจากเป็นตัวอักษรที่มีลักษณะที่ คล้ายกันคืออยู่ในท่ากำมือเช่นเดียวกันดังรูปที่ 4.1



รูปที่ 4.1 ผลจากการเพิ่มและลดพิกเทียบของมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม

พบว่า การนำข้อมูลที่ได้จากการผ่านคุณลักษณะต่างๆ ทั้ง 4 คุณลักษณะอันได้แก่ พื้นที่ของขอบภาพ 3 ด้าน, ความยาวของนิ้วจากจุดเซ็นทรอยด์, มุมระหว่างนิ้วมือ และค่าเดลต้าของมุมระหว่างนิ้วมือระหว่างเฟรมแรกและเฟรมสุดท้าย จะให้ผลลัพธ์ที่แตกต่างกันไปเมื่อมีการปรับค่าพารามิเตอร์ Testing, Validation และ Hidden layer

4.1.2.1 การเลือกใช้ค่า Hidden layer

เมื่อกำหนดให้ค่า Train และ Test เท่ากันและกำหนด Hidden layer เป็น 100 และ 10 ตามลำดับ (ค่าระหว่างนี้จะให้ผลลัพธ์ที่ไม่แตกต่างกันมากนัก) ผลจากการทดลอง 3 ครั้งเป็นไปตามตารางที่ 4.1 โดยค่าเฉลี่ยความถูกต้องของ Hidden layer = 100 และ Hidden layer = 10 มีค่าเป็น 89.97% และ 72.21% ตามลำดับ จะเห็นได้ว่าค่าเปอร์เซ็นต์ความถูกต้องของ

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

ตารางที่ 4.1 แสดงค่าเปอร์เซ็นต์ความถูกต้องของการใช้ Hidden layer ที่แตกต่างกัน

การใช้ Hidden layer = 100 ของการทดลองทั้ง 3 ครั้งจะมากกว่าการใช้ Hidden layer = 10 ดังนั้นผู้ทำการทดลองจึงเลือกใช้ Hidden layer = 100 ในการเข้าเรียนรู้ในโครงข่ายประสาทเทียม

4.1.2.2 การเลือกใช้ค่า Train : Test

เนื่องจากการทดลองนำข้อมูลเข้าไปเรียนรู้ในโครงข่ายประสาทเทียมในแต่ละครั้งนั้นจะให้ค่าเปอร์เซ็นต์ความถูกต้องที่แตกต่างกัน ผู้ทำการทดลองจึงจับคู่ใช้ค่า Test : Train เป็นค่าต่างๆ โดยใช้ Hidden layer = 100 ได้ผลการทดลองดังตารางที่ 4.2 จะเห็นได้ว่าค่าความถูกต้องของการใช้ค่า Test : Train ที่ 30% : 70% จะให้ความถูกต้องสูงที่สุดคือ 90.42% ผู้ทำการทดลองจึงเลือกใช้ค่านี้เพื่อทำการทดลองต่อไป

Test (%)	Train (%)	Accuracy (%)
20	80	73.77
30	70	90.42
40	60	82.93

ตารางที่ 4.2 แสดงการเลือกใช้ค่า Testing และ Training

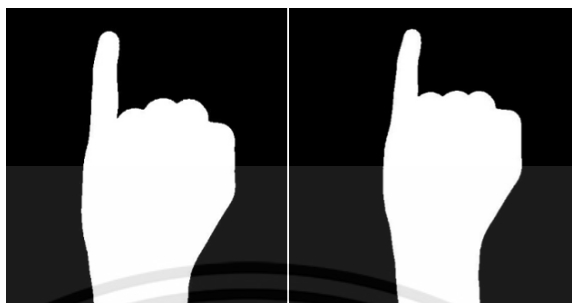
4.1.2.3 ผลค่าความถูกต้องที่ได้จากการทดลอง

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

ตารางที่ 4.3 แสดงเปอร์เซ็นต์ความถูกต้องของตัวอักษร A-Z

จากตารางที่ 4.3 จะเห็นได้ว่าจากการทดลองทั้งสามครั้งได้เปอร์เซ็นต์ความถูกต้องใกล้เคียงกันคือประมาณ 95% ตัวอักษรที่มีเปอร์เซ็นต์ความถูกต้องน้อย ได้แก่ J, M, N และ P ซึ่งมีเปอร์เซ็นต์ความถูกต้อง 90%, 69%, 82.3% และ 79% ตามลำดับ โดยตัวอักษร J ตอบผิดเป็น I , ตัวอักษร M และ N จะตอบผิดสลับกันและตัวอักษร P ตอบผิดเป็น Q จะเห็นได้ว่าเนื่องจากตัวอักษรดังกล่าวนี้มีความคล้ายคลึงกันมากตัวอย่างเช่น

- ตัวอักษร J และ I ซึ่งมีเฟรมแรกของภาพเหมือนกันทุกประการ



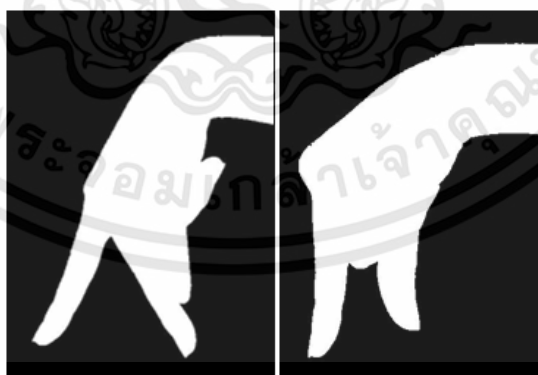
รูปที่ 4.2 แสดงภาพเฟรมแรกของตัว I (ซ้าย) และ J (ขวา)

- ตัวอักษร M และ N ซึ่งมีลักษณะคล้ายกันทั้งเฟรมแรกและเฟรมสุดท้าย



รูปที่ 4.3 แสดงภาพเฟรมแรกของตัวอักษร M (ซ้าย) และ N (ขวา)

- ตัวอักษร P และ Q ซึ่งมีลักษณะคล้ายกันทั้งเฟรมแรกและเฟรมสุดท้าย



รูปที่ 4.4 แสดงภาพเฟรมแรกของตัวอักษร P (ซ้าย) และ Q (ขวา)

4.1.3 ผลทดลองเปรียบเทียบกับการวิจัยอื่นที่เกี่ยวข้อง

ในการทดลองนี้เราทำการทดสอบโดยการให้ผู้ถูกทดสอบแสดงเฉพาะส่วนของมือบนหน้ากล้อง WEBCAM และมีพื้นหลังเป็นสีดำ ภายใต้ความสว่างแสงไฟปกติ ผลการทดลองเปรียบเทียบกับการวิจัยอื่นๆที่ใช้อัลกอริทึมที่แตกต่างกันในสภาพแวดล้อมและฐานข้อมูลเดียวกันเป็นดังตารางที่ 4.4 โดยที่ผลการทดลองของวิทยานิพนธ์เล่มนี้มีค่าความถูกต้องสูงถึง 95% ซึ่งถือว่ามีความสูงมากเมื่อเปรียบเทียบกับงานวิจัยอื่นๆที่ใช้อัลกอริทึมที่แตกต่างกันดังอ้างอิงในตาราง จะเห็นได้ว่าเปอร์เซ็นต์ความถูกต้องของงานวิจัยที่อ้างอิงสามารถจดจำตัวอักษรได้ทุกตัว (A-Z) นั้นมีความถูกต้องเพียง 56% ถึง 88.28% ส่วนงานวิจัยที่เหลือนั้นสามารถจดจำตัวอักษรได้เพียงเฉพาะบางตัวเท่านั้นอีกทั้งตัวอักษรส่วนใหญ่ก็มีลักษณะท่าทางางมือซึ่งถือเป็นกลุ่มตัวอักษรที่สามารถจดจำได้ง่าย

REFERENCE	FEATURE EXTRACTION	CLASSIFICATION	OUTPUT LETTERS	ACCURACY (%)
Proposed techniques	NWE, Fcen, AngF and delAng	ANN	A-Z	95
[17]	Fingertips detection	NA	B, D, F, I, K, L, U, V, W	70
[18]	Canny edge detection	PCA, ANN	A-E	55
[19]	Centroid of hand	Euclidean distance, Feature matching	A-Z	88.2
[20]	Area, Average height, Centroid and Distance from origin to centroid	ANN	A-Z	56

ตารางที่ 4.4 เปรียบเทียบผลของการใช้อัลกอริทึมที่แตกต่างกัน

ตัวอักษรที่มีอัตราการจดจำต่ำที่สุดคือ M และ N เนื่องจากทั้งสองตัวอักษรมีท่าทางที่คล้ายกันมากและบางงานวิจัยที่อ้างอิงในวิทยานิพนธ์เล่มนี้ก็พบปัญหาเดียวกันในการจดจำตัวอักษรทั้งสองตัว

อย่างไรก็ตามเปอร์เซ็นต์ความถูกต้องของตัวอักษรทั้งสองจากการใช้อัลกอริทึมของงานวิจัยนี้ยังอยู่ในเกณฑ์ที่ยอมรับได้

4.2 ระบบรู้จำคำในชีวิตประจำวัน

ผลการทดลองที่ได้จากการวิเคราะห์แยกคุณลักษณะของวิดีโอคำในชีวิตประจำวันภาษามือ 27 คำ ของการทดลองระบบรู้จำคำในชีวิตประจำวันภาษาอังกฤษได้แบ่งการทดลองออกเป็น 2 เทคนิคได้แก่ ผลการทดลองเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด และผลการทดลองเทคนิครู้จำคำโดยใช้เฟรมหลัก

4.2.1 ผลการทดลองเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด

ผลการทดลองหัวข้อนี้ทำโดยการนำอินพุตวิดีโอภาพไปคำนวณคุณลักษณะทั้งหมด 11 คุณลักษณะได้แก่ คุณลักษณะภายในในกลุ่ม (Intra-Feature) 8 คุณลักษณะ คือ ความกลมของวัตถุ, ทิศทางของวัตถุ, อัตราส่วนความยาวระหว่างแกนเอกและแกนโท, ระยะห่างระหว่างเซ็นทรอยด์ของหน้าและมือภายในเฟรม, ระยะห่างเซ็นทรอยด์ระหว่างมือซ้ายและมือขวาภายในเฟรม, อัตราส่วนระหว่างความกว้างและความยาวของ Blob ภายในเฟรม, จำนวนวัตถุภายในเฟรม และเวกเตอร์ตำแหน่งระหว่างหน้าและมือภายในเฟรม รวมทั้งคุณลักษณะระหว่างกลุ่ม (Inter-Feature) 3 คุณลักษณะคือ ระยะห่างเซ็นทรอยด์มือระหว่างเฟรมที่อยู่ติดกัน, ระยะห่างระหว่างมุมของ Blob ฝั่งซ้ายบนของมือระหว่างเฟรมที่อยู่ติดกัน และ เวกเตอร์เซ็นทรอยด์ของมือระหว่างเฟรมที่อยู่ติดกัน จากนั้นนำค่าแต่ละคุณลักษณะไปทำการนอร์มัลไลซ์ให้เหลือ 75 ค่า/ท่าทาง ซึ่งเป็นค่าเฉลี่ยของจำนวนเฟรมทั้ง 27 คำ ซึ่งจะได้ค่าคุณลักษณะรวมทั้งหมด 1783 ค่า/1ท่าทาง

