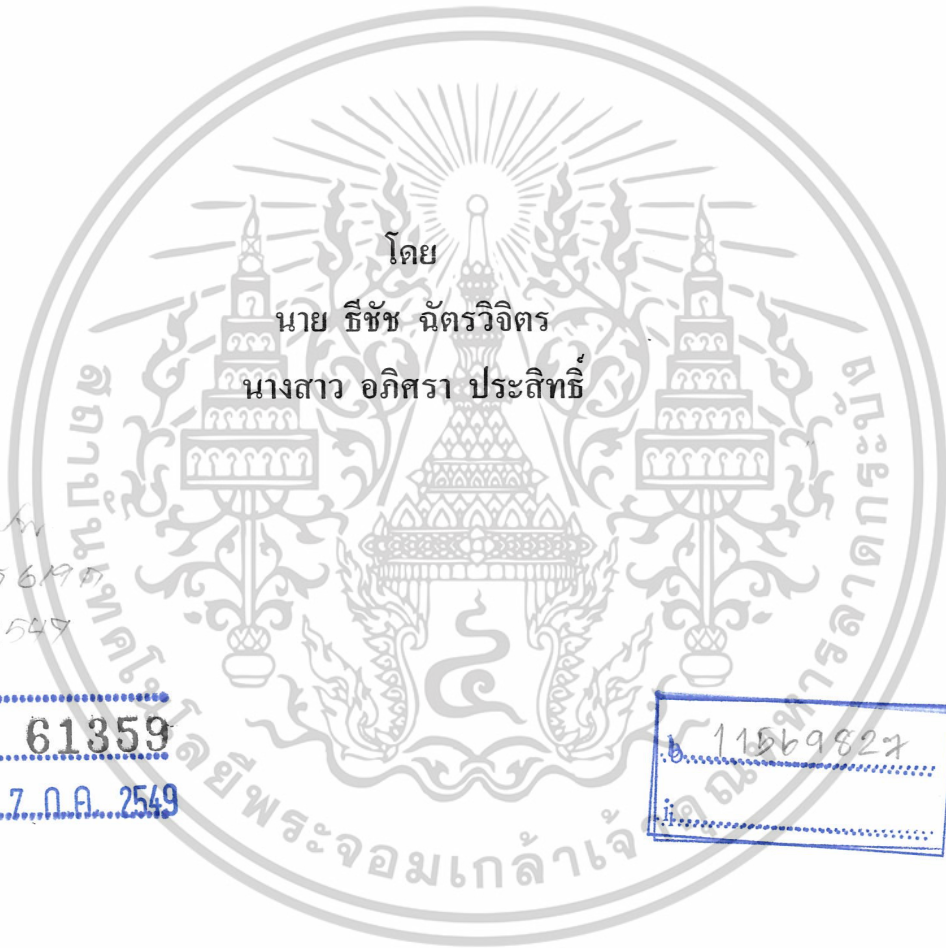


เกมไพ่ยูกิ

Yugi Card Game



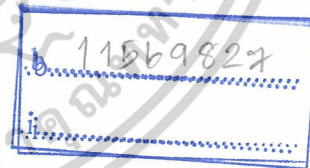
โดย  
นาย ธีชัช ฉัตรวิจิตร  
นางสาว อภิสรา ประสิทธิ์

๒/๗  
ธ 6/๙๓  
2547

เลขหมู่.....

เลขทะเบียน..... 61359

วัน,เดือน,ปี,17,ก.ค.,2549



ปริญญานิพนธ์นี้ส่วนหนึ่งของการศึกษาตามศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

## Yugi Card Game



By

Mr. Theechat Chatvijit

Miss Apisara Prasit

**A THESIS SUBMITTED IN PARTIAL OF THE REQUIREMENT FOR THE DEGREE OF  
BACHLOR IN DEPARTMENT OF INFORMATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2004**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบเสนอปริญญาบัตร

หัวข้อปริญญาบัตร	เกมไผ่ยูกิ	
ชื่อนักศึกษา	นาย ธีชัช ฉัตรวิจิตร	รหัสประจำตัว 45015848
	นางสาว อภิสรา ประสิทธิ์	รหัสประจำตัว 45015879
อาจารย์ที่ปรึกษา	รศ.ดร.ปิติเขต สุรักษา	
	อ.วิทวัส วิทยชำนานกุล	
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2547	

ปริญญาบัตรฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(รศ.ดร.ปิติเขต สุรักษา)

อาจารย์ที่ปรึกษาปริญญาบัตร

(อ.วิทวัส วิทยชำนานกุล)

อาจารย์ที่ปรึกษาปริญญาบัตรร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เกมไพ่ยูกิ		
ชื่อนักศึกษา	นาย ธีรัช	ฉัตรวิจิตร	รหัสประจำตัว 45015848
	นางสาว อภิสรา	ประสิทธิ์	รหัสประจำตัว 45015879
อาจารย์ที่ปรึกษา	รศ.ดร.ปิციเขต ผู้รักษา อ.วิทวัส วิทย์ชำนานุกุล		
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ		
ภาควิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2547		

### บทคัดย่อ

ปริญญานิพนธ์นี้ศึกษาการประมวลผลภาพ (Image Processing) โดยใช้การจับภาพด้วยกล้อง Web Camera จากนั้นนำภาพที่ได้มาผ่านกระบวนการรู้จำรูปแบบ (Pattern Recognition) โดยใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) แบบ Bidirectional Associative Memory ในการรู้จำภาพ และมีการเก็บข้อมูลเกี่ยวกับแแต้มของไฟแต่ละใบและรูปร่างของตัวละครไว้ในฐานข้อมูลอยู่ก่อนแล้ว และนำไปเขียน โปรแกรมจัดการ ในส่วนของเกมยูกิซึ่งมีกติกาแบบพื้นฐานคือการนำแแต้มมาเปรียบเทียบกันเพื่อดูผลว่าใครแพ้ใครชนะ ซึ่งเกมนี้สามารถเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องได้โดยใช้การสร้าง Socket เพื่อเชื่อมต่อและสามารถแสดงผลภาพของเกมออกมาได้ด้วยโปรแกรม Macromedia Flash MX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Thesis Title** Yugi Game

**Student** Mr. Theechat Chatvijit ID. 45015848  
Miss Apisara Prasit ID. 45015879

**Advisor** Assoc. Prof. Dr. Pitikhate Sooraksa  
Mr. Withawat Withayachumnankul

**Graduate Level** Bachelor Degree of Information Engineering

**Department** Information Engineering

**Academic Year** 2004

## ABSTRACT

This project presents a study about Image Processing and Pattern Recognition used for recognizing pictures captured from web camera. The algorithms are derived by means of Artificial Neural Networks with Bidirectional Associative Memory. Captured pictures on Yugi's card are stored in the database. The program compares Yugi's life points between two cards from the database to seek for who is the winner similar to the real Yugi's card game. This game can play on two computers linking by Socket Server and can display output on the monitors by using Macromedia Flash MX.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเป็นผลเนื่องมาจากความร่วมมือของคณะผู้จัดทำ และขอขอบพระคุณเป็นรายชื่อดังนี้ รศ.ดร. ปิติเขต สุรักษา สำหรับหัวข้อปริญญานิพนธ์ อ.วิทวัส วิทย์ชำนานุกุล และ ดร.สมเกียรติ อุดมธรรมากุล สำหรับคำแนะนำด้าน การประมวลผล ภาพ รศ.ดร. เอื้อน ปิ่นเงิน ภาควิชาวิศวกรรมคอมพิวเตอร์ สำหรับแนวคิด Decision Tree และ ทฤษฎีเกี่ยวกับ ปัญญาประดิษฐ์ ผศ.ดร. ยุทธนา คิดใจเดียว ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ ที่ให้ คำปรึกษาปัญหาต่างๆเกี่ยวกับ โครงข่ายประสาทเทียม และคำแนะนำในการใช้ Bidirectional Associative Memory ดร.นวกัก เอื้ออนันต์ ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น ที่เอื้อเพื่อ เอกสารประกอบการสอนวิชา วงจรข่ายนิเวรอนเทียม ทางอินเทอร์เน็ต ดร. สุพจน์ นิตย์สุวัฒน์ ภาควิชาครุศาสตร์คอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ดร. มนตรี กาญจนเดชา ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ และ อ.นิรุท อำนวยศิลป์ ภาควิชาวิทยาการคอมพิวเตอร์ สถาบันอินเทอร์เน็ตเทคโนโลยี สามพราน ที่ เอื้อเพื่อเอกสารประกอบการสอนวิชา Image Processing ทางอินเทอร์เน็ต อ.ณรงค์ศักดิ์ ศรีสม คณะวิทยาศาสตร์และเทคโนโลยี สถาบันราชภัฏเชียงใหม่ ที่เอื้อเพื่อเอกสารประกอบการสอนวิชา Data Communication ทางอินเทอร์เน็ต Mr. Farzad Pezeshakpour ที่เอื้อเพื่อโปรแกรม Video for MatLab ทางอินเทอร์เน็ต น้องศศิกันต์ ฉัตรวิจิตร สำหรับ Note Book ที่ใช้ในการสอบ โครงการงาน และขอขอบคุณเพื่อนๆ พี่ๆ ภาควิชาวิศวกรรมสารสนเทศ และเพื่อนๆ ภาควิชา อิเล็กทรอนิกส์ ที่มีส่วนร่วมและอำนวยความสะดวกในการจัดทำปริญญานิพนธ์นี้

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดาที่ให้การสนับสนุนการศึกษามาโดยตลอด

นายธีชัช ฉัตรวิจิตร  
นางสาวอภิศรา ประสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ฉ
สารบัญตาราง	ญ
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 แนวคิดและที่มาของปัญหา	1
1.2 จุดประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ผลที่คาดว่าจะได้รับ	1
1.5 วิธีการดำเนินงาน	2
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	<b>3</b>
2.1 ทฤษฎีการประมวลผลภาพ	3
2.2 การได้มาของรูปภาพ	9
2.3 กระบวนการประมวลผลภาพ	12
2.4 การปรับปรุงภาพ	16
2.5 การแปลงข้อมูลภาพในสองมิติ	20
2.6 การแยกข้อมูลภาพออกเป็นส่วนๆ	23
2.7 การแตกลักษณะเด่น	25
2.8 ทฤษฎีโครงข่ายประสาทเทียม	28
2.9 ทฤษฎีเกี่ยวกับ Network	66
<b>บทที่ 3 การออกแบบเกม Yugi'Oh</b>	<b>74</b>
3.1 ส่วนการออกแบบ	75
<b>บทที่ 4 ขั้นตอนการทดลองและผลการทดลอง</b>	<b>88</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุปและข้อเสนอแนะ	114
บรรณานุกรม	
ภาคผนวก ก. คู่มือและวิธีการใช้งาน	
ภาคผนวก ข. Action Script โปรแกรม Macromedia Flash MX 2004	
ภาคผนวก ค. Source Code โปรแกรม Java	
ภาคผนวก ง. Source Code โปรแกรม MatLab	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

เรื่อง	หน้า
รูปที่ 2.1 แสดงระบบสี HSV	5
รูปที่ 2.2 แสดงรูปสีในระบบ RGB และระบบ HSV	6
รูปที่ 2.3 แสดงรูปแบบแยกช่อง R , G และ B ในระบบ RGB แบบ Gray Scale	7
รูปที่ 2.4 แสดงรูปแบบแยกช่อง H , S และ V ในระบบ HSV แบบ Gray Scale	8
รูปที่ 2.5 แสดงระบบสี RGB 3 ช่อง	10
รูปที่ 2.6 แสดงผลจากการสุ่มตำแหน่งของภาพ	11
รูปที่ 2.7 แสดงผลภาพที่ผ่านการสุ่มตำแหน่งต่างกัน	11
รูปที่ 2.8 แสดงรูปดิจิทัลอินพุตในฟังก์ชัน 2 มิติ $f(x,y)$	12
รูปที่ 2.9 แสดงการกระทำแบบจุดต่อจุด	13
รูปที่ 2.10 แสดงการกระทำการเฉพาะแห่ง	14
รูปที่ 2.11 แสดงการกระทำทั้งหมด	14
รูปที่ 2.12 มาตริคขนาด 3x3	15
รูปที่ 2.13 แสดงอาเรย์และ Histogram ของภาพที่ยังไม่ปรับปรุง	18
รูปที่ 2.14 แสดงอาเรย์และ Histogram ของภาพที่ปรับปรุงแล้ว	18
รูปที่ 2.15 แสดงรูป grayscale ก่อนและหลังผ่านการ Histogram Equalization	19
รูปที่ 2.16 แสดงภาพต้นฉบับ และการใส่ 0 ลงไปในภาพ	20
รูปที่ 2.17 แสดงลักษณะการหมุนรอบแกน xy	21
รูปที่ 2.18 แสดงลักษณะการหมุนรอบแกน xy เมื่อจุด Pivot ไม่ได้ยู่ตำแหน่ง (0,0)	22
รูปที่ 2.19 แสดงผลการหมุนไฟ	22
รูปที่ 2.20 Image Histogram	24
รูปที่ 2.21 แสดงผลการ Bilevel Luminance Thresholding ไฟ	24
รูปที่ 2.22 ลักษณะทั่วไปทั้ง 2 สถานะของ Synapse	29
รูปที่ 2.23 ผู้คิดค้น โมเดล โครงข่ายประสาทเทียมตัวแรกของโลก	30
รูปที่ 2.24 แสดงค้นแบบ โครงข่ายประสาทเทียมตัวแรกของโลก ชื่อว่าแบบจำลอง McCulloch – Pitts	31
รูปที่ 2.25 เซลล์ประสาท	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาหรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า
รูปที่ 2.26 วิธีการเรียนรู้ทั่วไปสำหรับกลยุทธการเรียนรู้แบบพื้นฐาน	34
รูปที่ 2.27 การแบ่งชนิดของโครงข่ายประสาทเทียมตามกลยุทธการเรียนรู้	34
รูปที่ 2.28 การแบ่งชนิดของโครงข่ายประสาทเทียมตามวิธีการเรียนรู้	35
รูปที่ 2.29 การแบ่งชนิดของโครงข่ายประสาทเทียมตามการประยุกต์ใช้งาน	35
รูปที่ 2.30 การแบ่งชนิดของโครงข่ายประสาทเทียมตามสถาปัตยกรรมโครงข่าย	36
รูปที่ 2.31 บล็อกไดอะแกรมของ Associative Memory	44
รูปที่ 2.32 Feedforward Associative Network	45
รูปที่ 2.33 Prof. Bart Kosko	50
รูปที่ 2.34 แสดง Bidirectional Associative Memory	51
รูปที่ 2.35 ตัวอย่างสำหรับ bidirectional associative memory	60
รูปที่ 2.36 TCP/IP และ OSI	66
รูปที่ 2.37 ลักษณะการทำงานของ TCP	67
รูปที่ 2.38 ลักษณะการทำงานของ UDP	68
รูปที่ 2.39 โครงสร้างของคลาส (Class) แต่ละคลาส (Class)	70
รูปที่ 2.40 โครงสร้างการทำงานของโปรโตคอล TCP/IP	70
รูปที่ 2.41 การติดต่อระหว่าง Client และ Server	71
ในการเรียกข้อมูลเว็บผ่านทางบราวเซอร์	
รูปที่ 2.42 การติดต่อระหว่าง Client และ Server ผ่าน Socket	71
รูปที่ 2.43 การส่งข้อมูล(Broadcast) จากServer ไปให้ Client แต่ละตัว	72
รูปที่ 2.44 ขั้นตอนการติดต่อระหว่าง Client และ Server ผ่าน XML Socket	72
รูปที่ 3.1 แสดง Flowchart ของเกม Yugi'Oh	74
รูปที่ 3.2 แสดง Block Diagram ของส่วน Processing	75
รูปที่ 3.3 แสดงหน้าต่างโปรแกรม Video for MatLab	76
รูปที่ 3.4 แสดงการ Scan หาจุดภาพสีดำทางซ้ายสุดและขวาสุดของไฟ	77
รูปที่ 3.5 แสดงการ Scan หาจุดภาพสีดำทางล่างสุดและบนสุดของไฟ	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า
รูปที่ 3.6 แสดงการปรับแฉวและคอดัมน์เพื่อหาจุดทแยงมุมทั้ง 3	79
รูปที่ 3.7 แสดงการนับไปทางซ้ายจากจุดต่ำสุด(A)ไปจนคอดัมน์ไปตรงกับจุดซ้ายสุด (B) แล้วนับขึ้นจนเป็นจุดเดียวกัน	80
รูปที่ 3.8 แสดงการนับไปทางซ้ายจากจุดต่ำสุด(A)ไปจนคอดัมน์ตรงกับจุดขวาสุด แล้วนับขึ้นจนเป็นจุดเดียวกัน	81
รูปที่ 3.9 แสดงการนับไปทั้ง 4 ซ้อ วาดเป็นรูปจำลอง	82
ตามเหลี่ยมปีทาโกรัสเพื่อหามุม $\theta$	
รูปที่ 3.10 แสดงการจัดภาพส่วนที่ต้องการ	82
รูปที่ 4.1 แสดงภาพก่อนการหาขอบ , หลังการหาขอบ , thresholding 0.2 , thresholding 0.1 ตามลำดับ	88
รูปที่ 4.2 แสดง ผลรวมของ ไฟ้ 3 ไบ ที่ถ่าย 3 ครั้ง เฉพาะสีแดง สีเขียว สีน้ำเงิน ตามลำดับ	90-91
รูปที่ 4.3 แสดง ผลรวม จุดภาพ เฉพาะสีแดง ที่แบ่งเป็น 9 block	92
รูปที่ 4.4 แสดง ผลรวม จุดภาพ สีแดง สีเขียว สีน้ำเงิน ที่แบ่งเป็น 9 block ตามลำดับ	93-94
รูปที่ 4.5 แสดง Moment Invariant 14 ค่า ของไฟ้ 3 ไบ 1 ครั้ง ของช่อง R และ G ในระบบ RGB	95
รูปที่ 4.6 แสดง Moment Invariant 14 ค่า ของไฟ้ 81 ไบ 1 ครั้ง ของช่อง H และ S	96
รูปที่ 4.7 แสดง Moment Invariant order 5 ของไฟ้ 2 ไบ 7 ครั้ง ของช่อง H	97
รูปที่ 4.8 แสดง binary scale ที่มักจะให้ output ถูกต้อง	100
รูปที่ 4.9 แสดง binary scale ที่มักจะให้ output ผิด	100
รูปที่ 4.10 แสดงภาพ binary scale ที่ Thresholding จาก Hue	101
รูปที่ 4.11 แสดงผลที่ได้จากการรันโปรแกรม Java ที่ให้เครื่องนี้เป็น Server	108
รูปที่ 4.12 ฉากที่เริ่มเข้าสู่เกม	108
รูปที่ 4.13 ฉากสำหรับเลือกตัวละครที่จะเล่น	109
รูปที่ 4.14 กรณิที่การติดต่อล้มเหลว	109
รูปที่ 4.15 รอกการเชื่อมต่อกจากอีกฝ่ายหนึ่ง	109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

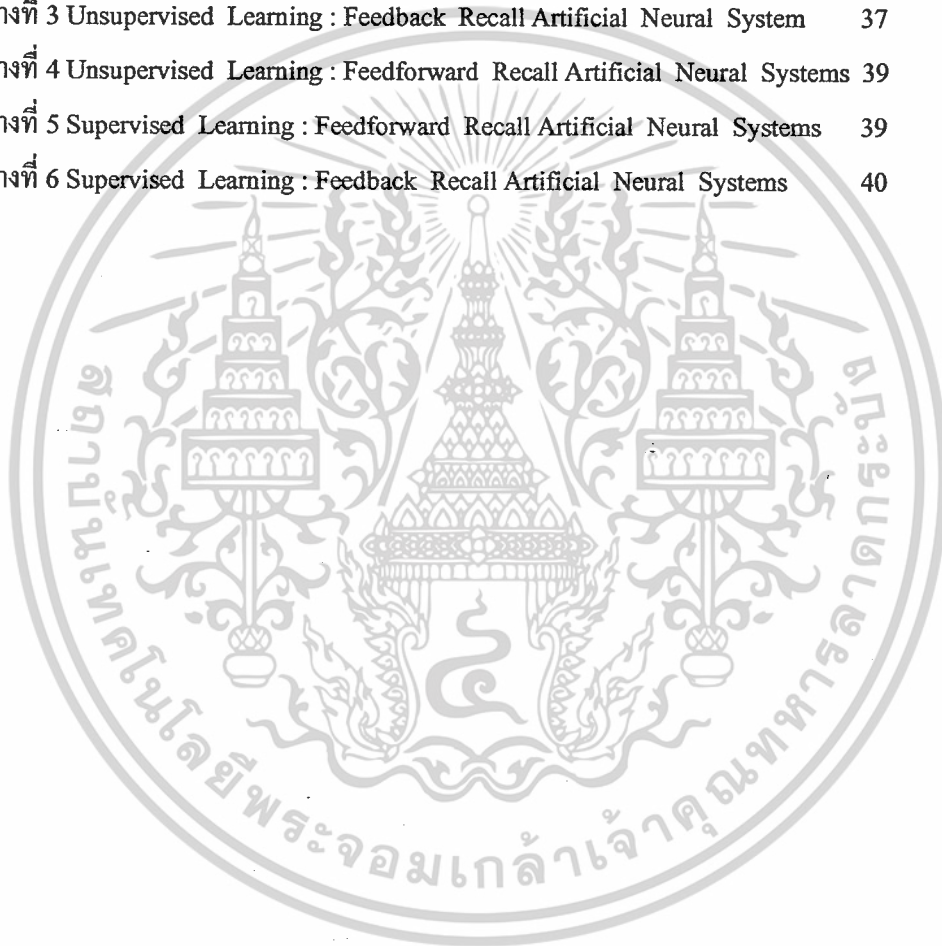
## สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า
รูปที่ 4.16 จะมีข้อความบอกให้เลือกใหม่เมื่อเลือกตัวซ้ำกัน	110
รูปที่ 4.17 ฉากการไหลคไฟที่ส่งค่ามาจากส่วนรู้จำของแต่ละฝ่าย	110
รูปที่ 4.18 ฉากรอการรับค่าจากอีกฝั่งหนึ่ง	111
รูปที่ 4.19 แสดงไฟและสถานะที่ส่งค่ามาจากแต่ละฝ่าย	112
รูปที่ 4.20 ฉากของการต่อสู้	112
รูปที่ 4.21 ฉากที่กลับมาไหลคไฟและแสดงยอดคะแนนคงเหลือ	112
รูปที่ 4.22 เป็นฉากที่บอกว่าผู้เล่นแพ้หรือชนะ	113



## สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 1 แสดง Histogram Equalization	17
ตารางที่ 2 การเปรียบเทียบ ระหว่าง โครงข่ายประสาททางชีวภาพ กับ โครงข่ายประสาทเทียม	32
ตารางที่ 3 Unsupervised Learning : Feedback Recall Artificial Neural System	37
ตารางที่ 4 Unsupervised Learning : Feedforward Recall Artificial Neural Systems	39
ตารางที่ 5 Supervised Learning : Feedforward Recall Artificial Neural Systems	39
ตารางที่ 6 Supervised Learning : Feedback Recall Artificial Neural Systems	40



### ญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 แนวคิดและที่มาของปัญหา

ในปัจจุบันวิทยาการของ การประมวลผลภาพ การจดจำรูปแบบ ได้มีการพัฒนาและประยุกต์ใช้กับเทคโนโลยีในด้านต่างๆ เช่นทางการแพทย์ ทางทหาร การพิสูจน์ตัวตน และอื่นๆ โดยใช้คอมพิวเตอร์ ช่วยในการแสดงผล เราจึงมีความคิดที่จะศึกษาในด้านการประมวลผลภาพ และ การจดจำรูปแบบ คอมพิวเตอร์กราฟฟิก เพื่อนำความรู้มาใช้ในโครงการนี้ โดยใช้กับเกมไฟ Yugi

### 1.2 จุดประสงค์

1.2.1 เพื่อศึกษาด้าน การประมวลผล ( Image Processing ) , การรู้จำรูปแบบ (Pattern Recognition ) , คอมพิวเตอร์กราฟฟิก (Computer Graphic ) และระบบเครือข่าย

1.2.2 เพื่อได้ความรู้ในการเขียนและพัฒนาโปรแกรมที่จะใช้งานต่างๆได้

1.2.3 เพื่อได้ความรู้เกี่ยวกับการสื่อสารผ่านระบบเครือข่าย

### 1.3 ขอบเขตของโครงการ

1.3.1 สามารถตรวจจับตัวละครในไฟได้ว่าเป็นตัวละครใด

1.3.2 สามารถแสดงผลเป็น 2 มิติในจอคอมพิวเตอร์ได้

1.3.3 สามารถให้ผู้เล่นเชื่อมต่อเพื่อต่อสู้กันแบบพื้นฐานระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องได้ โดยมีผลแพ้ชนะ พิจารณาจาก Life Point เฉพาะไฟ ที่เป็นตัวละครเท่านั้น

### 1.4 ผลที่คาดว่าจะได้รับ

1.4.1 สามารถนำความรู้ที่ศึกษาทางด้าน ด้าน การประมวลผลภาพ ( Image Processing ) , การรู้จำรูปแบบ (Pattern Recognition ) , คอมพิวเตอร์กราฟฟิก (Computer Graphic ) มาประยุกต์ใช้ให้เกิดประโยชน์ในด้านอื่นๆ ได้

1.4.2 ได้ความรู้ในการเขียน เครือข่ายท้องถิ่น (LAN) และพัฒนาโปรแกรมที่จะใช้งานต่างๆได้

1.4.3 เพื่อให้เกิดความบันเทิงแก่ผู้ใช้งาน โดยผู้ใช้งานจะมีอารมณ์ร่วมกับเกมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 วิธีการดำเนินงาน

### 1.5.1 ด้านการรับและจัดการกับภาพ

- ศึกษาเทคโนโลยี การประมวลผลภาพ ( Image Processing )
- ศึกษาและทดลองเขียน โปรแกรม MatLab

### 1.5.2 ด้านการรู้จำภาพ

- ศึกษาเทคโนโลยี การรู้จำรูปแบบ ( Pattern Recognition ) และ โครงข่ายประสาทเทียม

( Artificial Neural Networks )

- ศึกษาและทดลองเขียน โปรแกรม MatLab

### 1.5.3 ออกแบบและทดลอง การประมวลผลภาพ ( Image Processing )

### 1.5.4 ออกแบบและทดลอง Pattern Recognition

### 1.5.5 ด้านการสื่อสารระหว่างเครื่องคอมพิวเตอร์

- ศึกษาระบบเครือข่ายท้องถิ่น ( LAN ) ที่เชื่อมต่อแบบ Client/Server โดยอาศัยการทำงาน  
ของXMLSocket ที่มีใน Macromedia Flash MX ผ่าน Socket Server ที่ใช้ Java

### 1.5.6 ด้านโปรแกรมหลักในการควบคุม

- ศึกษาการเขียน ActionScript ในโปรแกรม Macromedia Flash MX

### 1.5.7 ออกแบบและทดลองเขียนโปรแกรมด้วยActionScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ทฤษฎีการประมวลผลภาพดิจิทัล ( Digital Image Processing )

การประมวลผลภาพดิจิทัล (Digital Image Processing ) จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิทัล(Digital Format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่าง ๆ ด้วยดิจิทัลคอมพิวเตอร์ได้ ในระบบของดิจิทัล อินพุตและเอาพุตของระบบจะอยู่ในรูปแบบดิจิทัลเท่านั้น

การวิเคราะห์ภาพดิจิทัล (Digital Image Analysis )จะเกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิทัล ซึ่งอินพุตของระบบจะเป็นข้อมูลภาพดิจิทัลและเอาพุตจะเป็นเครื่องหมายที่ใช้แทน ข้อมูลภาพดิจิทัลเหล่านั้น

##### 2.1.1 รูปร่างของภาพ (Image Shape)

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของ อะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ [17] สมมติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด  $M \times N$  ( $M$  แถว และ  $N$  คอลัมน์) ที่ใช้เก็บภาพขนาด  $M \times N$  จุด ( $M$  จุดในแนวนอน และ  $N$  จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image(5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

##### 2.1.2 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงินตัวอย่างระบบสีที่นิยมใช้กัน ได้แก่ ระบบ RGB (Red Green Blue) , HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงิน โดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode Ray Tube)

### 2.1.2.2 ระบบสี HSV

เป็นระบบสีอีกระบบ หรือ IHS โดย V คือ I IHS ย่อมาจาก Intensity Hue Saturation , Hue เป็นตัวแสดงถึง ส่วนสำคัญของของความยาวคลื่นกระตุ้นของสี Saturation มีความสัมพันธ์กับความบริสุทธิ์ของสี เช่น สีแดง ถ้าแดงสด  $S = 100\%$  ถ้าแดงจางๆ ( ชมพู )  $S = 50\%$  ส่วน Intensity คือ ระดับของความบริสุทธิ์ เช่น ถ้า Maximum intensity คือ ขาวล้วน Minimum intensity คือ ดำล้วน [22]

การแปลงจากระบบ RGB เป็น IHS หรือ HSV นี้ สูตรดั้งเดิมถูกคิดค้นโดย Tenebaum , Gravey , Weyl และ Wolf (1974)

$$H = \arccos \frac{0.5[(R-G) + (R-B)]}{\sqrt{(R-G)(R-G) + (R-B)(G-B)}}$$

ถ้า  $B > G$ ,  $H = 360 - H$  โดย H มีหน่วยเป็น องศา

นอกจากนี้ยังมีการแปลงระบบสี RGB เป็น IHS โดย ไม่มีหน่วยเป็นองศา ( Bajon , Cattoen , Kim (1985) ) เรียกว่า Bajon's Transformation

ถ้า  $\min(R,G,B)$  คือ B

$$H = \frac{G - B}{3(R + G - 2B)}$$

ถ้า  $\min(R,G,B)$  คือ R

$$H = \frac{B - R}{3(G + B - 2R)} + \frac{1}{3}$$

ถ้า  $\min(R,G,B)$  คือ G

$$H = \frac{R - G}{3(R + B - 2G)} + \frac{2}{3}$$

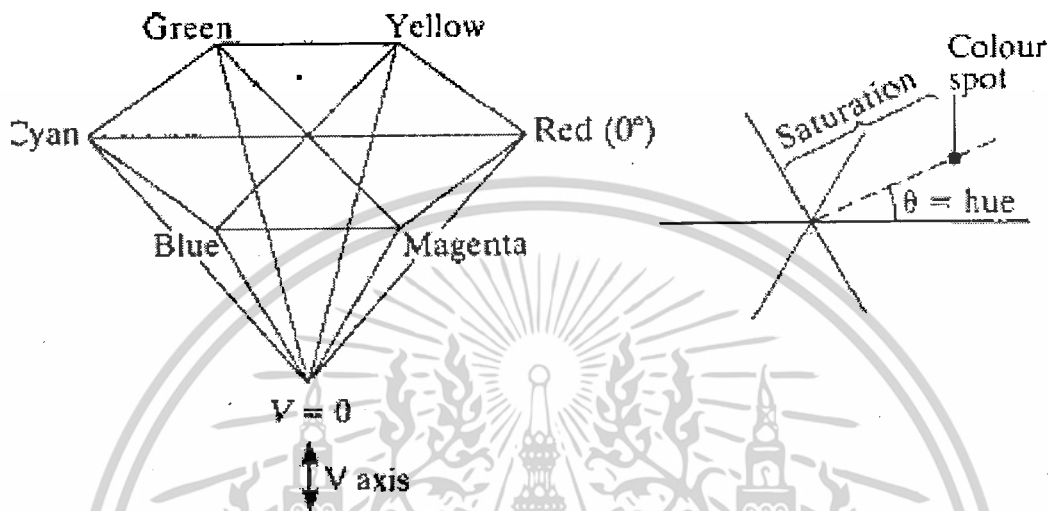
ส่วน Saturation หาได้โดย

$$S = 1 - \frac{\min(R, G, B)}{R + G + B}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Intensity หาได้โดย

$$I = \frac{R + G + B}{3} \quad \text{กรณี ความถี่ต่ำ จะได้ว่า} \quad I = R + G + B$$



รูปที่ 2.1 แสดงระบบสี HSV (ภาพจาก [17])

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้นำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0 ประโยชน์ที่สำคัญของระบบ HSV คือ H จะเหมาะกับการหาลักษณะเด่น (feature) (segmentation และ recognition) (Gagliardi, Hatch และ Sarkar (1985)) โดยทั้ง H และ S นั้นเหมาะกับการแสดงสีของวัตถุโดย ทั้งสองไม่ขึ้นอยู่กับ intensity ที่เปลี่ยนแปลงและไม่แปรปรวนกับเงา, shading, highlights (Perez และ Koch (1994))

ระบบ Machine Vision มักจะใช้ระบบสีนี้ในการพัฒนางานต่าง ๆ เช่น

Identification of colour-coded electronic components (Gordillo, 1985 ; Bajon, Cattoen and Liang 1986)

Quality control of frozen vegetable mixtures (Gagliardi, Hatch and Sarkar, 1985)

Robotic fruit harvesting (Slaughter and Harrell, 1987)

Bacteria culture manipulations (Massen, Bottcher and Leisinger, 1987)

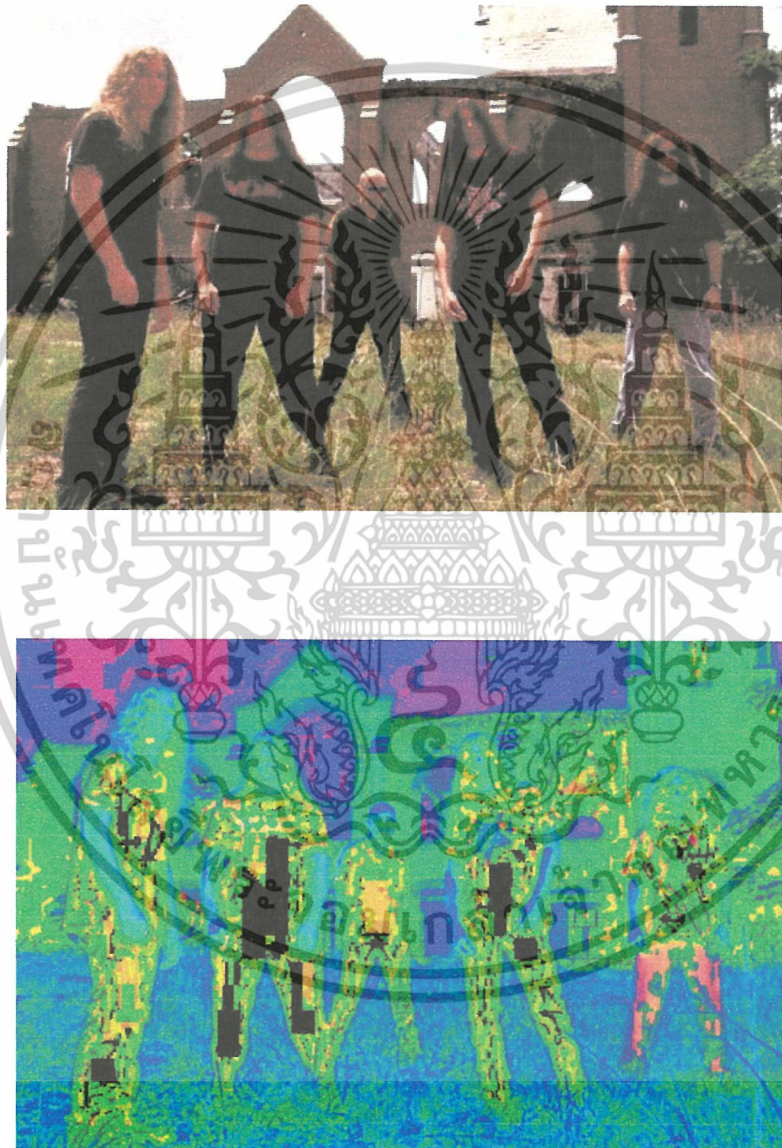
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fruit sorting and classification (Kay and De Jager, 1992)

Recognition of road landmarks (Kehtarnavaz, Griswold and Kang, 1993)

Discrimination of bottle crates (Milvang and Olafsdottir, 1993)