จากนั้นเข้าไปผ่านกระบวนการ Principal Components Analysis (PCA) โดยใช้ค่าความเชื่อมั่นที่ 95% (จะได้ข้อมูลที่มีขนาดเท่ากับ ขนาดของข้อมูลทดลอง X59 เวกเตอร์) จากนั้นจึงนำของข้อมูลเข้าเรียนรู้ในโครงข่ายระบบประสาทเทียมและได้มีการปรับค่าพารามิเตอร์เป็น Hidden layer = 100, Test = 30% , Train= 70% ผลที่ได้เป็นไปดังตารางที่ 4.5 และ 4.6 ซึ่งมีค่าความถูกต้องรวมอยู่ที่ประมาณ 83.15% โดยมีค่าที่มีความถูกต้องต่ำกว่า 70% อยู่ 8 คำ และค่าที่ถูกต้อง 100% อยู่ 10 คำ

WORDS	ACCURACY(%)	WORDS	ACCURACY(%)
GO	100	KETCHUP	90
YES	90	COLD	100
NO	100	DIFFERENT	95
SORRY	100	HAPPY	65
LOVE	95	STRONG	100
MORE	100	HUNGRY	35
NEED	100	PREGNANT	65
PLEASE	60	STOMACH	70
WITH	90	MEDICINE	65
HELP	100	INJECT	45
TEA	95	HEAL	55
EGG	100	COME	100
MILK	50	SAME	90
HOTDOG	90	AVERAGE	83.15

ตารางที่ 4.5 ผลการทดลองคำนวณคุณลักษณะของท่าทางโดยการกำหนดช่วงของเฟรมโดยอ้างอิงจากค่าเฉลี่ยของเฟรมท่าทางทั้งหมด

PCA	Go	Yes	No	Sorry	Love	More	Need	Please	With	Help	Tea	Egg	Milk	Hotdog	Ketchup	Cold	Different	Happy	Strong	Hungry	Pregnant	Stomach	Medicine	Inject	Heal	Come	Same
Go	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Yes	0	90	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sorry	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Love	0	0	0	0	95	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
More	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Need	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Please	0	0	15	25	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
With	0	0	0	0	5	5	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Help	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tea	0	0	0	0	0	0	0	0	0	0	95	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
Egg	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Milk	0	0	45	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	5	0	0	0
Hotdog	0	0	0	0	0	5	0	0	0	0	0	0	5	90	0	0	0	0	0	0	0	0	0	0	0	0	0
Ketchup	0	0	0	0	0	0	0	0	0	10	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0
Cold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Different	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0
Happy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	0	0	25	0	0	0	0	10	0
Strong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Hungry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	35	30	0	0	0	0	0	0
Pregnant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0	65	0	0	0	0	0	0
Stomach	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	25	0	0	0	0	70	0	0	0	0	0
Medicine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	65	0	0	10	0
Inject	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	30	0	0	0	0	45	0	0	0
Heal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	40	0	0	0	55	0	0
Come	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Same	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	90

ตารางที่ 4.6 ตาราง Confusion Matrix ของระบบรู้จำตัวอักษรภาษามือภาษาอังกฤษ

4.2.2 ผลการทดลองเทคนิครู้จำคำโดยใช้เฟรมหลัก

ผลการทดลองของหัวข้อนี้แบ่งการทดลองออกเป็น 4 การทดลองโดยการเปลี่ยนขนาดของช่วงทำทางที่สำคัญเป็น 4, 5, 6 และ 7 ช่วงตามลำดับ โดยอ้างอิงจากการเปลี่ยนแปลงทำทางของคำภาษามือที่นำมาใช้ทดลองซึ่งจะเปลี่ยนไม่เกิน 4 ครั้ง/ทำทาง โดยผู้ทำการทดลองได้นำข้อมูลเข้าเรียนรู้ในโครงข่ายระบบประสาทเทียมและได้มีการปรับค่าพารามิเตอร์เป็น Hidden layer = 100, Test = 30% , Train = 70% ผลที่ได้ของแต่ละการทดลองเป็นดังนี้

1. เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 4 ช่วง ได้ผลดังตารางที่ 4.7 จะเห็นได้ว่ามีค่าที่ตอบถูก 100% อยู่เพียง 6 คำ อีกทั้งยังมีค่าที่ตอบถูกต่ำกว่า 70% ถึง 4 คำ
2. เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 5 ช่วง ได้ผลดังตารางที่ 4.8 จะเห็นได้ว่ามีค่าที่ตอบถูก 100% อยู่เพียง 3 คำ และมีค่าที่ตอบผิดต่ำกว่า 70% อยู่ถึง 9 คำ ซึ่งมากกว่าเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 4 ช่วง ถึงเกือบ 2.4 เท่า
3. เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 6 ช่วง ได้ผลดังตารางที่ 4.9 จะเห็นได้ว่ามีค่าที่ตอบถูก 100% ถึง 17 คำ ซึ่งมีความมากที่สุด ในเทคนิคการรู้จำคำโดยใช้เฟรมหลัก และไม่มีคำใดเลยที่ได้เปอร์เซ็นต์ความถูกต้องต่ำกว่าร้อยละ 70
4. เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 7 ช่วง ได้ผลดังตารางที่ 4.10 จะเห็นว่า มีคำตอบที่ตอบถูก 100% ถึง 13 คำเท่ากับเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 6 ช่วง และไม่มีคำใดเลยที่ได้เปอร์เซ็นต์ความถูกต้องต่ำกว่าร้อยละ 70

เมื่อนำค่าเปอร์เซ็นต์ความถูกต้องของผลการทดลองทั้ง 4 มาเปรียบเทียบกับกันได้ผลเป็นไปดังตารางที่ 4.11 จะเห็นได้ว่า เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 6 ช่วง นั้นจะให้ค่าความถูกต้องสูงที่สุดคือ 95.19% ในขณะที่เมื่อเพิ่มช่วงเฟรมหลักเป็น 7 ช่วงนั้นจะได้ค่าความถูกต้องที่ต่ำลงคือ 93.70 % และที่การแบ่งเฟรมหลักออกเป็น 5 ช่วงจะให้ค่าความถูกต้องต่ำที่สุดคือ 75.18% และเมื่อนำผลการทดลองทั้งหมดมาเปรียบเทียบกับกันจะเห็นได้ว่า การแบ่งช่วงเฟรมหลักออกเป็นเลขคี่ (5 และ 7) จะทำให้ได้ผลลัพธ์ที่มีค่าต่ำลง ในขณะที่เมื่อเราแบ่งเฟรมหลักออกเป็นเลขคู่ (4 และ 6) จะให้ผลลัพธ์ของความถูกต้องที่มากขึ้น

4 INTERVALS	Go	Yes	No	Sorry	Love	More	Need	Please	With	Help	Tea	Egg	Milk	Hotdog	Ketchup	Cold	Different	Happy	Strong	Hungry	Pregnant	Stomach	Medicine	Inject	Heal	Come	Same	
Go	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	5
Yes	0	50	0	5	0	0	20	0	0	0	0	0	20	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
No	0	5	50	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sorry	0	0	0	80	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
Love	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	5	0	0	0	0
More	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
Need	0	25	15	0	0	0	40	0	5	0	0	0	10	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
Please	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
With	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
Help	0	0	0	0	0	0	0	0	5	55	5	0	0	0	0	0	0	0	0	5	5	0	20	0	0	0	0	5
Tea	0	0	0	0	0	0	0	0	0	0	85	0	0	0	5	0	0	0	0	0	5	0	5	0	0	0	0	0
Egg	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
Milk	0	0	0	0	0	0	0	0	0	0	0	5	90	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
Hotdog	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ketchup	0	0	0	0	0	5	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0
Cold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Different	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	5	0	0	0	0	0	0	0	0	0	0
Happy	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0
Strong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	90	0	0	0	0	0	0	0	0	0
Hungry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pregnant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Stomach	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	70	0	0	0	0	0	10
Medicine	0	0	0	0	5	0	0	0	0	5	10	0	0	0	0	0	0	0	0	5	5	0	70	0	0	0	0	0
Inject	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	95	0	0	0	0
Heal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Come	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	5	0	0	0	85	0	0
Same	0	0	0	0	0	0	0	0	0	0	5	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	90

ตารางที่ 4.7 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 4 ช่วง

5 INTERVALS	Go	Yes	No	Sorry	Love	More	Need	Please	With	Help	Tea	Egg	Milk	Hotdog	Ketchup	Cold	Different	Happy	Strong	Hungry	Pregnant	Stomach	Medicine	Inject	Heal	Come	Same	
Go	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	5
Yes	0	50	0	15	0	0	20	0	0	0	0	0	5	0	0	0	0	0	0	5	0	5	0	0	0	0	0	0
No	0	5	90	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sorry	0	0	0	95	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Love	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	5	0	15	0	
More	0	0	0	5	0	90	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Need	0	0	15	10	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	10	0	
Please	0	0	0	40	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
With	5	0	0	0	0	0	0	0	50	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
Help	0	0	0	0	0	0	0	0	0	55	15	5	0	0	0	0	0	0	0	5	0	5	15	0	0	0	0	0
Tea	0	0	0	0	0	0	0	0	0	0	90	5	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
Egg	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
Milk	0	0	10	0	0	0	0	0	0	0	0	0	75	0	0	0	0	5	0	5	0	5	0	0	0	0	0	0
Hotdog	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ketchup	5	0	0	0	0	5	0	0	0	0	20	0	0	0	55	0	0	0	0	0	0	0	5	0	10	0	0	0
Cold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	5	0	0	0	0	0	5	0	0	0
Different	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	50	5	0	25	0	0	0	0	0	0	0	15
Happy	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	60	0	0	0	35	0	0	0	0	0	0
Strong	5	0	0	0	0	0	0	5	0	0	0	0	0	0	0	5	0	0	70	0	0	0	0	0	10	0	5	0
Hungry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pregnant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	5	0	0	0	0	0	0
Stomach	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	5	0	0
Medicine	0	0	0	0	10	0	0	0	0	5	10	0	0	0	0	0	0	0	0	5	5	0	65	0	0	0	0	0
Inject	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Heal	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0
Come	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	95	0	0
Same	0	0	0	5	0	0	0	0	0	10	5	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	75

ตารางที่ 4.8 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 5 ช่วง

6 INTERVALS	Go	Yes	No	Sorry	Love	More	Need	Please	With	Help	Tea	Egg	Milk	Hotdog	Ketchup	Cold	Different	Happy	Strong	Hungry	Pregnant	Stomach	Medicine	Inject	Heal	Come	Same	
Go	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
Yes	0	85	5	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sorry	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Love	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
More	0	0	0	0	0	90	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
Need	0	5	5	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
Please	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
With	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Help	0	0	0	0	0	0	0	0	0	75	0	0	0	5	10	5	0	0	0	0	0	0	5	0	0	0	0	0
Tea	0	0	0	0	0	0	0	0	0	0	85	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	5	0
Egg	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Milk	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hotdog	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ketchup	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Cold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Different	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0
Happy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Strong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hungry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pregnant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Stomach	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Medicine	0	0	0	0	10	0	0	0	0	5	10	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0
Inject	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Heal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Come	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	95	0	0
Same	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0

ตารางที่ 4.9 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 6 ช่วง

7 INTERVALS	Go	Yes	No	Sorry	Love	More	Need	Please	With	Help	Tea	Egg	Milk	Hotdog	Ketchup	Cold	Different	Happy	Strong	Hungry	Pregnant	Stomach	Medicine	Inject	Heal	Come	Same	
Go	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
Yes	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No	0	0	95	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sorry	0	5	0	80	0	0	5	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Love	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
More	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	
Need	0	10	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
Please	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
With	0	0	0	0	0	0	0	0	95	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
Help	0	0	0	0	5	0	0	5	0	70	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0
Tea	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Egg	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Milk	0	10	15	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
Hotdog	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ketchup	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0
Cold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Different	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0
Happy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Strong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hungry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Pregnant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Stomach	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	90	0	0	0	0	0	0
Medicine	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0
Inject	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Heal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Come	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
Same	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	5	0	0	0	0	5	0	85	0

ตารางที่ 4.10 ตาราง Confusion Matrix ของลำดับของท่าทางสำคัญถูกแบ่งออกเป็น 7 ช่วง

WORDS	RESULT OF IMPORTANT KEY FRAME (%)			
	4	5	6	7
GO	100	90	90	95
YES	50	50	85	100
NO	50	90	100	95
SORRY	80	95	100	80
LOVE	90	75	100	95
MORE	95	90	90	95
NEED	40	60	85	85
PLEASE	95	50	100	100
WITH	95	50	100	95
HELP	55	55	75	70
TEA	85	90	85	100
EGG	95	95	100	100
MILK	90	75	100	70
HOTDOG	100	100	100	100
KETCHUP	95	55	100	90
COLD	100	90	100	100
DIFFERENT	95	50	95	95
HAPPY	95	60	100	100
STRONG	90	70	100	100
HUNGRY	100	100	100	100
PREGNANT	100	95	100	100
STOMACH	70	95	100	90
MEDICINE	70	65	75	90
INJECT	95	100	100	100
HEAL	100	95	100	100
COME	85	95	95	100
SAME	90	75	95	85
TOTAL	85.37	78.15	95.19	93.70

ตารางที่ 4.11 ผลการทดลองเทคนิครู้จำคำโดยใช้เฟรมหลักที่มีจำนวนช่วงต่างกัน

จากนั้นนำค่าเปอร์เซ็นต์ความถูกต้องข้างต้นและเวลาที่ใช้ในการประมวลผลของแต่ละการทดลองไปทำการวิเคราะห์หาค่า Fitness classifier โดยใช้สมการที่ 4.1 เพื่อใช้วิเคราะห์ประสิทธิภาพของวิธีการทดลองที่แตกต่างกันทั้ง 5 วิธี ซึ่งได้ผลดังตารางที่ 4.12

$$\text{Fitness}_{\text{classifier}} = \left(\alpha \times \left(\frac{\text{Accuracy}}{100} \right) + (1-\alpha) \left(\frac{\text{Time}_{\text{max}} - \text{Time}}{\text{Time}_{\text{max}}} \right) \right) \times 100 \quad (4.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

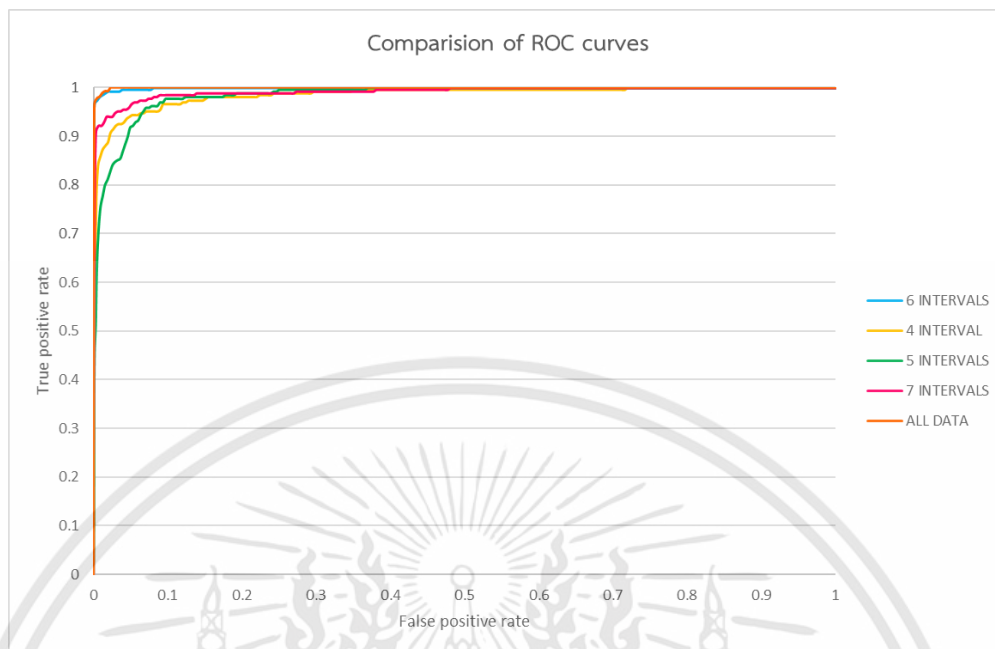
	METHODS				
	IMPORTANT KEY FRAMES				ALL FRAMES
	4	5	6	7	
ACCURACY (%)	85.37	78.15	95.15	93.70	83.15
NUMBER OF TRAINED VECTORS/WORD	96	120	144	168	59
COMPUTATIONAL TIME AVERAGE(SECONDS)	29.596 \pm 0.115	30.177 \pm 0.129	30.369 \pm 0.914	31.256 \pm 0.163	27.161 \pm 0.127
FITNESS CLASSIFIER AVERAGE (%)	66.81	63.16	76.88	75.84	66.56

ตารางที่ 4.12 เปรียบเทียบค่าความถูกต้องและเวลาในการประมวลผลของวิธีทั้งหมดโดยใช้เพียง 27 คำ

จากตารางข้างต้นจะเห็นว่าเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด (All frames) ใช้เวลาในการประมวลผลสั้นที่สุด (ทั้งนี้ขึ้นกับความเร็วของอุปกรณ์ที่ใช้ประมวลผลด้วย) คือประมาณ 27.16 วินาที ในขณะที่วิธีที่ใช้เวลาในการประมวลผลนานที่สุดคือเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 7 ช่วง คือประมาณ 31.25 วินาที แต่เมื่อพิจารณาที่ค่าความถูกต้องจะพบว่า เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 6 ช่วง ให้ค่าความถูกต้องสูงที่สุดคือประมาณ 95.15% และเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 5 ช่วง ให้ค่าความถูกต้องต่ำที่สุดคือประมาณ 78.15% เท่านั้น

จากข้อมูลข้างต้นเมื่อทำการหาค่า Fitness classifier เพื่อเปรียบเทียบประสิทธิภาพของวิธีการต่างๆ โดยกำหนดค่า $\alpha = 0.8$ ได้ผลว่า เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 6 ช่วง ให้ค่า fitness classifier สูงที่สุดคือประมาณ 76.88% ส่วนเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งเป็น 5 ช่วง จะให้ค่า Fitness classifier ต่ำที่สุดคือ 63.16%

และเมื่อนำตาราง Confusion matrix ของการเทคนิครู้จำคำโดยใช้เฟรมทั้งหมดและเทคนิครู้จำคำโดยใช้เฟรมหลักมาทำการพลอตกราฟ ROC (Receiver Operating Characteristic curve) ซึ่งใช้แสดงประสิทธิภาพในการเปรียบเทียบ Binary Classification หรือประสิทธิภาพการแยกกลุ่มซึ่ง จะได้กราฟดังรูปที่ 4.7



รูปที่ 4.5 แสดงการเปรียบเทียบกราฟ ROC ของการทดลองทั้งหมด

จากกราฟจะเห็นว่าเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด (สีส้ม) ให้ค่าพื้นที่ใต้กราฟใกล้เคียงกับเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 6 ช่วง (สีฟ้า) โดยมีค่าประมาณ 0.999 ในขณะที่เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 5 ช่วง (สีเขียว) จะให้ค่าพื้นที่ใต้กราฟต่ำที่สุดคือประมาณ 0.984 นั้นเอง

บทที่ 5

สรุปและวิจารณ์ผลงานวิจัย

วิทยานิพนธ์ฉบับนี้มุ่งเน้นการศึกษาเรื่อง การวิเคราะห์คุณลักษณะของภาพซึ่งได้จากท่าทาง ภาษามือภาษาอังกฤษทั้งที่เป็นตัวอักษรและคำศัพท์ในชีวิตประจำวันโดยใช้กล้อง 2 มิติในการรับวิดีโอ โดยเฉพาะอย่างยิ่งการสกัดคุณลักษณะจากขอบของภาพไบนารีและการจำแนกท่าทางเคลื่อนไหวและ ท่าทางหยุดนิ่งโดยคำนึงถึงลำดับของการเกิดท่าทาง เนื่องจากการศึกษางานวิจัยที่เกี่ยวข้องพบว่าปัจจุบัน ได้มีการพยายามพัฒนาอัลกอริทึมที่สามารถจดจำท่าทางภาษามือเพื่อใช้ในการสื่อสารระหว่างคนปกติ และคนที่มีปัญหาเรื่องการพูด และอัลกอริทึมส่วนใหญ่มักใช้กับกล้อง 3 มิติซึ่งมีความซับซ้อนและมีราคา แพง นอกจากนี้ยังมีงานวิจัยหลายชิ้นที่พยายามใช้กล้อง 2 มิติ แต่จำเป็นที่จะต้องมียูปรกรณ์ช่วยเช่น ถุงมือ และอุปกรณ์อื่นๆซึ่งประกอบไปด้วยอุปกรณ์ที่มีสายเชื่อมต่อกับจำนวนมากซึ่งทำให้ลำบากในการใช้งานจริง

งานวิจัยนี้จึงพยายามที่จะลดปัญหาทั้งทางด้านค่าใช้จ่ายคือใช้กล้อง 2 มิติซึ่งหาซื้อได้ตาม ท้องตลาด รวมทั้งไม่ใช้อุปกรณ์เสริมใดๆซึ่งทำให้เกิดความยุ่งยากในการทำท่าทางจริง โดยมีการแบ่งการ ทดลองออกเป็น 2 ส่วน คือระบบรู้จำตัวอักษรภาษาอังกฤษ และระบบรู้จำคำในชีวิตประจำวัน