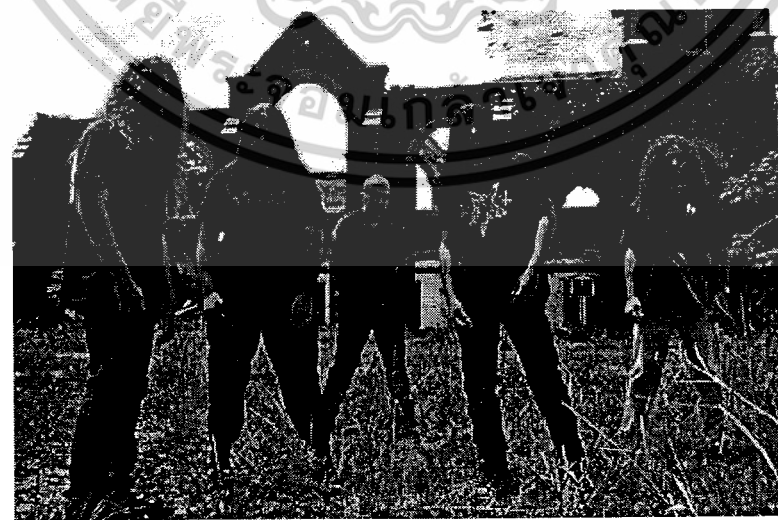
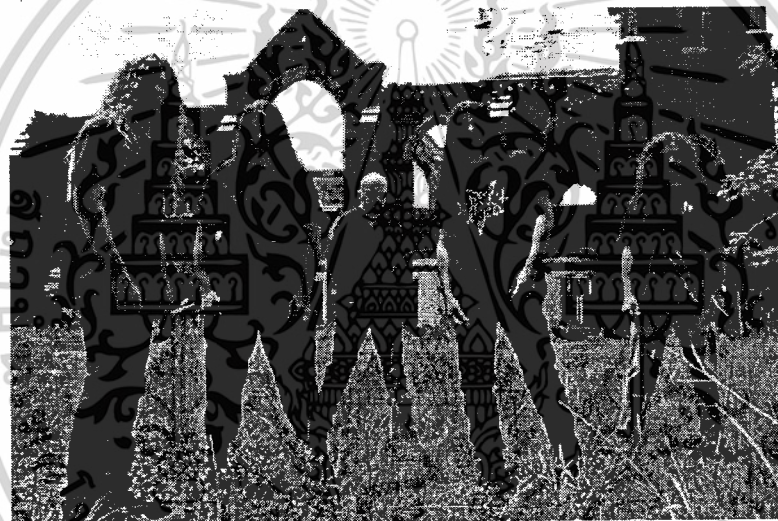
ต่อไปจะแสดงภาพเปรียบเทียบกันระหว่าง 2 ระบบ



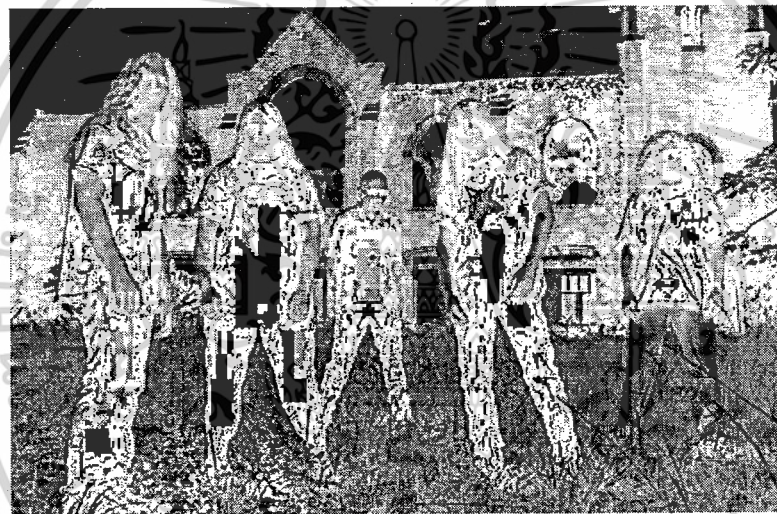
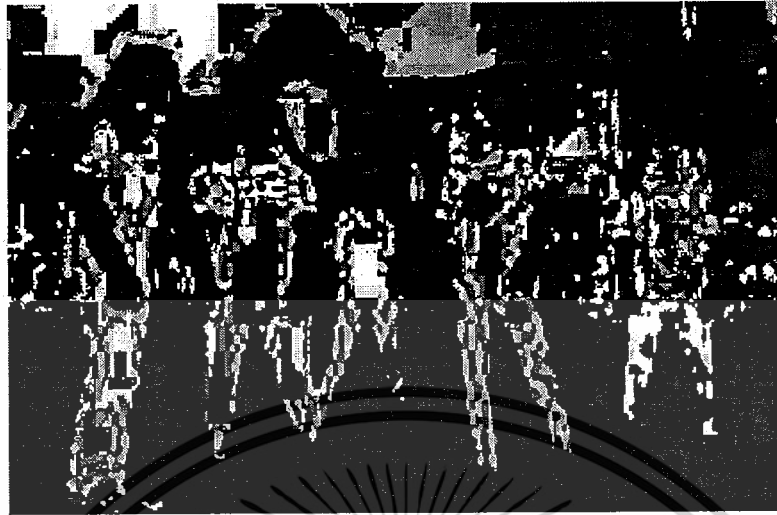
รูปที่ 2.2 แสดง รูปสีในระบบ RGB และระบบ HSV ตามลำดับ

(วง Cannibal Corpse ภาพจาก [21])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดง รูปแบบแยกช่อง R , G และ B ในระบบ RGB แบบ ระดับเทา ( Gray Scale )  
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดง รูปแบบแยกช่อง H , S และ V ในระบบ HSV แบบ ระดับเทา ( Gray Scale )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การแปลงภาพสีเป็น Grayscale

การแปลงเป็น ระดับเทา ( Gray Scale ) นั้น ในโครงการนี้ ทำเพื่อ นำไป ทำภาพ 2 ระดับ ( Binary Scale ) ต่อไป โดย แต่ละค่า  $f(x,y)$  ณ ตำแหน่งที่สมนัยกัน

$$I'(x, y) = \frac{R(x, y) + G(x, y) + B(x, y)}{3}$$

## 2.2 การได้มาของรูปภาพ ( Image Acquisition )

ส่วนมากเราจะคิดว่าส่วนมากภาพที่อยู่ในคอมพิวเตอร์จะเป็นรูปที่ได้มาจากกล้องวิดีโอ ดังเช่นรูปภาพที่เห็นอยู่ในโทรทัศน์ซึ่งในความเป็นจริงนั้นรูปภาพที่ได้มานั้นมาจากอุปกรณ์หลาย ๆ อย่าง มีวิธีการจัดเก็บที่แตกต่างกัน ดังนั้นในบทนี้จะกล่าวถึงอุปกรณ์ที่ใช้ในการนำรูปภาพเข้าสู่คอมพิวเตอร์ และวิธีจัดเก็บของแต่ละอุปกรณ์

### 2.2.1 รูปภาพที่ได้มาจากความเข้มของแสง (Intensity Images)

ความเข้มของแสงสามารถที่จะเปลี่ยนมาให้เป็นรูปภาพได้โดยใช้อุปกรณ์ทางด้านอิเล็กทรอนิกส์ที่มีการรับรู้ทางด้านแสง

#### 2.2.1.2 รูปภาพแบบสี (Color Images)

การนำภาพแบบมีสีเข้าสู่คอมพิวเตอร์จะเปรียบเหมือนการนำภาพเข้าสู่คอมพิวเตอร์ 3 รูปในเวลาเดียวกัน ซึ่งรายละเอียดขึ้นอยู่กับระบบชนิดของสี ในระบบ RGB จะใช้ความเข้มเป็น สีแดง สีเขียว และ สีน้ำเงิน ในกล้องถ่ายรูปจะทำการแยกแถบสีดังกล่าวออกจากกัน จึงทำให้ได้รูปภาพ 3 รูป เป็นของสีแดง สีเขียว และสีน้ำเงิน ตามลำดับ



รูปที่ 2.5 แสดงระบบสี RGB 3 ช่อง

### 2.2.1.3 การจับภาพ (Capture)

ผลลัพธ์ที่ได้จากกล้องวิดีโอส่วนมากจะเป็นสัญญาณในระบบ RGB ซึ่งถ้านำอุปกรณ์ดังกล่าวมาต่อกับคอมพิวเตอร์ จำเป็นต้องใช้อุปกรณ์เพิ่มเติมในการแปลงภาพดังกล่าวเข้าสู่คอมพิวเตอร์ โดยที่อุปกรณ์นั้นจะทำหน้าที่รับภาพเป็นเฟรม ๆ โดยอาจจะใช้หน่วยความจำของคอมพิวเตอร์เพื่อที่จะแปลงสัญญาณอนาลอกเป็นดิจิทัล หรืออาจมีหน่วยความจำเป็นของตนเอง ซึ่งหน่วยความจำดังกล่าวเรียกว่า Frame Buffer [17]

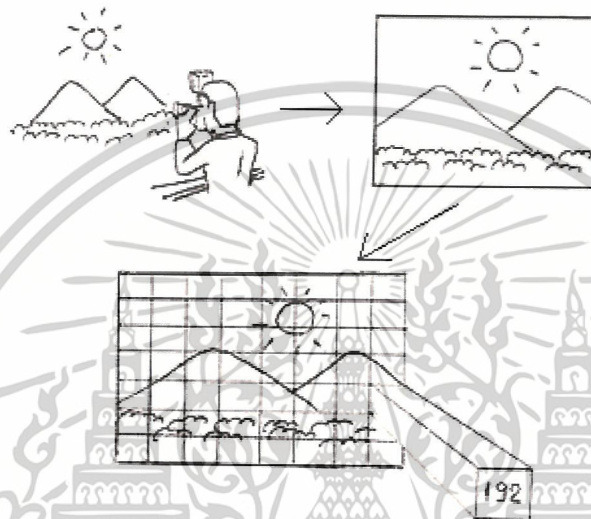
### 2.2.2 การแปลงภาพให้เป็นภาพเชิงดิจิทัล

ภาพดิจิทัลนั้นก็คือ ฟังก์ชัน 2 มิติ หรือ  $f(x,y)$  ของค่าความเข้มของแสง โดยที่  $x$  และ  $y$  คือค่าที่บอกถึงตำแหน่งในระบบพิกัดฉาก และค่าของฟังก์ชัน ณ ตำแหน่งใด ๆ จะเป็นสัดส่วนกับความสว่างของแสง ณ ตำแหน่งนั้น กระบวนการแปลงภาพให้เป็นภาพในเชิงดิจิทัลเราเรียกว่า Image Digitization มีกระบวนการ 3 ขั้นตอน คือ การบันทึกภาพ (Image Acquisition), การสุ่มเลือกจุดตำแหน่ง (Image Sampling) และ การประมาณค่าความเข้มของแสง (Image Quantization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.2.1. การสุ่มเลือกจุดตำแหน่ง (Image Sampling)

เป็นการแปลงภาพ 2 มิติที่ได้ให้เป็นภาพเชิงดิจิทัล โดยการสุ่มเลือกทางจุดตำแหน่ง หรือ Spatially Sampling โดยสุ่มเลือกเฉพาะบางตำแหน่งในภาพ ซึ่งถ้าเราสุ่มเลือกมาละเอียดภาพที่ได้ก็จะมีคุณภาพสูง หน่วยของการสุ่มเลือกก็คือ จุด หรือ Pixel นั่นเอง [6]



รูปที่ 2.6 แสดงผลจากการสุ่มตำแหน่งของภาพ (ภาพจาก[6])

เพราะฉะนั้น จึงจำเป็นที่จะต้องมีการสุ่มเลือกจุดตำแหน่ง หรือการทำ Image Sampling จากที่ได้กล่าวมานี้ ถ้าเราสุ่มเลือกจุดตำแหน่งถี่มากเท่าใด คุณภาพของภาพที่ได้ก็จะดีขึ้นเมื่อขยายเท่า ๆ กัน ดังรูปต่อไปนี้



รูปที่ 2.7 แสดงผลภาพที่ผ่านการสุ่มตำแหน่งต่างกัน (ภาพจาก[6])

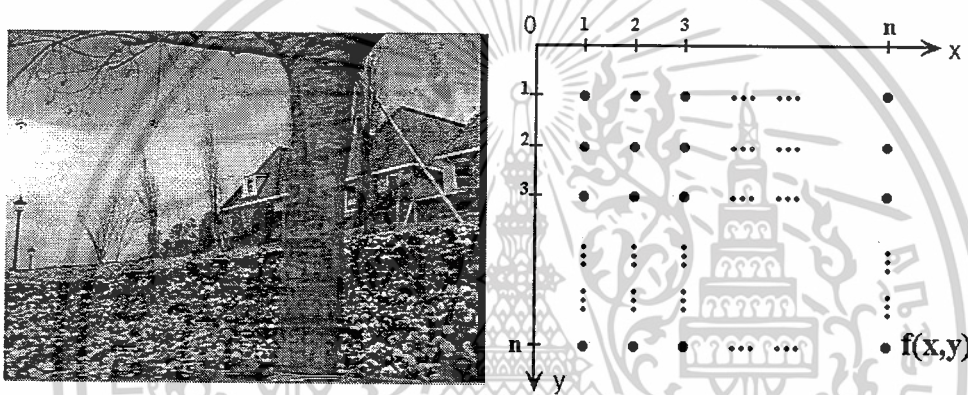
จากรูปทั้ง 3 รูป จะเห็นได้ว่า ภาพ ๆ เดียวกัน แต่เมื่อทำการสุ่มเลือกทางจุดมาไม่เท่ากัน ขยายออกมา คุณภาพของภาพที่ได้จึงต่างกัน ภาพที่ 1 สุ่มเลือกมา 128x128 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสุ่มเลือกทางตำแหน่งนี้ ถ้าในระยะความกว้างและความสูงของภาพ สุ่มมาละเอียดมาก ๆ เช่น สุ่มมา 128 จุด คุณภาพของภาพก็จะดีขึ้น แต่ก็ต้องใช้เนื้อที่ในการเก็บข้อมูลภาพมากขึ้น และขนาดของภาพก็จะมากขึ้นด้วย

### 2.2.3 ภาพเชิงดิจิทัล (Digital Image) คือ

ฟังก์ชัน 2 มิติ  $f(x,y)$  ของค่าความเข้มของแสง โดยที่  $x$  และ  $y$  คือค่าแสดงตำแหน่งในระบบพิกัดฉากค่าของฟังก์ชัน  $f$  ณ ตำแหน่ง  $(x,y)$  จะเป็นสัดส่วนกับความสว่างของแสง ณ ตำแหน่งนั้น [25]



รูปที่ 2.8 แสดงรูปดิจิทัล ในฟังก์ชัน 2 มิติ  $f(x,y)$  (ภาพจาก [25])

## 2.3 กระบวนการประมวลผลภาพ

กระบวนการประมวลผลภาพ มี 2 วิธี ใหญ่ๆ คือ

### 2.3.1 กระบวนการในโดเมนความถี่ (frequency domain)

กระบวนการนี้จะเป็นการนำการแปลงฟูริเยร์ มาประยุกต์ใช้ 3 ขั้นตอนดังนี้

- นำภาพมาผ่านการแปลงฟูริเยร์
- นำผลการแปลงฟูริเยร์ มาทำการใดๆ ตามต้องการ
- แปลงฟูริเยร์กลับ

กระบวนการในโดเมนความถี่นี้ สามารถนำมาใช้ประโยชน์ได้หลายอย่าง ซึ่งเร็วกว่า ในโดเมนระนาบ ซึ่งสามารถศึกษาได้จากหนังสือ Digital Image Processing

### 2.3.2 กระบวนการในโดเมนระนาบ (spatial domain)

เป็นการกระทำต่อจุดภาพโดยตรง ในรูปแบบของคณิตศาสตร์ กระทำกับภาพง่ายกว่าแบบโดเมนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

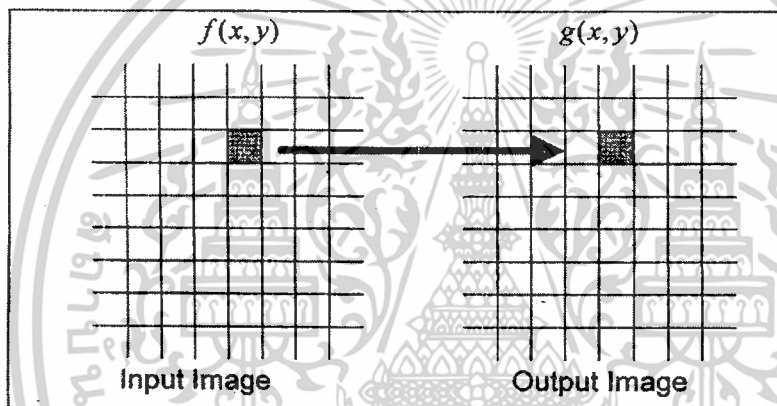
ทั้ง 2 กระบวนการ นี้ สามารถใช้ได้ในเป้าหมายเดียวกัน เช่น การกรองภาพ สามารถทำได้ทั้ง 2 กระบวนการ ในโครงการนี้ใช้กระบวนการในโดเมนระนาบเท่านั้น

### 2.3.3 ประเภทการกระทำภาพ

การกระทำภาพสามารถแบ่งออกได้เป็น 3 ประเภท คือ [8]

**2.3.3.1 การกระทำแบบจุดต่อจุด ( point operator )** ค่าความเข้มแสงของแต่ละพิกเซลของภาพผลลัพธ์ ขึ้นอยู่กับค่าความเข้มแสงของพิกเซลในภาพนำเข้า ณ ตำแหน่งที่สมนัยกัน ลักษณะการกระทำแบบนี้ ได้แก่ การปรับความสว่าง ความคมชัด การบวก ลบ คูณ ทหาร ภาพ ต่างๆ ( spatial domain )

$$g(x_i, y_i) = \tau[f(x_i, y_i)] \text{ เมื่อ } \tau \text{ เป็นการกระทำภาพใดๆ}$$

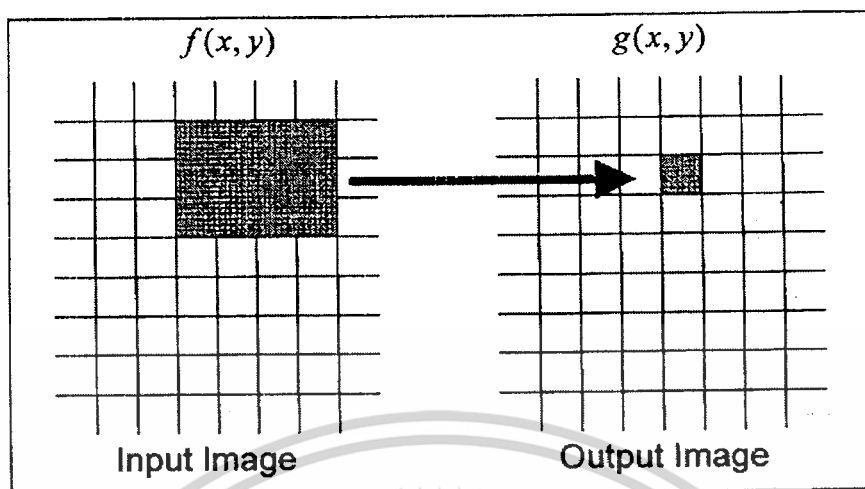


รูปที่ 2.9 แสดงการกระทำแบบจุดต่อจุด (ภาพจาก [8])

**2.3.3.2 การกระทำเฉพาะบริเวณ (Local Operations)** สำหรับการกระทำแบบนี้อัตราความเข้มแสงของพิกเซลแต่ละจุดในภาพผลลัพธ์จะขึ้นกับค่าความเข้มแสงของกลุ่มพิกเซลที่อยู่ในบริเวณเดียวกัน (Neighborhood Pixels) ในภาพนำเข้า ลักษณะการกระทำภาพประกอบประเภทนี้ได้แก่ การหาขอบ (Edge Detection) การกรองสัญญาณใน โดเมนระยะทาง (Special Filtering) เป็นต้น

ถ้า  $f(x,y)$  และ  $g(x,y)$  เป็นภาพนำเข้าและผลลัพธ์ตามลำดับ ค่าของพิกเซล  $g(x,y)$  จะมีค่าดังนี้

$$g(x_i, y_i) = \tau[\text{neighborhood of } f(x_i, y_i)]$$

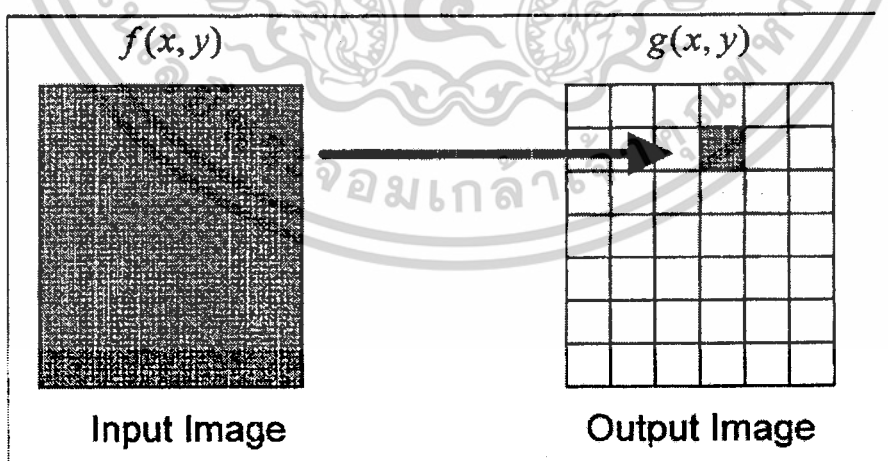


รูปที่ 2.10 แสดงการกระทำเฉพาะแห่ง (ภาพจาก[8])

2.3.3.3. การกระทำทั้งหมด (Global Operations) การกระทำแบบนี้ ค่าความเข้มแสงในแต่ละพิกเซลของภาพผลลัพธ์ (Output Image) จะขึ้นกับค่าความเข้มแสงของพิกเซลทุกตัวในภาพนำเข้า ดังรูปที่ 3 ลักษณะการกระทำภาพประเภทนี้ได้แก่ การเทรชโฮลดิ้ง (Thresholding) การทำฮิสโตแกรม (Histogram) เป็นต้น

ถ้า  $f(x,y)$  และ  $g(x,y)$  เป็นภาพนำเข้าและภาพผลลัพธ์ตามลำดับ ซึ่งค่าของพิกเซล  $g(x,y)$  จะมีค่าดังนี้

$$g(x_i, y_i) = \tau[f(x_i, y_i) \text{ for all } i]$$



รูปที่ 2.11 แสดงการกระทำทั้งหมด (ภาพจาก[8])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.4 การคูณประสานหรือการคอนโวลูชัน (Convolution)

การคอนโวลูชันในเชิงการประมวลผลภาพดิจิทัล จะเป็นการกระทำระหว่างภาพนำเข้า  $f(x,y)$  ที่มีขนาด  $N \times N$  เมื่อ  $N$  เป็นเลขจำนวนเต็มใด ๆ กับมาสก์ (Mask)  $m(x,y)$  ซึ่งจะเป็นภาพที่มีขนาด  $M \times M$  เมื่อ  $M$  เป็นเลขจำนวนเต็มใด ๆ (ปกติจะเป็นเลขคี่) และมีขนาดน้อยกว่า  $N$  มาก ๆ เช่น  $3 \times 3$   $5 \times 5$   $7 \times 7$   $9 \times 9$  หรือ  $11 \times 11$  เป็นต้น ผลลัพธ์ความเข้มแสงใหม่ที่ได้จากการทำคอนโวลูชันจะถูกเก็บไว้ในภาพผลลัพธ์  $g(x,y)$  ซึ่งขั้นตอนการทำคอนโวลูชันจะเป็นไปตามสมการ [8]

$$g(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f(i,j)m(x-i,y-j)$$

จากสมการจะเห็นว่าการทำงานคอนโวลูชันจะมีวิธีการเป็นขั้นตอนดังนี้

1. กลับ (Flip) มาสก์เทียบกับตำแหน่งจุดตรงกลางของมาสก์ เมื่อใช้มาสก์มีขนาดเป็นเลขจำนวนเต็มคี่ ซึ่งโดยปกติมาสก์ที่ใช้ในการประมวลผลภาพดิจิทัลจะมีลักษณะสมนัยทั้งในแนว  $x$  และ  $y$  ทำให้มาสก์ที่ได้จะมีลักษณะเหมือนเดิม ดังนั้นขั้นตอนนี้จึงมักจะถูกละไว้และการกำหนดมาสก์ที่ได้จะกำหนดในลักษณะที่เป็นมาสก์หลังจากการกลับแล้ว

2. เลื่อนมาสก์ให้ไปทับภาพนำเข้า โดยให้จุดตรงกลางของมาสก์ตรงกับพิกเซล  $f(x,y)$

3. ทำการคูณค่าความเข้มแสงระหว่างพิกเซลของมาสก์และพิกเซลของภาพนำเข้าที่ ณ ตำแหน่งเดียวกันแบบจุดต่อจุด

4. บวกผลคูณที่ได้ในขั้นตอนที่ 3 ทั้งหมด และนำผลลัพธ์ที่ได้ไปใส่เป็นค่าความเข้มสว่างของภาพผลลัพธ์ ณ ตำแหน่งเดียวกับพิกเซลของภาพนำเข้า นั่นคือ  $g(x,y)$

5. เลื่อนตำแหน่งไปที่พิกเซลถัดไปของภาพนำเข้า และทำซ้ำตั้งแต่ขั้นตอนที่ 2 ถึง 4 จนกระทั่งครบพิกเซลทุกตัวในภาพนำเข้า

ตัวอย่างเช่น ถ้ามีมาสก์

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$$

รูปที่ 2.12 มาสก์ขนาด  $3 \times 3$

เมื่อนำมาสก์นี้ไปทาบทิกเซล  $p$  ณ ตำแหน่ง  $(x,y)$  ของภาพนำเข้า  $f(x,y)$  ผลของการทำคอนโวลูชัน จะได้ค่าความเข้มแสงของทิกเซล  $q$  ณ ตำแหน่งเดียวกันของภาพผลลัพธ์  $g(x,y)$  มีค่าดังนี้

$$q = g(x,y) = Af(x-1,y-1)+Bf(x-1,y)+Cf(x-1,y+1)+Df(x,y-1)+Ef(x,y)+Ff(x,y+1),$$

$$Gf(x+1,y-1)+Hf(x+1,y)+If(x+1,y+1)$$

จากนั้นมาสก์จะเลื่อนไปที่ตำแหน่งถัดไปทางขวามือ นั่นคือ  $f(x,y+1)$  โดยจะทำไปเรื่อย ๆ จนครบทุกทิกเซลในภาพนำเข้า

จากนั้นขั้นตอนการทำคอนโวลูชันจะเห็นได้ว่า ถ้าเราต้องการให้มาสก์ทุกตัวทาบทลงบนภาพนำเข้าได้พอดี เราจะต้องข้ามไม่ทำการคอนโวลูชันทิกเซลในแถวแรก ๆ หรือแถวท้าย ๆ และหลักแรก ๆ หรือหลักท้าย ๆ ซึ่งจำนวนที่หายไปจะขึ้นกับขนาดของมาสก์ ตัวอย่างเช่น มาสก์ขนาด  $3 \times 3$  การทำคอนโวลูชันจะไม่ทำกับทิกเซลแถวแรกและแถวสุดท้าย และในหลักแรกและหลักสุดท้าย การคอนโวลูชันเป็นพื้นฐานของการกระทำการทางการประมวลผลดิจิทัลหลายประเภทในโดเมนเวลา

## 2.4 การปรับปรุงภาพ ( Image Enhancement )

การปรับปรุงภาพคือการทำให้อภาพที่ได้มาในสภาพที่ไม่ดี เช่น มืด หรือ สว่างเกินไป โดยพิจารณาจากฮิสโตแกรม ( Histogram ) ของภาพ ซึ่ง Histogram จะเป็นตัวแสดงการกระจายของจุดภาพ ว่าภาพนี้ควรปรับปรุงอย่างไร มีหลายวิธี เช่น Linear min/max scaling และอื่นๆ จะขอกล่าวเพียง 2 วิธีที่ใช้ในโครงการคือ Histogram Equalization และ Linear min/max scaling

### 2.4.1 Histogram Equalization

เป็นการปรับปรุงภาพที่มีความคมชัดของภาพทำให้ภาพมีรายละเอียดมากขึ้น โดยทำให้ histogram ที่กระจุกอยู่เปลี่ยนไปและกระจายตัวทั่วแกน intensity ซึ่งจะทำให้ภาพมีรายละเอียดยิ่งขึ้น

Histogram Equalization มี Algorithm ดังนี้

1. นับจำนวนจุดภาพทั้งหมดคือ 1 ระดับความเข้ม เช่นจุดภาพจุดที่ 1, 5, 200, 279 มีระดับความเข้ม 7

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

จุดภาพจุดที่ 3, 4, 6, 21, 300 มีระดับความเข้ม 4

หมายความว่า ระดับความเข้ม 7 มีจุดภาพอยู่ 4 จุด

ระดับความเข้ม 4 มีจุดภาพอยู่ 5 จุด

ทำเช่นนี้จนครบทุกระดับความเข้มของภาพ

2. เรียงลำดับระดับความเข้มจากน้อยสุดไปมากที่สุดดังนั้นจำนวนจุดภาพก็จะถูกเรียงด้วย
3. ทำการบวกจำนวนจุดภาพไปเรื่อยจนครบจำนวนจุดภาพทั้งหมด
4. นำจุดภาพที่บวกได้ในแต่ละแถวมาหารด้วยจำนวนจุดภาพทั้งหมด
5. นำผลหารแต่ละข้างมาคูณด้วยระดับความเข้มสูงสุด
6. หาค่า floor และ ceiling ของผลหารนั้น ๆ ทุกตัว โดยถ้าจุดทศนิยมต่ำกว่า 5 ให้ปัดลงตั้งแต่ 5 ขึ้นไปให้ปัดขึ้นถ้าการปัดเศษปัดแล้วเท่ากันให้นำจำนวนจุดภาพมาบวกกันเป็นคำตอบ
7. นำค่าจากข้อ 6 มาสร้าง Histogram ใหม่

ตารางที่ 1 แสดง Histogram Equalization

Gray level intensity	F pixe	f	$\frac{f}{N}$	$D_m \times \left(\frac{f}{N}\right)$	Floor & ceiling
0	8	8	0.16	1.12	1
1	7	15	0.3	2.1	2
2	6	21	0.42	2.94	3
3	4	25	0.5	3.5	4
4	10	35	0.7	4.9	5
5	2	37	0.74	5.18	5
6	8	45	0.9	6.3	6
7	5	50	1	7	7

61359

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

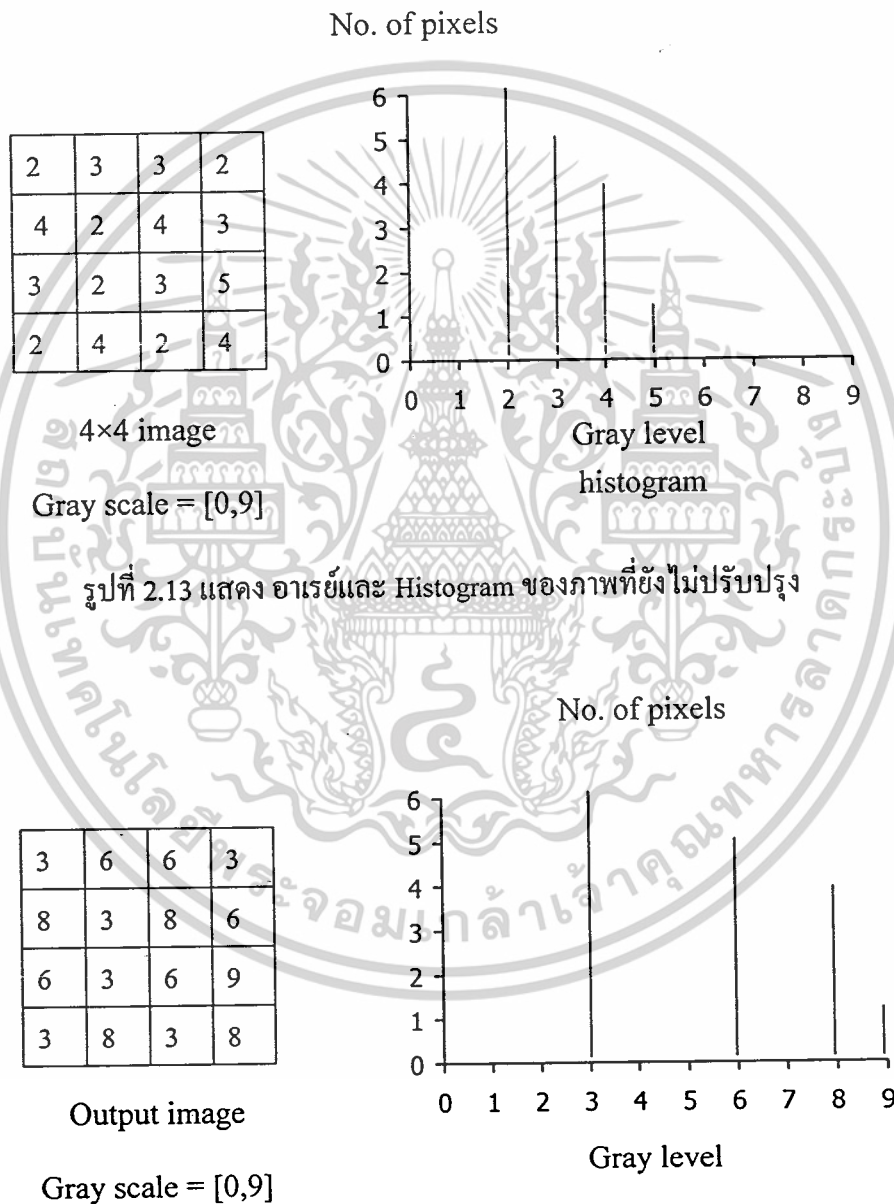
โดยที่

$F$  pixel คือ จำนวนจุดภาพ

$f$  คือ ผลรวมของจำนวนจุดภาพแต่ละจุด

$f/n$  คือ  $\frac{\text{ผลรวมของจำนวนจุดภาพแต่ละจุด}}{\text{จำนวนจุดภาพทั้งหมด}}$

$D_m$  คือ ค่า gray level intensity (ในตารางนี้คือ 7)



### Histogram Equalization

รูปที่ 2.14 แสดง อาร์เรย์และ Histogram ของภาพที่ปรับปรุงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพก่อนทำ Histogram Equalization

ภาพหลังทำ Histogram Equalization

รูปที่ 2.15 แสดง รูป Gray Scale ก่อนและหลังผ่านการ Histogram Equalization

#### 2.4.2 Linear min/max scaling

เป็นการปรับปรุงภาพแบบหนึ่ง ซึ่งทำให้ Histogram ที่กระจุกตัวกัน กระจายออก โดยขั้นแรกต้องหา จุดที่มีความสว่างน้อยที่สุดและมากที่สุด เป็นค่า min และ max มีสูตรคำนวณดังนี้

$$I'(x, y) = (I(x, y) + a) * b$$

$$a = -\min$$

$$b = \frac{255}{(\max - \min)}$$

$I'(x, y)$  เป็นภาพที่ถูกปรับปรุงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 การแปลงข้อมูลภาพในสองมิติ ( Two Dimensional Geometric Transformation )

Transformation เป็นการแปลงข้อมูลภาพซึ่งในบทนี้จะกล่าวถึงสำหรับในสองมิติเท่านั้น การแปลงภาพเป็นสิ่งจำเป็นมากสำหรับ Digital การประมวลผลภาพ ( Image Processing ) เนื่องจากเป็นกระบวนการที่จะนำไปสู่การวิเคราะห์ภาพ ( Digital Image Analysis )

### 2.5.1 การแปลงข้อมูลภาพแบบพื้นฐาน

การแปลงข้อมูลภาพพื้นฐาน ( Basic Transformation ) กล่าวเพียง 2 ชนิดคือ

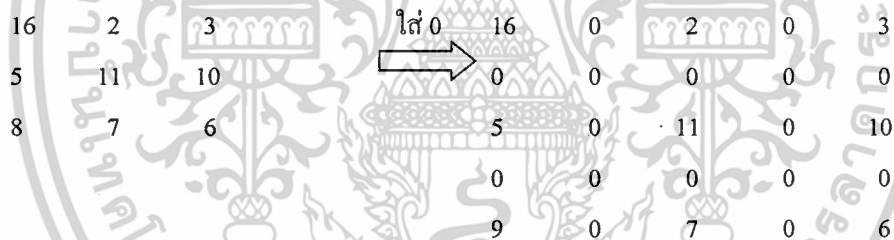
- การย่อและขยายภาพ ( Scaling )
- การหมุนภาพ ( Rotation )

#### 2.5.1.1 การย่อและขยายภาพ ( Scaling )

##### 2.5.1.1.1 Enlargement by spatial filtering

ขั้นตอนแรก ใส่ 0 ลงไปในภาพ

รูปต้นฉบับ



รูปที่ 2.16 แสดงภาพต้นฉบับ และ การใส่ 0 ลงไปในภาพ

จากนั้นนำภาพที่ใส่ 0 แล้ว ไป Convolution กับ

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

จะเห็นว่าภาพมีขนาดเป็น 2 เท่า ถ้าต้องการขยายภาพ 4 เท่า ดังนั้นต้อง ทำเช่นนี้ 2 ครั้ง

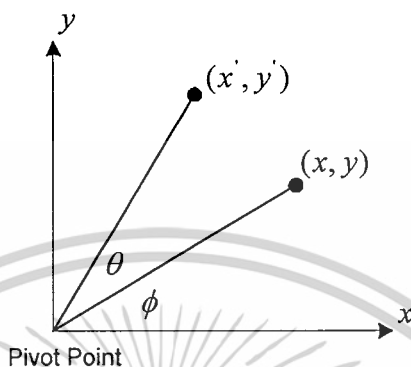
##### 2.5.1.1.2 Scaling Smaller

เป็นการย่อภาพโดยใช้การลบจุดภาพสลับกันไป เรียกว่า subsampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.1.2 การหมุนภาพ(Rotation)

เป็นการหมุนตำแหน่งของภาพในระนาบ xy รอบจุด Pivot Point(จุดหมุน) [17]