5.1 สรุปและวิจารณ์ระบบรู้จำตัวอักษรภาษาอังกฤษ

5.1.1 วิจารณ์ผลงานวิจัย

ภาพที่จะนำมาวิเคราะห์นั้นต้องทำให้เห็นแต่ภาพมือที่ชัดเจนเพื่อให้สามารถใช้วิธีการ เปรียบเทียบจุดสีเพื่อแยกมือออกจากพื้นหลังได้ดี เนื่องจากคุณลักษณะที่ใช้สกัดมาจากขอบมือของภาพ ไบนารีซึ่งเน้นไปที่การหาตำแหน่งของฝ่ามือและนิ้วมือ ดังนั้นผู้ทำท่าทางจึงต้องแสดงท่าทางที่ถูกต้องและ ตรงในแนวระนาบเดียวกับกล้อง กล่าวคือต้องไม่บิดมือไปทางซ้ายหรือขวามากเกินไปซึ่งทำให้ข้อมูลที่ได้ ผิดเพี้ยนเนื่องจากไม่เห็นฝ่ามือหรือหลังมือที่ชัดเจน โดยตัวอักษรที่เป็นปัญหาส่วนใหญ่มักมาจาก ตัวอักษรแบบกำมือซึ่งไม่สามารถเห็นปลายนิ้วมือได้ชัดเจนต้องอาศัยข้อมูลจากข้อนิ้ว อีกทั้งตัวอักษรใน กลุ่มนี้ส่วนใหญ่มักมีความแตกต่างกันที่ข้อมูลภายในของฝ่ามือซึ่งไม่สามารถแยกออกมาได้อย่างชัดเจนใน กล้อง 2 มิติ ในขณะที่กลุ่มตัวอักษรแบบมือกลับมีค่าความถูกต้องสูงถึง 100% หลายตัวอักษรด้วยกัน เนื่องจากการเห็นปลายนิ้วที่ชัดเจนและมีความแตกต่างของความยาวของแต่ละนิ้วค่อนข้างมาก

ด้านการประมวลผลข้อมูลพบว่าการประมวลผลภาพนั้นต้องใช้เครื่องมือที่มีความเร็วสูงในการคำนวณข้อมูลและวิเคราะห์ผล จึงจำเป็นต้องใช้คอมพิวเตอร์ที่สมรรถนะสูงเพื่อให้การประมวลผลเป็นไปอย่างรวดเร็วและมีประสิทธิภาพมากขึ้น

5.1.2 สรุปผลการทดลองงานวิจัย

การวิเคราะห์จำแนกตัวอักษรภาษาอังกฤษของภาษามือออกจากกันพบว่าวิธีที่นำมาใช้สกัดคุณลักษณะมีทั้งสิ้น 4 วิธีซึ่งประกอบด้วยพื้นที่ของขอบภาพ 3 ด้าน, ความยาวนิ้วมือจากจุดเซ็นทรอยด์, มุมระหว่างนิ้วมือของเฟรมแรกและเฟรมหลัง และค่าเดลต้าของมุมระหว่างนิ้วมือของเฟรมหน้าและเฟรมสิ้นสุด รวมทั้งหมด 25 คุณลักษณะ โดยข้อมูลทั้งหมดถูกแบ่งออกเป็น 2 กลุ่มคือชุดข้อมูลสำหรับการ Training เพื่อสร้างโมเดลและ ชุดข้อมูลสำหรับ Testing เพื่อทดสอบประสิทธิภาพของโมเดลและ Validation ในขั้นตอนของการสร้างโมเดลเราเลือกใช้วิธีโครงข่ายประสาทเทียมแบบ Two-layer feed-forward บนโปรแกรม MATLAB ซึ่งอัลกอริทึมที่นำเสนอสามารถแยกกลุ่มตัวอักษรออกจากกันได้อย่างมีประสิทธิภาพด้วยค่าความถูกต้องถึงร้อยละ 95 จะเห็นได้ว่าความถูกต้องจากการทดลองมีประสิทธิภาพสูงแม้ว่าจะมีตัวอักษรที่มีความผิดพลาดอยู่บ้างแต่ก็อยู่ในเกณฑ์ที่ยอมรับได้ อีกทั้งการประมวลผลยังเป็นไปด้วยความถูกต้องและรวดเร็วอีกด้วย

5.2 สรุปและวิจารณ์ระบบรู้จำคำในชีวิตประจำวัน

5.2.1 วิจารณ์ผลงานวิจัย

หัวข้อวิจารณ์ผลงานวิจัยในระบบรู้จำคำในชีวิตประจำวันถูกแบ่งออกเป็น 2 หัวข้อได้แก่

5.2.1.1. วิจารณ์เทคนิครู้จำคำโดยใช้เฟรมทั้งหมด ข้อดีของเทคนิคนี้คือใช้เวลาในการประมวลผลเร็วที่สุดจากทุกวิธีดังแสดงในตารางที่ 4.11 แต่วิธีนี้มีข้อเสียก็คือมีขั้นตอนในการประมวลผลมากกว่าเทคนิครู้จำคำโดยใช้เฟรมหลักเนื่องจากต้องผ่านทั้งกระบวนการปรับขนาดของข้อมูลให้มีขนาดเท่ากันเสียก่อน (เมื่อใช้คำศัพท์มากขึ้นอาจจะต้องเปลี่ยนค่ากลางอีกด้วย) และยังต้องผ่านขั้นตอนการลดมิติของข้อมูลด้วยวิธี PCA ก่อนนำเข้าเรียนรู้ในโครงข่ายประสาทเทียมด้วย เนื่องจากข้อมูลมีความซับซ้อนเพราะต้องคำนวณคุณลักษณะของทุกเฟรมภาพในช่วงของท่าทาง โดยผู้ทำการทดลองพบว่าเฟรมภาพที่อยู่ติดกันนั้นจะให้ค่าคุณลักษณะที่ไม่แตกต่างกันมากหรือแทบไม่แตกต่างกันเลยหากใช้ค่าเฟรมเรทสูงๆ (ซึ่งแก้ไขได้ด้วยการใช้เฉพาะเฟรมหลัก)

อีกทั้งเมื่อคำนวณค่า Fitness Classifier ยังให้ค่าไม่ถึง 70% ซึ่งเกือบจะน้อยที่สุดในบรรดาวิธีการทั้งหมด (เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 5 ช่วง ให้ค่า Fitness Classifier ต่ำที่สุดซึ่งต่ำกว่าเพียง 3%) แต่เมื่อวิเคราะห์ที่กราฟ ROC กลับพบว่ามีความแม่นยำที่ได้กราฟสูงสุดซึ่งบ่งบอกว่าวิธีการนี้สามารถแยกค่าในคลาสของตัวเองได้ดีที่สุด

5.2.1.2. วิจัยเทคนิครู้จำคำโดยใช้เฟรมหลัก วิธีการนี้ถือเป็นการแก้ปัญหาการใช้ข้อมูลซ้ำของ เทคนิครู้จำคำโดยใช้เฟรมทั้งหมด กล่าวคือจะเป็นการใช้เฉพาะท่าทางที่สำคัญซึ่งเป็นเอกลักษณ์ในช่วงการเปลี่ยนแปลงท่าทางของคำศัพท์นั้นๆ โดยการทดลองนี้ได้ทำการแบ่งช่วงของท่าทางที่สำคัญออกเป็น 4 การทดลองคือ 4, 5, 6 และ 7 ช่วงด้วยกันด้วยกัน ผลที่ได้คือเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 6 ช่วง จะให้คำศัพท์ที่ตอบถูกต้อง 100% มากที่สุดและมีค่าความถูกต้องถูกที่สุดในเทคนิครู้จำคำโดยใช้เฟรมหลัก อีกทั้งยังสูงกว่าวิธีการเทคนิคการรู้จำคำโดยใช้เฟรมทั้งหมด แต่ใช้เวลาในการประมวลผลมากกว่าถึง 3 วินาทีเช่นเดียวกัน เนื่องจากเสียเวลาในการคำนวณเพื่อเลือกเฟรมหลักของแต่ละช่วงท่าทาง ในขณะที่เทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 7 ช่วง กลับให้ค่าความถูกต้องที่น้อยกว่าแบบ 6 ช่วงและใช้เวลาในการประมวลผลนานกว่า 1 วินาทีอีกด้วย เมื่อนำค่า Fitness Classifier และกราฟ ROC มาวิเคราะห์ พบว่าเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 6 ช่วง ให้ค่าสูงที่สุดอีกด้วย

5.2.2 สรุปผลการทดลองงานวิจัย

การวิเคราะห์จำแนกคำภาษาอังกฤษในชีวิตประจำวันของภาษามือได้แบ่งการทดลองออกเป็น 2 เทคนิคได้แก่ เทคนิครู้จำคำโดยใช้เฟรมทั้งหมดซึ่งค่าคุณลักษณะจะถูกนำไปนอมัลไลซ์ให้มีจำนวนเท่ากับ 75 ค่า/1 คุณลักษณะ แล้วจึงเข้าสู่กระบวนการ PCA เพื่อลดจำนวนข้อมูลเข้าก่อนนำไปจำแนกประเภทของคำศัพท์ และเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งแบ่งการทดลองของลำดับท่าทางที่สำคัญเป็น 4, 5, 6 และ 7 ช่วง ซึ่งทั้งสองวิธีต่างก็มีการแบ่งข้อมูลที่ใช้ Test และ Train รวมถึงจำนวน Hidden layer และชนิดของฟังก์ชันในชั้นซ่อนและชั้นเอาต์พุตเหมือนกันทุกประการรวมถึงคุณลักษณะที่ใช้ในการจำแนกก่อนนำเข้าเรียนรู้ด้วยโครงข่ายประสาทเทียมด้วย จากการทดลองพบว่าเทคนิครู้จำคำโดยใช้เฟรมหลักซึ่งถูกแบ่งออกเป็น 6 ช่วงให้ค่าความถูกต้องที่มีประสิทธิภาพสูงที่สุดแม้จะใช้เวลามากกว่าเทคนิครู้จำคำโดยใช้เฟรมหลักเล็กน้อยแต่เหมาะสมสำหรับนำไปใช้งานมากกว่าเนื่องจากเทคนิครู้จำคำโดยใช้เฟรมทั้งหมดมีขั้นตอนการประมวลผลที่ซับซ้อนกว่าโดยต้องผ่านทั้งกระบวนการปรับขนาดของข้อมูลและ

ขั้นตอนการลดมิติของข้อมูลก่อนนำเข้าสู่การเรียนรู้ด้วยโครงข่ายประสาทเทียม โดยเมื่อเปรียบเทียบกับประสิทธิภาพของความถูกต้องที่ได้แล้ว เทคนิครู้จำคำโดยเทคนิครู้จำคำซึ่งถูกแบ่งออกเป็น 6 ช่วงยังคงเหมาะสมแก่การนำไปใช้งานมากกว่าเทคนิครู้จำคำโดยใช้เฟรมทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] J. Ravkiran, M. Kavi, M. Suhas, R. Dheeraj, S. Sudheeder, P. V. Nitin, “Finger Detection for Sign Language Recognition,” **International MultiConference of Engineers and Computer Scientists (IMECS)**, Vol 1, Hongkong, March 18-20, 2009.
- [2] J. R. Pansare, S. H. Gawande and M. Ingle, “Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background,” **Journal of Signal and Information Processing**, No. 3. pp. 364-367, 2012
- [3] R. Rajeshree, D. Dharmpal and K. Manesh, “Hand Gesture Recognition by Thinning Method,” **International Conference on Digital Image Processing**, pp.284-287, Bangkok, Thailand, March 7-9, 2009
- [4] Md. A. Rahman, Ahsan-Ul-Ambia and Md. Aktaruzzama, “Recognition Static Hand Gestures of Alphabet in ASL,” **International Journal of Computer and Information Technology (IJCIT)**, Vol. 02, Issue 01, pp.75-77, 2011
- [5] K. Alina, L. Laura and R. Bodo, “Real-time sign language recognition using a consumer depth camera,” **IEEE International Conference on Computer Vision Workshops**, Sydney, NSW, Australia ,2-8 Dec. 2013
- [6] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual interpretation of hand gestures for human-computer interaction: A review.” **IEEE Trans. on Pattern Recognition and Machine Intelligence**, vol. 19 issue 7, pp 677-695, July 1997
- [7] B. Y. Said, A. Eric, M. Franck and K. Richard, “Dynamic hand gesture recognition based on 3D pattern assembled trajectories,” **IEEE International Conference on Image Processing Theory, Tools and Applications**, Montreal, Canada, pp.1-6, 2017
- [8] B. Y. Said, A. Eric, K. Richard and M. Franck, “HIF3D: Handwriting-Inspired Features for 3D skeleton-based action recognition,” **International Conference on Pattern Recognition (ICPR)**, pp.980-985, December 4-8, 2016