รูปที่ 2.17 แสดงลักษณะการหมุนรอบแกน xy

จากรูปที่ 2.17 จะได้ว่า

$$x = r \cos(\phi) \quad (1)$$

$$y = r \sin(\phi) \quad (2)$$

และ

$$x' = r \cos(\phi + \theta) = r(\cos \phi \cos \theta - \sin \phi \sin \theta) \quad (3)$$

$$y' = r \sin(\phi + \theta) = r(\sin \phi \cos \theta + \cos \phi \sin \theta) \quad (4)$$

เพราะฉะนั้นจากสมการที่ (3) และ (4) จะได้สมการของการหมุนรอบจุด Pivot Point ดังนี้คือ

$$x' = x \cos(\theta) - y \sin \theta$$

$$y' = x \sin(\theta) + y \cos \theta$$

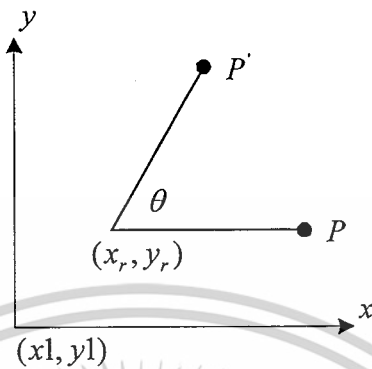
ซึ่งสามารถเขียนให้อยู่ในรูปแบบของเมตริกได้ มีลักษณะดังนี้คือ  $P' = R.P$  เมื่อ

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad P = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{และ} \quad R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การหมุนภาพเมื่อจุด Pivot ไม่ได้อยู่ที่จุด Origin

พิจารณาเมื่อจุด Pivot ไม่ได้อยู่ในตำแหน่ง  $(0,0)$  (ย้ายไปอยู่ที่ตำแหน่ง  $(x_r, y_r)$ )



รูปที่ 2.18 แสดงลักษณะการหมุนรอบแกน  $xy$  เมื่อจุด Pivot ไม่ได้อยู่ในตำแหน่ง  $(0,0)$

วิธีการในการหมุนภาพเมื่อจุดหมุนไม่ได้อยู่ที่จุด Origin สามารถทำได้ดังนี้คือ

1. ทำการเปลี่ยนจุด Pivot ไปยังจุด Origin

$$x1 = x - x_r$$

$$y1 = y - y_r$$

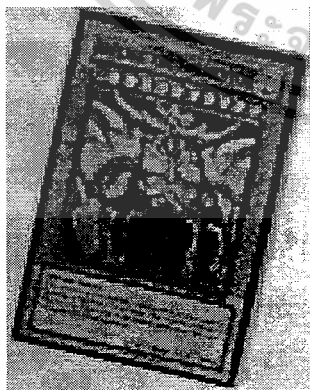
ทำการหมุนรอบจุด Origin

2. ย้ายกลับไปยังจุดเดิม โดยการบวกด้วย  $x_r$  และ  $y_r$

3. สมการการหมุนรอบจุด Pivot ใด ๆ ที่ไม่ใช่จุด Origin มีลักษณะดังนี้คือ

$$x' = (x - x_r) \cos \theta - (y - y_r) \sin \theta + x_r$$

$$y' = (x - x_r) \sin \theta + (y - y_r) \cos \theta + y_r$$



รูปที่ 2.19 แสดงผลการหมุนไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การแยกข้อมูลภาพออกเป็นส่วน ๆ ( Image Segmentation )

การทำการแยกส่วน (Segmentation) จะทำให้สามารถแยกข้อมูลภาพของส่วนที่ต้องการออกมาได้ (ข้อมูลที่มีลักษณะเหมือนกับข้อมูลตัวอย่าง) วิธีการพื้นฐานสำหรับการ Segmentation คือการพิจารณาแอมพลิจูดของภาพ (Image amplitude) (ได้แก่การพิจารณาความสว่างของภาพ สำหรับภาพแบบ Gray scale และความแตกต่างของสีสำหรับภาพสี) การ Segmentation มีหลายวิธี เช่น Amplitude segmentation methods, Region segmentation methods และ Boundary detection [17] จะขอกกล่าวแค่ 1 วิธีที่ใช้งาน คือ Amplitude segmentation methods

### 2.6.1 Amplitude Segmentation Methods

สำหรับการ Segmentation ในหัวข้อนี้จะเป็นการพิจารณาความเข้มของจุดต่าง ๆ ภายในภาพ (pixel) ซึ่งผลของการ segment จะขึ้นอยู่กับวิธีการ Threshold ของส่วนประกอบที่เป็นความเข้มหรือสีของภาพ ซึ่งการ Thersholding มีอยู่หลายวิธีด้วยกัน จะขอกกล่าวแค่ 1 วิธีที่ใช้งาน คือ Bilevel Luminance Thresholding

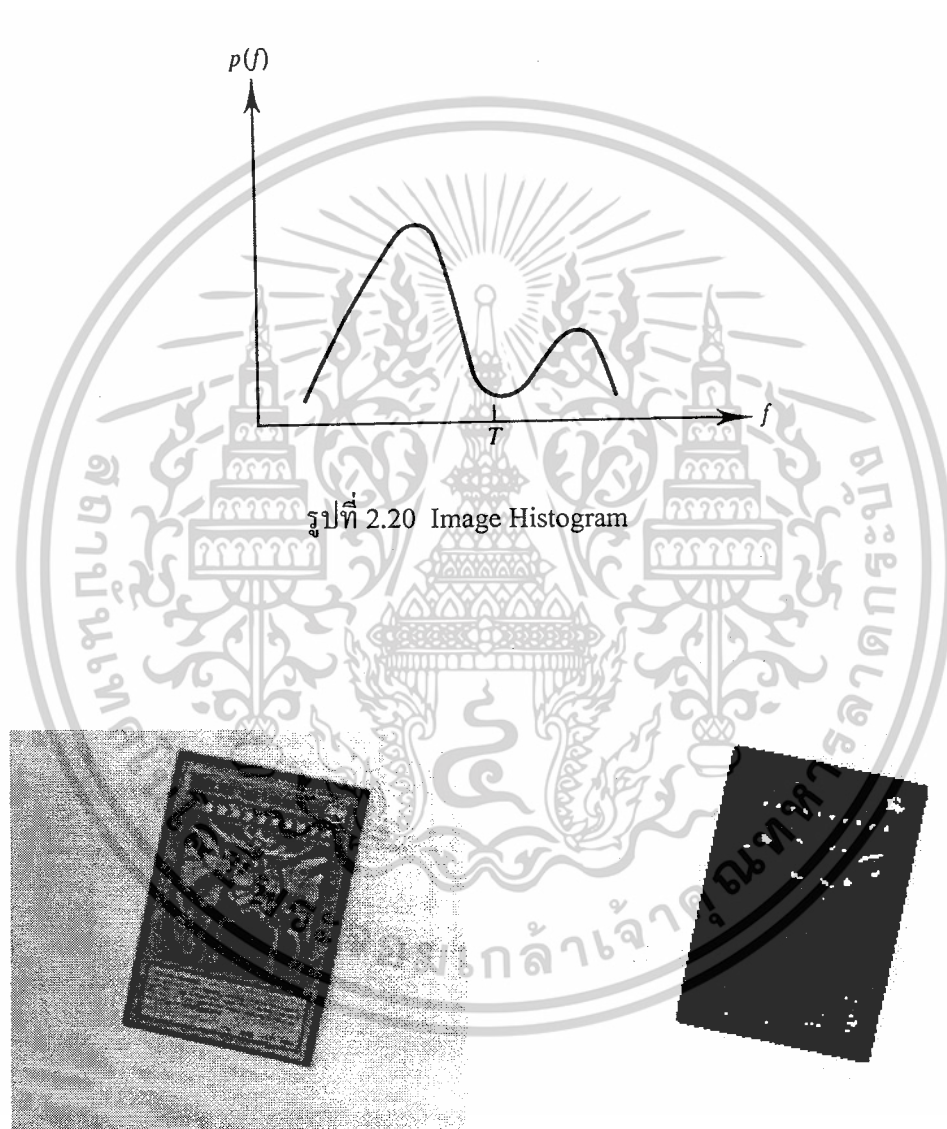
#### 2.6.1.1 Bilevel Luminance Thresholding

สำหรับภาพบางชนิดจะมีลักษณะวัตถุที่เราสนใจซึ่งมีความเข้มที่คงที่เมื่อเทียบกับพื้นหลังตัวอย่างได้แก่ ภาพของตัวอักษร (Text) เป็นต้น ซึ่งภาพเหล่านี้จะมีความเข้มของวัตถุที่เราสามารถแยกออกพื้นหลังได้อย่างชัดเจน (มีความเข้มขึ้นสองระดับได้แก่ความเข้มของวัตถุและความเข้มของพื้นหลัง)

การทำการ Segmentation สามารถทำได้โดยการกำหนดค่า Threshold ซึ่งเป็นค่าความเข้มให้มีค่าที่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้ตัวอย่างเช่น ภาพของตัวอักษรที่มีความเข้มของตัวอักษรเป็น 0 (สีดำ) และมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่า Threshold จึงควรมีค่าเท่ากับ 128 เพื่อที่จะให้สามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่า Threshold จะขึ้นอยู่กับ Histogram ของภาพ ตามรูปที่ 2.16 แสดงการหาค่า Threshold โดยค่า Threshold ควรที่จะเลือกค่า Histogram ที่อยู่ที่จุดต่ำสุดที่อยู่ระหว่างจุดสูงสุด (peaks)

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

เมื่อ  $g(x,y)$  เป็นข้อมูลภาพ ณ ตำแหน่งที่  $x,y$   
 $T$  เป็นค่า Threshold



รูปที่ 2.20 Image Histogram

รูปที่ 2.21 แสดงผลการ Bilevel Luminance Thresholding ไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 การแตกลักษณะเด่น ( Feature Extraction )

การแตกลักษณะเด่น มีด้วยกันหลายวิธี เช่น Contour Following , Fourier Descriptor และอื่นๆ ที่ใช้ในโครงการนี้ คือ Moment Invariant

### 2.7.1 Moment Invariant

โมเมนต์ระยะทาง (Spatial Moments)

จากทฤษฎีความน่าจะเป็น Moment ของฟังก์ชันความหนาแน่นของความน่าจะเป็นร่วม (Joint Probability Density Function)  $P(x,y)$  นิยามได้ดังนี้ [3] [14]

$$m_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^i y^j p(x,y) dx dy$$

Central Moment นิยามได้ดังนี้

$$\mu_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-x')^i (y-y')^j p(x,y) dx dy$$

$x'$  และ  $y'$  คือค่าเฉลี่ยของ  $x$  และ  $y$  ตามลำดับ

จากความสัมพันธ์โมเมนต์ข้างบนได้ถูกนำมาใช้กับการวิเคราะห์รูปร่าง โดย Hu – Alt ซึ่งเป็นวิธีที่บรรยายคุณสมบัติของวัตถุในเทอมของพื้นที่ , ตำแหน่ง , ความหันเห ซึ่งค่อนข้างจะเที่ยงตรง ดังนั้นสามารถแทนฟังก์ชันความหนาแน่นของความน่าจะเป็นร่วม  $P(x,y)$  ด้วย ฟังก์ชันของภาพ  $f(x,y)$  สามารถเขียนได้ใหม่คือ

$$m_{ij} = \sum_x \sum_y x^i y^j f(x,y) \quad \text{โดย } i, j \text{ คือ } 1, 2, 3, \dots$$

Order ของ moment คือ  $i + j$

$x$  และ  $y$  คือ pixel coordinators

$f(x,y)$  คือ image function

Central Moment สามารถเขียนใหม่ได้คือ

$$\mu_{ij} = \sum_x \sum_y (x-x')^i (y-y')^j f(x,y)$$

$$\text{โดย } x' = \frac{m_{10}}{m_{00}} \quad \text{และ } y' = \frac{m_{01}}{m_{00}}$$

$x', y'$  เรียกว่าจุด Centroid เป็น parameter สำคัญในการบอกตำแหน่งวัตถุ ซึ่งเป็นจุดที่มีความสว่างมวล (Brightness Mass) บน, ล่าง ซ้ายและขวา มีขนาดเท่ากัน

$m_{00}$  คือ Zero – Order Moment มีสมการดังนี้

$$m_{00} = \sum_x \sum_y f(x,y)$$

$m_{10}, m_{01}$  คือ First – Order Moment แสดงได้ดังนี้

$$m_{10} = \sum_x \sum_y x \cdot f(x,y) ; x \text{ คูณกับ } f(x,y) \text{ แบบจุดต่อจุด}$$

$$m_{01} = \sum_x \sum_y y \cdot f(x,y) ; y \text{ คูณกับ } f(x,y) \text{ แบบจุดต่อจุด}$$

$f(x,y)$  คือความสว่างของจุดภาพหรือเท่ากับขนาดของ  $f(x,y)$  ตามแนว  $x$  และ  $y$  ตามลำดับ  
เช่น ภาพ  $f(x,y)$  มีขนาด  $20 \times 30$ ,  $x$  มีค่า 1 ถึง 20,  $y$  มีค่า 1 ถึง 30

จาก Central Moment สามารถพัฒนาเป็น Normalized

Central Moments คือ

$$\eta_{ij} = \frac{\mu_{ij}}{(\mu_{00})^\lambda}$$

$$\text{โดย } \lambda = \frac{(i+j)}{2} + 1, \quad (i+j) = 2, 3, \dots$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น Central Moment สามารถแสดงได้ดังนี้

$$\mu_{00} = m_{00}$$

$$\mu_{10} = \sum_x \sum_y (x-x')(y-y')^0 f(x,y)$$

$$\mu_{01} = \sum_x \sum_y (x-x')^0 (y-y')^1 f(x,y)$$

$$\mu_{20} = \sum_x \sum_y (x-x')^2 (y-y')^0 f(x,y)$$

$$\mu_{02} = \sum_x \sum_y (x-x')^0 (y-y')^2 f(x,y)$$

$$\mu_{11} = \sum_x \sum_y (x-x')^1 (y-y')^1 f(x,y)$$

$$\mu_{30} = \sum_x \sum_y (x-x')^3 (y-y')^0 f(x,y)$$

$$\mu_{03} = \sum_x \sum_y (x-x')^0 (y-y')^3 f(x,y)$$

$$\mu_{12} = \sum_x \sum_y (x-x')^1 (y-y')^2 f(x,y)$$

$$\mu_{21} = \sum_x \sum_y (x-x')^2 (y-y')^1 f(x,y)$$

Derived Moment Invariant แสดงได้ดังนี้

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} +$$

$$(3\eta_{21} - \eta_{03})(\eta_{21} - \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

$$\phi_6 = (\eta_{20} - \eta_{02})\{(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} +$$

$$4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} +$$

$$(3\eta_{21} - \eta_{03})(\eta_{21} - \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถนำ  $\phi$  แต่ละตัวมา take log และ take absolute ที่ต้อง take log เพื่อลด range ที่ไม่นิ่ง และ take absolute เพื่อหลีกเลี่ยงค่าที่ติดลบของ complex number [14]

Moment Invariant มีประโยชน์ของตำแหน่งสำหรับการจดจำวัตถุ ซึ่งทนต่อการแปรผันของตำแหน่ง ขนาด และความหันเห (ภาพในกระจก) โดยไม่ว่าภาพจะอยู่ตำแหน่งไหน ขนาดเท่าใด ค่าของ Moment Invariant มีค่าใกล้เคียงกันสามารถประยุกต์ใช้ในด้าน Character Recognition , Aircraft Identification และ Scene - Matching

## 2.8 ทฤษฎีโครงข่ายประสาทเทียม

### 2.8.1 Machine Learning คืออะไร

โดยทั่วไป Machine Learning คือการทำให้เครื่องจักรสามารถเรียนรู้จากประสบการณ์ , เรียนรู้จากตัวอย่าง ซึ่งการเรียนรู้จะสามารถปรับปรุงประสิทธิภาพของ Intelligent system , Machine Learning สร้างพื้นฐานสำหรับระบบที่ปรับตัวได้ สิ่งที่น่าสนใจในการสร้าง Machine Learning คือ โครงข่ายประสาทเทียม ( Artificial Neural Networks ) และ Genetic Algorithm [18]

### 2.8.2 โครงข่ายประสาทเทียมคืออะไร

โครงข่ายประสาทเทียม คือ การจำลอง Model ที่เป็นเหตุเป็นผลบนพื้นฐานของสมองมนุษย์ สมองประกอบด้วย การเชื่อมต่อกันมากมายของเซลล์ประสาท โดยส่วนประมวลผลเรียกว่า "Neural" ในสมองมนุษย์มี Neural ประมาณ  $10^{11}$  หน่วย และ  $60^{19}$  รอยต่อ (Synapse) ระหว่างตัวมัน (Shepherd and Koch, 1990) สมองจึงสามารถปฏิบัติการฟังก์ชันต่างๆ ได้เร็วกว่าเครื่องคอมพิวเตอร์ที่เร็วที่สุดในปัจจุบันนี้ [18]

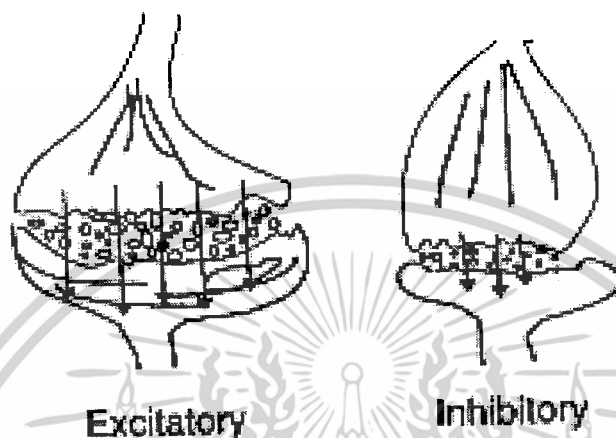
โดยทั่วไปแต่ละ Neuron มีโครงสร้างง่ายมากๆ แต่เมื่อมันมีจำนวนมากมหาศาลจึงทำให้มีประสิทธิภาพ โครงข่ายประสาทประกอบด้วย soma , dendrite , และ axon โดยที่ dendrite ทำหน้าที่เป็นตัวรับสัญญาณเข้าโครงข่ายฯ ตัว axon ทำหน้าที่ส่งผ่านสัญญาณจาก soma ไปยัง soma ตัวอื่น ตัว soma ทำหน้าที่เป็นตัวประมวลผลโดยรับสัญญาณมาจากหลายๆ axon [18]

สัญญาณถูกแพร่โดยกระบวนการไฟฟ้าเคมี ซึ่ง synapse จะเป็นตัวเปลี่ยนศักย์ไฟฟ้าในตัว cell เมื่อศักย์ไฟฟ้ามีค่ามากกว่าค่าค่าหนึ่ง (กระตุ้น) มันจะถูกส่งต่อไปยัง axon อันต่อไป ดังนั้น Synapse จะเป็นตัวลดหรือเพิ่มศักย์ไฟฟ้า (กระตุ้นหรือยับยั้ง) ดังนั้น โครงข่ายฯ จะสามารถปรับปรุงเปลี่ยนแปลงได้ตลอด ซึ่งเป็นธรรมชาติของสมองมนุษย์ [18]

สมองนั้นเป็นระบบที่ซับซ้อนมาก , ไม่เป็นเชิงเส้นและทำงานแบบขนาน ข้อมูลต่างๆ จะถูกเก็บแบบเป็นความรู้ในกรณีทั่วไปไม่เจาะจง กรณีใดกรณีหนึ่งหรือเจาะจงน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอยต่อระหว่าง Neuron เป็นตัวตัดสินหาคำตอบที่ถูกหรือผิด ซึ่งขึ้นอยู่กับทักษะการเรียนรู้ และประสบการณ์ของคนนั้นๆ ดังนั้น Synapse จะเป็นตัวแปรสำคัญในการหาคำตอบ



รูปที่ 2.22 ลักษณะทั่วไปทั้ง 2 สถานะของ Synapse (รูปจาก [20])

การฝึกฝนเล่นดนตรี เช่น เล่นกีตาร์จะต้องมีการฝึกการวางมือ วางนิ้ว, Synchronous กันระหว่างมือ 2 ข้าง เริ่มแรกจะไม่สามารถเล่นได้เลย ถ้ามีการฝึกฝนอย่างต่อเนื่องทำให้สมองมีการสั่งงานประสาท มือ, นิ้ว, กล้ามเนื้อที่จำเป็นในการเล่นบ่อยๆ มากๆ หลายๆ รอบจะทำให้ Synapse มีการส่งสัญญาณไฟฟ้าบ่อยครั้งขึ้น ทำให้รอยต่อใหญ่ขึ้นส่งไฟฟ้าได้ดีขึ้นและเมื่อถึงจุดๆ หนึ่ง เราก็จะสามารถเล่นได้คล่องแคล่ว กลับกัน ถ้าเราเคยเล่นได้แล้วไม่ได้ฝึกฝนอีกเป็นเดือนเป็นปี จะทำให้ไม่มีการส่งไฟฟ้าตาม Synapse นั้นทำให้ Synapse เล็กลง ส่งไฟฟ้าได้ไม่ดีหรือขาดไปในที่สุด (Network Pruning) [1] เมื่อกลับมาเล่นใหม่เราจะไม่สามารถเล่นได้คล่องแคล่วหรือดีดังเดิม

ฮอดคิน (A.L.Hodkin) และ ฮักซเลย์ (A.F.Huxley) ใช้เครื่องมือที่เรียกว่าไมโครอิเล็กโตรดสอดเข้าไปในปลายข้างหนึ่งของเอกซอนในเซลล์ประสาทปลาหมึกจึงค้นพบว่า มีการกระตุ้นและยับยั้งขึ้นอยู่กับสัญญาณมากกว่าหรือน้อยกว่า ค่า Threshold หรือไม่ [9] [10] ทำให้สามารถจำลองการทำงานของเซลล์ประสาทได้

ศาสตร์ทางด้านโครงข่ายประสาทเทียม ได้เริ่มมีการคิดค้นและวิจัยมาตั้งแต่ปี ค.ศ. 1940 โดยจะขอลำดับเหตุการณ์ในช่วงเวลาที่ก่อให้เกิดสถาปัตยกรรมของโครงข่ายประสาทเทียมชนิดต่างๆ ในปี ค.ศ. 1943 วอร์เรน แมคคูลล็อก (Warren McCulloch) และวอลเตอร์ พิตท์ (Walter Pitts) ได้ออกแบบนิรอนที่ถือเป็นต้นกำเนิดและพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของแมคคูลลอคและพิทท์ ประกอบด้วยชั้นของโครงข่ายจำนวน 2 ชั้นคือชั้นอินพุทและชั้นเอาต์พุท โครงข่ายจะมีการเชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาต์พุทโดยผ่านน้ำหนัก (weight) ซึ่งค่าน้ำหนักจะถูกกำหนดค่าไว้ตายตัว การทำงานเบื้องต้นของโครงข่ายคือนำไปใช้จำลองฟังก์ชันทางลอจิก พื้นฐานคือฟังก์ชัน AND ฟังก์ชัน OR ฟังก์ชัน NOT และฟังก์ชัน XOR โดยการนำเสนอการกำหนดค่าน้ำหนักเพื่อให้โครงข่ายทำงานได้กับการจำลองฟังก์ชันดังกล่าว



Warren Sturgis McCulloch  
1898-1969



Walter Pitts  
1923-1969

รูปที่ 2.23 ผู้คิดค้น โมเดล โครงข่ายประสาทเทียมตัวแรกของโลก  
(ภาพจากเว็บไซต์ [13])

ในปี ค.ศ. 1949 โดนาลด์ เฮบบ์ (Donald Hebb) ได้ออกแบบกฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมเป็นคนแรก กฎการเรียนรู้นี้ก็คือนั่นตอนของกระบวนการปรับแต่งค่าน้ำหนักเพื่อทำให้เกิดค่าเอาต์พุทที่ต้องการซึ่งเป็นแนวทางในการพัฒนาการเรียนรู้รูปแบบอื่นๆ ตามมา

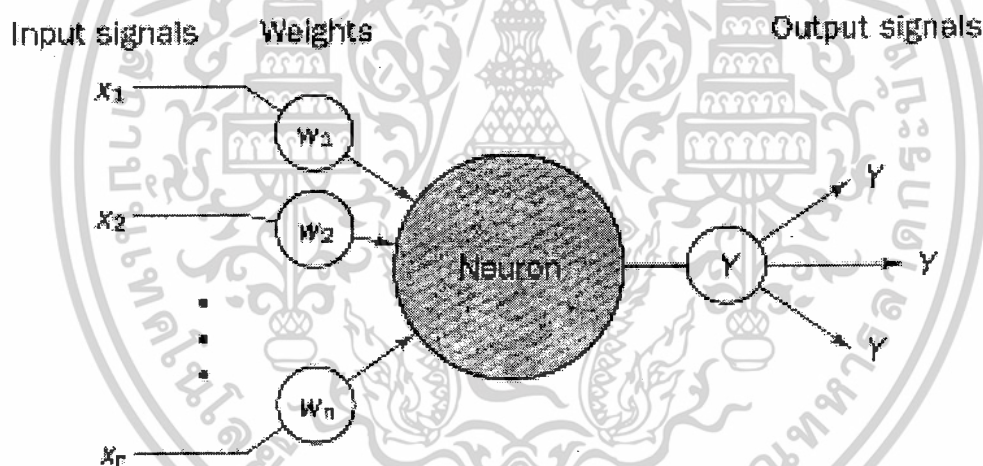
ในปี ค.ศ. 1957 แฟรงค์ โรเซนบลัทท์ (Frank Rosenblatt) ได้นำเสนอรูปแบบที่สำคัญรูปแบบหนึ่งของโครงข่ายประสาทเทียมนั่นก็คือรูปแบบที่เรียกว่าเพอร์เซพตรอน เป็นสถาปัตยกรรมอันประกอบไปด้วยชั้นอินพุทต่อผ่านน้ำหนักไปยังชั้นนิวตรอนที่ชั้นเอาต์พุท และใช้กฎการเรียนรู้ของเพอร์เซพตรอนในการปรับแต่งค่าน้ำหนักที่มีประสิทธิภาพดีกว่ากฎการเรียนรู้ของเฮบบ์

ในปี ค.ศ. 1960 เบอ์นาร์ด วิดโรว์ (Bernard Widrow) และมาเชียน ฮอฟฟ์ (Marcian Hoff) ได้พัฒนาการเรียนรู้ซึ่งใกล้เคียงกับการเรียนรู้ของเพอร์เซพตรอนและนำกฎนี้ไปใช้กับระบบหรือรูปแบบที่ชื่อว่าอะคาไลน์ รูปแบบมาคาไลน์คือรูปแบบที่พัฒนามาจากอะคาไลน์โดยวิดโรว์เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี ค.ศ. 1972 ทูโว โคโฮเนน (Teuvo Kohonen) ได้เริ่มพัฒนาการเรียนรู้แบบจัดตัวเอง (Self – Organized) สถาปัตยกรรมของโครงข่ายโคโฮเนนนี้ประกอบด้วยจำนวนชั้น 2 ชั้น คือชั้นอินพุตและชั้นเอาต์พุต การเชื่อมโยงที่ชั้นอินพุตไปยังนิวตรอนที่ชั้นเอาต์พุตนั้นจะต่อผ่านน้ำหนักข้อแตกต่างจากโครงข่ายอื่นๆ คือกระบวนการเรียนรู้ที่สามารถจัดกลุ่มข้อมูลอินพุตได้ด้วยของมันเองและการเรียนรู้ี้เรียกว่าการเรียนรู้แบบแข่งขัน (Competitive Learning)

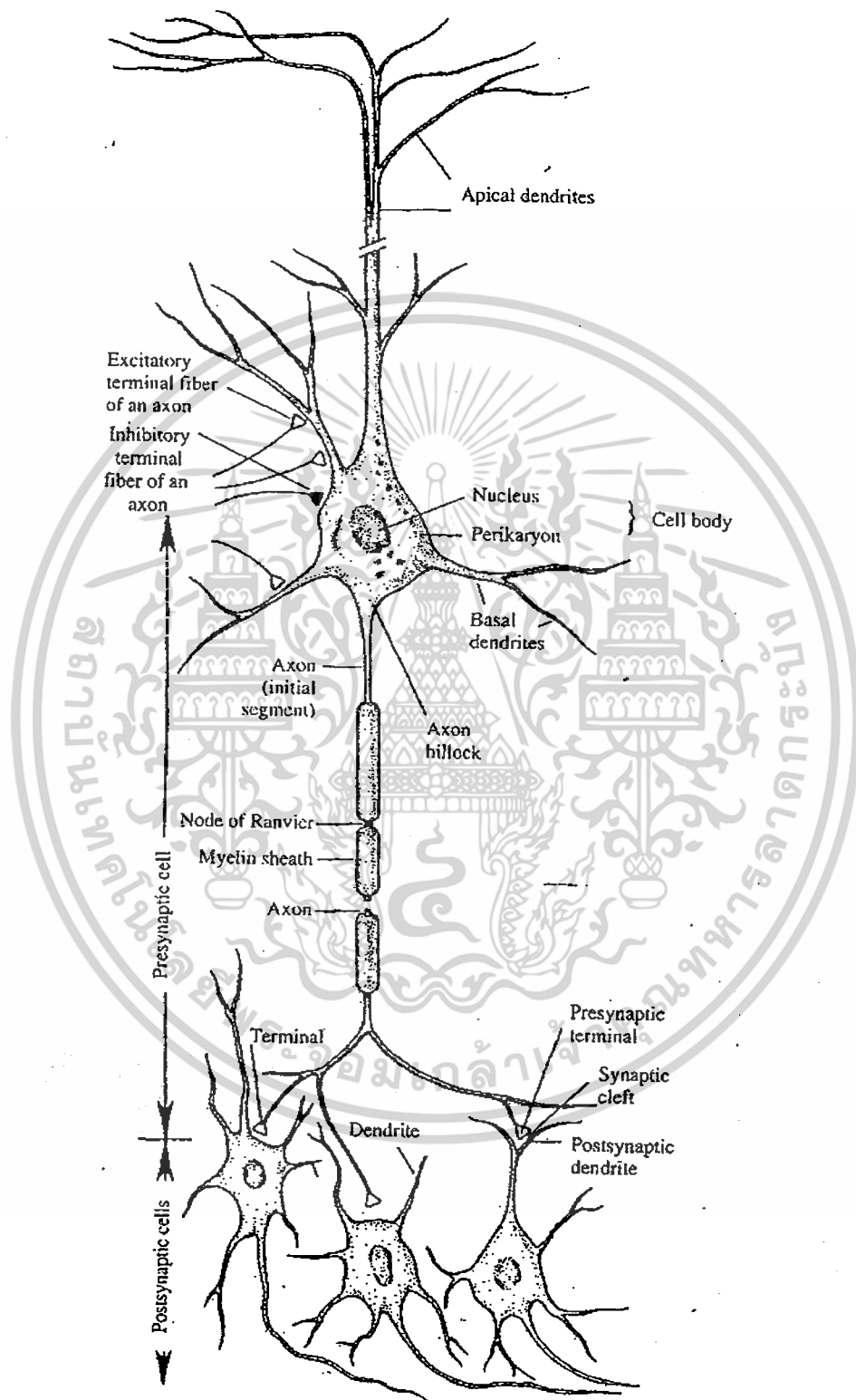
หลังจากปี ค.ศ. 1980 เป็นต้นมาได้มีการพัฒนารูปแบบของโครงข่ายประสาทเทียมชนิดต่างๆ เช่น รูปแบบจัดตัวเองที่เรียกว่า ART คิดค้นโดย สตีเฟน กรอสเบิร์ก (Stephen Grossberg) และ เกย์ล คาร์เพนเตอร์ (Gail Carpenter) รูปแบบโครงข่ายฮอปฟิลด์ คิดค้นโดย จอห์น ฮอปฟิลด์ (John Hopfield) นีโอคอกนิวตรอน คิดค้นโดย คุนิฮิโกะ ฟูกูชิมะ (Kunihio Fukushima) รูปแบบโบลตแมนน์แมชชีน (Boltzmann Machine) คิดค้นโดยนักวิจัยหลายคน และรูปแบบที่นิยมใช้กันมากคือรูปแบบแพร่ย้อนกลับ (Back – Propagation) คิดค้นโดยเดวิด ปาร์กเกอร์ (David Parker) และเดวิด รุมลฮาร์ด (David Rumelhard)



รูปที่ 2.24 แสดงต้นแบบโครงข่ายประสาทเทียมตัวแรกของโลก ชื่อว่าแบบจำลอง McCulloch – Pitts (ภาพจาก[18])

ต่อไปจากแสดงภาพเซลล์ประสาทของสิ่งมีชีวิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 เซลล์ประสาท (ภาพจาก [24])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงสรุปเป็นตารางได้ดังนี้

ตารางที่ 2 การเปรียบเทียบ ระหว่าง โครงข่ายประสาททางชีวภาพ กับ โครงข่ายประสาทเทียม

Biological Neural Networks	Artificial Neural Networks
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

ลักษณะที่สำคัญของเซลล์ประสาทของสิ่งมีชีวิตที่นำไปสู่การพัฒนาโครงข่ายประสาทเทียมคือ [3]

- 1) ตัวเซลล์ประสาทหรือหน่วยประมวลผลสามารถรับสัญญาณได้หลายสัญญาณ
- 2) ความแรงของสัญญาณอาจถูกควบคุมได้ที่บริเวณ Synapse
- 3) หน่วยประมวลผลทำการรวมสัญญาณอินพุตที่ผ่านการปรับความแรงของสัญญาณ
- 4) ในบางสถานการณ์ ตัวประสาทส่งสัญญาณเอาที่พุทออกไป สัญญาณเดียว
- 5) สัญญาณจากตัวประสาทอาจส่งไปยังเซลล์ประสาทอื่นหลายเซลล์
- 6) ความแรงของสัญญาณที่ผ่าน Synapse สามารถปรับเปลี่ยนได้ตามประสบการณ์
- 7) สัญญาณที่ส่งผ่าน Synapse อาจเป็นแบบยับยั้ง (Inhibitory) หรือกระตุ้น (Excitatory)

### 2.8.3 Artificial Neuron Network (ANN) เรียนรู้ได้อย่างไร

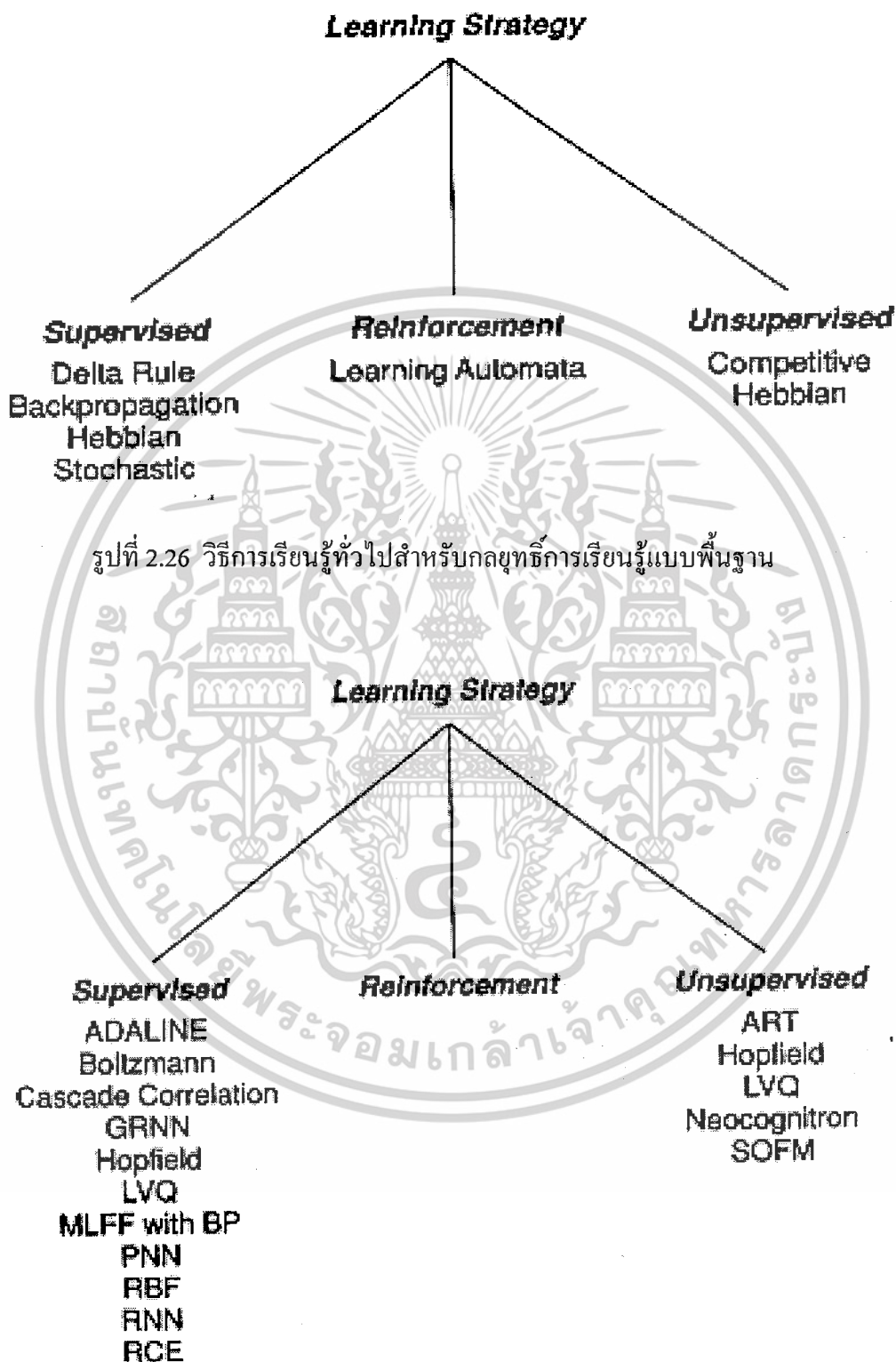
Artificial Neuron Network เป็นการเชื่อมต่อของ link ที่ link จะมีค่า Weight อยู่ ค่า Weight เป็นสิ่งสำคัญในการเรียนรู้ หมายถึง long-term memory ดังนั้น การเรียนรู้ของ ANN คือการหา Weight ที่ดีที่สุดทุก ๆ Weight ซึ่งจะต้องมีการปรับหลายครั้ง [18]

จาก McCulloch – Pitts Model จะสังเกตว่า จะต้องปรับค่า Weight ที่มากระทำกับ input เพื่อให้ได้ output ที่ต้องการ

เนื่องจากนักวิจัยหลายท่านคิดค้น Model ANN มากมาย ในปัจจุบันมีเป็นร้อยชนิด แต่ละ model ต้องมีกฎการเรียนรู้ซึ่งมีหลายแบบเช่นกัน เพื่อหา Weight ที่ดีที่สุด และเหมาะสมกับ Model

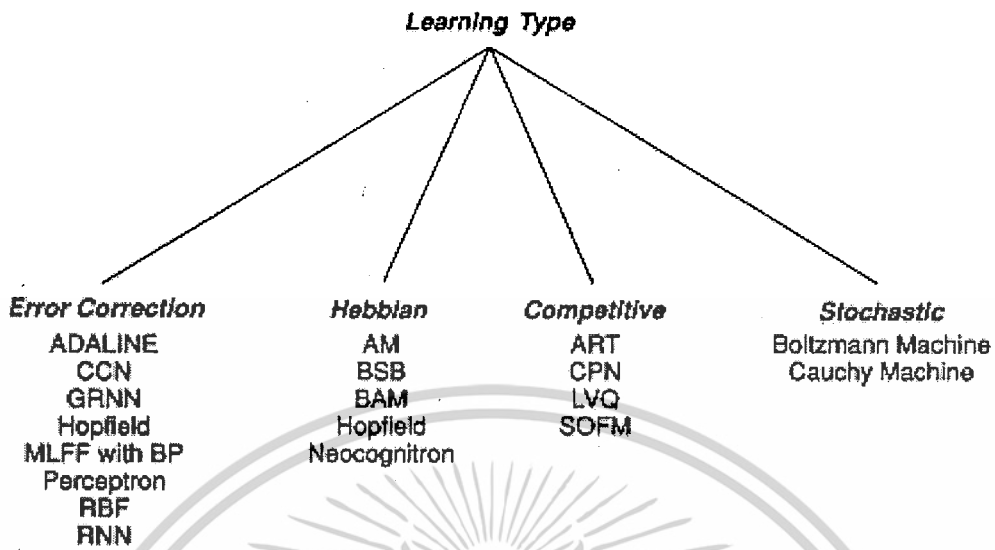
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลยุทธ์การเรียนรู้ปกติมี 3 แบบ สามารถจำแนกได้ดังนี้ [19]

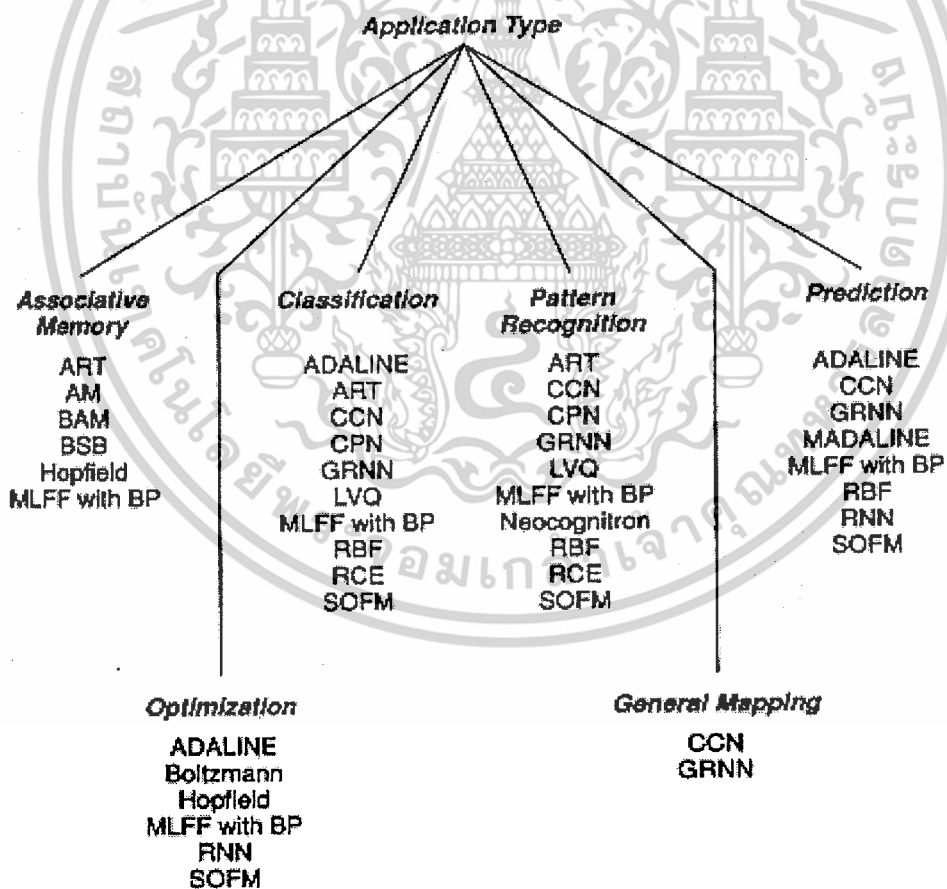


รูปที่ 2.27 การแบ่งชนิดของโครงข่ายประสาทเทียมตามกลยุทธ์การเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

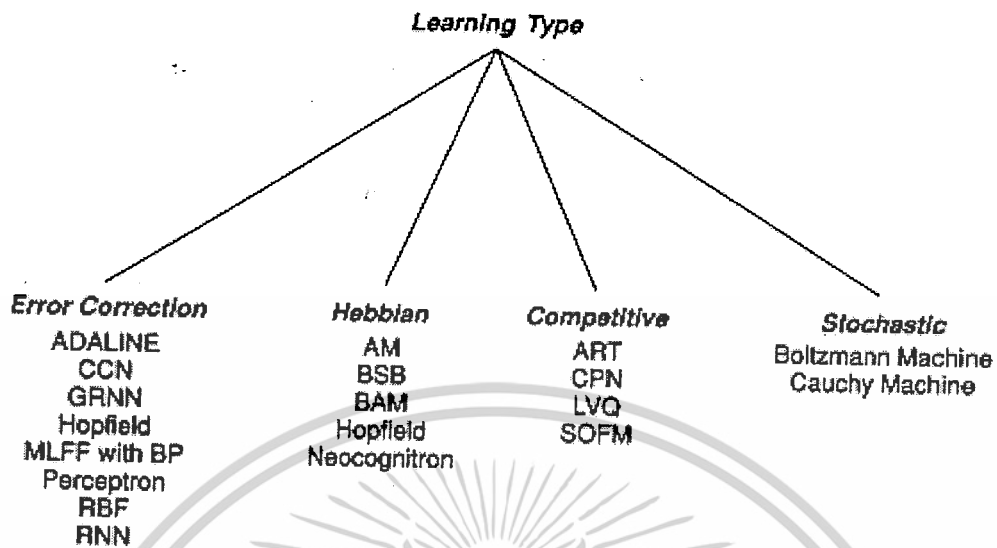


รูปที่ 2.28 การแบ่งชนิดของโครงข่ายประสาทเทียมตามวิธีการเรียนรู้

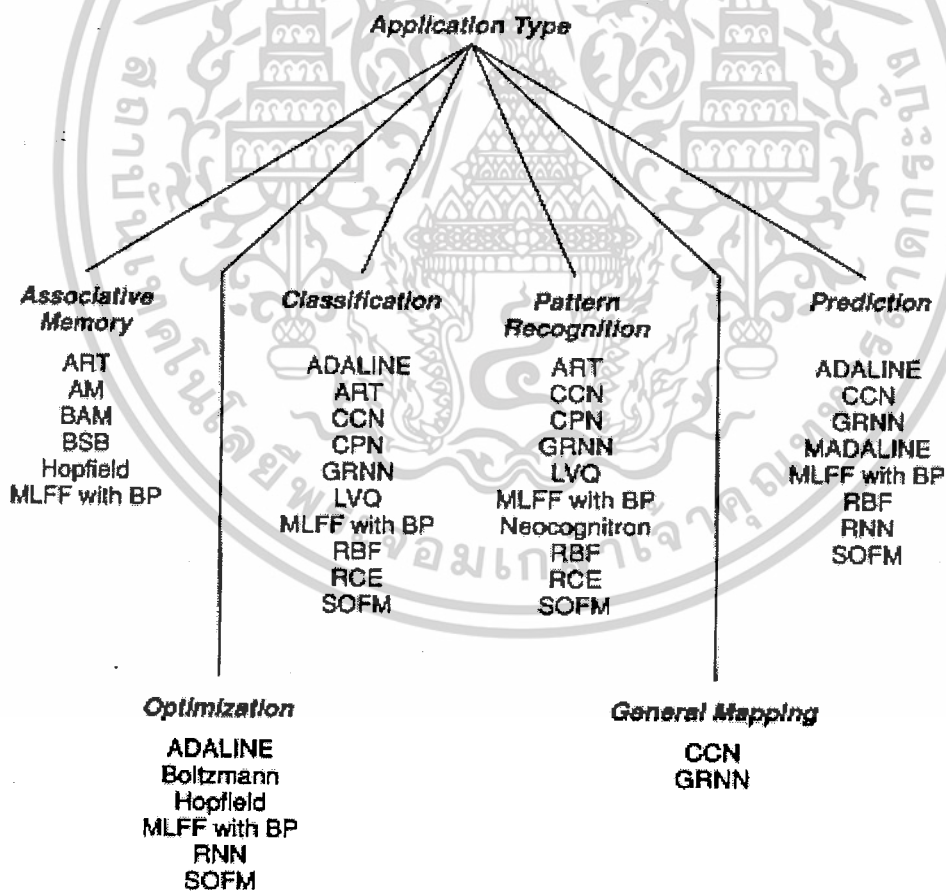


รูปที่ 2.29 การแบ่งชนิดของโครงข่ายประสาทเทียมตามการประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

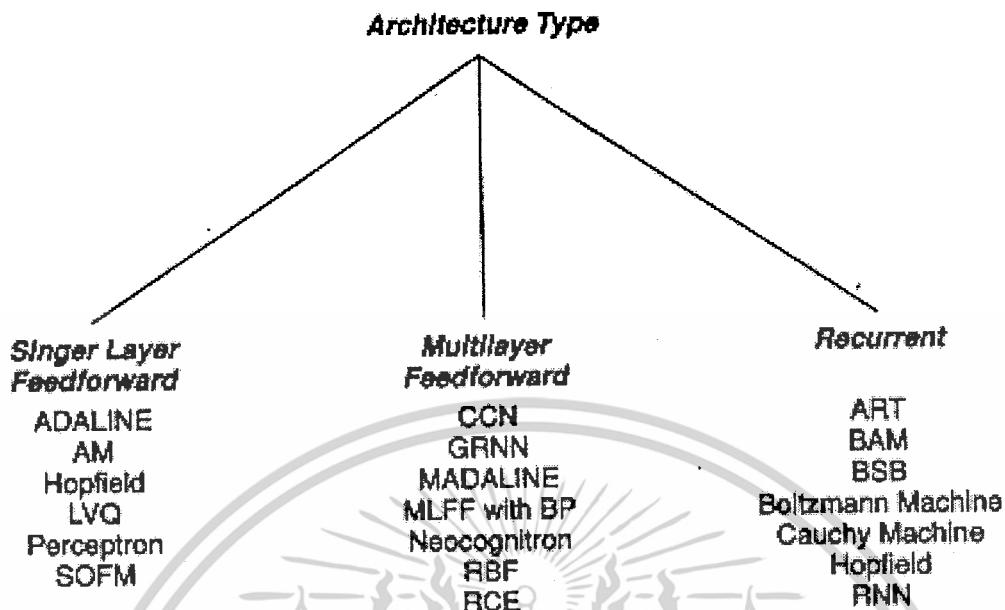


รูปที่ 2.28 การแบ่งชนิดของโครงข่ายประสาทเทียมตามวิธีการเรียนรู้



รูปที่ 2.29 การแบ่งชนิดของโครงข่ายประสาทเทียมตามการประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.30 การแบ่งชนิดของโครงข่ายประสาทเทียมตามสถาปัตยกรรมโครงข่าย

ADALINE (Adaptive Linear Neural Element)

ART (Adaptive Resonant Theory)

AM (Associative Memories)

BAM (Bidirectional Associative Memory)

Boltzmann Machine

BSB (Brain – State – in – a – Box)

CCN (Cascade Correlation)

Cauchy Machine

CPN (Counter Propagation)

GRNN (Generalized Regression Neural Network)

Hamming

Hopfield

LVQ (Learning Vector Quantization)

MADALINE

MLFF with BP (Multilayer Feedforward Backpropagation)

Neocognitron

NLN (Neurologic Networks)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Perceptron

PNN (Probabilistic Neural Network)

RBF (Radial Basis Function)

RNN (Recurrent Neural Networks)

จากรูปข้างต้น จะเห็นว่า model ต่าง ๆ ก็เหมาะกับกฎการเรียนรู้ต่างไป และเหมาะกับ Application ต่างๆ กันไป

Singlelayer Feedforward เป็นโครงสร้างชั้นเดียวที่มีการส่งสัญญาณไปข้างหน้าเท่านั้น

Multilayer Feedforward เหมือน Singlelayer Feedforward แต่มี layer มากกว่า 2 ชั้น เรียกว่า “ชั้นซ่อน” หรือ “hidden layer”

Recurrent เป็นโครงสร้างที่ Axon มีการป้อนสัญญาณกลับมาเข้า neuron ที่ต่ำกว่า เช่น ป้อนเข้า hidden layer หรือ ป้อนเข้า input layer

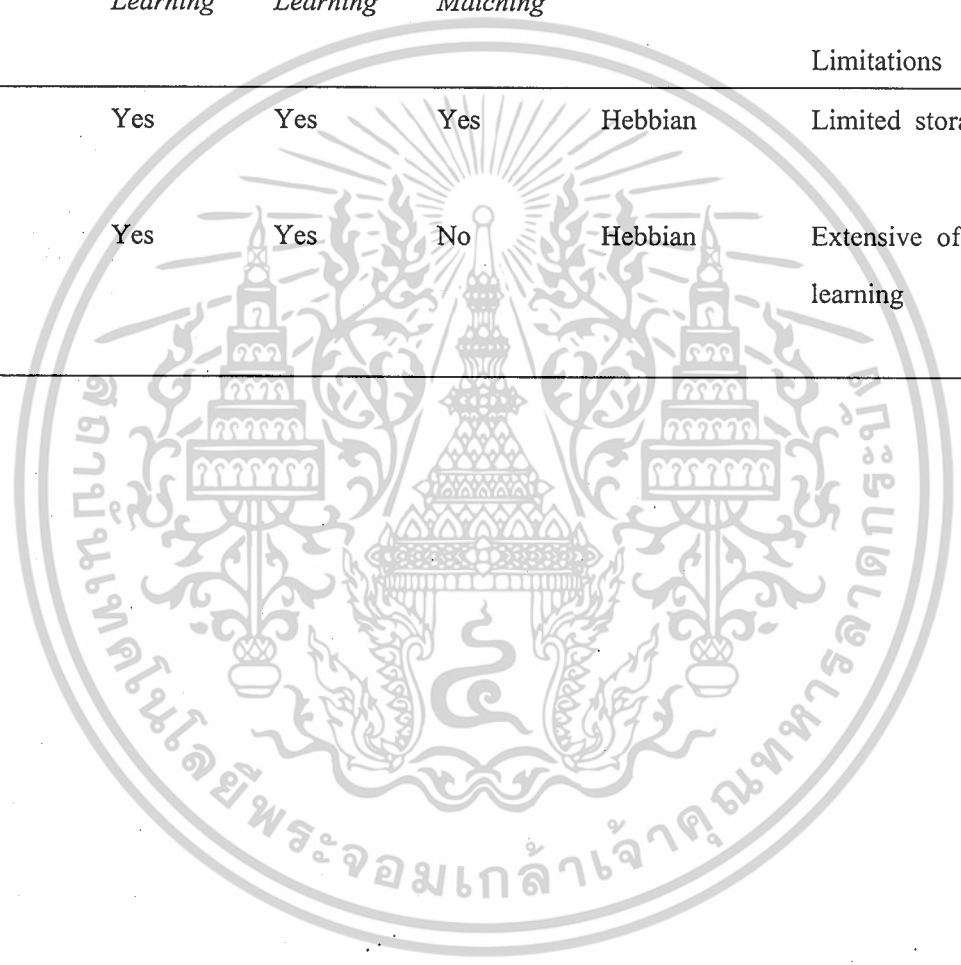
ต่อไปแสดงตารางการประยุกต์ใช้งาน Artificial Neuron Network แบ่งตามกลยุทธ์การเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3 Unsupervised Learning: Feedback Recall Artificial Neural System [19]

Paradigm	Problem <i>Type Solved</i>	On – line <i>Learning</i>	Off – line <i>Learning</i>	Pattern <i>Matching</i>	Type <i>Learning</i>	Limitations	Implementation
AG	Unsupervised data classification	Yes	No	Yes	Hebbian	Limited storage	Optical and electronic
SG	Human information processing	Yes	No	Yes	Hebbian	Limited storage	Electronic
ARTI	Classify complex patterns	Yes	No	Yes	Competitive decay	Restricted to binary	Neural computer
DH	Speech processing	No	Yes	No	Hebbian		Optical electronic
CH	Combinatorial optimization	No	Yes	Yes	Hebbian or competitive	Restricted to binary	VLSI Optical electronic
BAM	Image recognition	No	Yes	Yes	Hebbian	Limited storage	VLSI
TAM	Store spatial – temporal patterns	No	Yes	No	Hebbian	Extensive off-line learning	Optical and neural computer
ABAM	Process analog patterns in continuous time	Yes	No	Yes	Hebbian	Extensive off-line learning Limited storage	Computer simulation Neural computer

Paradigm	Problem <i>Type Solved</i>	On – line <i>Learning</i>	Off – ling <i>Learning</i>	Pattern <i>Matching</i>	<i>Learning</i>	Limitations	Implementation
CABAM	Combinatorial optimization	Yes	Yes	Yes	Hebbian	Limited storage	Computer simulation
FCM	Combinatorial optimization	Yes	Yes	No	Hebbian	Extensive off-line learning	Computer simulation



ตารางที่ 4 Unsupervised Learning: Feedforward Recall Artificial Neural Systems [19]

Paradigm	Problem <i>Type Solved</i>	On – line <i>Learning</i>	Off – line <i>Learning</i>	Pattern <i>Matching</i>	Type <i>Learning</i>	Limitations	Implementation
LM	Process Monitoring	No	Yes	Yes	Hebbian	None reported	Mechanical electronic magnetic
DR	Prediction	No	Yes	Yes	Hebbian	Restricted to discrete time	Robot simulation
LAM	System control	No	Yes	Yes	Hebbian	Limited storage	Computer simulation
OLAM	Signal processing	No	Yes	Yes	Hebbian	Extensive off-line learning	Optical
FAM	Knowledge processing	Yes	No	Yes	Hebbian	Limited storage	Computer simulation and silicon chip

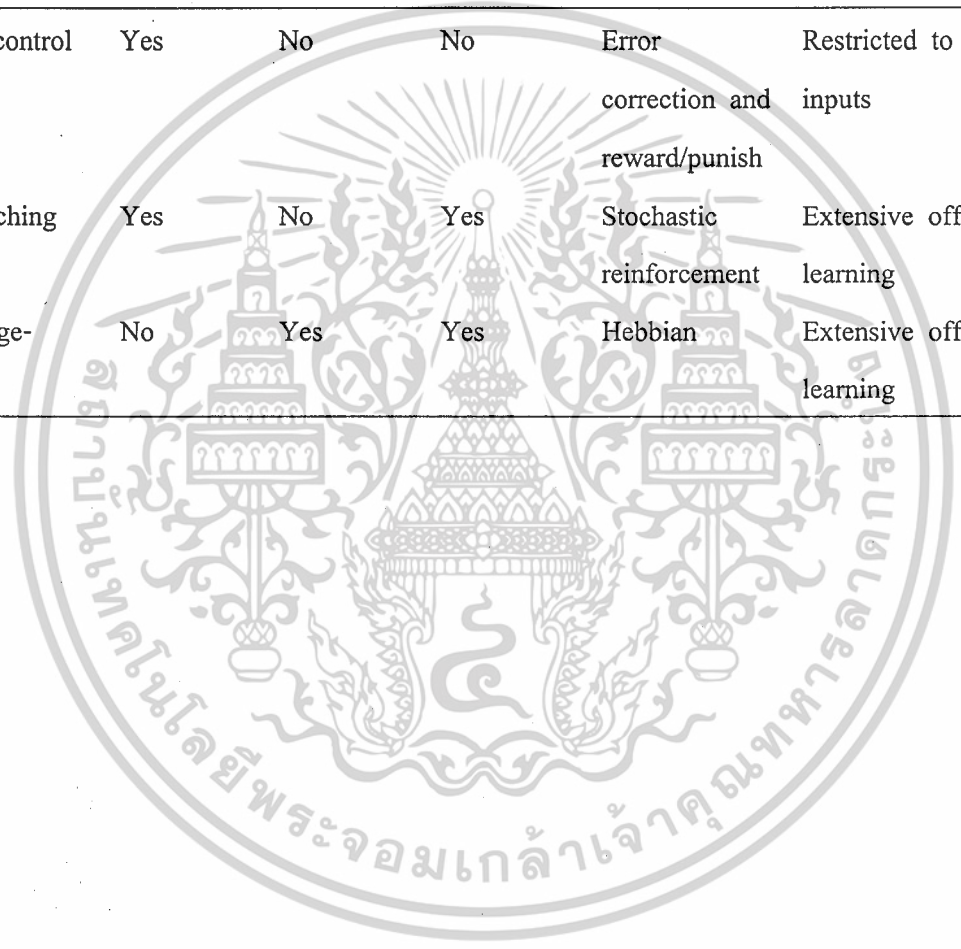
ตารางที่ 5 Supervised Learning: Feedforward Recall Artificial Neural Systems [19]

Paradigm	Problem <i>Type Solved</i>	On – line <i>Learning</i>	Off – line <i>Learning</i>	Pattern <i>Matching</i>	Type <i>Learning</i>	Limitations	Implementation
BSB	Real – time classification	No	Yes	Yes	Error correction	Extensive off-line learning	Computer simulation

ตารางที่ 6 Supervised Learning: Feedback Recall Artificial Neural Systems [19]

Paradigm	Problem <i>Type Solved</i>	On – line <i>Learning</i>	Off – line <i>Learning</i>	Pattern <i>Matching</i>	Type <i>Learning</i>	Limitations	Implementation
Perceptron	Prediction	No	Yes	Yes	Error correction	None reported	Mark I Perceptron machine
ADALINE/ MADALINE	Prediction noise canceling	No	Yes	Yes	Error correction	Extensive off-line learning	Magnetic optical
BP	Crypting character- recognition	No	Yes	Yes	Error correction	Extensive off-line learning	Optical electronic VLSI
AVQ	Self-organization of data	No	Yes	Yes	Error correction	Extensive off-line learning	Neural-computer VLSI
CPN	Self-programming	No	Yes	Yes	Hebbian	Lack of control w/o star mapping	Neural computer
BM	Combinatorial optimization	No	Yes	Yes	Hebbian and simulated annealing	Extensive off-line learning	Optical electronic VLSI
CM	Combinatorial optimization Problem	No	Yes	Yes	Hebbian and simulated annealing	Extensive off-line learning	Optical

Paradigm	Type Solved	On – line Learning	Off – ling Learning	Pattern Matching	Type Learning	Limitations	Implementation
AHC	Prediction control	Yes	No	No	Error correction and reward/punish	Restricted to bipolar inputs	Computer simulation
ARP	Pattern-matching control	Yes	No	Yes	Stochastic reinforcement	Extensive off-line learning	Computer simulation
SNMF	Speech-image-recognition	No	Yes	Yes	Hebbian	Extensive off-line learning	Computer simulation



<i>Acronym</i>	<i>Definition</i>
AG	Additive Grossberg
SG	Shunting Grossberg
ART1	Binary Adaptive Resonance Theory
DH	Discrete Hopfield
CH	Continuous Hopfield
BAM	Discrete Bidirectional Associative Memory
TAM	Tempotal Associative Memory
ABAM	Adaptive Bidirectional Associative Memory
CABAM	Competitive ABAM
FCM	Fuzzy Cognitive Map
LM	Learning Matrix
DR	Drive Reinforcement
LAM	Linear Associative Memory
OLAM	Optimal Linear Associative Memory
FAM	Fuzzy Associative Memory
BSB	Brain-State-in-a-Box
PERCEP	Perceptron
ADALINE	Adaptive Linear Element
MADALINE	Multiple Adaptive Linear Element
BP	Back Propagation
AVQ	Adaptive Vector Quantizer
CPN	Counterpropagation
BM	Boltzman Machine
CM	Cauchy Machine
ARP	Associative Reward-Penalty
SNMF	Spatiotemporal Nearest-Matched Filter

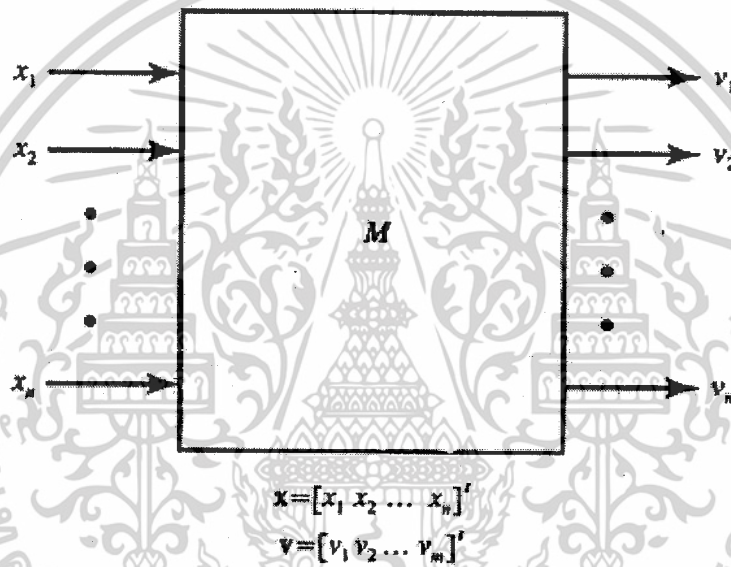
สำหรับ Model , กฎการเรียนรู้, กลยุทธ์การเรียนรู้ รายละเอียดต่างๆ สามารถศึกษาได้จาก Text book ทั่วไป ในปริณญาณิพนธ์นี้ ใช้ ANN ชนิด Associative Memory ที่ใช้ Hebbian Learning ในการจดจำภาพไผ่ยูกิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.4 Associative Memories

Associative Memory เป็น Artificial Neural Network ชนิดหนึ่งซึ่งเป็นการจำลองความจำของมนุษย์ในลักษณะการจำหลาย ๆ สิ่งไว้ร่วมกัน และเมื่อรับ input เข้ามาในรูปแบบเดียวกันจะนำ input เข้าไปสืบค้น (retrieval) ในความจำทั้งหมด แล้วส่ง output ที่มีความสัมพันธ์กับ input ออกมา

ในระหว่างการสืบค้นจากการกระตุ้นด้วย input ที่ไม่รู้จัก เรียกว่า key pattern ลักษณะเหมือนกับ search argument



รูปที่ 2.31 บล็อกไดอะแกรมของ Associative Memory [26]

จากรูป 2.31 เป็น Block diagram ทั่วไปของ Associative Memory ทั่วไปซึ่งเป็นการสร้างจาก input vector  $X$  และ output vector  $V$  เป็นไปดังสมการ

$$V = M[X]$$

โดย  $M$  เป็น nonlinear matrix – type operator (Neural Memory Model)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้าง M ต้องใช้ Algorithm เรียกว่า recording หรือ storage algorithm vector ต้นแบบจะถูกเก็บใน memory storage algorithm จะเกิดจาก 1 หรือ 2 set ของ vector ต้นแบบ ซึ่ง algorithm ขึ้นอยู่กับชนิดของ Associative Memories [26]

สมมุติว่า P คือ stored pair ของ association นิยามได้

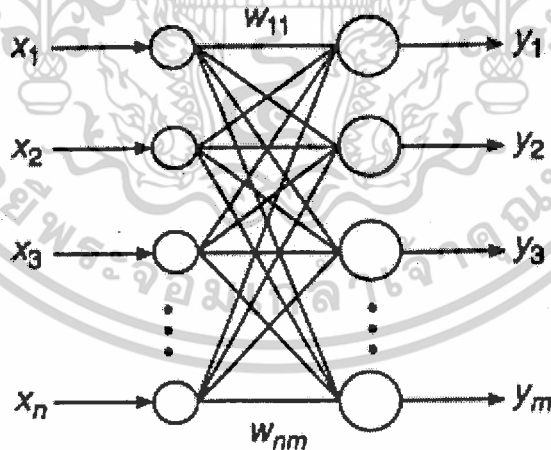
$$X^{(i)} \rightarrow V^{(i)}, \quad i = 1, 2, \dots, P$$

จะเห็นว่า Associative Memories ประกอบด้วยเซตของคู่ลำดับที่สัมพันธ์กัน เช่น ชื่อคน = X กับรูปถ่ายของคนคนนั้น = V โดยคู่ลำดับนี้เรียกว่า Exemplar [5] ดังนั้น input X ที่ไม่รู้จักจะถูกนำมาเป็น key pattern โดยเทียบกับ input ต้นแบบใน memory ดังนั้น input prototype ที่จะถูกเลือกจะมีความใกล้เคียงกับ unknow input vector X ที่สุดโดยใช้ระยะ Hamming มาเป็นตัววัดตั้งสมการ [20]

$$d(X', X^P) = \min_i \{d(X', X^i)\}$$

#### 2.8.4.1 Feedforward Associative Network

เป็น Associative Network ที่ส่งสัญญาณไปหน้าเท่านั้น มี layer เพียง 1 layer



รูปที่ 2.32 Feedforward Associative Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า input X แพร่ผ่าน Weight W และได้ Output Y ดังสมการ

$$\text{Net}_j = \sum_{i=1}^n W_{ij} X_j = W_i X$$

โดย X เป็น input vector ,  $W_i$  weight matrix [20]

#### 2.8.4.2 Learning Algorithms สำหรับ Associative Memory

##### Hebb's Learning Rule

Hebb's Learning Rule เป็นกฎการเรียนรู้พื้นฐานจากประโยคที่ว่า “ถ้า neuron ที่ i ส่งอิทธิพลต่อ neuron j ซ้ำๆ synapse ระหว่าง neuron ทั้งสองจะแข็งแรงขึ้น” [18] สามารถเขียนเป็นสมการคือ

$$\Delta W_{ij} = \alpha X_i Y_j$$

โดย  $\alpha$  คือ Normalizing Constant หรือ stepsize เกิดจาก Learning rate และ forgetting factor ใช้ควบคุมการเรียนรู้ [18]

กรณี matrix

$$W = \alpha Y(X)^t$$

โดย W จะเก็บคู่ของ Pattern ที่ mapping ไว้  $(X^p, Y^p)$ , P = pattern ดังนั้นเพื่อความถูกต้อง ควรเขียนเป็น

$$W^p = \alpha Y^p (X^p)^t$$

เพราะฉะนั้น Weight รวมของทุกคู่ pattern  $(X^p, Y^p)$  คือ

$$W = \sum_{p=1}^P W^p$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไป Associative Memories มีอยู่ 3 ชนิด [5]

### 1. Heteroassociative Memories

$$Y_i = M[X_i]$$

โดย  $X_i$  และ  $Y_i$  เป็น pattern คนละชนิดกัน เช่น ชื่อกับรูปถ่ายหรือชื่อกับเบอร์โทรศัพท์ โดย input  $X$  ใดๆ ที่ป้อนเข้าไปจะให้ค่า  $Y_i$  โดยที่  $X$  ใกล้เคียง  $X_i$  มากที่สุด

เช่น  $X =$  ทักษิ

$$m[\text{ทักษิ}] = y = \text{ชินวัตร}$$

### 2. Autoassociative Memories

$$X_i = m[X_i]$$

โดย  $X_i$  เป็น pattern ชนิดเดียวกัน เช่น ชื่อหรือหน้าคนรู้จักที่เพิ่งเจอกันใหม่ input  $X$  ใดๆ ที่ป้อนเข้าไป จะให้ค่า  $X_i$  ที่ใกล้เคียงกับ  $X$  มากที่สุด

เช่น  $X =$  ทักษิ

$$m[\text{ทักษิ}] = X = \text{ทักษิ}$$

### 3. Interpolative Associative Memories

$$Y_i = m[X_i]$$

โดย input  $X$  ใดๆ ที่ป้อนเข้าไปโดย  $X = X_i + d$

$m[X_i]$  จะป้อน  $Y_i + e$  ออกมา เช่น

$X =$  ทักษิ

$$m[\text{ทักษิ}] = Y + e = \text{ชินวัตร}$$

ในเรื่อง Associative Memories นี้จะใช้ Vector เช่น

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}, \quad \text{โดย } X = \text{vector } X \text{ และ } X_n = \text{Component of vector } X$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.4.3 Feedforward Heteroassociative Memories Networks

เป็น Network ที่มีโครงสร้างเหมือน Feedforward Associative Memories เพียงแต่ input และ output ต้นแบบไม่ใช่ชนิดเดียวกันโดยมีการสร้าง  $W^p$  (Correlation Matrix) [20]

$$W^p X^p = t^p (x^p)' \cdot X^p = t^p$$

จะสังเกตว่า ด้วยหลักการของ Matrix จะสามารถหาคู่ของอีกตัวได้เสมอ เพราะ Weight ถูกสร้างจาก input transpose เมื่อ vector ตั้งจากกันคูณกันจะหักล้างกัน ด้วยสมการ

$$x_i' x_j = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}, X' = \text{transpose of } X$$

ดังนั้น เราสามารถสร้าง mapping function ได้โดยง่าย [5]

$$m[X] = (Y_1 X_1' + Y_2 X_2' + \Lambda + Y_L X_L')$$

ถ้า  $X = X_2$  จะได้ว่า

$$\begin{aligned} m[X] = m[X_2] &= (Y_1 X_1' + Y_2 X_2' + \Lambda + Y_L X_L') X_2 \\ &= Y_1 X_1' X_2 + Y_2 X_2' X_2 + \Lambda + Y_L X_L' X_2 \\ &= Y_1 \delta_{12} + Y_2 \delta_{22} + \Lambda + Y_L \delta_{L2} \\ &= Y_2 \end{aligned}$$

โดยทั่วไป input vector มักจะไม่เป็น orthonormal vector ทำให้ไม่ได้ output ตามที่ต้องการ

#### Signum function

เป็น function threshold ที่อยู่ภายใน node ของ Associative Memories Network ดังสมการ

$$Y_j = \text{Sgn} \left( \sum_{i=1}^n W_{ij} X_i \right) = \begin{cases} +1 & \text{if } W_{ij} X_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

ซึ่ง Associative Memory มักจะใช้ vector ที่ประกอบจาก Component ที่เป็น Bipolar (กรณี 1, 0 ก็มี (Binary))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1

ให้  $+ \equiv +1$  และ  $- \equiv -1$ ,  $X^p$  = vector input,  $t^p$  เป็น vector output

$$X^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad X^2 = \begin{bmatrix} -1 \\ +1 \\ +1 \\ -1 \end{bmatrix} \quad X^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \quad X^4 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \end{bmatrix}$$

$$t^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix} \quad t^2 = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \quad t^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix} \quad t^4 = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}$$

$$W^1 = t^1(X^1)^T = \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \begin{bmatrix} + & - & - & - \end{bmatrix} = \begin{bmatrix} + & - & - & - \\ - & + & + & + \\ - & + & + & + \end{bmatrix}$$

$$W^2 = \begin{bmatrix} - \\ + \\ - \\ - \end{bmatrix} \begin{bmatrix} - & + & + & - \end{bmatrix} = \begin{bmatrix} + & - & - & + \\ - & + & + & - \\ + & - & - & + \end{bmatrix}$$

$$W^3 = \begin{bmatrix} - \\ - \\ - \\ + \end{bmatrix} \begin{bmatrix} - & - & + & + \end{bmatrix} = \begin{bmatrix} + & + & - & - \\ + & + & - & - \\ - & - & + & + \end{bmatrix}$$

$$W^4 = \begin{bmatrix} + \\ - \\ - \\ + \end{bmatrix} \begin{bmatrix} - & - & - & + \end{bmatrix} = \begin{bmatrix} - & - & - & + \\ + & + & + & - \\ - & - & - & + \end{bmatrix}$$

$$W = W^1 + W^2 + W^3 + W^4 = \begin{bmatrix} 2 & -2 & -4 & 0 \\ 0 & 4 & 2 & -2 \\ -2 & -2 & 0 & 4 \end{bmatrix}$$

$$Wx^1 = \begin{bmatrix} 2 & -2 & -4 & 0 \\ 0 & 4 & 2 & -2 \\ -2 & -2 & 0 & 4 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} +8 \\ -4 \\ -4 \end{bmatrix} \quad \text{ดังนั้น, } y^1 = \text{sgn} \left( \begin{bmatrix} +8 \\ -4 \\ -4 \end{bmatrix} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Y^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix}$$

$$y^2 = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \quad y^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix} \quad y^4 = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}$$