- [9] A. Delaye and E. Anquetil, “Hbf49 feature set: A first unified baseline for online symbol recognition,” **Pattern Recognition**, vol. 46, no. 1, pp. 117–130, 2013.
- [10] S. Lidia, C. Nicola, T. Alessandro, G. L. José, F. d. l. Eusebio, F. C. Juan and P. Javier, “Dynamic Gesture Recognition Using a Smart Glove in Hand-Assisted Laparoscopic Surgery,” **Technologies** 2018, 6, 8
- [11] M. K. Bhuyan, D. Ajay Kumar, Karl F. MacDorman and Yuji Iwahori, “A novel set of features for continuous hand gesture recognition,” **Journal on Multimodal User Interfaces** 8(4), 8 oct. 2012
- [12] L. Hervé and L. Derek, “Real-Time Hand Gesture Recognition Using Range Cameras,” **The Computer Modelling Group LTD and the Natural Sciences and Engineering Research Council of Canada**
- [13] P. K. Neelamma, Y. M. Ravi and P. Jagadeesh, “Comparison between HSV and YCbCr Color Model Color-Texture based Classification of the Food Grains,” **International Journal of Computer Applications (0975 – 8887)**, Volume 34, No.4, November 2011
- [14] D. Chai, and K.N. Ngan, "Face segmentation using skin-color map in videophone applications" **IEEE Trans. on Circuits and Systems for Video Technology**, 9(4): 551-564, June 1999.]
- [15] ตะวัน ชุนอาสา, “ระบบตรวจจับและตีความปลายจำกัดความเร็วในเวลากลางคืนด้วยเทคนิค Haar-like Feature Detection”
- [16] L. A. Noor, K. Z. Rafiqul and H. M Mokhtar, “Comparative Study of Skin Color Based Segmentation Techniques,” **International Journal of Applied Information Systems (IJ AIS)**, Foundation of Computer Science FCS, New York, USA, Volume 5– No. 10, August 2013

- [17] J. Ravkiran, M. Kavi, M. Suhas, R. Dheeraj, S. Sudheeder, P. V. Nitin, “Finger Detection for Sign Language Recognition,” **International MultiConference of Engineers and Computer Scientists (IMECS)**, Vol 1, Hongkong, March 18-20, 2009.
- [18] A. Sharmila Konwar, B. Sagarika Borah, C. Dr.T.Tuithung, “An American Sign Language Detection System using HSV Color Model and Edge Detection,” **International Conference on Communication and Signal Processing**, pp. 743-746, Melmaruvathur, India, April 3-5, 2014
- [19] J. R. Pansare, S. H. Gawande and M. Ingle, “Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background,” **Journal of Signal and Information Processing**, No. 3. pp. 364-367, 2012
- [20] Md. A. Rahman, Ahsan-Ul-Ambia and Md. Aktaruzzama, “Recognition Static Hand Gestures of Alphabet in ASL,” **International Journal of Computer and Information Technology (IJCIT)**, Vol. 02, Issue 01, pp.75-77, 2011



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดรู้จำตัวอักษรภาษาอังกฤษ

```

close all; clc; clear all;
path= 'E:\Project\USE\Database3\MN';
filelist = ls([path '*.jpg']);
filelist = cellstr(filelist);
IM1=imread('E:\Project\USE\Database3\BG1.jpg');
c=1; n=1;
for z= 2253
    disp(z)
    IM2=imread ([path filelist{z}]);
    [out]=imgrethresh(IM2,IM1);
    imshow(out)
    Start=Handprocess(out);
    [out,crop]=imgrethresh(IM2,IM1);
    Stop=Handprocess(out);
    del=abs (Start(:,7:10) - Stop(:,7:10));
    deld = abs (Start(:,11:15) - Stop(:,11:15));
    sumd = sum (deld);
    AFINAL(:,n)=[transpose(Start); transpose(Stop(:,7:15)); transpose(del); transpose(deld); sumd];
    n=n+1;
end
c=c+20;
h2 = msgbox('DONE!!');

```

โค๊ดรู้จำคำภาษาอังกฤษโดยใช้เฟรมทั้งหมด

```

clear all, close all, clc
path= 'E:\Pookkii work\Master Degree\Project M\Database\3_DATABASE\Mild_fbf\';
pathsave='E:\Pookkii work\Master Degree\Project M\Database\3_DATABASE\Mild_fbf\';
filelist = ls([path '*.jpg']);
filelist = cellstr(filelist);
load ('M_1.mat');
Left = [ ];
Right =[ ];U=1;
for i = 2:2:length(M_3)
    disp('-----')
    disp(['No.Video =',num2str(i)])
    k=1;q=1;
    s= M_3(i,1)+2;
    f= M_3(i,end)-2;
    for w=s:f
        Delta = f-s;
        disp(w)
        IM1=imread([path filelist{w-1}]);
        IM2=imread([path filelist{w}]);
        H1= deleteFace2(IM1);
        [H2 C2] = deleteFace2(IM2);
        Cface(k,:)= C2;
        stF = regionprops (H1,
'BoundingBox','Centroid','MajorAxisLength','MinorAxisLength','Perimeter','Area','Orientation','EquivDiameter');
        stL = regionprops(H2,
'BoundingBox','Centroid','MajorAxisLength','MinorAxisLength','Perimeter','Area','Orientation','EquivDiameter');
        HandNum(q,:)= length(stL);
        q=q+1;
        %%% After knew that st(n)=st(n+1)|| size(stF,1)< size(stL,1) %%%
    if w==s
        if size(stF,1)==size(stL,1) || size(stF,1)< size(stL,1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

st = stL;
j=1;
if size(st,1)>1
    st1=st(1).BoundingBox(1,1);
    st2=st(2).BoundingBox(1,1);
    if st1(1)<st2(1)
        CenLeft(k,:)= round(st(1).Centroid(1:2));
        CenRight(k,:)=round(st(2).Centroid(1:2));
        BoundL(k,:)= st(1).BoundingBox;
        BoundR(k,:)= st(2).BoundingBox;
        RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
        RoundR(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
        OrientL(k,:)= ((st(1).Orientation)+90)/180;
        OrientR(k,:)= ((st(2).Orientation)+90)/180;
        AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        k=k+1;
    elseif st1(1)>st2(1)
        CenLeft(k,:)= round(st(2).Centroid(1:2));
        CenRight(k,:)=round(st(1).Centroid(1:2));
        BoundL(k,:)= st(2).BoundingBox;
        BoundR(k,:)= st(1).BoundingBox;
        RoundL(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
        RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
        OrientL(k,:)= ((st(2).Orientation)+90)/180;
        OrientR(k,:)= ((st(1).Orientation)+90)/180;
        AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        k=k+1;
    end
elseif size(st,1)==1
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)= [0 0];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= [0 0 0 0];
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = 0;
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= 0;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= 0;
    k=k+1;
end
elseif size(stF,1)> size(stL,1) %% 2 hands --> 1 hand
    st = stL;
    j=2;
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    k=k+1;
end

```

%% check after the beginning whether hand is still the same no. or change from 1--> 2 hands

%%

elseif w~=1&&j==1

%%

if size(stF,1)==size(stL,1) || size(stF,1)< size(stL,1) %% equal or 1 hand --> 2 hands

st = stL;

j=1;

if size(st,1)>1

```

st1=st(1).BoundingBox(1,1);
st2=st(2).BoundingBox(1,1);
    if st1(1)<st2(1)
        CenLeft(k,:)= round(st(1).Centroid(1:2));
        CenRight(k,:)=round(st(2).Centroid(1:2));
        BoundL(k,:)= st(1).BoundingBox;
        BoundR(k,:)= st(2).BoundingBox;
        RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
        RoundR(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
        OrientL(k,:)= ((st(1).Orientation)+90)/180;
        OrientR(k,:)= ((st(2).Orientation)+90)/180;
        AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        k=k+1;
    elseif st1(1)>st2(1)
        CenLeft(k,:)= round(st(2).Centroid(1:2));
        CenRight(k,:)=round(st(1).Centroid(1:2));
        BoundL(k,:)= st(2).BoundingBox;
        BoundR(k,:)= st(1).BoundingBox;
        RoundL(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
        RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
        OrientL(k,:)= ((st(2).Orientation)+90)/180;
        OrientR(k,:)= ((st(1).Orientation)+90)/180;
        AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        k=k+1;
    end

```

```

elseif size(st,1)==1
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)= [0 0];
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= [0 0 0 0];

```

```

RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
RoundR(k,:) = 0;
OrientL(k,:)= ((st(1).Orientation)+90)/180;
OrientR(k,:)= 0;
AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
AxisratioR(k,:)= 0;
k=k+1;
end
elseif size (stF,1)> size(stL,1) %% 2 hands --> 1 hand
st = stL;
j=2;
CenLeft(k,:)= round(st(1).Centroid(1:2));
CenRight(k,:)=round(st(1).Centroid(1:2));
BoundL(k,:)= st(1).BoundingBox;
BoundR(k,:)= st(1).BoundingBox;
RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
OrientL(k,:)= ((st(1).Orientation)+90)/180;
OrientR(k,:)= ((st(1).Orientation)+90)/180;
AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
k=k+1;
end
%%% check after hand change from 2 hands --> 1 hand
elseif w~=1&&j==2
%%
if size(stF,1)==size(stL,1) %% 2 hands --> 1 hand
st = stL;
j=2;
CenLeft(k,:)= round(st(1).Centroid(1:2));
CenRight(k,:)=round(st(1).Centroid(1:2));
BoundL(k,:)= st(1).BoundingBox;
BoundR(k,:)= st(1).BoundingBox;

```

```

RoundL(k,:)= (4*pi*st(1).Area)/(st(1).Perimeter^2);
RoundR(k,:)= (4*pi*st(1).Area)/(st(1).Perimeter^2);
OrientL(k,:)= ((st(1).Orientation)+90)/180;
OrientR(k,:)= ((st(1).Orientation)+90)/180;
AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
k=k+1;
elseif size(stF,1)< size(stL,1) %%1 hand --> 2 hands
st = stL;
j=1;
st1=st(1).BoundingBox(1,1);
st2=st(2).BoundingBox(1,1);
if st1(1)<st2(1)
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)=round(st(2).Centroid(1:2));
    BoundL(k,:)= st(1).BoundingBox; %%Delx/Dely
    BoundR(k,:)= st(2).BoundingBox;
    RoundL(k,:)= (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:)= (4*pi*st(2).Area)/(st(2).Perimeter^2);
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= ((st(2).Orientation)+90)/180;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
    k=k+1;
elseif st1(1)>st2(1)
    CenLeft(k,:)= round(st(2).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(2).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:)= (4*pi*st(2).Area)/(st(2).Perimeter^2);
    RoundR(k,:)= (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(2).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;
    AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;

```

```

        AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        k=k+1;
    end
end
end

```

```

end
end

```

```

CenLeft1 = imresize(CenLeft(:,1)',[1 75]);
CenLeft2 = imresize(CenLeft(:,2)',[1 75]);
CenRight1 = imresize(CenRight(:,1)',[1 75]);
CenRight2 = imresize(CenRight(:,2)',[1 75]);
FarrayL = [CenLeft1' CenLeft2'];
FarrayR = [CenRight1' CenRight2'];
Cface1 = round(imresize(Cface(:,1)',[1 75]));
Cface2 = round(imresize(Cface(:,2)',[1 75]));
BoundratioL1=imresize(BoundL(:,1)',[1 75]);
BoundratioL2=imresize(BoundL(:,2)',[1 75]);
BoundratioL3=imresize(BoundL(:,3)',[1 75]);
BoundratioL4=imresize(BoundL(:,4)',[1 75]);
BoundratioR1=imresize(BoundR(:,1)',[1 75]);
BoundratioR2=imresize(BoundR(:,2)',[1 75]);
BoundratioR3=imresize(BoundR(:,3)',[1 75]);
BoundratioR4=imresize(BoundR(:,4)',[1 75]);
BoundFL = [BoundratioL1' BoundratioL2' BoundratioL3' BoundratioL4'];
BoundFR = [BoundratioR1' BoundratioR2' BoundratioR3' BoundratioR4'];
Cfacef = [Cface1' Cface2'];

```

```

%%%%%%%%%%%%%% Features %%%%%%%%%%%%%%%

```

```

Round1 = imresize(RoundL',[1 75]); %75
Round2 = imresize(RoundR',[1 75]); %75
Orient1 = imresize(OrientL',[1 75]);%75
Orient2 = imresize(OrientR',[1 75]);%75

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Axisratio1 = imresize(AxisratioL',[1 75]);%75
Axisratio2 = imresize(AxisratioR',[1 75]); %75
[distBL ratioL]=Bdist(BoundFL); %148
[distBR ratioR]=Bdist(BoundFR); %148
FdistL = Cdist(FarrayL); %74
FdistR = Cdist(FarrayR); %74
FdistLR = Cdsame(FarrayL,FarrayR); %75
[FvecL1 FvecL2] = Cvec(FarrayL); %148
[FvecR1 FvecR2] = Cvec(FarrayR); %148
[FvecFL1 FvecFL2] = Cvecf(Cfacef,FarrayL); %148
[FvecFR1 FvecFR2] = Cvecf(Cfacef,FarrayR);
HandFaceL = Cdsame(Cfacef,FarrayL); %75
HandFaceR = Cdsame(Cfacef,FarrayR); %75
FstNum = (imresize(HandNum',[1 75]))'./2; %75
%%
ZFinal(U,:) = [Round1 Orient1 Axisratio1 distBL' ratioL' FdistL' FdistLR' FvecL1' FvecL2' FvecFL1'
FvecFL2' HandFaceL' Round2 Orient2 Axisratio2 distBR' ratioR' FdistR' FvecR1' FvecR2' FvecFR1'
FvecFR2' HandFaceR' FstNum'];
%
U=U+1;
end

```

โค้ดรู้จำคำภาษาอังกฤษโดยใช้เฟรมหลัก

```

clear all, close all, clc
path= 'E:\Pookkii work\Master Degree\Project M\Database\4_DATABASE\Pookkii_fbf\';
pathsave='E:\Pookkii work\Master Degree\Project M\Database\4_DATABASE\Pookkii_fbf\';
filelist = ls([path '*.jpg']);
filelist = cellstr(filelist);
load ('P_4.mat');
m=1;
for i= 1:length(P_4)
    s= P_4(i,1)+2;%%%%
    f= P_4(i,end)-2;%%%%
    q=1;
        disp(i)
        for w=s:f
            disp(w)
            IM1=imread([path filelist{w-1}]);
            IM2=imread([path filelist{w}]);
            H1= deleteFace(IM1);
            H2 = deleteFace(IM2);
            IM3 = H2-H1;
            IM4 = sum(logical (IM3(:)));
            final(q,:)= IM4;
            q=q+1;
        end
        y=ceil(length(final)/6); % 6. %%%
        c=1;
        for D = 1:8
            if D < 6
                p =find(final==min(final(c:c+y-1)));
                a = p(p<=c+y-1 & p>=c);
                a=a(1,:);
                a=s+a;
                Frame(D,:) = a;
                c=c+y;
            end
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elseif D==6;
z=length(final)-c;
p=(find(final==min(final(c:c+z))));
a = p(p<=c+z & p>=c);
a=a(1,:);
a=s+a;
c=c+z;
Frame(D,:) = a;
elseif D==7
a=Frame(1,1);
Frame(D,:) = a;
elseif D==8
a=Frame(2,1);
Frame(D,:) = a;
end
end %%% for D = 1:8
FFrame(:,m) = Frame;
m=m+1;
clear final
end %%% for z= 1:length(P_1)

FCenLeft =[ ]; FCenRight=[ ];FBoundL=[ ];FBoundR=[ ];FCface=[ ];FRoundL=[ ];
FRoundR=[ ];FOrientL=[ ]; FOrientR=[ ];FAXisratioL=[ ]; FAXisratioL=[ ];FAXisratioR =[ ];
FSideL=[ ]; FSideR=[]; FHandnum=[ ];

for z = 1:size(FFrame,2)
k=1; q=1; disp(z)
for s = 1:7
IM1 = imread([path fileList{FFrame(s,z)}]);
IM2 = imread([path fileList{FFrame(s+1,z)}]);
[H1 C1]= deleteFace2(IM1);
[H2 C2]= deleteFace2(IM2);clear C2
Cface(k,:)= C1;

```

```

stF = regionprops(H1,
'BoundingBox','Centroid','MajorAxisLength','MinorAxisLength','Perimeter','Area','Orientation','EquivDiameter');
stL = regionprops(H2,
'BoundingBox','Centroid','MajorAxisLength','MinorAxisLength','Perimeter','Area','Orientation','EquivDiameter');
HandNum(q,:)= length(stL);
q=q+1;
%% Start to run feature From any s%%
if s==1
%%
if size(stF,1)==size(stL,1) || size(stF,1)< size(stL,1) %% equal or 1 hand --> 2 hands
st = stL;
j=1;
if size(st,1)>1
st1=st(1).BoundingBox(1,1);
st2=st(2).BoundingBox(1,1);
if st1(1)<st2(1)
CenLeft(k,:)= round(st(1).Centroid(1:2));
CenRight(k,:)=round(st(2).Centroid(1:2));
BoundL(k,:)= st(1).BoundingBox; %%Delx/Dely
BoundR(k,:)= st(2).BoundingBox;
RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
RoundR(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
OrientL(k,:)= ((st(1).Orientation)+90)/180;
OrientR(k,:)= ((st(2).Orientation)+90)/180;
AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
k=k+1;

elseif st1(1)>st2(1)
CenLeft(k,:)= round(st(2).Centroid(1:2));
CenRight(k,:)=round(st(1).Centroid(1:2));
BoundL(k,:)= st(2).BoundingBox;

```

```

BoundR(k,:)= st(1).BoundingBox;
RoundL(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
OrientL(k,:)= ((st(2).Orientation)+90)/180;
OrientR(k,:)= ((st(1).Orientation)+90)/180;
AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
k=k+1;
end

elseif size(st,1)==1
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)= [0 0];
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= [0 0 0 0];
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = 0;
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= 0;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= 0;
    k=k+1;
end

elseif size(stF,1)> size(stL,1) %% 2 hands --> 1 hand
    st = stL;
    j=2;
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;

```

```

AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
k=k+1;

end

%% check after the begining whether hand is still the same no. or change from 1--> 2 hands
elseif s~=1&&j==1
%%
if size(stF,1)==size(stL,1) || size(stF,1)< size(stL,1) %% equal or 1 hand --> 2 hands
    st = stL;
    j=1;
    if size(st,1)>1
        st1=st(1).BoundingBox(1,1);
        st2=st(2).BoundingBox(1,1);
        if st1(1)<st2(1)
            CenLeft(k,:)= round(st(1).Centroid(1:2));
            CenRight(k,:)=round(st(2).Centroid(1:2));
            BoundL(k,:)= st(1).BoundingBox;
            BoundR(k,:)= st(2).BoundingBox;
            RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
            RoundR(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
            OrientL(k,:)= ((st(1).Orientation)+90)/180;
            OrientR(k,:)= ((st(2).Orientation)+90)/180;
            AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
            AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
            k=k+1;
        elseif st1(1)>st2(1)
            CenLeft(k,:)= round(st(2).Centroid(1:2));
            CenRight(k,:)=round(st(1).Centroid(1:2));
            BoundL(k,:)= st(2).BoundingBox;
            BoundR(k,:)= st(1).BoundingBox;
            RoundL(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
            RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
            OrientL(k,:)= ((st(2).Orientation)+90)/180;
            OrientR(k,:)= ((st(1).Orientation)+90)/180;
        end
    end
end

```

```

        AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        k=k+1;
    end

elseif size(st,1)==1
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)= [0 0];
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= [0 0 0 0];
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = 0;
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= 0;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= 0;
    k=k+1;
end
elseif size(stF,1)> size(stL,1) %% 2 hands --> 1 hand
    st = stL;
    j=2;
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    k=k+1;
end
%% check after hand change from 2 hands --> 1 hand

```

```

elseif s~=1&&j==2
%%
if size(stF,1)==size(stL,1) %% 2 hands --> 1 hand
    st = stL;
%    FSide1 = imfindside(H2,stL,1);
    j=2;
    CenLeft(k,:)= round(st(1).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(1).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(1).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;
    AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    k=k+1;
elseif size(stF,1)< size(stL,1) %%1 hand --> 2 hands
    st = stL;
    j=1;
    st1=st(1).BoundingBox(1,1);
    st2=st(2).BoundingBox(1,1);
    if st1(1)<st2(1)
        CenLeft(k,:)= round(st(1).Centroid(1:2));
        CenRight(k,:)=round(st(2).Centroid(1:2));
        BoundL(k,:)= st(1).BoundingBox;
        BoundR(k,:)= st(2).BoundingBox;
        RoundL(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
        RoundR(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
        OrientL(k,:)= ((st(1).Orientation)+90)/180;
        OrientR(k,:)= ((st(2).Orientation)+90)/180;
        AxisratioL(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
        AxisratioR(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
        k=k+1;