ในที่นี้จะไม่กล่าวถึง Autoassociative Memories เนื่องจากโครงงานใช้ Heteroassociative Memories ซึ่งสามารถศึกษาได้จาก text book ทั่วไป

#### 2.8.4.4 Recurrent Associative Memories

เป็น Associative Memories ที่มีการส่งสัญญาณย้อนกลับเข้ามาใน Network เช่น Hopfield Network, Bidirectional Associative Memories, Multidirectional Associative Memories, Brain State – In – a – Box และ Temporal associative memory

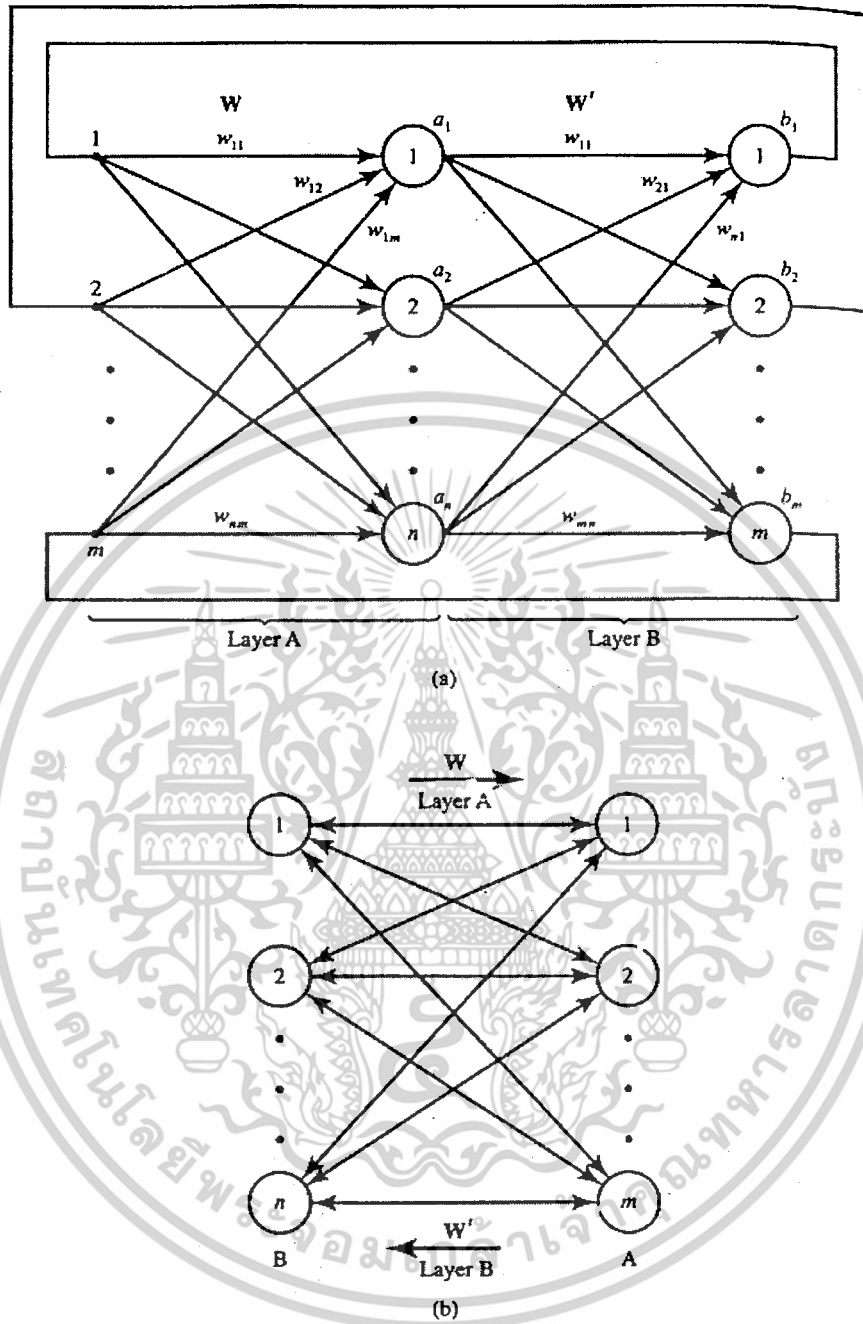
##### 2.8.4.4.1 Bidirectional Associative Memory (BAM)

BAM เป็น Heteroassociative Memories ประกอบไปด้วย 2 layer ที่ใช้ในการส่งผ่านสัญญาณไปหน้าและย้อนกลับ สร้างขึ้นโดย Bart Kosko (1987)



รูปที่ 2.33 Prof. Bart Kosko

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 แสดง Bidirectional Associative Memory: (a) general diagram and (b) simplified diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด Weight  $W$  นั้น แสดงไว้ดังสมการ

$$W = \sum_{P=1}^P W^P$$

โดย

$$W^P = Y^P(X^P)^t$$

ซึ่ง  $X^P$  และ  $Y^P$  เป็น input และ output vector ที่เป็นคู่สัมพันธ์ต่อกัน ดังนั้นถ้าต้องการหา output vector โดยป้อน input vector  $X$  ใด ๆ

$$Y = \text{Sgn}(WX)$$

สมการนี้ถือเป็นช่วงสัญญาณไปหน้า (feedforward part) ต่อไปจะทำการหา input vector  $X$  ที่สัมพันธ์กับ input vector  $Y$  จากสมการที่แล้ว

$$X = \text{Sgn}((W)^t \cdot Y)$$

สมการนี้เป็นช่วงสัญญาณกลับ feedback Part) โดย Signum function คือ

$$\text{Sgn}(Y_i) = \begin{cases} +1 & Y_i > 0 \\ 0 & Y_i = 0 \\ -1 & Y_i < 0 \end{cases}$$

Signum function นี้อาจเปลี่ยนเป็น activation function BAM จะทำการคำนวณผลลัพธ์ใน feedforward Part และ feed back Part ซ้ำ ๆ จนกว่าจะเกิดสถานะเสถียร กล่าวคือ Vector  $X$  ที่ได้จาก feed back part มีค่าซ้ำในรอบที่แล้ว

$$Y_i = \begin{cases} +1 & \text{if } WX_i > \theta \\ Y_i & \text{if } |WX_i| < \theta \\ -1 & \text{if } WX_i < \theta \end{cases}$$

โดย  $\theta \neq 0$  เป็นค่า threshold

ซึ่งอาจใช้ activation แบบอื่นได้ เช่น Sigmoid function กรณีใช้ discrete activation function การทำให้เป็น Bipolar (+1, -1) จะมีประสิทธิภาพกว่า binary (1, 0) (Kosko, 1987) [20] ซึ่งมันจะช่วยลด Noise หรือ crosstalk term ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงสามารถสรุปการทำงานของ BAM ได้ดังนี้ [5]

1. กำหนดค่าเริ่มต้นของ unknow input vector  $X(t)$  และ output vector  $Y(t)$
2. Feedforward Part ป้อน vector  $X(t)$  เข้าทาง input layer คำนวณและ update ค่า  $Y(t+1)$  แทนที่  $Y(t)$  ( $t$  คือหมายเลขจำนวนรอบในการคำนวณ)
3. Feedback part ป้อน  $Y(t+1)$  จากข้อ 2 เข้าทาง output layer คำนวณและ update ค่า  $X(t+1)$  แทนที่  $X(t)$
4. คำนวณซ้ำในข้อ 2-3 จนกว่า  $X(t+n)$  ในรอบล่าสุดจะมีค่าเท่ากับ  $x(t+n-1)$  ในรอบที่แล้ว หรือ  $t > t_{\max}$  ซึ่งก็คือ BAM จะทำการคำนวณผลลัพธ์ใน feedforward Part และ feed back Part ซ้ำ ๆ จนกว่าจะเกิดสถานะเสถียร กล่าวคือ Vector  $X$  ที่ได้จาก feed back part มีค่าซ้ำในรอบที่แล้ว

กรณีทำข้อ 2-3 ไม่หลุด loop ให้ตั้งจำนวนรอบมากที่สุดไว้ ดังนั้นค่า  $Y(t+1)$  ล่าสุดเมื่อการทำงานเสร็จสิ้นจะเป็นค่าที่มีความสัมพันธ์กับ input  $X$  มากที่สุด ซึ่งกรณี Unknow input vector  $X$  ที่ใกล้เคียงกับ input vector  $X$  ต้นแบบมากที่สุด (ระยะ Hamming) น้อยที่สุดจะทำให้เทอม  $Y_L X_L X$  ในสมการมีค่ามากที่สุดเพราะ Vector  $X$  ที่ใกล้กับ  $X_L$  ต้นแบบจะเกือบตั้งฉากกัน ทำให้มีค่าใกล้เคียงหรือมากกว่าตัว  $Y X X$  พจน์อื่นๆ ดังนั้นจะส่งผลให้แนวโน้ม output เป็น  $Y_L$  มีมากที่สุด เมื่อเทียบกับ part term อื่น (แต่คำตอบอาจไม่ได้เป็น  $Y_L$  เพราะ crosstalk อาจใหญ่กว่า) โดยถ้า Unknow input vector  $X$  ที่เข้ามามี Error มากจะทำให้ระยะ Hamming มีค่ามากกว่า vector  $X$  ต้นแบบตัวที่ต้องการ ทำให้ Unknow vector ตัวนี้ไปใกล้กับ input vector ต้นแบบของตัวอื่นทำให้ได้ output  $Y$  ที่ผิดออกมา