```

```

elseif st1(1)>st2(1)
    CenLeft(k,:)= round(st(2).Centroid(1:2));
    CenRight(k,:)=round(st(1).Centroid(1:2));
    BoundL(k,:)= st(2).BoundingBox;
    BoundR(k,:)= st(1).BoundingBox;
    RoundL(k,:) = (4*pi*st(2).Area)/(st(2).Perimeter^2);
    RoundR(k,:) = (4*pi*st(1).Area)/(st(1).Perimeter^2);
    OrientL(k,:)= ((st(2).Orientation)+90)/180;
    OrientR(k,:)= ((st(1).Orientation)+90)/180;
    AxisratioL(k,:)= st(2).MinorAxisLength/st(2).MajorAxisLength;
    AxisratioR(k,:)= st(1).MinorAxisLength/st(1).MajorAxisLength;
    k=k+1;
end
end
end %%s==1
end %% s=1:7
%%%% Extra calculation features%%% |1|2|3| I=in frame , B=Between frame
%%
FCface= [FCface Cface];
FCenLeft= [FCenLeft CenLeft];
FCenRight = [FCenRight CenRight];
FBoundL = [FBoundL BoundL] ;
FBoundR = [FBoundR BoundR];
FHandnum = [FHandnum HandNum(1:6,:)];
FRoundL= [FRoundL RoundL(1:6,:)];
FRoundR= [FRoundR RoundR(1:6,:)];
FOrientL=[FOrientL OrientL(1:6,:)];
FOrientR= [FOrientR OrientR(1:6,:)];
FAxisratioL = [FAxisratioL AxisratioL(1:6,:)];
FAxisratioR = [FAxisratioR AxisratioR(1:6,:)];
end %% z=1:size(FFrame,2)
FdistHFL = Cdsame(FCface,FCenLeft); %%B(Distance between Face and Lhand)*
FdistHFR = Cdsame(FCface,FCenRight); %%B(Distance between Face and Rhand)*
FdistLR = Cdsame(FCenLeft,FCenRight); %%B(Distance between Lhand and RHand)*

```

```

FdistCL = Cdist(FCenLeft); %B(Centroid distance)
FdistCR = Cdist(FCenRight); %B(Centroid distance)
[FdistBL FratioL]=Bdist(FBoundL); %B(Cornor distance) %B(Ratio width&length) **
[FdistBR FratioR]=Bdist(FBoundR); %B(Cornor distance) %B(Ratio width&length) **
FHandnum = FHandnum/2;
[FvecL1 FvecL2] = Cvec(FCenLeft); %l
[FvecR1 FvecR2] = Cvec(FCenRight); %l
[FvecFL1 FvecFL2] = Cvecf(FCface,FCenLeft); %l
[FvecFR1 FvecFR2] = Cvecf(FCface,FCenRight); %l
%%%% Finally %%%
ZFinal = [FRoundL' FRoundR' FOrientL' FOrientR' FAxisratioL' FAxisratioR' ...
          FdistHFL' FdistHFR' FdistLR' FdistCL' FdistCR' FdistBL' FratioL' FdistBR' ...
          FratioR' FvecL1' FvecL2' FvecR1' FvecR2' FvecFL1' FvecFL2' FvecFR1' FvecFR2' FHandnum'];

```

A Novel Feature Extraction for American Sign Language Recognition Using Webcam

Ariya Thongtawee¹, Onamon Pinsanoh², Yuttana Kitjaidure³

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Bangkok, Thailand

57011508@kmitl.ac.th¹, 57011504@kmitl.ac.th², kkyuttan@gmail.com³

Abstract— Sign language is physical communication for contributing the meaning instead of using voice to demonstrate communicator's opinion. This paper introduces a simple and efficient algorithm for feature extraction to recognize American Sign Language alphabets from both static and dynamic gestures. The proposed algorithm comprises of four different techniques: Number of white pixels at the edge of the image (NwE), Finger length from the centroid point (Fcen), Angles between fingers (AngF) and Differences of angles between fingers of the first and last frame (delAng). After extracting features from video images, an Artificial Neural Network (ANN) is used to classify the signs. The result of these experiments is achieved up to 95% recognition rate, which is clearly to be the highest accuracy comparing with the other research worked in this field.

Keyword— American Sign Language (ASL), Hand gesture recognition, Feature Extraction, Artificial Neural Network, Webcam.

I. INTRODUCTION

Sign language is a language that employs signs made with the hands and other movements, including facial expressions and postures of the body, used primarily by people who are deaf. Although, there are several research working on a method for American sign language recognition to let the deaf people and the people who do not know sign language to communicate each other to reduce the gap during communication, but they are still focused on opened fingers postures or some alphabets which do not cover all the letters in American sign language.

Our paper aims to bridge this gap by introducing an algorithm which can accurately identify American Sign Language of all 26 alphabets, which has both static and dynamic (J and Z) gestures by using inexpensive image processing equipment. This algorithm can be automatically identified the alphabets without using any sensor gloves or 3D camera. Only a computer with a webcam are worked together in capturing, processing and classifying to recognize the signs.

Various studies are invented different techniques to extract features of the hands to interpret American Sign Language alphabets. Most of them can be recognized only opened finger or some postures [1], [2] and achieve the result only 60-95%; moreover, some research needs special equipment [3], [4] or 3D camera [5] which makes even much cost in processing.

In addition, some research even needs a lot of cameras to increase the accuracy [6]. Even though some research can detect all alphabets in ASL, but their recognition rate is still low.

Thus, we offer a simple techniques of image processing algorithm based on image contour with high accuracy, fast and easy to understand in real usability.

Our work has focused on extracting features from the boundary of binary images in gestures representing alphabets. Using just the edge of the image is another challenge because we do not know any information inside the hand which is even harder to find the real fingers in closed hand gestures. However, this method of extracting the features is able to use with any kind of 2D camera and fast enough for using in real-time processing.

II. PROPOSED METHODOLOGY

Our algorithm is divided into 4 steps, starting with capturing video and converting to images for pre-processing to improve image quality, after that specifying palm positions to separate them from the fingers. The next step is to extract features from the images by using contour-based techniques and the last step is to classify by using Artificial Neural Network (ANN) which

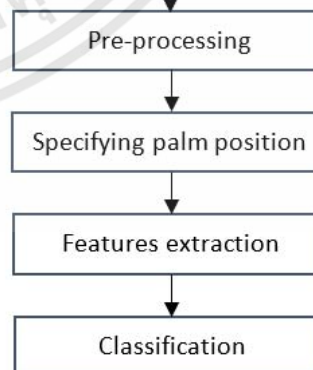


Fig. 1: Overview of the proposed system

uses to recognize signs to alphabets. The proposed system is shown in Fig. 1.

A. Pre-processing

This method aims to improve image quality by transforming the image to grey scale then binary and removes noise to get only the clear image then segmentation is done using background removing (original image minus its background) as shown in Fig. 2 then crop only the hand part by using its Blob.

B. Specifying Palm Position

We start by finding the edge in image by using ‘Canny’ algorithm, which has adaptivity to find local contents of the image. This process reduces problems related with edge discontinuities by identifying strong edges and preserving the relevant weak edges. This makes us easier to find the exact contour of hand.

We now draw the possible biggest circle by using Eq.1 and use its center to be ‘centroid’ and the point that breaks the palm into 2 sections left (has 2 fingers) and right (has 3 fingers) as shown in Fig. 3.

$$x^2+y^2=r^2 \quad (1)$$

Automatically adjusting the orientation of the hand is needed because our techniques required the fingers to be located in vertical axis. By using the concept that the palm position must be located above the wrist (the edge where the white pixels are) and its principle axis should be located as closely to vertical as shown in Fig.4.

C. Features Extraction

We use 4 different techniques, include 3 digits from the number of white pixels in binary image along the edge as shown in Fig. 5, 10 digits of finger length from the centroid point, 8 digits of angles between fingers and 4 digits of angle differences between the first and last frame.

Number of white pixels at the edge of the image (NwE)

This technique is to compute the number of white pixels

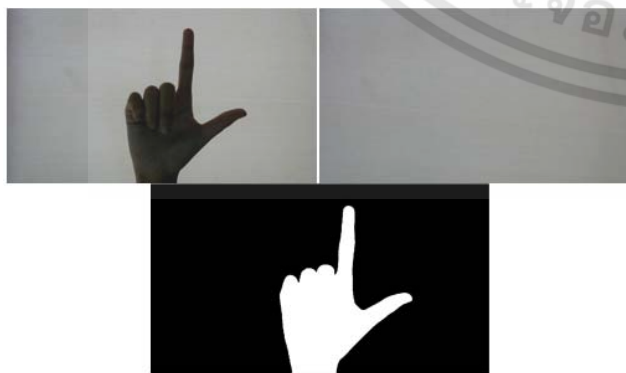


Fig. 2. Background removing

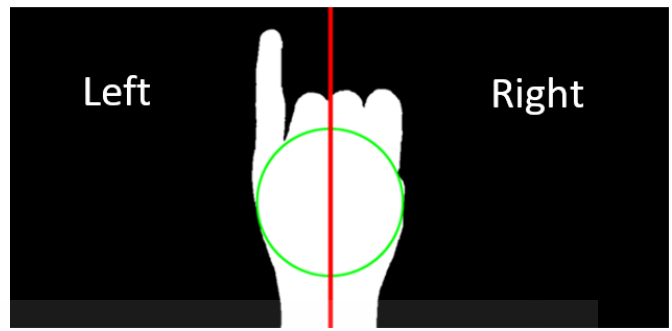


Fig. 3. Image of palm position

along the edge of binary images from both first and last frame to separate which side of the signs is received, then encode them to binary.

Finger length from the centroid point (Fcen)

First, we start to calculate the angles between fingertips from the centroid by finding the peaks from the hand contour in images and assume them as fingertips of hand. Then we calculate the angles between them from both the first and last frame.

Given (x_1, y_1) as the first peak, (x_2, y_2) as the second peak and let (cx, cy) be the centroid point. The Euclidean distance between these points are given by the Eq. 2 and Eq. 3

$$|D_1| = \sqrt{(x_1 - cx)^2 + (y_1 - cy)^2} \quad (2)$$

$$|D_2| = \sqrt{(x_2 - cx)^2 + (y_2 - cy)^2} \quad (3)$$

Finally, the angle between fingers is given by Eq.4

$$\theta = \cos^{-1}\left(\frac{|D_1 \cdot D_2|}{|D_1| \cdot |D_2|}\right) \quad (4)$$

After identifying the centroid point and angles between consecutive peaks, we continue to find all real fingertips from these values from both first and last frame. If the obtained peaks do not cover all 5 fingertips we must add or delete with these following methods to obtain 5 fingertips.