จากขั้นตอนการทำงานของ BAM สามารถนำมาเขียนโปรแกรมได้ดังนี้

```

x = reshape(x,109*113,1); % ทำ unknow input vector (bipolar) เป็น Matrix หลัก
xold = x; % กำหนด x old
yold = ones(size(y1),1); % กำหนดค่าเริ่มต้น y old
counter = 0;
terminate = 0;

while terminate ~= 1
    nety = (w')*xold; % feedforward part , w คือ ค่า weight
    check_size_y = size(nety);
    for i = 1:1:check_size_y(1,1) % signum function
        if nety(i,1) > 0
            nety(i,1) = 1;
        end
        if nety(i,1) < 0
            nety(i,1) = -1;
        end
    end
    ynew = nety;

    netx = w*ynew; %backward part , feed กลับหา x
    check_size_x = size(netx);
    for i = 1:1:check_size_x(1,1) % signum function
        if netx(i,1) > 0
            netx(i,1) = 1;
        end
    end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if netx(i,1) < 0
    netx(i,1) = -1;
end
end
xnew = netx;

if xnew == xold    % ตรวจสอบเงื่อนไขการเข้าสู่สภาวะเสถียร
    terminate = 1;
else
    xold = xnew;    % update ค่า x , y เพื่อ เก็บเป็นตัวเปรียบเทียบ
    yold = ynew;    % ในรอบต่อไป
end
counter = counter + 1;
end
ต่อไปจะแสดงตัวอย่างในการคำนวณของ BAM

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1 [5]

$$\left. \begin{aligned} x_1 &= [1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1]'; \\ y_1 &= [1 \ -1 \ -1 \ -1 \ -1 \ 1]'; \\ x_2 &= [1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1]'; \\ y_2 &= [1 \ 1 \ 1 \ 1 \ -1 \ -1]'; \end{aligned} \right\} \begin{array}{l} \text{Exemplar No. 1} \\ \text{Exemplar No. 2} \end{array}$$

$$W = y_1 * x_1' + y_2 * x_2' \quad \text{Weight matrix}$$

$$W = \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix}$$

Iteration 1 :

$$\text{net}^y = \begin{bmatrix} 4 \\ -12 \\ -12 \\ -12 \\ -4 \\ 12 \end{bmatrix} \quad Y_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \text{net}^x = \begin{bmatrix} 4 \\ -8 \\ -8 \\ 8 \\ -4 \\ 8 \\ 4 \\ -8 \\ -4 \\ 8 \end{bmatrix} \quad X_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

Final state :  $X_{\text{new}} = X_1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 (ตัวอย่างนี้ไม่ใช้การ transpose W แต่ใช้การ transpose input vector X เป็นเมตริกซ์แถวแทน) [20]

$$\begin{aligned}(x^1)^T &= [1 \ -1 \ 1 \ -1 \ 1 \ -1] & (y^1)^T &= [1 \ 1 \ -1 \ -1] \\ (x^2)^T &= [1 \ 1 \ 1 \ -1 \ -1 \ -1] & (y^2)^T &= [1 \ -1 \ 1 \ -1]\end{aligned}$$

$$W^1 = x^1(y^1)^T = \begin{bmatrix} + & + & - & - \\ - & - & + & + \\ + & + & - & - \\ - & - & + & + \\ + & + & - & - \\ - & - & + & + \end{bmatrix} \quad W^2 = x^2(y^2)^T = \begin{bmatrix} + & - & + & - \\ + & - & + & - \\ + & - & + & - \\ - & + & - & + \\ - & + & - & + \\ - & + & - & + \end{bmatrix}$$

$$W = W^1 + W^2 = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix}$$

$$(x^1)^T = [1 \ -1 \ 1 \ -1 \ 1 \ -1] \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ -4 \\ -8 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$(y^1)^T W^T = W y^1 = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 4 \\ -4 \\ 4 \\ -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = x^1$$

$$(x^*)^T = [1 \ 1 \ 1 \ -1 \ -1 \ -1]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x^*W = \begin{bmatrix} - & + & + & - & - & - \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 4 \\ -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = y^* = y^2$$

$$(y^*)^T W^T = W y^* = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} + \\ - \\ + \\ - \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ -4 \\ -4 \\ -4 \\ -4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = x^2$$

ตัวอย่างที่ 3 ให้ a เป็น input vector และ b เป็น output vector [26]

$$a^{(1)} = [1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^T,$$

$$b^{(1)} = [1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1]^T$$

$$a^{(2)} = [1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1]^T,$$

$$b^{(2)} = [1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^T$$

$$a^{(3)} = [1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1]^T,$$

$$b^{(3)} = [1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1]^T$$

$$a^{(4)} = [-1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]^T,$$

$$b^{(4)} = [-1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W = \begin{bmatrix} 4. & -4. & -2. & 2. & -2. & 4. & -2. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 4. & 0. & 2. & -4. \\ 2. & -2. & -4. & 0. & 0. & 2. & 0. \\ 0. & 0. & -2. & 2. & 2. & 0. & -2. \\ 0. & 0. & 2. & 2. & -2. & 0. & -2. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ 0. & 0. & -2. & -2. & 2. & 0. & 2. \\ -2. & 2. & 4. & 0. & 0. & -2. & 0. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ -4. & -4. & -2. & 2. & -2. & 4. & -2. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 0. & 0. & 2. & 2. & 2. & 0. & -2. \end{bmatrix}$$

$$a^1 = [-1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1]$$

$$b^2 = \text{Sgn}[-16 \ 16 \ 0 \ 0 \ 32 \ -16 \ 0]'$$

$$b^2 = [-1 \ 1 \ 0 \ 0 \ 1 \ -1 \ 0]'$$

$$a^3 = \text{Sgn}[-14 \ -10 \ -10 \ -6 \ -6 \ 2 \ -2 \ 10 \ 2 \ 6 \ 10 \ 10 \ -14 \ -10 \ -10 \ 2]'$$

$$a^3 = [-1 \ -1 \ -1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1]'$$

$$b^4 = \text{Sgn}[-28 \ 28 \ 8 \ -8 \ 40 \ -28 \ 8]'$$

$$b^4 = [-1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]'$$

$$a^5 = \text{Sgn}[-20 \ -10 \ -10 \ -14 \ -10 \ -4 \ -4 \ 10 \ 4 \ 10 \ 10 \ 10 \ -20 \ -10 \ -10 \ 0]'$$

$$a^5 = [-1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]'$$

$$b^6 = b^4$$

$$a^7 = a^5$$

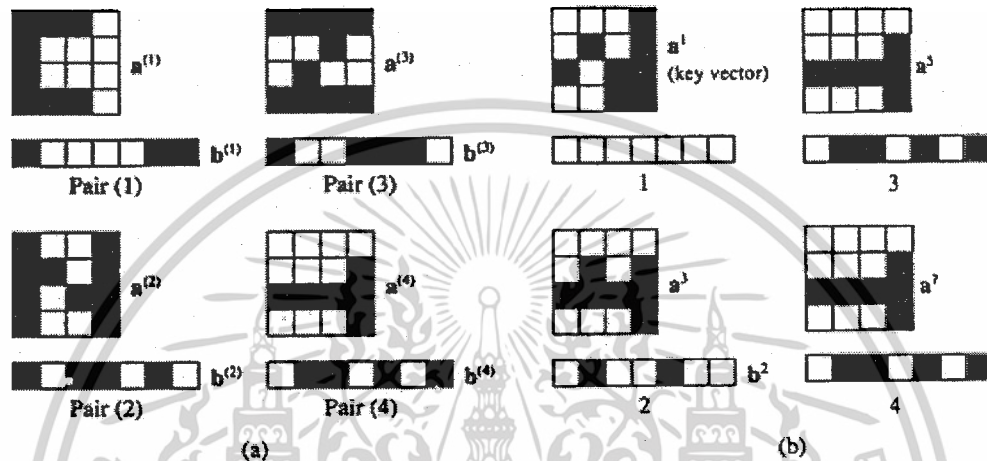
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$HD(a^1, a^{(4)}) = 4$$

$$HD(a^1, a^{(1)}) = 12$$

$$HD(a^1, a^{(2)}) = 4$$

$$HD(a^1, a^{(3)}) = 10$$



รูปที่ 2.35 ตัวอย่างแบบภาพสำหรับ Bidirectional Associative Memory: (a) association stored and (b) four steps of retrieval of pair 4.

Error function ของ BAM สามารถมองในลักษณะของ Energy ได้ โดยที่ network พยายามจะลดค่า energy ให้ต่ำสุดเท่าที่จะทำได้ การพิสูจน์ว่า BAM มีโอกาสเข้าสู่สถานะเสถียร ต้องนิยาม Energy function ของ BAM มาพิสูจน์ [5]

Energy Function

Energy Function ของ BAM ในรูปของ input-output vector สามารถเขียนในรูป

$$E(x,y) = -y^i w x$$

โดย x และ y เป็น input, output vector ใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Energy function ในรูปของ function components คำนวณได้จาก

$$E = - \sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j$$

### The BAM : Energy Function : Lyapunov Function

ทฤษฎีของ Lyapunov กล่าวไว้ว่าเราสามารถหา bounded function ของ Dynamics system ได้ โดยทุก ๆ ครั้งที่ state variables มีการปรับค่าแล้วทำให้ function นี้มีค่าลดลงตลอดเวลา แสดงว่า ระบบนี้สามารถมีค่าตอบที่เสถียรได้

เราเรียก function นี้ว่า Lyapunov function หรือ Energy function ทฤษฎีนี้ประยุกต์ใช้กับ BAM network และ BAM energy function จะได้

1. การเปลี่ยนแปลงใด ๆ ของ  $x$  และ  $y$  ระหว่างการประมวลของ Network จะส่งผลให้ Energy function มีค่าลดลง
2. Energy ของ Network จะมีค่ามากกว่าหรือเท่ากับ  $E_{\min}$  เสมอ โดย

$$E_{\min} = - \sum_{i,j} |W_{ij}|$$

3. เมื่อ  $E$  มีการเปลี่ยนแปลงค่า จะเปลี่ยนแปลงเป็นค่าที่จำกัด (finite value)

### The BAM : Energy Function : Lyapunov Function (cont.)

ทฤษฎีของ Lyapunov หมายความว่า

$$\text{รอบ } 0 \quad E_0 = E(x_0, y_0)$$

$$\text{รอบ } 1 \quad y_1 = \phi(x_0, y_0) \quad x_1 = \phi^{-1}(x_0, y_1) \quad E_1 = E(x_1, y_1)$$

$$\text{รอบ } 2 \quad y_2 = \phi(x_1, y_1) \quad x_2 = \phi^{-1}(x_1, y_2) \quad E_2 = E(x_2, y_2)$$

$$\text{รอบ } n \quad y_n = \phi(x_{n-1}, y_{n-1}) \quad x_n = \phi^{-1}(x_{n-1}, y_n) \quad E_n = E(x_n, y_n)$$

ถ้า  $E_0 \geq E_1 \geq E_2 \geq \Lambda \geq E_n$  ระบบนี้จะมีค่าตอบที่เสถียร (ค่าตอบนี้ไม่จำเป็นต้องถูกต้อง แม้ว่าค่าตอบจะ converge เข้าสู่ค่าที่แน่นอนก็ตาม)

โดย  $\phi(x, y)$  เป็น function ในการหา output (feedforward part)

$\phi^{-1}(x, y)$  เป็น function ในการหา input (feedback part)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### The BAM : Prove of the BAM Energy Theorem

1. การเปลี่ยนแปลงใด ๆ ของ  $x$  และ  $y$  ระหว่างการประมวลผลของ Network จะส่งผลให้ Energy function มีค่าลดลง

พิสูจน์: พิจารณา ที่ component  $y_k$

$$\begin{aligned} E &= - \sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j \\ &= - \sum_{j=1}^n y_k w_{kj} x_j - \sum_{\substack{i=1 \\ i \neq k}}^m \sum_{j=1}^n y_i w_{ij} x_j \end{aligned}$$

ในกรณีที่ component  $y_k$  เปลี่ยนเป็น  $y_k^{new}$

$$E^{new} = - \sum_{j=1}^n y_k^{new} w_{kj} x_j - \sum_{\substack{i=1 \\ i \neq k}}^m \sum_{j=1}^n y_i w_{ij} x_j$$

จะได้การเปลี่ยนแปลงของ energy เป็น

$$\begin{aligned} \Delta E &= E^{new} - E = (y_k - y_k^{new}) \sum_{j=1}^n w_{kj} x_j \\ &= (y_k - y_k^{new}) \cdot net_k^y \end{aligned}$$

โดย  $net_k^y$  คือ output vector ก่อนเข้า signum function

จากสมการของ output node

$$Y_k^{new} = \text{sgn}(net_k^y) = \begin{cases} +1 & net_k^y > 0 \\ y_k & net_k^y = 0 \\ -1 & net_k^y < 0 \end{cases}$$

กรณีแรก  $y_k$  เปลี่ยนจาก -1 เป็น  $y_k^{new} = 1$

$$\begin{aligned} \Delta E &= (y_k - y_k^{new}) \cdot net_k^y \\ &= -2 \cdot net_k^y \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่า  $y_k$  เปลี่ยนจาก -1 เป็น  $y_k^{new} = 1$  ได้ในกรณีเดียวคือ

$$net_k^y > 0$$

ดังนั้น

$$\Delta E < 0$$

กรณีที่ 2  $y_k$  เปลี่ยนจาก 1 เป็น  $y_k^{new} = -1$

$$\Delta E = (y_k - y_k^{new}) \cdot net_k^y$$

เพราะว่า  $y_k$  เปลี่ยนจาก 1 เป็น  $y_k^{new} = -1$  ได้ในกรณีเดียวคือ

$$net_k^y < 0$$

ดังนั้น

$$\Delta E < 0$$

กรณีที่ 3  $y_k$  ไม่มีการเปลี่ยนแปลง

$$\Delta E = 0$$

2. Energy ของ Network จะมีค่ามากกว่าหรือเท่ากับ  $E_{min}$  เสมอ โดย

$$E_{min} = -\sum_{i,j} |w_{ij}|$$

พิสูจน์

$$\begin{aligned} E &= -\sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j \\ &\geq -\sum_{i=1}^m \sum_{j=1}^n |y_i| |w_{ij}| |x_j| = -\sum_{i=1}^m \sum_{j=1}^n |w_{ij}| \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตัวอย่าง Energy Function

พิจารณาตัวอย่างที่ 1 จากค่า  $W$  สามารถหา Minimum Energy ( $E_{\min}$ ) ได้จากสมการ

$$E_{\min} = -\sum_{i,j} |W_{ij}| = -64$$

ต่อไปจะหา Energy ระหว่าง unknow input vector  $x$  และ output vector  $y$  ที่เกิดจาก unknow input vector  $x$  (feedforward part) จากสมการ

$$E(x,y) = -y' \cdot w \cdot x = 40$$

จากนั้นหา  $E(x,y)$  ระหว่าง input vector  $x$  ตัวใหม่ที่เกิดจาก feedback part โดย output vector  $y$  ด้วยสมการ

$$E(x,y) = -y'wx = -64$$

ดังนั้น  $E(x,y)$  ที่คำนวณได้มีค่าเท่ากับ  $E_{\min}$  แล้วคือ  $-64$  แสดงว่า Network อยู่ในสถานะเสถียรแล้วทำให้ output vector  $y$  ตัวล่าสุดเป็นคำตอบที่ได้

<p>Iteration 0:</p> $X_{\text{new}} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$	$y_{\text{new}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$	<p>Iteration 1:</p> $X_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$	$y_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$
<p><math>E = 40</math></p>		<p><math>E = -64</math></p>	
<p><math>E_{\min} = -64</math></p>			

จะเห็นว่า Network จะเข้าสู่สถานะเสถียร เมื่อ Energy เท่ากับ Energy ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ BAM สามารถเก็บ Pattern pair ได้ก็ต้องมีขอบเขตในการเก็บเช่นกัน

Kosko (1988) แสดงให้เห็นว่าขีดจำกัดสูงสุดของ BAM ที่สามารถเก็บ pattern มากที่สุดแล้ว สืบค้นได้ถูก (ให้ output vector ถูกต้อง) คือ  $\min(m,n)$  โดยถ้าจำนวน pattern มีมากกว่า  $n$  จะทำให้ noise มีค่ามากกว่าเทอม signal ดังนั้นจึงสามารถประมาณความจุที่เก็บจำนวน pattern แบบหยาบซึ่งประมาณโดย heuristic คือ

$$p \leq \min(m,n)$$

โดย  $p$  คือจำนวน pattern,  $m,n$  เป็น dimension ของ matrix weight [26]

เพื่อความเป็นแบบแผนสามารถเขียนใหม่ได้เป็น

$$p = \sqrt{\min(m,n)}$$

ดังนั้น ประสิทธิภาพของ BAM ขึ้นอยู่กับคุณสมบัติของ pattern ต้นแบบว่าเปลี่ยนแปลงจากตัวต้นแบบมากแค่ไหน [26]

จึงมีวิธีการที่จะประกันว่า output vector ที่ vecall มาได้ จะมีความถูกต้อง ด้วยวิธี Dummy Augmentation ที่เสนอโดย Wang (1990) [26] แนวคิดของ Wang อาศัยหลักการของ Energy Minima ( $x^i, y^j$ ) โดย  $i = 1, 2, \dots, P$  โดยการสร้างตัว input-output vector ต้นแบบโดยมีเงื่อนไขว่า

$$HD(x^i, y^j) = \frac{n}{2}$$

$$HD(y^i, y^j) = \frac{m}{2}$$

โดย  $i = 1, 2, \dots, P, j = 1, 2, \dots, P$  และ  $x$  คือ input vector  $y$  คือ output vector พิจารณาได้ว่า input vector และ output vector แต่ละตัวต้องมีระยะ Hamming ห่างกันเท่ากับ  $\frac{n}{2}$

และ  $\frac{m}{2}$  ตามลำดับ โดย  $m$  และ  $n$  คือ Dimension ของ matrix weight

อย่างไรก็ตาม ความต้องการที่จะสร้าง augmenting ทั้งคู่ของ vector นั้น จะต้องเพิ่มข้อมูลให้มากขึ้น เพื่อให้เข้าเงื่อนไขของ Dummy Augmentation (Wang (1990)) จากผลการทดลองของ Wang มีเปอร์เซ็นต์ความถูกต้องมากขึ้น แต่ข้อเสียก็คือจะต้องสร้าง Weight ที่ใหญ่ขึ้นเนื่องจากต้องเพิ่มข้อมูลในกลุ่ม input-output vector ให้เข้าเงื่อนไข [26]

Associative Memories นั้นมีข้อดีคือไม่ต้องมีการ train หลายรอบเพื่อให้ได้มาซึ่งค่า weight ที่เหมาะสม โดย Associative Memories จะ train เพียงแค่ 1 ครั้งเท่านั้น แต่ข้อเสียคือ อาจจะไม่เก็บข้อมูลได้น้อยถ้าต้องการความถูกต้องสูง หรือถ้าเก็บ pattern ไว้มากโอกาสที่ input vector จะมีระยะ Hamming ใกล้กับ input vector ต้นแบบตัวอื่นที่ไม่ต้องการมีมากขึ้น

Associative Memories ยังมีอีกหลายชนิด เช่น Associative Memories (FAM), Adaptive BAM (ABAM), Competitive ABAM (CABAM), Radom ABAM (RABAM), Multidirectional Associative Memories (MAM), Temporal Associative Memories (TAM) และอื่น ๆ เหมาะกับงานด้านการเลียนแบบความทรงจำที่ Associate กันของมนุษย์

## 2.9 ทฤษฎีเกี่ยวกับ Network

ความหมายของโปรโตคอลคือ ข้อกำหนดในการสื่อสารของคอมพิวเตอร์หรืออุปกรณ์เครือข่าย ในระบบเครือข่ายขนาดใหญ่อาจมีเส้นทางการเชื่อมโยงระหว่างกันเป็นจำนวนมาก ข้อมูลที่ส่งออกไปอาจไม่ได้ใช้เส้นทางเดียวกันตลอด ดังนั้นจึงมีข้อกำหนดของโปรโตคอล Transmission Control Protocol / internet Protocol (TCP/IP) [2]

### 2.9.1 โปรโตคอล TCP/IP

เป็นโปรโตคอลที่พัฒนาขึ้นมาสำหรับใช้บนอินเทอร์เน็ต โดยที่โปรโตคอล TCP/IP มีการจัดการการทำงานเป็นชั้นหรือ Layer และ จะทำงานเทียบได้กับ OSI หลาย ๆ Layer ปนกันซึ่งในแต่ละ layer ประกอบด้วย Process Layer, Host-to-Host Layer, Internetwork Layer , Network Interface Layer ซึ่งในแต่ละกลไกของโปรโตคอล TCP/IP จะมีโปรโตคอลอื่น ๆ ในชุดของ TCP/IP รวมทำงานอยู่ด้วย

		Layer	
ftp, telnet, mail application	Process Layer	Application	7
		Presentation	6
โปรโตคอล TCP, UDP	Host-to-Host Layer	Session	5
		Transport	4
โปรโตคอล IP	Internetwork Layer	Network	3
		Data Link	2
เคเบิล Ethernet, Token-Ring	Network Interface Layer	Physical	1

TCP/IP Stack

OSI Model

รูปที่ 2.36 TCP/IP และ OSI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.1.1 Process Layer

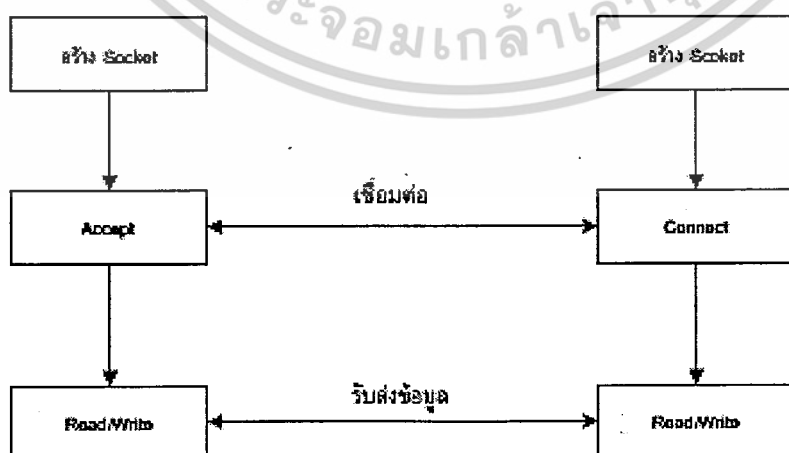
ทำงาน 2 หน้าที่เทียบได้กับ Application Layer และ Presentation Layer ในชั้นนี้จะรองรับการทำงานของแอปพลิเคชันต่างๆและจะรองรับให้โปรโตคอลอื่นทำงานได้หลายโปรเซสและหลายโปรโตคอลได้พร้อมกัน ทำให้ผู้ใช้สามารถเปิดโปรแกรมได้หลายๆอย่างพร้อมกันได้

### 2.9.1.2 Host-to-Host Layer

การทำงานของ Layer นี้จะมีการสร้างการเชื่อมต่อกันระหว่างแอปพลิเคชันกับ Host-to-Host Layer โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า Port หรือ socket และในแต่ละแอปพลิเคชันก็จะสร้างการเชื่อมต่อผ่าน port ได้พร้อมกันหลายแอปพลิเคชัน ซึ่งในการใช้งาน post ของแต่ละแอปพลิเคชันที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้และแต่ละโปรโตคอลจะมีการใช้งาน post หมายเลขต่างๆ ไม่ซ้ำกัน

เมื่อแอปพลิเคชันทำงานผ่านโปรโตคอลในชั้น Process Layer จะมีการส่งผ่านข้อมูลไปยัง Host-to-Host Layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน port ที่กำหนดทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่อง Server ที่ให้บริการจะมีการทำงานอยู่หลายโปรเซสที่แตกต่างกันก็ตาม หรือมีเครื่อง Client เข้ามาใช้งานพร้อมกันจำนวนมากและหลายแอปพลิเคชันในเวลาเดียวกัน ในชั้น Host-to-Host Layer หรือ Transport Layer ของ TCP/IP นี้จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกัน คือ โปรโตคอล TCP และ โปรโตคอล UDP (user Datagram Protocol) ในการส่งผ่านข้อมูลลงไปชั้นถัดไป

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆ ก่อน แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งหายไปก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง



รูปที่ 2.37 ลักษณะการทำงานของTCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล UDP เป็นโปรโตคอลที่มีคุณสมบัติแตกต่างจากโปรโตคอล TCP ทำงานอยู่ใน Host-to-Host Layer ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบทั้งสองด้านไม่ต้องอาศัยการสร้างช่องทางการเชื่อมต่อกัน (connectionless) ระหว่างเครื่อง Sever และเครื่อง Client โดยที่ไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องในการรับส่งข้อมูลนั้นด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณตอบกลับข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูลเมื่อเป็นเช่นนี้แอปพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลจะต้องสร้างกระบวนการตรวจสอบขึ้นมาด้วย



รูปที่ 2.38 ลักษณะการทำงานของUDP

### 2.9.1.3 Internetwork Layer

มีหน้าที่ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำหน้าที่เป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บนอินเทอร์เน็ต คือ โปรโตคอล IP นอกจากนี้ในชั้น Internetwork Layer ยังมีโปรโตคอลอื่นทำงานด้วยอีก 2 ชนิด คือ โปรโตคอล Internet Control Message Protocol (ICMP) และโปรโตคอล Address Resolution Protocol (ARP)

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้จะมีเครือข่ายเชื่อมต่อกันในอินเทอร์เน็ตเป็นล้าน ๆ เครือข่ายก็ตาม เนื่องจากโปรโตคอล IP มีข้อมูลตำแหน่งปลายทางที่จะส่งข้อมูลไปให้โดยทำงานร่วมกันอุปกรณ์ Router เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ โปรโตคอล IP จะทำงานแบบ packet switching คือ มีการส่งข้อมูลผ่าน switch ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่างๆ ผ่านสวิตช์นี้ไปเรื่อย ๆ กว่าที่จะถึงปลายทาง ตัววงจรผ่านหรือสวิตช์นี้อาจเป็น Gateway หรือ Router ในระบบเครือข่ายก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอล ICMP มีหน้าหลัก คือ การแจ้งหรือแสดงข้อความจากระบบเพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ปัญหาส่วนใหญ่ที่พบ คือ ส่งไปไม่ได้ เป็นต้นดังนั้น โพรโทคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบเครื่อง Sever ที่ให้บริการหรืออุปกรณ์ที่อยู่ในเครือข่ายอินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping ที่เราเรียกจะใช้งานโพรโทคอล ICMP แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง

โพรโทคอล ARP ถูกเรียกใช้งานในโพรโทคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น Sever ที่เชื่อมต่อในเครือข่ายอินเทอร์เน็ตในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card หรือ Lan Card ติดตั้งอยู่ที่ Lan Card จะมีหมายเลขประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใครเพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานโพรโทคอล TCP/IP จะต้องมีการกำหนดหมายเลข IP Address ประจำตัวเพื่อใช้อ้างอิงกันและโพรโทคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงในระดับการทำงานที่ Internetwork Layer

กลไกของโพรโทคอล IP ในการส่งข้อมูลหรือ IP Datagram ไปยังเครือข่ายอินเทอร์เน็ตนั้น โพรโทคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่งข้อมูลนั้นจะเป็น เครื่องที่อยู่บนเครือข่ายเดียวกันหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยโพรโทคอล IP จะตรวจสอบ IP Datagram ว่าส่วนที่เป็นค่าหมายเลขเครือข่าย Network Address จะเหมือนกันค่าหมายเลขเครือข่ายของ IP Address ต้นทางหรือไม่ ถ้าค่าเดียวกันแสดงว่าเป็นการส่งข้อมูลภายในเครือข่ายเดียวกัน ถ้าค่าต่างกัน แสดงว่าต้องส่งข้อมูลไปยังเครือข่ายที่คนละเครือข่ายกัน

IP Address ประกอบด้วยเลขฐานสองจำนวน 32 บิต แบ่งออกเป็นสี่ส่วน แต่ส่วนมี 8 บิต เมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0-255) IP Address จะนำเอาหมายเลขทั้งสี่ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุด ดังนั้นหมายเลขทั้งหมดที่เป็นไปได้โดยที่ค่าไม่ซ้ำกัน คือ 256 หรือ 4,294,967,296 จำนวน มีค่าหมายจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้มากำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิง

IP Address แบ่งได้ 2 กลุ่ม คือ กลุ่มใช้งานเป็นรหัสประจำเครือข่าย (Network Address) และกลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Address) IP Address ในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่รหัสกลุ่มประจำเครือข่ายซ้ำกันไม่ได้ ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิง

IP Address สามารถแบ่งได้ 5 คลาส ได้แก่ คลาส A, B, C, D, และ E โดยคลาส A, B, และ C เป็นแอดเดรสสำหรับใช้งานทั่วไป คลาส D ใช้สำหรับกรณีพิเศษ และคลาส E สำรองไว้สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

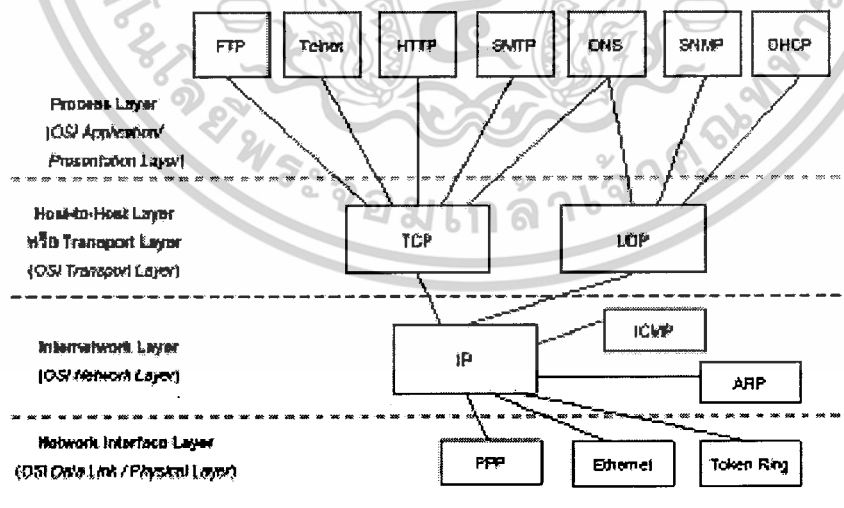
ในอนาคตโดยในที่นี้จะพิจารณาเฉพาะในกรณีของคลาส A, B และ C เท่านั้นการแบ่งคลาส (Class) สามารถแบ่งได้ดังนี้

	MSB		LSB		
Class A	0		Network ID (7)	Host ID (24)	
Class B	1	0	Network ID (14)	Host ID (16)	
Class C	1	1	0	Network ID (21)	Host ID (8)

รูปที่ 2.39 โครงสร้างของคลาส(Class)แต่ละคลาส(Class)

#### 2.9.1.4 Network Interface Layer

การทำงานระดับล่างสุดต่อจาก Internetwork Layer จะเป็นการแปลงข้อมูล IP Datagram ให้อยู่ในรูปแบบที่เหมาะสม และแปลงเป็นสัญญาณไฟฟ้าส่งยังเครือข่ายต่อไปซึ่งในชั้น Network Interface Layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วเป็นการรวม 2 Layer เข้าด้วยกันคือ Data Link Layer และ Physical Layer กล่าวโดยสรุปคือ การทำงานในชั้นต่าง ๆ ตามโครงสร้างของ โปรโตคอล TCP/IP จะมีลักษณะดังรูปที่ 2.40



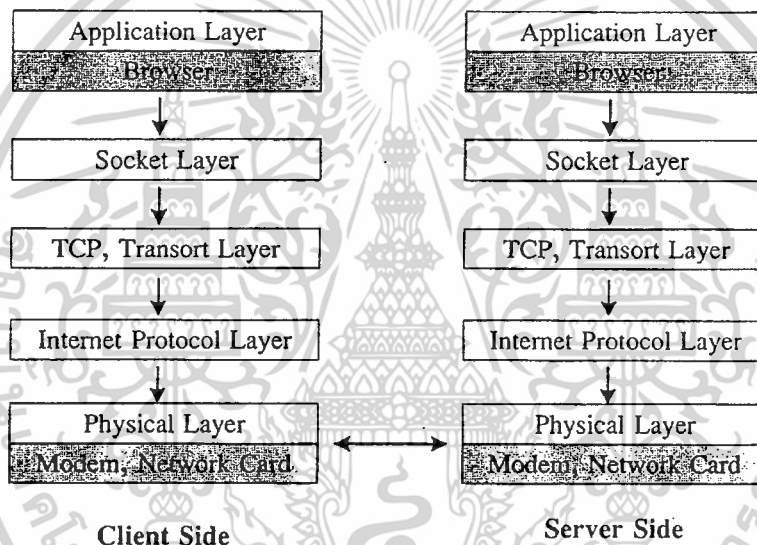
รูปที่ 2.40 โครงสร้างการทำงานของโปรโตคอล TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9.2 ความหมายของ Socket

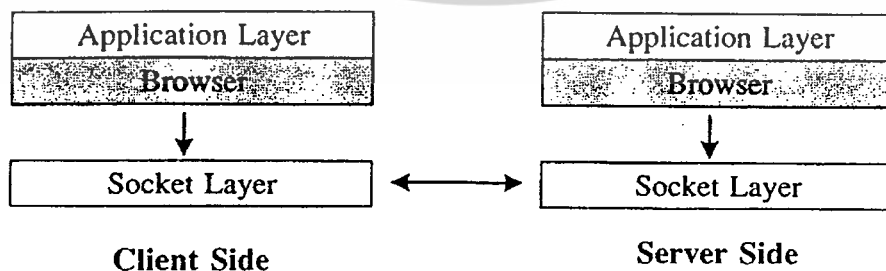
Socket เป็นช่องทางหนึ่งที่จะทำให้เกิดการเชื่อมต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องได้ โดยสามารถรับการเชื่อมต่อจำนวนมากได้ในคราวเดียวกันและสามารถแลกเปลี่ยนข้อมูลกันระหว่างการเชื่อมต่อได้เพราะเนื่องจาก Socket Server เป็นแอปพลิเคชันที่ทำงานตลอดเวลา ซึ่งนำไปใช้กับโปรแกรมการ Chat หรือเกมแบบเล่นได้หลายคน (Multiplayer) เพื่อให้การส่งข้อมูลเป็นไปอย่างต่อเนื่อง

โดยทั่วไปการเรียกดูผ่านทางบราวเซอร์โดยปกติมักมีการติดต่อระหว่าง Client และ Server ก็จริงแต่ไม่ได้ใช้ Socket ในการติดต่อถึงกันโดยตรง การทำงานโดยปกติจะเป็นไปตามรูปที่ 2.41



รูปที่ 2.41 การติดต่อระหว่าง Client และ Server ในการเรียกข้อมูลเว็บผ่านทางบราวเซอร์ [4]

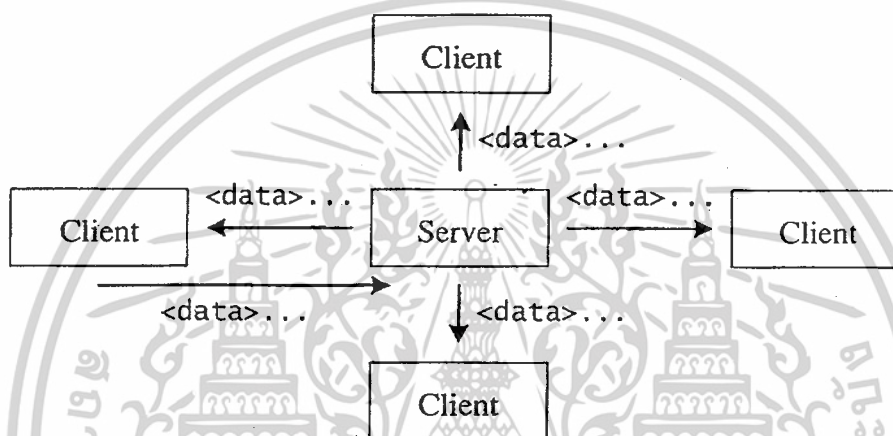
แต่การทำงานของ XML Socket บน Flash จะทำให้ Client และ Server ติดต่อผ่านทาง Socket Layer โดยตรงซึ่งจะทำให้การทำงานเป็นไปอย่างรวดเร็วแสดงดังรูปที่ 2.42



รูปที่ 2.42 การติดต่อระหว่าง Client และ Server ผ่าน Socket [4]

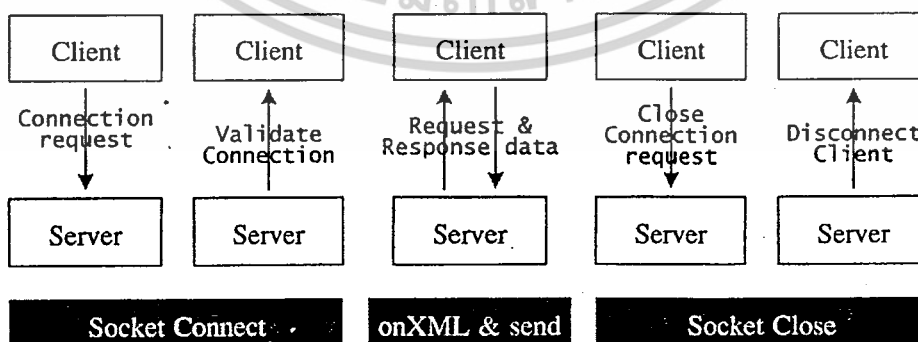
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การทำงานในลักษณะนี้ยังประกอบไปด้วย โปรแกรมฝั่ง Client ซึ่งคือ ActionScript และ โปรแกรมฝั่ง Server ซึ่งมีอยู่หลายภาษาที่มีความสามารถในการทำงานกับ Socket เช่น Java , C หรือ Visual Basic เป็นต้น การทำงานในลักษณะนี้โปรแกรมทางฝั่ง Server จะถูกสั่งให้ทำงานตลอดเวลา เพื่อตรวจสอบการร้องขอจากผู้ใช้งานและเปิดการติดต่อ และการส่งข้อมูลจาก Server กลับไปให้ Client แต่ละเครื่องนั้นก็จะทำงานผ่านทาง Socket ที่ตัว Server จะทำการกระจาย (Broadcast) ข้อมูลไปยัง Client ทุกๆเครื่องพร้อมๆกันเมื่อมีข้อมูลใหม่เกิดขึ้น



รูปที่ 2.43 การส่งข้อมูล(Broadcast) จากServer ไปให้ Client แต่ละตัว [4]

การทำงานระหว่าง Client และ Server แต่ละคู่ผ่านทาง Socket โดยจะอาศัย ออบเจ็ค XML Socket นั้นก็จะมีลักษณะเหมือนกันคือเริ่มจาก Client ขอร้องติดต่อกับ Server เมื่อการติดต่อ(connect)ผ่านทาง Socket สำเร็จ(onConnect) ก็จะอยู่ในขั้นตอนที่มีการรับส่งข้อมูลกัน (onXML และ send) และเมื่อ Client เลิกใช้ก็จะเป็นขั้นตอนการปิด Socket สำหรับการติดต่อระหว่างกัน (Close และ onClose)



รูปที่ 2.44 ขั้นตอนการติดต่อระหว่าง Client และ Server ผ่าน XML Socket [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.3 Client / Server Network

การสื่อสารแบบ Client / Server Network นี้สามารถใช้ได้ทั้งในการต่อแบบ LAN หรือ WAN ตัวอย่างการสื่อสารแบบ Client / Server Network ที่เห็นได้ชัดที่สุดจะเกี่ยวข้องกับระบบฐานข้อมูลจะถูกเก็บไว้ที่เครื่องเน็ตเวิร์กเซิร์ฟเวอร์และ DBMS หากผู้ใช้ที่อยู่เครื่อง Client อยากรู้ข้อมูลที่อยู่ในเครื่อง Server โดยสร้างเงื่อนไขของข้อมูลนั้นขึ้นมาและรีควีสไปยังที่เครื่อง Server ส่วนเครื่อง Server ก็จะส่งข้อมูลที่ประมวลผลได้กลับมายังเครื่อง Client โดยที่เครื่อง Client ก็จะมีซอฟต์แวร์ที่จะจัดการข้อมูลที่ได้นำเสนอออกมาในรูปแบบที่เข้าใจง่าย

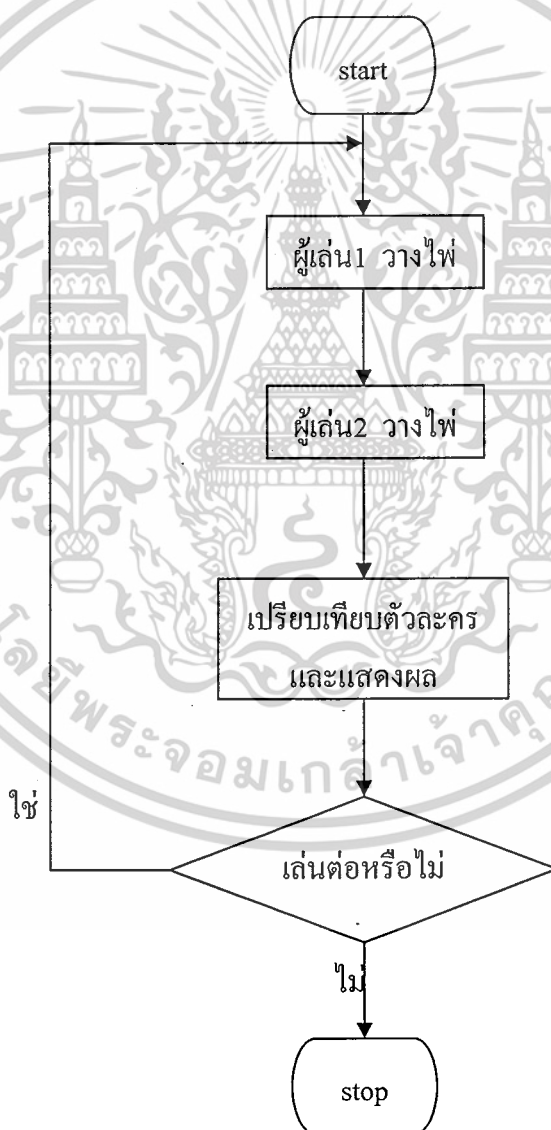


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การออกแบบเกมไฟ Yugi

เกม Yugi'Oh นี้ เป็นเกมที่เล่นได้ 2 คน ผ่านทางคอมพิวเตอร์โดยสื่อสารกันด้วยระบบ LAN รับภาพโดยกล้อง Web Camera ควบคุมโดยโปรแกรม Visual C++ โดยผลิตกันเล่นเป็นเทิร์น จะใช้ไฟจริง ครั้งละ 5 ใบ ในการเล่น จากนั้นตัวละครในไฟจะถกเรียกขึ้นมา เปรียบเทียบกัน ตามกติกาของเกมนี้ ดังนั้นสามารถเขียนโปรแกรมไฟ Yugi'Oh ได้ดังนี้



รูปที่ 3.1 แสดง flowchart ของเกม Yugi'Oh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 ส่วนการออกแบบ

ในการออกแบบเกมนี้ เราจะแบ่งส่วนการทำงานออกได้ดังนี้

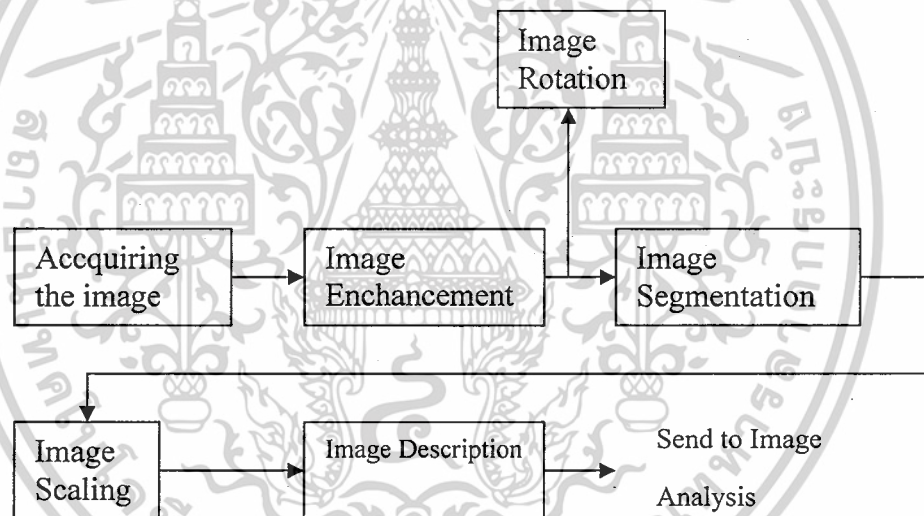
#### 3.1.1. ส่วน Preprocessing

ส่วนนี้รับภาพด้วยโปรแกรม Video for MatLab ของ Farzad Pezeshkpour และใช้หลักการด้าน Image Processing กระทำกับภาพหลังจากการรับภาพมาจนถึงก่อน ส่วนแตกลักษณะเด่น ( Feature Extraction )

#### 3.1.2 ส่วน Image Analysis

ส่วนนี้มีหน้าที่วิเคราะห์ภาพ โดยรับ ลักษณะเด่น จาก ส่วนก่อนหน้า เพื่อวิเคราะห์ว่าไฟโบนี่ชื่ออะไร จากนั้นจึงส่ง ชื่อ ของไฟที่วิเคราะห์ได้ ไปสู่โปรแกรมหลักเพื่อเรียกรายละเอียดต่างๆ ของไฟโบนี่ย้อนออกมาและแสดงผล

ดังนั้นจึงสามารถแสดงส่วนแรกนี้ ในรูป block diagram ได้ดังนี้

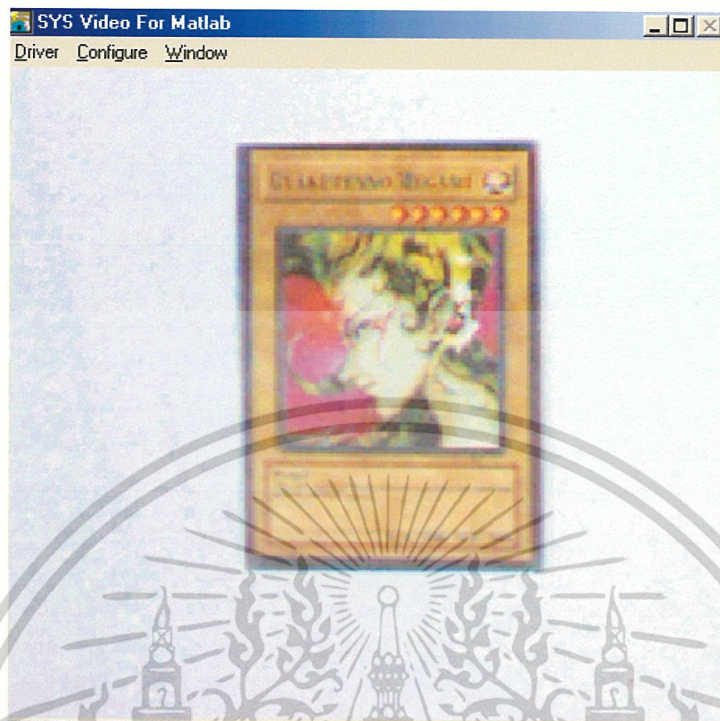


รูปที่ 3.2 แสดง Block Diagram ของส่วน Preprocessing

รายละเอียดของแต่ละบล็อกมีการทำงานดังนี้

#### ส่วน Acquiring the image

รับภาพมาได้เป็นเมตริกซ์ มีขนาดมิติเท่ากับขนาดความละเอียดของกล้อง เป็นภาพสีในระบบ RGB



รูปที่ 3.3 แสดง หน้าต่าง โปรแกรม Video for MatLab ในการ preview ภาพไฟ

### ส่วน Image Segmentation

1. แยกภาพแต่ละช่อง สี RGB เก็บไว้ในแต่ละเมทริกซ์ แปลงภาพในระบบ RGB เป็น ระบบ HSV
2. แปลงภาพเป็น Gray Scale
3. หาค่า Threshold โดยใช้ค่า median ลบด้วยค่า ปรับแต่ง แล้ว Thresholding ภาพนั้น จะได้เป็นภาพ 2 ระดับ ( Binary Scale )
4. นับจำนวนไฟ และหาจุดต่างๆ ของไฟแต่ละใบ ซึ่งจะนำมาเป็นตำแหน่งในการแยกไฟ เนื่องจาก เกมส้นต้องวางไฟทีละ 5 ใบและทำการวิเคราะห์ไฟทีละใบ โดยแยกไฟเป็น 1 ใบ ซึ่งมี Algorithm ดังนี้

### Algorithms การแยกไฟ

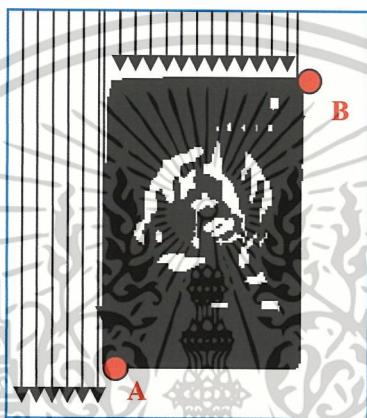
เนื่องจากไฟมีลักษณะการวางไฟ ใกล้เคียง กัน แนวเดียวกัน ในที่นี้จะคิดเพียงแคไฟ 1 ใบ

1. ตรวจสอบจุดดำจุดแรกโดย scan จาก บนลงล่าง ซ้ายไปขวา เพื่อหาจุดซ้ายสุดของไฟ เมื่อเจอให้ save ตำแหน่งนั้นไว้ให้เป็น A แล้วหยุดทำงาน ถ้าไม่เจอก็เพิ่มคอลัมน์ไปทีละ 1 คอลัมน์จนกว่าจะเจอ
2. กลับไปแถวแรก ในคอลัมน์ที่เจอจุดดำจุดแรกใหม่
3. ทำข้อ 1 ซ้ำ จะเจอจุดดำให้ save ตำแหน่งนั้นไว้ (ให้เป็น B)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำข้อ 2 ซ้ำ ณ จุดนี้ทำไปเรื่อย ๆ จนกว่าการ scan จะสิ้นสุดแถว ของ array ดังนั้นแสดงว่าจะได้จุดแรก (จุดซ้ายสุด) (B) และจุดที่ 2 เป็นตำแหน่งขวาสุดของภาพ (เกิดจากการ save ทับ ไปเรื่อย ๆ ตัวสุดท้าย ที่ save ไว้ จะเป็นตำแหน่งขวาสุด (ให้เป็น C)

กรณีไฟ 5 ไบก็ใช้ Algorithm เดียวกัน แต่แยก save จุด ซ้าย, ขวา ของไฟ เป็น 5 ชุด โดยเมื่อได้ของไฟไบแรกแล้วก็ให้เพิ่มคอลัมน์ 1 คอลัมน์ และเริ่ม Algorithm ข้อ 1 ใหม่ ก็จะได้จุดซ้ายขวาของไฟ 5 ไบ



รูปที่ 3.4 แสดง การ scan หาจุดภาพสีดำทางซ้ายสุดและขวาสุดของไฟ

#### Algorithm การหาจุดล่างสุดของไฟ

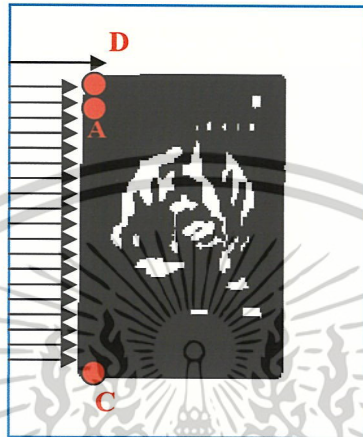
1. ให้เริ่มต้นที่จุดซ้ายสุดของไฟ ไบแรก
2. เพิ่มแถว 1 แถว
  - 2.1 ถ้าจุดนั้นเป็นจุดดำ ให้ save ตำแหน่งนี้ไว้แล้วทำข้อ 2 ใหม่
  - 2.2 ถ้าเป็นจุดขาว ให้บวกคอลัมน์ 1 คอลัมน์ ถ้าเป็นจุดดำให้ save ตำแหน่งนี้ทับตำแหน่งก่อนหน้านั้น(ให้เป็นC) แต่ถ้าเป็นจุดขาว ให้เพิ่มคอลัมน์ไปเรื่อย ๆ จนกว่าจะเจอจุดดำ ในการเพิ่มคอลัมน์นี้ให้พิจารณาว่า เกิน 30 คอลัมน์หรือไม่ ถ้าเกินให้หยุดทำงานไม่ต้อง save ค่า 30 นี้ ใช้เป็นตัวกำหนดขอบเขตของไฟว่าคิดแค่ 1 ไบ เพราะถ้าเกินกว่า 30แล้วเจอจุดดำ แสดงว่าเป็นของไฟไบถัดไป (กรณีนี้ค่า 30 ทดลองที่ 144 x 176) ถ้าความละเอียดมากกว่านี้ ค่าก็จะมากกว่า 30)
3. ทำข้อ 2 ใหม่ จนกว่า จะสิ้นสุดแถวของ array (ในที่นี้ใช้ความละเอียด 144 x 176 ,144 จะเป็นจุดสิ้นสุดของ array)

กรณีไฟ 5 ไบก็ใช้ Algorithm เดียวกันทั้งหมด และจะได้จุดล่างสุดของไฟ 5 ไบ 5 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Algorithm หาจุดบนสุดของไฟ

1. ใช้ Algorithm เดียวกันกับหาจุดล่างสุดของไฟ แต่ใช้การลบแถวทีละ 1 แถวแทน ส่วนในข้อ 3 ของ Algorithm หาจุดล่างสุดของไฟ ให้ทำงานกว่าจะถึงจุดเริ่มต้นแถว (คือแถวที่ 1) (ให้เป็น D) กรณี ไฟ 5 ใบก็ใช้ Algorithm เดียวกัน ดังนั้นจะได้จุดบนสุดของไฟมา 5 จุด



รูปที่ 3.5 แสดง การ scan หาจุดภาพสีค่าทางล่างสุดและบนสุดของไฟ

### การคัดลอกภาพ

ภาพที่เรานำมา segmentation นั้นใช้ภาพ binary scale เพื่อสะดวกในการตรวจสอบจุด แต่ภาพ binary นั้น ยังไม่สามารถนำมาวิเคราะห์ได้ เราจะนำภาพ gray scale มาใช้งานแทน โดยตำแหน่งจุดที่เรา segmentation มาได้จะมีตำแหน่งเดียวกับภาพ gray scale ต่อไปนี้จะเป็นการคัดลอกภาพ gray scale มาโดยกำหนดจากจุดที่หามาได้ก่อนหน้านี้

### การคัดลอกไฟ 1 ใบ

1. ณ ตำแหน่งจุดซ้ายสุด นำคอลัมน์ลบ 1 จะได้จุดซ้ายสุดใหม่ที่คอลัมน์ลดลง 1 คอลัมน์
  2. ณ ตำแหน่งจุดขวาสุดของไฟ นำคอลัมน์บวก 1 จะได้จุดขวาสุดใหม่ที่คอลัมน์เพิ่มขึ้น 1 คอลัมน์
  3. ณ ตำแหน่งบนสุด นำแถวลบ 1 จะได้จุดบนสุดใหม่ที่แถวลดลง 1 แถว
  4. ณ ตำแหน่งล่างสุด นำแถบบวก 1 จะได้จุดล่างสุดใหม่ที่แถวเพิ่มขึ้น 1 แถว
- ณ ตอนนี้จะ ได้จุดต่าง ๆ ของไฟทั้ง 4 จุดมาใหม่ และใช้ Algorithm เดียวกัน ทั้ง 5 ใบ
5. ณ จุดซ้ายสุดใหม่ให้ลบแถว ให้แถวอยู่แถวเดียวกับจุดบนสุดของไฟ จะได้จุดบนซ้ายสุด
  6. ณ จุดซ้ายสุดใหม่ ให้บวกแถวให้แถวอยู่แถวเดียวกับจุดล่างสุดของไฟ จะได้จุดล่างขวาสุด
  7. นำจุดบนขวาสุด ลบกับจุดบนซ้ายสุดเพื่อหาระยะคอลัมน์
  8. นำจุดล่างซ้ายสุดลบกับจุดบนซ้ายสุดเพื่อหาระยะแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. สร้าง array 2 มิติที่มีขนาดแถวเท่ากับระยะแถวของข้อ 9 และขนาดคอลัมน์เท่ากับระยะคอลัมน์ของข้อ 8

10. ไปจุดบนซ้ายสุดแล้ว คัดลอกรูปภาพทั้งหมดเรียงกันไปลงสู่ array ใหม่ที่สร้างขึ้นมา โดยจุดบนซ้ายสุด

กรณีไฟ 5 ไบก็ทำเช่นนี้ 5 ครั้งจะได้ array ใหม่ 5 array เป็นภาพ gray scale และ H, S, V ทั้ง 3 ช่อง



รูปที่ 3.6 แสดงการปรับแถวและคอลัมน์เพื่อหาจุดทแยงมุมทั้ง 3

5. ตรวจสอบความเอียงของไฟและแก้ไขความเอียงโดยการหมุนภาพ ซึ่งมี Algorithm ดังนี้

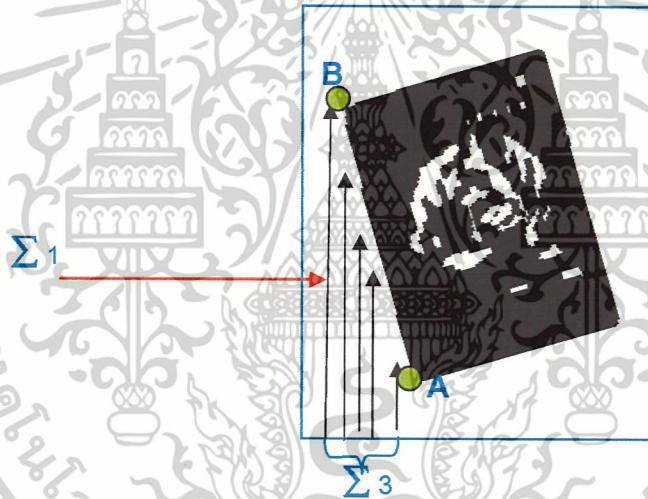
### ส่วน Image Rotation

ไฟยุคโบราณ ผู้เล่น เล่นต้องวางไฟลงบนแผ่นกระดาษหลายครั้ง ซึ่งจะเกิดการเอียงส่งผลต่อการวิเคราะห์ไฟต่อไปได้ จึงต้องตรวจสอบความเอียงและแก้ไขความเอียงนั้น โดยจะอนุญาตให้เอียงได้ค่าหนึ่งที่ยอมรับได้ ถ้าเอียงมากกว่าค่าที่กำหนดนั้นก็จะแก้ไขโดยการหมุนภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

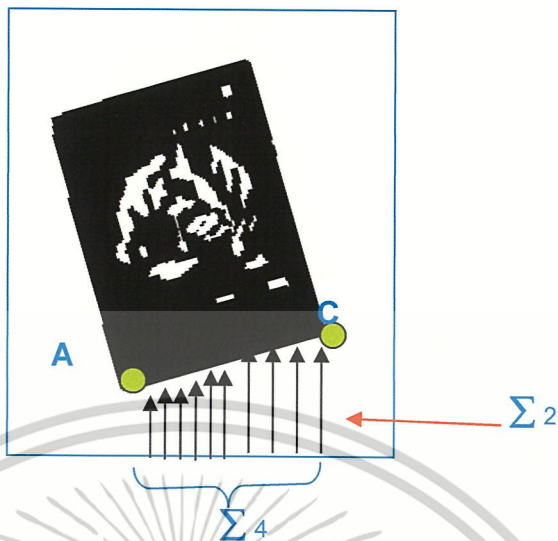
### Algorithm ตรวจสอบความเอียง

1. ณ จุดต่ำสุดของไฟที่หามาได้จากการคัดลอกภาพ ให้ลบคอดีมน์ทีละ 1 คอดีมน์ จนกว่าคอดีมน์ที่ลบนี้เท่ากับคอดีมน์ของจุดซ้ายสุดของไฟโบนนี่ซึ่งจุดซ้ายสุดนี้ได้มาจากการคัดลอกโดยการลบคอดีมน์นี้ให้ นับการลบเพิ่มขึ้นทีละ 1 ด้วย และเก็บผลการนับไปทางซ้ายนี้ไว้ (ให้เป็น  $\Sigma 3$ )
2. ทำเช่นเดียวกับ ข้อ 1 แต่ใช้การบวกคอดีมน์ ทีละ 1 แทน ไปเท่ากับคอดีมน์ของจุดขวาสุด และเก็บผลการนับไปทางขวานี้ไว้ (ให้เป็น  $\Sigma 4$ )
3. ณ จุดที่เท่ากับคอดีมน์ของจุดซ้ายสุดให้ลบแถวจนจุดนั้นเป็นจุดเดียวกับจุดซ้ายสุดของไฟ ทุก ๆ การลบแถวให้นับค่าการลบนี้ด้วยและเก็บผลการนับไว้ (ให้เป็น  $\Sigma 1$ )
4. ณ จุดที่เท่ากับคอดีมน์ของจุดขวาสุดให้ทำเช่นเดียวกับข้อ 3 จนจุดนั้นเป็นจุดเดียวกับจุดขวาสุดของไฟ และเก็บผลการนับนี้ไว้ (ให้เป็น  $\Sigma 2$ )



รูปที่ 3.7 แสดงการนับไปทางซ้ายจากจุดต่ำสุด (A) ไปจนคอดีมน์ตรงกับจุดซ้ายสุด (B) แล้วนับขึ้นจนเป็นจุดเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงการนับไปทางซ้ายจากจุดต่ำสุด (A) ไปจนตลอดนี้ตรงกับจุดขวาสุด แล้วนับขึ้นจนเป็นจุดเดียวกัน

5. นำผลการนับของข้อ 3 และ 4 มาเปรียบเทียบกัน ถ้าผลการนับข้อ 3 มากกว่าผลการนับข้อ 4 และผลการนับทั้งสองนี้มีค่าเกิน 13 (ค่าความเอียงที่ยอมรับได้) แสดงว่าภาพเอียงซ้าย ให้คำนวณหามุมเอียงนี้โดยใช้พีทาโกรัสทางด้านตรงข้ามตรงข้ามมุม โดย

$$\text{ด้านตรงข้ามมุม } \theta = \sqrt{\text{ผลการนับในข้อหนึ่ง}^2 + \text{ผลการนับในข้อสาม}^2}$$

6. คำนวณหามุมเอียงโดย

$$\theta = \cos^{-1} \left[ \frac{\text{ผลการนับในข้อสอง}}{\text{ด้านตรงข้ามมุม } \theta} \right] \left[ \frac{180}{\pi} \right]$$

และทำการหมุนภาพด้วยคำสั่งใน MatLab โดยใช้ค่ามุมเป็น  $-\theta$

7. จากข้อ 5 ถ้าผลการนับข้อ 3 น้อยกว่าผลการนับข้อ 4 และผลการนับทั้งสองมีค่าเกิน 8 แสดงว่าภาพเอียงขวา ดังนั้นเช่นเดียวกับข้อ 5

$$\text{ด้านตรงข้ามมุม } \theta = \sqrt{\text{ผลการนับในข้อสอง}^2 + \text{ผลการนับในข้อสี่}^2}$$

8. คำนวณหามุมเอียง

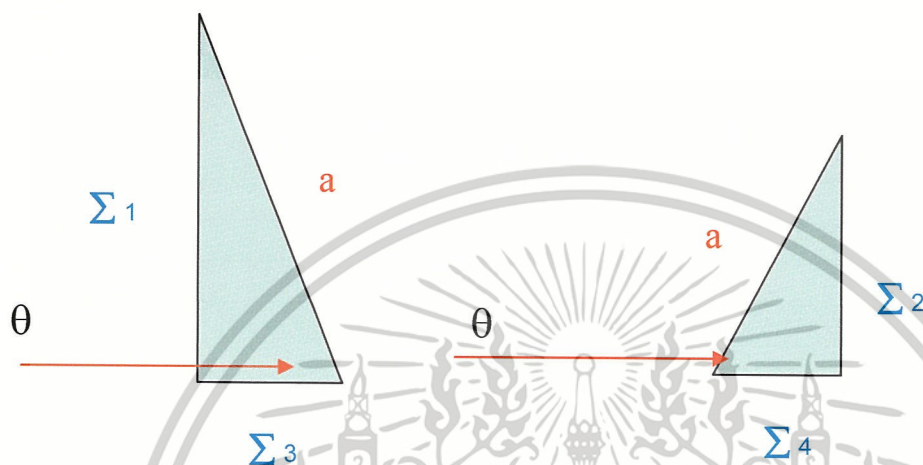
$$\theta = \cos^{-1} \left[ \frac{\text{ผลการนับในข้อหนึ่ง}}{\text{ด้านตรงข้ามมุม } \theta} \right] \left[ \frac{180}{\pi} \right]$$

และทำการหมุนภาพด้วยคำสั่ง ใน MatLab โดยใช้ค่ามุมเป็น  $\theta$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนภาพ โดยจัดการหมุนทั้ง 3 ช่องสี ซึ่งใช้มุมเดียวกันทั้งหมด

กรณีที่ต้องใช้ค่า Threshold ก็เพราะว่าการถ่ายภาพจาก Web Camera นี้ภาพที่เป็น binary scale อาจจะมีจุดดำเล็ก ๆ ประปรายตามขอบของไฟทำให้โปรแกรมมองเห็นว่าภาพนั้นเอียงและคำนวณมุมออกมามากเกินไป ทำให้ผลการหมุนภาพผิดไป



รูปที่ 3.9 แสดงการนับไปทั้ง 4 ซี่ง วาดเป็นรูปจำลอง สามเหลี่ยมพีทาโกรัสเพื่อหามุม  $\theta$

### ส่วน Image Scaling

หลังจากผ่านการปรับปรุงภาพแล้ว จะ Scaling ภาพ ให้อยู่ที่มาตรฐานเดียวกันทุกภาพ ในที่นี้คือ  $320 * 240$  ซึ่งภาพที่ใหญ่หรือเล็กกว่าที่เข้ามาจากกล้อง Web Camera จะถูกเปลี่ยนเป็นขนาดเดียวกัน

1. โดยใช้คำสั่ง `imresize(A, [320 240], 'bilinear');` โดย A เป็น ภาพที่นำมา Scaling
2. ทำการตัดภาพ (Crop) เป็นการตัดภาพจากภาพต้นฉบับมาเพียง 1 ส่วน โดยต้องรู้ระยะการตัดภาพทั้ง 4 ระยะแล้วตัดลอกภาพมา



รูปที่ 3.10 แสดงการตัดภาพ ส่วนที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

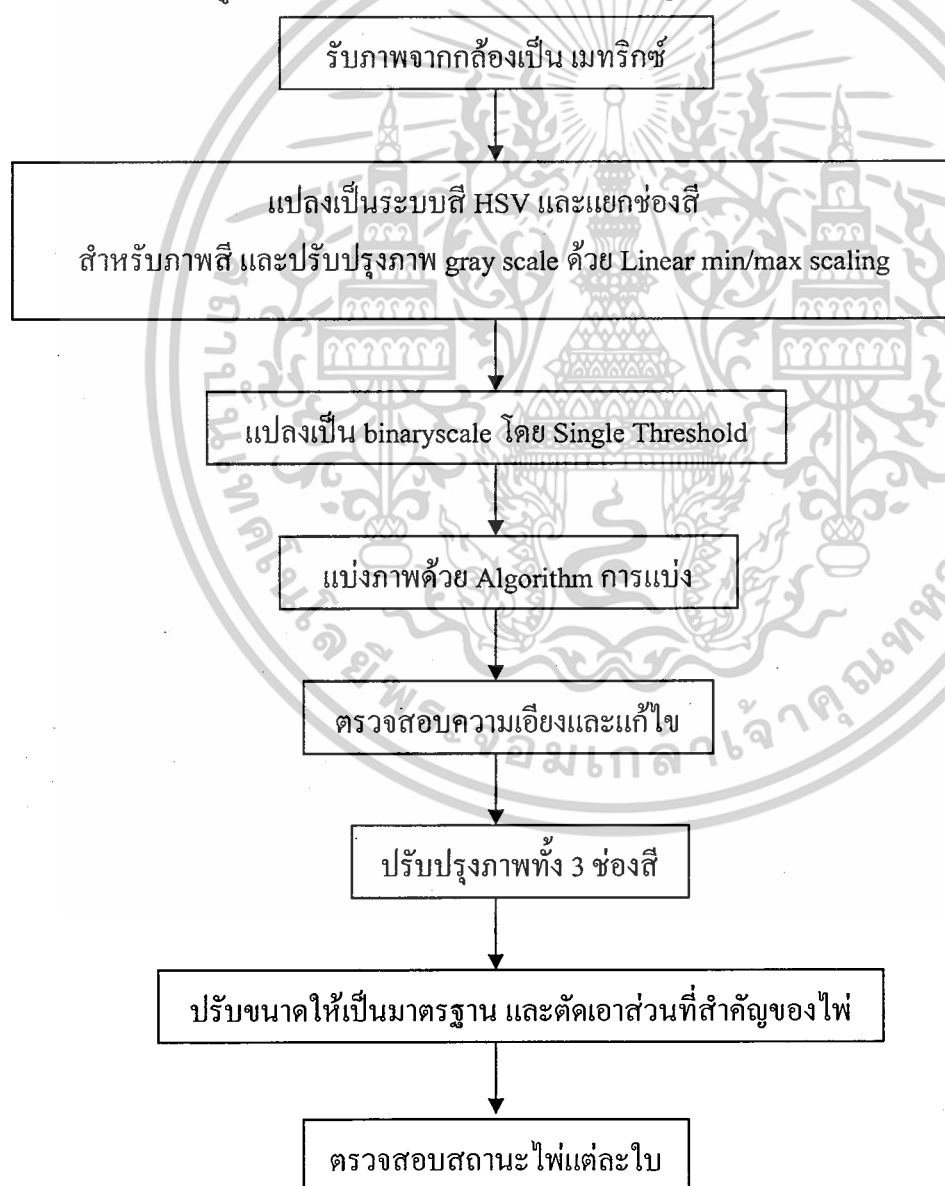
## ส่วน Image Description

เป็นส่วนบรรยายลักษณะของไฟ โดยแบ่งเป็น 2 ส่วนคือ ส่วนวิเคราะห์สถานะไฟ และ ส่วนการแตกลักษณะเด่น (Feature Extraction)

1. ไฟยุคิ ขณะเล่นจะมี 2 สถานะคือ attack และ defend โดย attack จะเป็นภาพแนวตั้ง ส่วน defend จะเป็นภาพแนวนอน การวิเคราะห์สถานะของไฟนี้จะตรวจสอบขนาดของ Matrix ของภาพ binary ที่แบ่งภาพไฟมา 1 ใบแล้ว

ถ้าจำนวนแถวมากกว่าจำนวนคอลัมน์แสดงว่าเป็นสถานะ attack ถ้าน้อยกว่าจะเป็น defend ดังนั้นการทำงานในส่วน Preprocessing นี้ สามารถเขียนเป็น flowchart ได้ดังนี้

รูปที่ 3.13 แสดง flowchart ส่วนของ Preprocessing ที่ทำถึงในตอนที่ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วน Image Analysis

หลังจากตรวจสอบสถานะไฟเรียบร้อย ต่อไปก็จะถึงขั้นตอนสำคัญที่สุด คือ Feature Extraction

Feature Extraction คือ การสกัด ข่าวดสารที่สำคัญจากข้อมูล ซึ่งมีประโยชน์ต่อการรู้จำ ดังนั้น Feature ที่สกัดออกมาจึงควรมีความเป็นหนึ่ง (Unique) ยิ่งมากยิ่งดี ซึ่งก็คือลักษณะเด่นนั่นเองก่อน การ Feature Extraction สิ่งที่สำคัญอีกอย่างก็คือ Feature Selection เนื่องจากมีกรรมวิธีในการสกัด Feature หลายแบบ ดังนั้น จึงสมควรที่จะทดลองหลาย ๆ แบบ เพื่อหา feature ที่มีความเป็นหนึ่งมากที่สุด โดยในการทดลองนี้ ไฟฟูยี ซึ่งเป็นข้อมูลที่เป็นภาพของตัวละครอยู่ตรงกลาง จึงมีขั้นตอนดังนี้

### การตัดภาพ (Crop)

ใช้คำสั่ง `imcrop` ใน MatLab ตัดภาพเอาเฉพาะส่วนที่เป็นภาพตัวละครออกมา โดยการกำหนดขอบเขตการตัดจะใช้วิธีลองผิดลองถูก และ heuristic จนกว่าจะได้ตามขอบเขตที่ต้องการ

### Feature Selection

การวิเคราะห์ feature แบบโครงสร้างโดยที่ใช้ Contour Following หรือ Furier Descriptor ดังนั้น จึงต้องหาโครงสร้าง โดยใช้การหาขอบภาพ (Edge Detection) โดยใช้การ Convolution Mask กับภาพที่ Gray Scale ที่ตัดมา

การวิเคราะห์ feature แบบพิจารณาจุดภาพ วิเคราะห์โดยนำภาพ Gray Scale หรือ R,G,B มาหาผลรวมจุดภาพ แล้วนำผลรวมมาวิเคราะห์ด้วยการ plot graph หรือแตกภาพเป็น Block 9 อัน แล้วหาผลรวมของแต่ละอัน จากนั้นนำผลรวมมาวิเคราะห์ด้วยการ plot graph

การวิเคราะห์ด้วยวิธี Momont Invariant [3][14] Monent Invariant เป็นการบรรยายคุณสมบัติของภาพในเทอมของพื้นที่, ตำแหน่ง, ความหันเห ค่อนข้างเพียงตรง ทนต่อการแปรผันของตำแหน่ง, ขนาด และความหันเห (เช่นภาพในกระจกไม่ว่าภาพอยู่ในตำแหน่งไหน ขนาดเท่าใด ค่าจะมีความใกล้เคียงกันโดย Monent Invariant จะมี 7 Order ค่าแต่ละ order จะมีความใกล้เคียงกัน โดยจะหา Moment ทั้งในระบบ RGB และ HSV ดังนั้นจะได้ 14 ค่า ในแต่ละระบบภาพ และนำมา plot graph

วิธีการสกัด feature ที่กล่าวมาเป็นวิธีพื้นฐานยังมีการสกัด feature แบบอื่นอีก เช่น Discrete Cosine Tranform (DCT), Furier Tranform (FT), Wavelet Tranform (WT), Karhunen\_Loeve Tranform และอื่น ๆ โดยเฉพาะ 2 ตัวหลังให้ผลลัพธ์ดีมาก

เมื่อเลือก feature ที่ดีที่สุดสำหรับงานได้แล้ว ก็ต้องนำมาวิเคราะห์อีกทีโดยการทำแบบซ้ำ ๆ โดยถ่ายรูปไม่ตามตำแหน่งต่าง ๆ หลาย ๆ ครั้ง แล้ววิเคราะห์ว่า feature แปรปรวนไปทับ feature อื่น

หรือไม่ โดยจะลอง 3 ถึง 5 ถึง 10 ครั้ง เมื่อได้ feature ที่ต้องการแล้วก็นำมาเข้าส่วนรู้จำเพื่อจดจำต่อไป

### โครงข่ายประสาทเทียม (ANN)

ใช้โครงข่ายประสาทเทียมในการจดจำ feature ที่ได้มา

#### - Multilayer Feedforward Network

โดยใช้ Back propagation with Variable Learning Rate and Momentum term Algorithm เพื่อเร่งการเข้าสู่จุดที่ Error น้อยสุดที่ยอมรับได้ ตามทฤษฎีในการหลีกเลี่ยง Local Minima [15] โดยป้อน feature เข้าไปทาง input ของ Network (ต้อง Normalisation ด้วยค่าที่มากที่สุดของแต่ละ input ก่อน) แล้วจึง train Network ซึ่งโครงสร้างส่วนใหญ่จะมี hidden layer ไม่เกิน 3 ชั้น และต้องทำการสุ่มจำนวน hidden node ของ โดยในการทดลองนี้จะเริ่มด้วย hidden layer 2 ชั้น จำนวนโหนดให้ค่อนข้างมาก ประมาณ 30-50 โหนด Output มี 1 โหนด ซึ่งถ้า Network สามารถเรียนรู้ได้จนจบ ก็ค่อยๆ ตัดโหนดใน hidden ทั้งที่ละ 1 ถึง 2 โหนด (Network Pruning) จน Network สามารถเรียนรู้ได้หรือขาลง ให้เพิ่มโหนดเข้าไปทีละ 1 โหนด จนกว่าจะเรียนรู้ได้ (Network Growing) [12] หรืออาจใช้สูตรช่วยในการหา Node ที่ตัดก็ได้ [22] หรือสูตรประมาณจำนวน Node ที่จำเป็น [11]

#### - Associative Memories

นำ feature มาเข้ารหัสด้วย BCD Code และสร้าง input/output Vector ขึ้นมาสร้าง Correlation Matrix โดย feature ต้องนำมาหาค่าเฉลี่ยเพื่อสร้าง input vector ดันแบบเพียง 1 อัน

ขั้นตอนต่อมาคือการทดลองแบบซ้ำๆ เพื่อหาไฟโบที่มักจะผิด เช่น ผิดตรงกับอะไร, ทำการแก้ปัญหา ซึ่งขั้นตอนนี้ใช้เวลานานพอสมควร ต้อง plot graph โดยดูคำสั่งการ plot จาก [7]

### GUI (Graphic User InterFace)

ออกแบบด้วย GUI ของ MatLab โดยมีปุ่มที่สำคัญๆ และการแสดงผลของ Web Camera ศึกษาจาก Help ของโปรแกรม และ ที่ถ่ายที่ไฟ โบบทที่ 4 ผลการทดลองตามการวางแผนใน บทที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 ส่วนของเกมที่จะเล่นและการเชื่อมต่อระหว่างคอมพิวเตอร์ 2 เครื่อง ส่วนของฉากแต่ละฉากที่จะใช้แสดงผลของเกม

ในการออกแบบเฟรมของเกมนี้จะต้องสร้างเรื่องราวต่างๆที่จะเกิดขึ้นในเกมว่ามีเหตุการณ์ใดเกิดขึ้นในเกมของเราบ้าง โดยจะสร้างโดยใช้โปรแกรม Macromedia Flash MX โดยจะแบ่งออกได้ดังนี้

1. ออกแบบฉากการเริ่มเข้าสู่เกมโดยรับค่าจากผู้เล่น
2. ออกแบบฉากการเลือกตัวผู้เล่น
3. ออกแบบการรับค่าหมายเลขไฟและสถานะของไฟที่ได้มาจากส่วนรู้จำ
4. ออกแบบฉากการนำภาพของไฟมาแสดงผลและค่าScoreของไฟ
5. ออกแบบฉากแสดงผลที่ได้จากการต่อสู้
6. ออกแบบฉากการจบของเกมว่าฝ่ายใดแพ้หรือชนะหรือออกจากเกม
7. ออกแบบและตกแต่งเพิ่มเติมเพื่อความสวยงาม

#### ส่วนการออกแบบกติกาเพื่อใช้ทดสอบการรับค่าต่างๆ

ในการออกแบบส่วนนี้จะใช้เพื่อทดสอบค่าที่ได้รับมาจากส่วนรู้จำว่าตรงกันกับค่าของเดิมที่เก็บไว้หรือไม่และสามารถทดสอบผลที่ได้จากการต่อสู้ว่าจากนั้นจะออกมาแบบใดและมีผลของเดิมอย่างไร โดยในโปรแกรมของกติกาส่วนนี้จะแบ่งออกเป็น 4 กรณีคือ