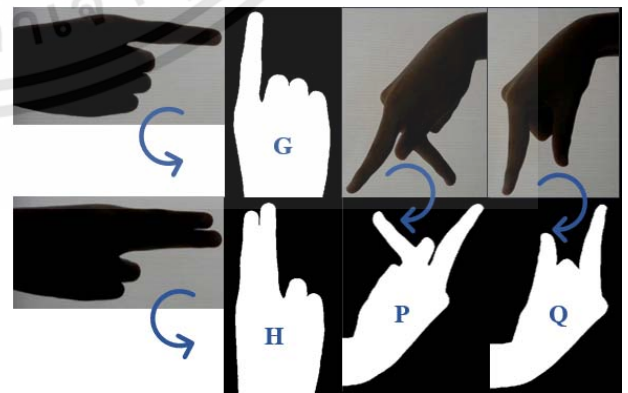


Fig. 4. Rotation of G, H, P and Q

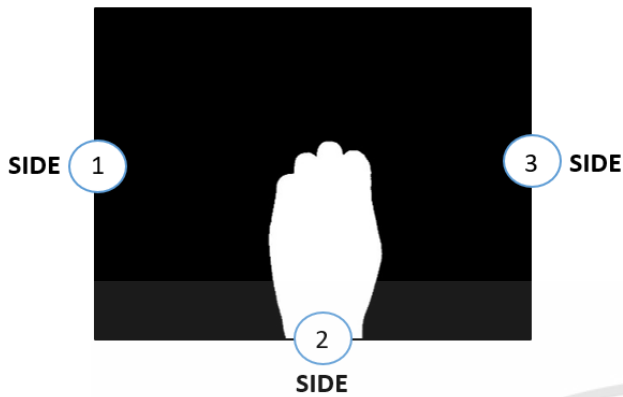


Fig. 5. Three sides of white pixel along the edge image

a.) *Adding*: If the number of peaks is less than 5 we have to compute the average angle between the peaks and use this value to examine which side has to be add, left or right, based on the idea that the left side should have 2 fingers and the right side should have 3 fingers then use the green point to represent the adding point as shown in Fig. 6.

b.) *Deleting*: If the number of peaks is greater than 5 we also have to compute the average angle between peaks and calculate the variance from the mean of each angle then examine which angle has the greatest variation and delete it as shown in Fig. 7.

Finally, we calculate the distances from the centroid point to those 5 fingertips (F_{cen}) obtained from the above methods. According to this method, some gestures may get the artificial fingertips, which always occur in closed hand group such as E, M, N, O, S and T. Although the signs may seem to similar to each other but the contour of them are different. Therefore, whether we create the unreal fingertips the F_{cen} of them are not the same as shown in Fig. 8.

Angles between fingers ($AngF$)

After obtaining exactly 5 fingertips derived from the previous section, we have to calculate the angles between fingertips again to separate the closed hand and opened hand by using the Eq. 1-Eq. 3.

Differences of angles between the first and last frame ($delAng$)

Since the American Sign Language alphabets have both dynamic (J and Z) and static (the rest) signs, thus we use this

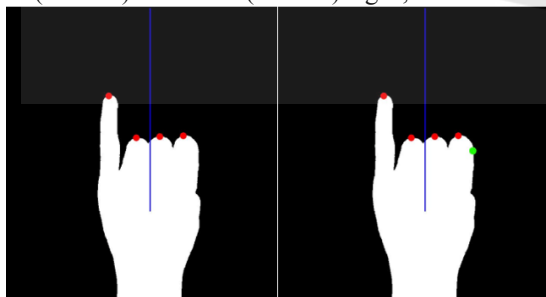


Fig. 6. Before and after adding peak

technique to separate them. The concept is whenever there is no movement in the sign, the differences of angles between the first and last frame will be close to zero. On the other hand, whenever the moving sign is detected, the differences of angles between both frames will be large.

This technique is easy to understand and practical. Therefore, our work can deal with the dynamic sign which is the problem normally occurring in using a 2D camera.

D. Classification

We use Artificial neural network (ANN) which is a computing system made up of several simple, interconnected processing elements, which process information in MATLAB program to classify the alphabets into 26 classes.

All data are divided into 3 groups: training and validation dataset for modeling and testing dataset for testing the performance of the model. Each gesture will be captured for 2 seconds and uses frame rate of 30 frames per second.

After that, the features that are analyzed from the first and last frame are taken together therefore the total features per gesture are 25, include 3 values of number of white pixels at edge of the image, 10 values of finger length from the centroid point, 8 values of angles between fingers and 4 values of differences of angles between fingers of the first and last frame.

The architecture of this network is two-layer feed-forward, with sigmoid function at hidden layer and softmax function at the output layer. Softmax function will create a competition between neurons on the same layer which means the strongest activating neuron will suppress the weaker ones, therefore the correct class will have the highest probability (close to 1) then we use the competitive layer, which consists of 26 nodes that represent each class to decide the answer. As a result, the accuracy of the recognition rate is increase.

Neuron network will be trained using conjugate- gradient backpropagation then selecting the appropriate percentage of testing, training, validation, and hidden layer parameters to optimize the neural network for best efficiency.

III. RESULTS EVALUATION AND DISCUSSION

This work was implemented using Matlab version 9.0 (R2016a) on 5th Generation Intel® Core™ i7 NVIDIA® GeForce® 940M 2G GFX @ 2.4GHz processor machine, Window 10, 8GB RAM and use Logitech HD webcam C310 with resolution 640x480



Fig. 7 Before and after deleting peak

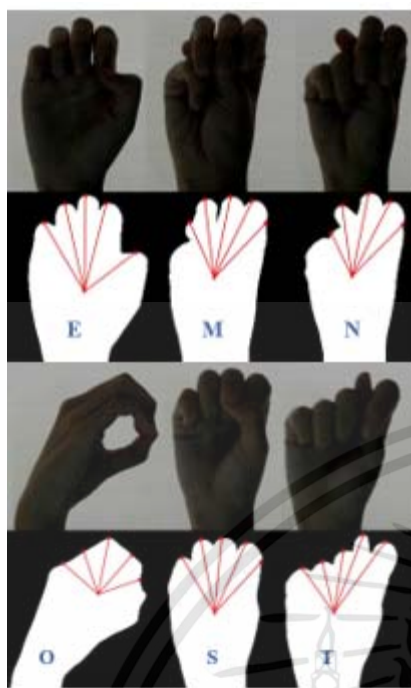


Fig. 8. Results from adding and deleting fingertips in closed hand group

In the experiments, we tested multiple times using samples from ten different users by asking them to show only the hand part in front of the dark background under the normal lighting. Based on the values of features obtained from many users, we found that the parameters should be as follow: Hidden layer = 100, Testing = 30%, Validation = 15% and Training = 55% which is the most suitable for our experiments.

The results of ASL recognition using many algorithms are shown in Table.1. The percentages of accuracy in our research were approximate of 95%, which is very high comparing with the research that was referenced. As you see the research that can recognize all the alphabets (A-Z) only has detection rate at 88.28% to 90%, moreover the remaining research can detect only some alphabets and most of them are opened finger, which obviously easier to recognize.

The alphabets with lowest recognition rate in our research are M and N because they are both very similar postures and some research referred in this paper has the same problem to recognize them. However, our recognition rate for M and N still has satisfactory results and efficient enough for usability.

IV. CONCLUSION

In this paper, a novel approach to hand recognition based on edge of binary image using 2D camera is proposed. Our algorithm uses only 4 parameters to extract features from hand, include Number of white pixels at edge of the image (NwE), Finger length from the centroid point (Fcen), Angles between fingers (AngF) and Differences of angles between fingers of the first and last frame (delAng) then ANN is used to classify the sign into 26 classes of American alphabets.

TABLE I. PERFORMANCE RESULTS

Reference	Feature extraction	Classification	Output letters	Accuracy (%)
Proposed techniques	NWE, Fcen, AngF and delAng	ANN	A-Z	95
1	Boundary tracing and Fingertips detection	NA	B, D, F, I, K, L, U, V, W	95
2	Canny edge detection	PCA, ANN	A-E	60
7	Centroid, Area of hand	Euclidean distance, Feature matching	A-Z	90
8	Average Height, Centroid and Euclidian Distance of the Centroid from the Origin	ANN	A-Z	80.28
9	Haar features and skin color segmentation	Edge orientation Histogram (EOH)	A-Z	88.26

This algorithm is fast and easy to understand in real usability, moreover the accuracy obtained from the experiments is very high although some alphabets have errors but are acceptable. Furthermore, our work can apply with fingerspelling which is the process of spelling out words by using hand shapes that correspond to the letters of the word since the fingerspelling is often used such as people's names, places, titles, organizations and brands etc.

V. FUTURE WORK

In the future work, we expect to extend our work to American Sign Language words which are dynamic hand gestures by using the same webcam. The purpose is to develop the algorithm that can be used in real communication by using the concept of inexpensive tools.

VI. REFERENCES

- [1] J. Ravkiran, M. Kavi, M. Suhas, R. Dheeraj, S. Sudheeder, P. V. Nitin, "Finger Detection for Sign Language Recognition," Proceedings of International MultiConference of Engineers and Computer Scientists (IMECS), Vol 1, Hongkong, March 18-20, 2009.
- [2] A. Sharmila Konwar, B. Sagarika Borah, C. Dr.T.Tuithung, "An American Sign Language Detection System using HSV Color Model and Edge Detection," International Conference on Communication and Signal Processing, pp. 743-746, Melmaruvathur, India, April 3-5, 2014

- [3] O. Serban, R. Christoph, S. Bochao, "Automatic static hand gesture recognition using tof cameras," 20th European Signal Processing Conference (EUSIPCO), pp. 2748-2751, Bucharest, Romania, August 27 - 31, 2012.
- [4] J. Shubham, S. Pratik, B. Avinash, H. Parag, W. Shutosh, "Review on Hand Gesture Recognition using Sensor Glove," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 5, Issue 11, pp.563-565, November 2016.
- [5] J. L. Rahejaa, A. Mishrab, and A. Chaudharyc, "Indian Sign Language Recognition Using SVM," Pattern Recognition and Image Analysis, Vol. 26, No. 2, pp. 434-441, 2016.
- [6] P.V.V.Kishore MIEEE, M.V.D.Prasad and Ch.Raghava Prasad, R.Rahul," 4-Camera Model for Sign Language Recognition Using Elliptical Fourier Descriptors and ANN", SPACES-2015, Dept of ECE, K L UNIVERSITY.
- [7] J. R. Pansare, S. H. Gawande and M. Ingle, "Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background," Journal of Signal and Information Processing, No. 3. pp. 364-367, 2012
- [8] Md. A. Rahman, Ahsan-Ul-Ambia and Md. Aktaruzzama, "Recognition Static Hand Gestures of Alphabet in ASL," International Journal of Computer and Information Technology (IJCIT), Vol. 02, Issue 01, pp.75-77, 2011
- [9] J. R. Pansare and M. Ingle, "Vision-Based Approach for American Sign Language Recognition Using Edge Orientation Histogram," International Conference on Image, Vision and Computing (ICIVC), Aug 3-5, 2016

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวอริยา ทองทวี
วัน เดือน ปีเกิด	14 ธันวาคม 2538 ที่กรุงเทพมหานคร
ที่อยู่	113 ซ.รังสิต-ปทุมธานี12 หมู่บ้านรัตนโกสินทร์ 200 ปี ตำบล ประชาธิปัตย์ อำเภอ ธัญบุรี จังหวัด ปทุมธานี 12130 โทร. 0988281668
ประวัติการศึกษา	2560 วิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	Signal and Image processing
ทุนการศึกษาที่เคยได้รับ	
พ.ศ 2558 – 2560	ทุนพานาโซนิค บัณฑิตวิศวกรรมศาสตร์ Panasonic Future Gear Scholarship
พ.ศ 2561 – 2562	ทุนเรียนต่อหลักสูตรวิศวกรรมศาสตรบัณฑิตสำหรับนักศึกษาที่ได้รับ เกียรตินิยมอันดับหนึ่ง