กรณีที่ 1 ผู้เล่นฝ่าย A ลงไฟให้สถานะเป็น Attack และผู้เล่นฝ่าย B ลงไฟให้สถานะเป็น Attack จะทำให้

Scoreรวมของแต่ละฝ่าย = Scoreเดิมของแต่ละฝ่ายนั้น - ค่าเต็มAttackของฝ่ายตรงข้าม

กรณีที่ 2 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Attack และผู้เล่น B ลงไฟให้สถานะเป็น Defence จะทำให้

แยกออกเป็น 2 กรณีย่อยๆ คือ

- 1) กรณีที่ค่า Attack มากกว่าค่า Defence

Scoreรวมฝ่ายผู้เล่นB = Score เดิมของฝ่ายB - (ค่าAttackของฝ่ายA- ค่าDefenceของฝ่ายB)

Scoreรวมฝ่ายผู้เล่นA = Score เดิมของฝ่ายA

- 2) กรณีที่ค่า Attack น้อยกว่าค่า Defence

Scoreรวมของแต่ละฝ่าย=Scoreเดิมของแต่ละฝ่ายนั้น

กรณีที่ 3 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Defence และผู้เล่น B ลงไฟให้สถานะเป็น Attack จะทำให้

แยกออกเป็น 2 กรณีย่อยๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) กรณีที่ค่า Attack มากกว่าค่า Defence

Scoreรวมฝ่ายผู้เล่นA = Score เดิมของฝ่ายA - (ค่าAttackของฝ่ายB- ค่าDefenceของฝ่ายA)

Scoreรวมฝ่ายผู้เล่นB = Score เดิมของฝ่ายB

2) กรณีที่ค่า Attack น้อยกว่าค่า Defence

Scoreรวมของแต่ละฝ่าย= Score เดิมของแต่ละฝ่ายนั้น

กรณีที่ 4 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Defence และผู้เล่น B ลงไฟให้สถานะเป็น Defence จะทำให้

Scoreรวมของแต่ละฝ่าย = Score เดิมของแต่ละฝ่ายนั้น

โดยกรณีทั้งหมดนี้ โปรแกรมจะทำการตรวจสอบผลลัพธ์ของScoreรวมของแต่ละฝ่ายว่าฝ่ายใดมีค่าScore ต่ำกว่าศูนย์หรือมีค่าติดลบหรือไม่ ถ้ามีโปรแกรมนั้นจะทำการกระโดดไปยังเฟรมที่เป็นฉากจบของเกมและถามว่าต้องการที่จะเริ่มเล่นเกมนี้ใหม่อีกหรือไม่ ถ้าต้องการเล่นต่อให้ทำคลิกที่ปุ่มเพิ่มเริ่มเกมใหม่

ออกแบบการเชื่อมต่อโดยอาศัยSocket Server

ขั้นตอนนี้จะเป็นการออกแบบการเชื่อมต่อ โดยอาศัย XMLSocket ที่สามารถใช้งานในโปรแกรม Macromedia Flash MX ได้แต่ก่อนอื่นจะต้องทำการรัน โปรแกรมJava ที่เขียนสำหรับทำให้เครื่องที่รันJava อยู่บนเครื่อง Server เพื่อให้ใช้งานในการเชื่อมต่อSocket ได้ โดยมีขั้นตอนการเชื่อมต่อดังนี้

1. ที่เครื่องServer หมายเลขIP Addressคือ192.168.0.1ทำการเปิดport 8000 โดยรันโปรแกรม Server.Javaเอาไว้
2. เปิดโปรแกรมFlash ที่ถูก Publish ให้เป็นไฟล์ที่มีนามสกุล .HTML หรือ .swf
3. เริ่มเข้าสู่เกม
4. เลือกตัวละครที่จะเล่น ส่วนนี้จะเป็นการตรวจสอบUser เพื่อไม่ให้เลือกซ้ำตัวเดียวกัน โดยจะส่งค่าของUserคนแรก que เลือกไปยังServer
5. รอกการเชื่อมต่อจากอีกฝ่ายหนึ่งถ้ามีการเลือกตัวละครที่จะเล่นซ้ำกันให้ฟ้อง เมจเสกขึ้นมาเพื่อเลือกใหม่อีกครั้ง
6. เมื่อเชื่อมต่อทั้งสองฝ่ายแล้วจึงเริ่มเฟรมที่วางไฟ
7. ส่งค่าเต็มของไฟและเมจเสกต่างๆในเกมของแต่ละฝั่งเพื่อนำไปคำนวณต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ขั้นตอนการทดลองและผลการทดลอง

การทดลองจะเริ่มจาก Feature Selection ตามขั้นตอนในบทที่ 3

#### การทดลองที่ 1

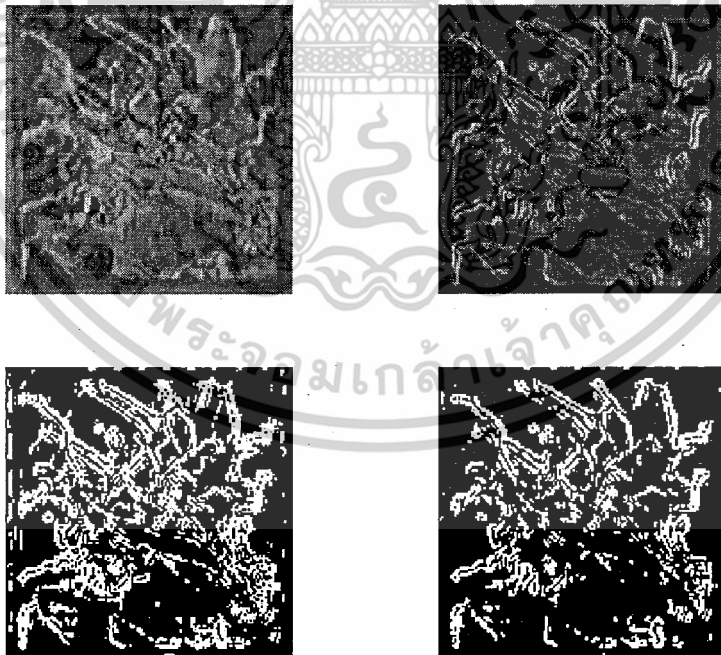
การหาขอบภาพ เพื่อที่จะให้ได้โครงสร้างของภาพ

#### ขั้นตอนการทดลอง

1. นำ Maak Sobel มา Convolution กับภาพ gray scale ที่ตัดมา
2. แปลงผลที่ได้เป็น binary Scale โดยใช้วิธี Bilevel Luminance Thresholding

#### ผลการทดลอง

จะเห็นว่า โครงสร้างของขอบที่ได้ไม่ต่อเนื่อง ไม่เป็น loop ปิดและเมื่อแก้ค่า threshold ให้มากขึ้นจะทำให้เพื่อตัดจุดประปรายไม่สำคัญทิ้ง จะทำให้ขอบภาพอื่นขาด เมื่อลดค่า threshold ลงจะทำให้เพื่อให้ขอบติดกันจะเพิ่มจุดภาพประปรายมากขึ้นดังนั้นการวิเคราะห์โครงสร้าง ไม่ได้ผล



รูปที่ 4.1 แสดงภาพก่อนการหาขอบ , หลังการหาขอบ , thresholding 0.2 , thresholding 0.1

ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทดลองที่ 2

การพิจารณาคุณภาพแบบ gray scale

### ขั้นตอนการทดลอง

1. นำภาพ gray scale มาหาผลรวมจุดภาพ แล้วเปรียบเทียบกันหลายๆ ภาพ โดยผ่าน Histogram Equalisation ก่อน
2. พิจารณาค่า ผลรวมนั้น

### ผลการทดลอง

ผลรวมจุดภาพ ไฟเบอร์ 7      2.2535e+004

ผลรวมจุดภาพ ไฟเบอร์ 10      2.5520e+004

ผลรวมจุดภาพ ไฟเบอร์ 3      2.2953e+004

ค่าผลรวมจุดภาพที่มีค่าใกล้เคียงกัน ดังนั้น การหาผลรวมจุดภาพแบบนี้ไม่ได้ผล

## การทดลองที่ 3

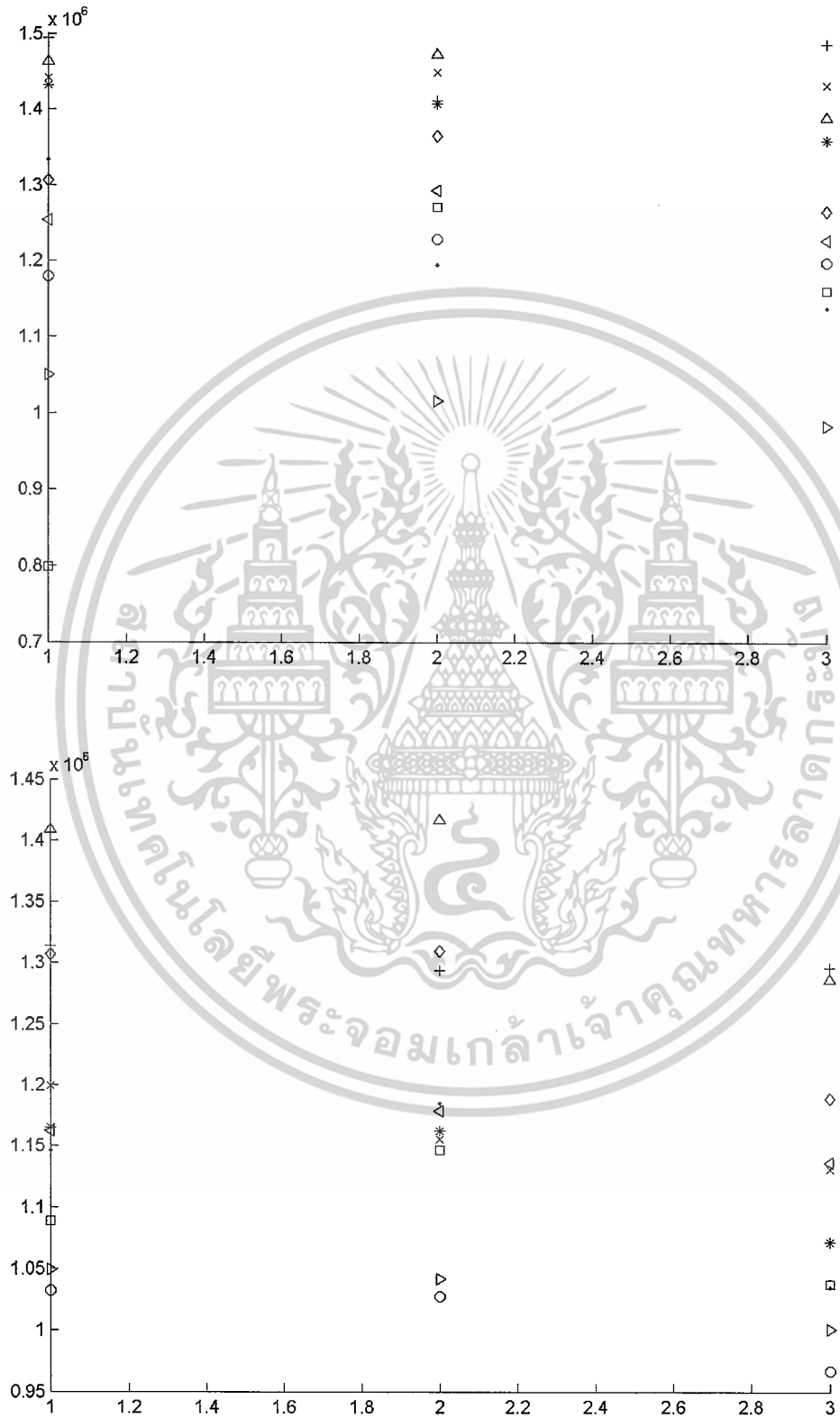
การวิเคราะห์ผลรวมจุดภาพแบบ R,G,B ทำ 3 ครั้ง

### ขั้นตอนการทดลอง

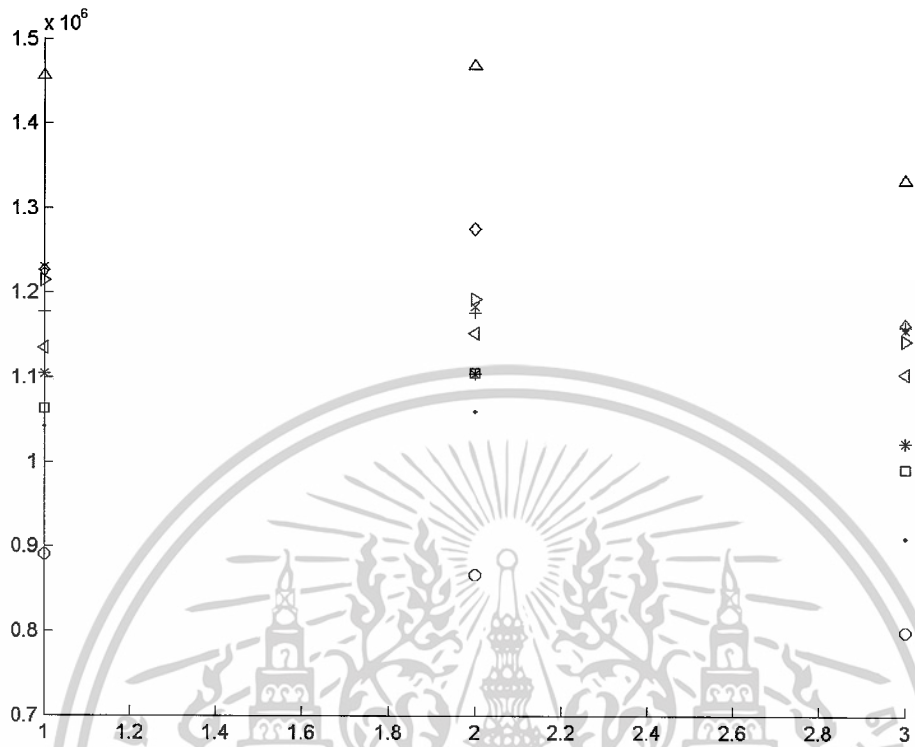
การทดลองเหมือนการทดลองที่ 2 แต่วิเคราะห์ 3 ภาพ R,G,B

### ผลการทดลอง

เมื่อ plot graph และคำนวณจุดภาพ 3 ครั้ง จะเห็นว่าเริ่มมีการทับกัน สังเกตว่าไฟ 1 ไบเป็น 1 สัญลักษณ์ ทั้งหมด 10 ไบ สัญลักษณ์ ค่อนข้างใกล้กัน ไม่เกาะกลุ่ม (ต้องมองโดยรวมเป็น 1 แถว จาก 3 แถว)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงผลรวมของ ไฟล์ 3 ใบ ที่ถ่าย 3 ครั้ง เฉพาะสีแดง สีเขียว สีน้ำเงิน ตามลำดับ โดยแกน x คือ R G B สามสี แกน y คือค่าผลรวม

#### การทดลองที่ 4

การวิเคราะห์ผลรวมของรูปภาพ แบบ R,G,B แบบขอย่อย เป็น Block

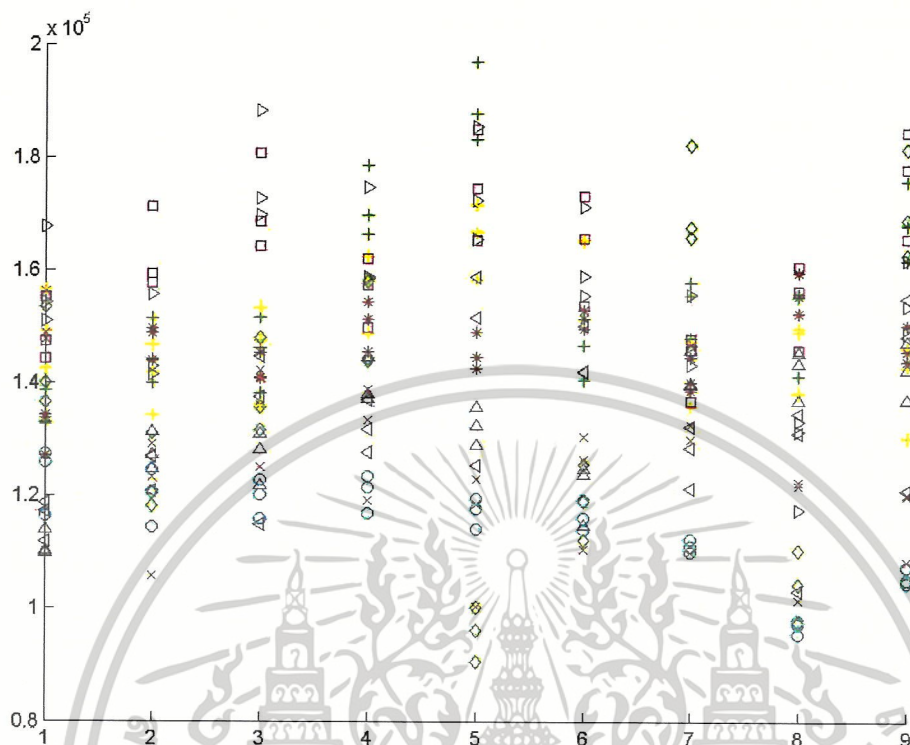
##### ขั้นตอนการทดลอง

1. แบ่งภาพเป็น 9 ส่วน ย่อย เท่า ๆ กัน แล้วหาผลรวมแบบ R,G,B
2. นำผลมา Plot graph

##### ผลการทดลอง

ผลรวมของแต่ละภาพค่อนข้างต่างกัน แต่เมื่อถ่ายภาพ 3 ครั้งค่าผลรวมจะเริ่มทับกันมากขึ้น แสดงว่ามีรบกวน feature อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง ผลรวม จุดภาพ เฉพาะสีแดง ที่แบ่งเป็น 9 block แกน x คือ จำนวน Block แกน y คือค่าผลรวมของแต่ละ block

### สรุปผลการทดลอง

การทดลองที่ 4 ก่อนข้าง ได้ผล แต่ว่า เมื่อถ่ายภาพหลายครั้งมีการรบกวนระหว่าง feature นี้อย่างมากจากแสง

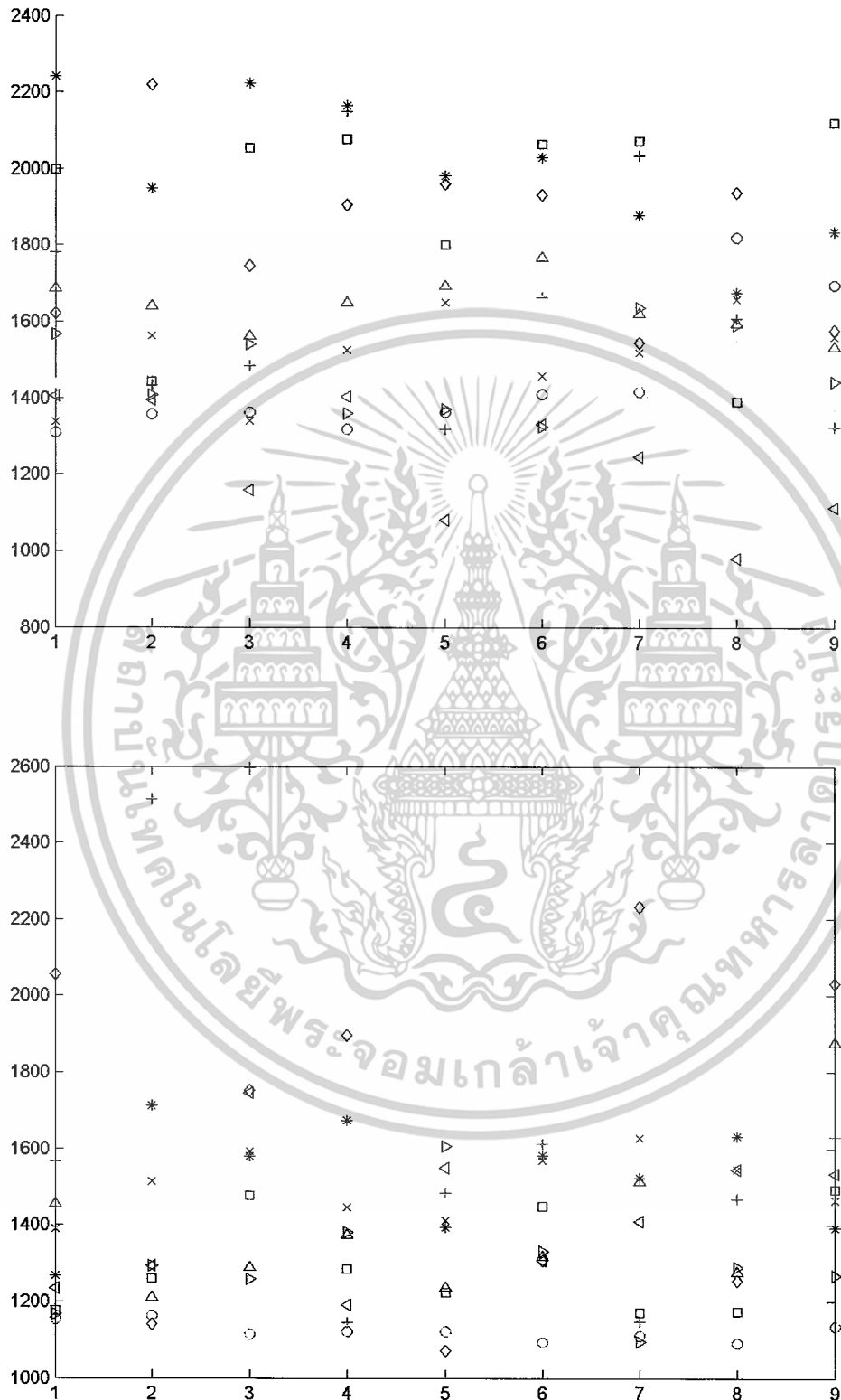
### การทดลองที่ 5

การหาผลรวมจุดภาพแบบอัตราส่วน (Ratio)

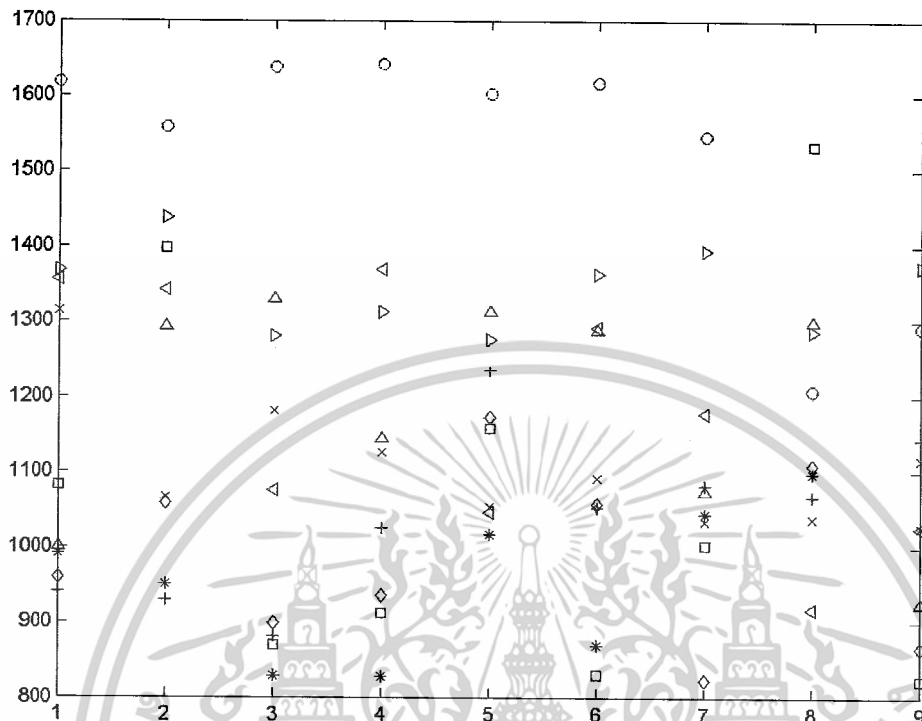
#### ขั้นตอนการทดลอง

1. นำภาพ R,G,B มาหารกันเชิงสมาชิก (จุดต่อจุด (ใช้เครื่องหมาย ./ ใน MatLab))  
เช่น R/G , G/B , B/R
2. หาผลรวมของจุดภาพแล้ว plot graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดง ผลรวม จุดภาพ สีแดง สีเขียว สีน้ำเงิน ที่แบ่งเป็น 9 block ตามลำดับ แกน x คือ จำนวน Block แกน y คือค่าผลรวมของแต่ละ Block

#### ผลการทดลอง

จะสังเกตว่า ก่อนข้างที่จะแยกชั้น ได้ดี กว่าแบบผลรวม แต่ มี feature บางตัวใกล้เคียงกัน ถ้า plot มากกว่า 1 ครั้ง จะเห็น การทับกันของ feature

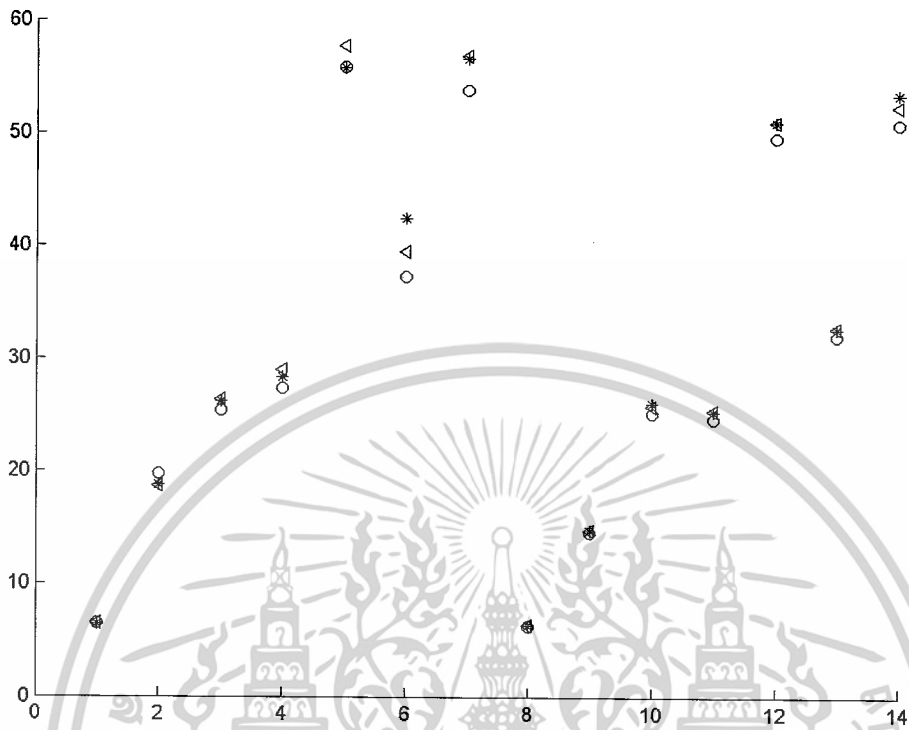
#### การทดลองที่ 6

Moment Invariant RGB system

#### ขั้นตอนการทดลอง

1. นำภาพมาหา Moment Invariant ทั้ง 7 order ของภาพ RGB
2. พิจารณาค่า Moment ที่ได้มา
3. plot graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดง Moment Invariant 14 ค่าของไฟ้ 3 โบ 1 ครั้ง ของช่อง R และ G ในระบบ RGB

#### ผลการทดลอง

Plot เพียง แค่ 3 โบ 1 ครั้ง สังกัดว่า ค่า Moment Invariant มีค่าใกล้เคียงกันมาก สังกัดว่า Order 6 ของ R และ Order 7 ของ G แยกกัน แต่ ก็น้อยเกินไป

#### สรุปผลการทดลอง

จะเห็นว่าแสงนั้นจะรบกวนภาพระบบ RGB มากทีเดียว ถึงตอนนี้พอสรุปได้ว่า ภาพในระบบ RGB ไม่ทนต่อความแปรปรวนของแสง จึงจะใช้ระบบ HSV แทน

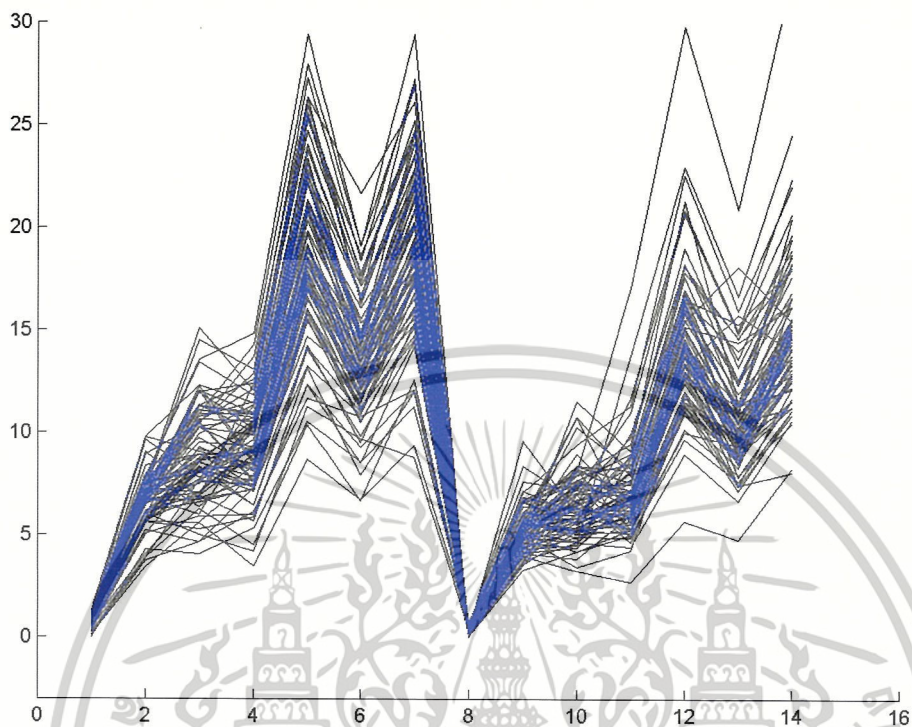
#### การทดลองที่ 7

Moment Invariant HSV system

#### ขั้นตอนการทดลอง

ขั้นตอนแบบเดียวกับการทดลองที่ 6 แต่ใช้ภาพระบบ HSV โดยใช้แค่ช่อง H และ S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดง Moment Invariant 14 ค่าของไฟ 81 ใบ 1 ครั้ง ของช่อง H และ S แกน x คือ จำนวน Order แกน y คือ ค่า Moment

#### ผลการทดลอง

เมื่อ plot จำนวนใบน้อยๆ จะเห็นว่า มัน แยกชั้น ได้ดี จึงทำการ plot หลายๆ ใบจะเห็นว่า Moment ใน order 5 และ 7 ของทั้ง 2 ช่อง มีการแยกชั้นกัน แสดงว่าภาพในระบบนี้ ทนต่อความแปรปรวนแสงพอสมควร

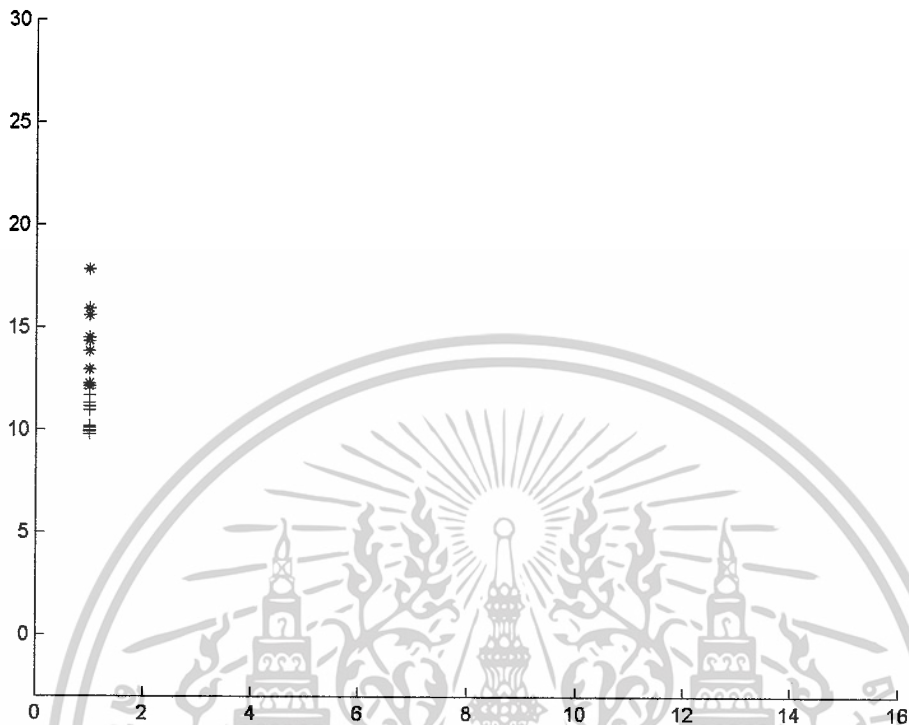
#### การทดลองที่ 8

ทดสอบ feature โดยถ่ายภาพหลายๆ ครั้ง

#### ขั้นตอนการทดลอง

1. หาค่า Moment Invariant เฉพาะ order 5 และ 7 โดยถ่ายภาพ หลายๆ ครั้ง
2. plot graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดง Moment Invariant order 5 ของไฟ 2 ใบ 7 ครั้ง ของช่อง H

#### ผลการทดลอง

ค่า feature มีการรบกวนกันของ feature แต่ในที่นี้พิจารณาเพียง 2 ใบ ถ้าสังเกตควบคู่กับ graph ของผลการทดลองที่ 7 จะเห็นว่ามีการรบกวนกันของ feature

#### สรุปผลการทดลอง

จะเห็นว่า การทดลองที่ผ่านมา feature ล้วนแต่มีความแปรปรวน ไปรบกวน feature อื่น แต่ผลจากการทดลองที่ 8 นั้นมีความแปรปรวนน้อยสุด ดังนั้นจะลองเอา feature มาเข้าโครงข่ายประสาทเทียมแบบ Multilayer Feedforward Network ในขั้นตอนต่อไป

#### การทดลองที่ 9

Multilayer Feedforward Network

##### ขั้นตอนการทดลอง

1. กำหนด parameter ต่าง ๆ เช่น Learning rate , จำนวนรอบ Momentum term , Error รวม โดย ใช้ hidden layer 2 ชั้น เป็น 130,230 ตามลำดับ
  2. ทำการ train ด้วย feature order 5,7 ของ hue และ order 7 ของ Saturate
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ผลการทดลอง

Network สามารถ train เข้าสู่ global Minima ได้ แต่การทดลอง test ด้วยข้อมูลใหม่ นั้น Network ให้ค่าที่ผิดพลาดออกมาแทบทุกครั้ง เนื่องจาก feature ใหม่ ห่างจาก feature ตัวอย่างที่ train

### สรุปผลการทดลอง

เมื่อพิจารณาจาก graph ใน การทดลองที่ 7 และ 8 ก็จะทำให้เห็นว่า เส้นการตัดสินใจของ ANN ตัดตรงรอยต่อระหว่าง feature ที่ใกล้กัน แต่ประสิทธิภาพของ Network จะต่ำมากเนื่องจาก ระยะจากข้อมูล ถึงเส้นตัดสินใจน้อยมาก เมื่อมี unknow เข้ามา จะทำให้ ไปตกในอีกด้านหนึ่งของเส้นตัดสินใจ ทำให้ตัดสินใจผิด

แสงเป็นตัวการสำคัญที่ทำให้ feature มีความแปรปรวนมาก ดังนั้น จึงต้องหาวิธีใหม่ คดขยการวิเคราะห์ ภาพ แบบ Binary โดยอาศัยการวิเคราะห์จากลูปภาพ เพราะเมื่อแสงมีผลต่อ feature ภาพที่ถูกแปลงเป็นแบบ Binary จะหักล้างความสว่างซึ่งแปรปรวนอยู่ตลอดได้ แต่ภาพ binary ก็จะมีการเปลี่ยนแปลงจุดขาวดำอยู่บ้าง โดยนำภาพ Binary นี้มาสร้าง Correlation Matrix ของ Bidirectional Associative Memory

### การทดลองที่ 10

ทดลองโปรแกรม ของ BAM

#### ขั้นตอนการทดลอง

1. เขียน โปรแกรมตาม Algrithm ของ BAM
2. ลอง test กับตัวอย่างใน Text book

### ผลการทดลอง

เป็นไปตามตัวอย่างใน Text book

### การทดลองที่ 11

ทดลองนำภาพ Binary มาใช้กับ BAM

#### ขั้นตอนการทดลอง

1. นำภาพ binary มาแปลงเป็น input vector โดยทำเป็น Matrix หลัก และ เปลี่ยนค่าที่เป็น 0 ให้เป็น -1
2. สร้าง out put vector ขึ้นมาเอง โดยจะทดลองแค่ 3 ใบก่อน
3. นำมาป้อนเข้า Network และทดลอง test หลาย ๆ ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

Network ให้ out put ถูกทั้ง 3 ไบ ทุกครั้ง

**การทดลองที่ 12**

เพิ่มจำนวน pattern ให้ Network

ขั้นตอนการทดลอง

เพิ่ม pattern ให้กับ Correlation Matrix เป็น 5 ไบ และ 10 ไบ ตามลำดับขั้นตอนการทดลอง เหมือนกับการทดลองที่ 11 แต่เพิ่มการสร้าง input /output vector

ผลการทดลอง

การทดลอง 5 ไบ และ 10 ไบ ทุกไบ ทุกครั้ง

**การทดลองที่ 13**

เพิ่ม pattern เป็น 20 pattern และ output vector เป็นการเข้ารหัส แบบ BCD Code

ขั้นตอนการทดลอง

เหมือนกับการทดลองที่ 12

ผลการทดลอง

มีไฟไบที่ผิดคือ 4 ผิดเป็น 2, 2 ผิดเป็น 3) 13 ผิดเป็น 3, 14 ผิดเป็น 0 ม, 16 ผิดเป็น 1, 17 ผิดเป็น 0, 18 ผิดเป็น 0, 19 ผิดเป็น 3, 20 ผิดเป็น 3

สรุปผลการทดลอง

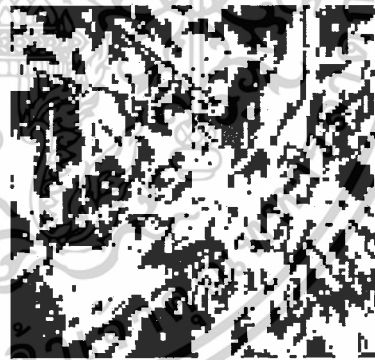
เมื่อมีการเพิ่ม pattern มากขึ้น ทำให้ความเสถียรภาพ Network ลดลง เพราะโอกาสที่ระยะ hamming ของ unknow input ไปใกล้เคียงกับ input vector ตัวอื่น ของตัวที่ไม่ต้องการ ซึ่งเป็นปกติของ BAM เช่น ไฟไบที่ 4 มีระยะ hamming ใกล้เคียงกับไฟไบที่ 2 เมื่อ unknow ของไฟไบที่ 4 เข้ามา ซึ่งเกิด error จากการถ่ายภาพ ทำให้ไปใกล้เคียงกับ Vector input ต้นแบบไฟไบที่ 2

จากการวิเคราะห์ภาพ Binary ที่ถูกสร้างจากgray scale ภาพไฟไบที่ Network ให้ Input ถูกต้อง มักจะมีลักษณะภาพที่เป็นกลุ่มก้อน เป็นเนื้อเดียวกันมาก ๆ ส่วนภาพในที่ผิดมักจะเป็นภาพที่พื้นหลัง ลาย ๆ โครงสร้าง ไม่ค่อยเป็นกลุ่มก้อน เช่น



รูปที่ 4.8 แสดงภาพ binary scale ที่มักจะให้  
output ถูกต้อง

รูปที่ 4.9 แสดงภาพ binary scale ที่มักจะ  
ให้output ผิด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงมีแนวคิดที่จะใช้ภาพในระบบ HSV ช่อง Hue มาทำเป็น Binary เพราะทนต่อการเปลี่ยนแปลงของแสงกว่า ภาพจากระบบ RGB และภาพที่มีลักษณะเป็นกลุ่มก้อนกว่า Binary ที่มาจาก RGB



รูปที่ 4.10 แสดงภาพ binary scale ที่ Thresholding จาก Hue

#### การทดลองที่ 14

การหาค่า threshold ของภาพ HSV ช่อง Hue

##### ขั้นตอนการทดลอง

ทดลองแทนค่าคงที่มา 1 ค่า แบบลองผิดลองถูกแล้วใช้การหาค่าต่อไปแบบ Heuristic จนกว่าจะได้ภาพที่ binary เป็นกลุ่มก้อนซึ่งอาจต้องทดลองโดยป้อนเข้า Network ละดูอัตราการผิด

##### ผลการทดลอง

ได้ค่า threshold เท่ากับ 0.66 โดยทดลองที่ 0.5, 0.7, 0.65 ตามลำดับ 0.66 ให้อัตราการผิดน้อยที่สุด

#### การทดลองที่ 15

ทดสอบป้อน binary ที่ได้จากการทดลองที่ 14 เข้า BAM

##### ขั้นตอนการทดลอง

วิธีการเหมือนการทดลองที่ 13

##### ผลการทดลอง

มีไฟฟัด 3 ใบจาก 20 ใบคือ 18 ผิดเป็น 22, 19 ผิดเป็น 3 และ 20 ผิดเป็น 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### สรุปผลการทดลอง

จะเห็นว่ามีการผิดพลาด กว่าเดิมมาก เพราะฉะนั้นจะใช้ภาพช่อง Hue ในการสร้าง input vector ต้นแบบ แต่มีการผิดนอกเหนือจาก output vector (22)

จากผลการทดลองที่ได้จึงเกิดแนวคิดของการตัดตัวเลือกขึ้น ซึ่งมีแนวคิดมาจาก Decision tree ดังนั้น จะนำภาพที่ผิดในช่อง Hue มาวิเคราะห์ในแบบ binary scale ที่ได้จาก gray scale ของ RGB อีกที แต่ภาพที่ถูกเช่นในกรณีการทดลองที่ 15 3 และ 4 จะต้องถูกนำมาคิดด้วย (ใช้สร้าง Correlation Matrix รวม) ดังนั้นภาพที่ถูกก็จะถูก ป้อนเข้าการวิเคราะห์แบบ binary ที่ได้จาก gray scale ของ RGB ด้วย

### การทดลองที่ 16

ทดสอบแนวคิด Decision tree

#### ขั้นตอนการทดลอง

1. นำหมายเลขภาพไฟที่ผิดจากการทดลองที่ 15 มาสร้าง Correlation Matrix
2. ทำการป้อนเข้า Network

#### ผลการทดลอง

จากไฟใบที่ผิดจากการทดลองที่ 15 คือ 3,4, 18, 19  
ใบที่ผิดคือ 19 ผิดเป็น 3 เพียง 1 ใบ

#### สรุปผลการทดลอง

จากการทดลองแนวคิด Decision tree ใช้ได้ผลดี ดังนั้นจะใช้แนวคิดนี้ต่อไป

การใช้แนวคิดนี้ สามารถมองได้เป็น การต่อ BAM อนุกรมกัน โดย weight คนละชุด

### การทดลองที่ 17

ทดสอบไฟ 30 ใบ ระหว่าง Binary scale จาก gray scale (RGB)

#### ขั้นตอนการทดลอง

การทดลองเหมือนการทดลองที่ 13

#### ผลการทดลอง

Binary scale จาก gray scale ถูก 9 ใบ

Binary scale จาก Hue ถูก 13 ใบ

แต่มี output vector แปลกปลอม นอกเหนือจากที่สร้างเป็น Correlation Matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### สรุปผลการทดลอง

จะใช้ Hue เริ่มต้น และใช้ Binary จาก gray scale ตามลำดับและเกิดข้อสงสัยเกี่ยวกับ output vector แปลกปลอม เนื่องจากไม่เกิดกับ Matrix ขนาดเล็กแต่ใน การทดลองนี้มี Correlation Matrix ขนาดใหญ่มาก เพราะ input vector ยาวมาก

ดังนั้น จึงเกิดความคิดว่า Vector แปลกปลอมเกิดจาก ขนาดของ weight ที่ใหญ่หรือไม่

### **การทดลองที่ 18**

ทดสอบ ย่อขนาด Correlation Matrix

#### ขั้นตอนการทดลอง

1. นำภาพที่จะนำมาทดสอบมาลดขนาดด้วยวิธี Scaling Smaller ในบทที่ 2 โดยลดลงเป็น อัตราส่วนที่เท่ากัน ลดลงประมาณ 3 เท่า
2. ทำการทดลองเหมือนการทดลองที่ 13 แต่ใช้ไฟ 30 ไบ

#### ผลการทดลอง

Hue มี output ถูก 13 ไบ ไม่มีพวก out put vector ลดลง

#### สรุปผลการทดลอง

จะเห็นได้ว่าขนาดของ Matrix ที่ใหญ่เกินไป ส่งผลเสียในการประมวลผลของ BAM

### **การทดลองที่ 19**

ทดลองย่อภาพ ให้เล็กลงกว่าเดิมอีก 2 เท่า

#### ขั้นตอนการทดลอง

1. ย่อภาพ ดันฉบับให้เล็กลงกว่าเดิม
2. นำมาสร้าง Correlation Matrix แล้ว test ดู

#### ผลการทดลอง

ไฟผิด 11 ไบ แย่กว่า แบบยังไม่ลดขนาด

#### สรุปผลการทดลอง

การลดขนาดภาพลงมาก ๆ จะทำให้สูญเสีย feature เป็น โครงสร้างเมื่อมองเผินๆ อาจมีค่า ใกล้เคียงกัน จึงสรุปว่าใช้ภาพขนาดเดิมไม่ต้องลดขนาด

## การทดลองที่ 20

ทดลองหาความเสถียร ของระบบ

### ขั้นตอนการทดลอง

1. ทำการทดลองเหมือนการทดลองที่ 13
2. ใช้ correlation matrix ที่ย่อขนาดแล้ว
3. ใช้การวิเคราะห์ในช่อง Hue ก่อน แล้วตามด้วย Binary
4. ทำการทดลองหลาย ๆ ครั้ง เพื่อหาไฟไบที่ผิดบ่อย

### ผลการทดลอง

การทดลองที่ 19 นี้ใช้ระยะเวลาานต้องถ่ายภาพและจัดบันทึกมากพอสมควร ผลการทดลอง คือ การผิดของไฟมักจะไม่ค่อยคงที่ (ต่อไบ) และอัตราการถูกแปรเปลี่ยนอยู่ในช่วง 12 ถึง 16 ไบ ตลอด และยังพบจากสรุปผลการทดลอง ว่ารอบแรกในการ feed ค่า output รอบแรกถูกแต่งรอบต่อไป ผิดอีกด้วย

### สรุปผลการทดลอง

เนื่องจากการผิดของไฟไม่คงที่จึงยากที่จะนำหมายเลขไฟไบที่ผิดมาสร้าง Matrix เพื่อการวิเคราะห์ครั้งที่ 2 อีกปัญหาคือแสงถ้าการถ่ายภาพเพื่อบันทึก input vector ต้นแบบ มักขึ้นอยู่กับแสงมาก เช่น การทดลองกระทำในห้องพัก ปิดประตูหน้าต่างทุกบานเพื่อควบคุมแสง และใช้ไฟจากหลอดฟลูออโรสเซนส์ และบันทึก input vector ตอนบ่าย และลอง test หลังจากบันทึกเสร็จจะมีอัตราถูกมาก แต่เมื่อปิดเครื่องไป แล้วทดลองตอนมี output ที่ได้จะมีการผิดมาก บางครั้งต้องถ่าย 2 ครั้งจึงจะถูก

จากสรุปผลการทดลอง แสดงว่า รบกวนมากที่สุดและเหตุผลใหญ่ที่ output คือ unknow input vector จะสามารถผิดจาก input vector ต้นแบบ ได้น้อยมาก เพราะ Network จะต้องจำ Pattern ที่มาก ตามทฤษฎีของ Associative Memories

จึงเกิดแนวคิดการ Clustering หรือแบ่งหมู่ตามลักษณะเด่นของมันเอง โดยจะเห็นว่าไฟยูกินั้น 30 ไบที่วิเคราะห์จะมีสรอบนอกอยู่ 4 สีคือ น้ำเงิน, ม่วง, น้ำตาล, เหลือ เพราะฉะนั้นจึงจะ Clustering แยกชนิดสีออกก่อน เพื่อลดภาระการรู้จำของ BAM ดังนั้นจึงสันนิษฐานได้ว่าถ้า BAM มี Correlation Matrix ที่เก็บ Pattern น้อยลง น่าจะมี output ที่ถูกมากขึ้น ( Network มีความเสถียรมากขึ้น )

### การทดลองที่ 21

วิเคราะห์แยกชนิด สี ขอบไฟ

#### ขั้นตอนการทดลอง

1. เก็บจุดภาพให้เข้าไปในส่วนสีไฟมา 30 จุด
2. หาผลรวมของจุดภาพ 30 จุดนั้น แล้วปิดขึ้นหรือลงชั้นกับทศนิยม
3. หาทั้ง 4 สี แล้วนำมา plot graph

#### ผลการทดลอง

ได้ค่าผลรวมทั้งสี่ค่าโดยค่าของสีน้ำเงิน กับ ม่วง ใกล้เคียงกัน และ น้ำตาล กับ เหลือง ใกล้เคียงกัน

### การทดลองที่ 22

วิเคราะห์ ผลรวม เพื่อหาขอบเขตที่แปรปรวน

#### ขั้นตอนการทดลอง

ทำการทดลองเหมือนการทดลองที่ 21 สีละ 10 ครั้งทำการปิดเศษตามจุดทศนิยม แล้วนำค่ามาพิจารณา

#### ผลการทดลอง

ได้ขอบเขตของแต่ละสีมา โดยแต่ละสีแปรปรวน  $\pm 1$  เท่านั้น

#### สรุปผลการทดลอง

ผลรวมปิดเศษของทุกสีไม่แปรปรวนมาก สามารถกำหนดขอบเขตได้แน่นอน สามารถสร้างกฎ IF-THEN จัดการได้

หลังจากสามารถ Clustering สีไฟออกมาได้ 4 ชุด ขั้นตอนต่อไปต้องทดสอบ Network โดยสร้าง Network 4 ตัว โดยมี Correlation Matrix แบ่งตามไฟที่อยู่ในกลุ่ม 4 นั้นเช่น Correlation Matrix ตัวแรกเก็บไฟสีน้ำเงิน ก็ต้องประกอบไปด้วย 4 pattern pain หรือ Correlation Matrix ตัวที่ 2 เก็บไฟสีม่วง ก็ต้องประกอบไปด้วย 9 pattern pain

### การทดลองที่ 23

ทดสอบ BAM แบบแบ่งกลุ่มตามสี

#### ขั้นตอนการทดลอง

เหมือนการทดลอง 13 แต่ทำ 4 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

Out put ออกมาเป็นที่น่าพอใจ โดยกลุ่มสี น้ำเงินและน้ำตาลถูกทุกใบ กลุ่มสีม่วงและสีเหลือง ผิด 3 ใบ และ 1 ใบ ตามลำดับ

สรุปผลการทดลอง

การ Clustering จะช่วยลดความจำของ BAM จากเกิดต้องจำ 30 ใบ ถูกนำมาแบ่งออกเป็น 4 Correlation Matrix ได้ผลพอสมควรตามทฤษฎีของ BAM

การทดลองที่ 24

ทดสอบความเสถียรของ Network  
ขั้นตอนการทดลอง

เหมือนการทดลองที่ 20

ผลการทดลอง

Output ที่ได้ค่อนข้างแปรปรวนพอสมควรยังมีใบผิดอยู่บ้างแต่พอยอมรับได้

การทดลองที่ 25

ปรับปรุง Output ที่ผิด  
ขั้นตอนการทดลอง

1. หาหมายเลขไฟใบที่ผิด โดยการทดลองและจดบันทึกมา
2. นำ Output ที่ผิด จาการจดบันทึก
3. นำมาสร้าง weight ย่อยแบบ Binary scale ของ gray scale
4. ไฟที่ผิดนอกกลุ่ม (out put vector ไม่ได้ถูกประกอบขึ้นจากกลุ่มเดิม) เช่น ไฟใบที่ 2 กลุ่มเหลืองให้output เป็น 6 ซึ่ง 6 ไม่ได้อยู่ในกลุ่มเหลือง ให้ map ด้วยกฎ IF-THEN แก่ค่าให้ถูก
5. ไฟที่ผิดในกลุ่มให้ทำข้อ 3

ผลการทดลอง

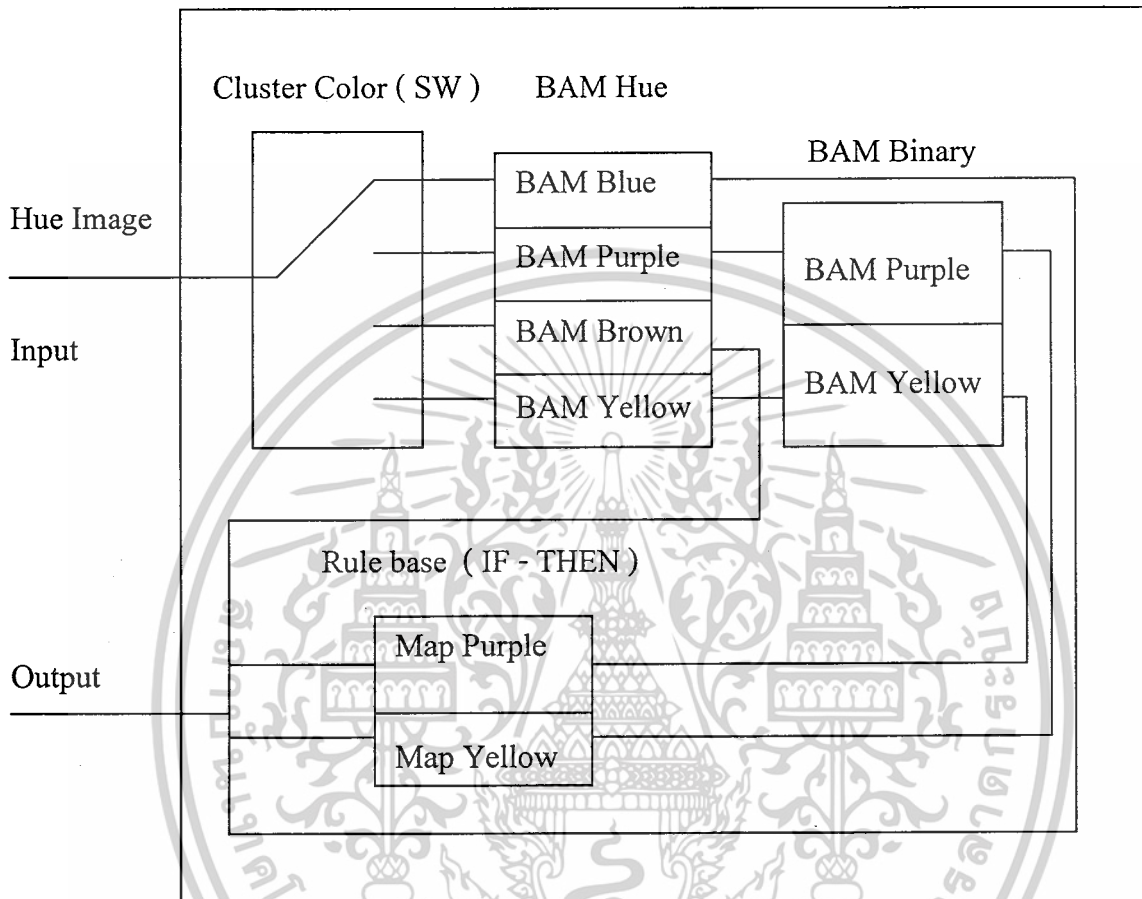
Network ให้ Output ถูกทุกใบ

สรุปผลการทดลอง

แนวคิด Decision tree ค่อยๆ ตัด choice ให้ผลลัพธ์เป็นที่น่าพอใจ ดังนั้น จะใช้วิธีการนี้ในการสร้างระบบต่อไป

ดังนั้นจึงสามารถ สร้าง block diagram ในส่วน Image Analysis ที่อ้างไว้จากบทที่ 3 ได้ดังนี้

### Image Analysis



### การทดลองที่ 26

ทดลองการเชื่อมต่อของเกม

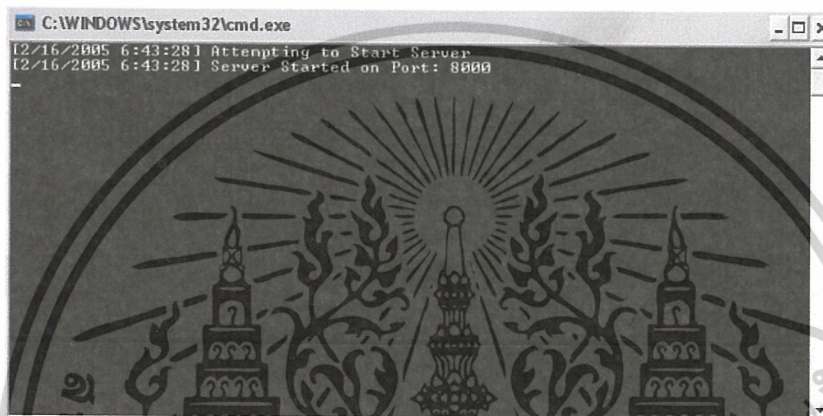
#### ขั้นตอนการทดลอง

1. ทำการรันโปรแกรมJavaที่ทำให้เครื่องที่รันนั้นเป็นServerโดยจะให้เครื่องนี้มีหมายเลข โฮสต์ เป็น 192.168.0.1 และเปิดการเชื่อมต่อที่Port = 8000 จะได้ผลดังรูปที่ 4.11
2. เริ่มเข้าสู่ตัวเกมดังรูปที่ 4.12 เมื่อทำการคลิกที่ปุ่มStart จะทำให้เข้าสู่เกมและเปลี่ยนหน้าเฟรมต่อไป
3. เลือกเล่นตัวละครที่เราจะเล่นดังรูปที่ 4.13 ผู้เล่นคนแรกจะเลือกตัวละครคนใดก็ได้ แต่คนที่เลือกทีหลังจะไม่สามารถเลือกซ้ำกับคนแรกที่เลือกไว้ ในส่วนนี้ถ้าServerไม่เปิดการเชื่อมต่อหรือไม่สามารถเชื่อมต่อกับServerได้จะขึ้นข้อความว่าการเชื่อมต่อล้มเหลวดังรูปที่4.14
4. รอกการเชื่อมต่อจากอีกฝ่ายหนึ่งดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โปรแกรมจะทำการตรวจสอบจำนวนผู้เล่นและตัวละครที่เลือกกว่าแต่ละฝ่ายเลือกเลือกซ้ำกันหรือไม่ ถ้าผู้เล่นซ้ำกันจะมีเมจแสงฟุ้งขึ้นมาดังรูปที่ 4.16 ผู้เล่นที่เลือกทีหลังก็จะต้องเลือกใหม่อีกครั้ง ถ้าผู้เล่นเลือกไม่ซ้ำกันจะกระโดดไปฉากที่ใช้เล่นและโหลดไฟจากการส่งค่าของส่วนรู้จำดังรูปที่ 4.17

#### ผลการทดลอง

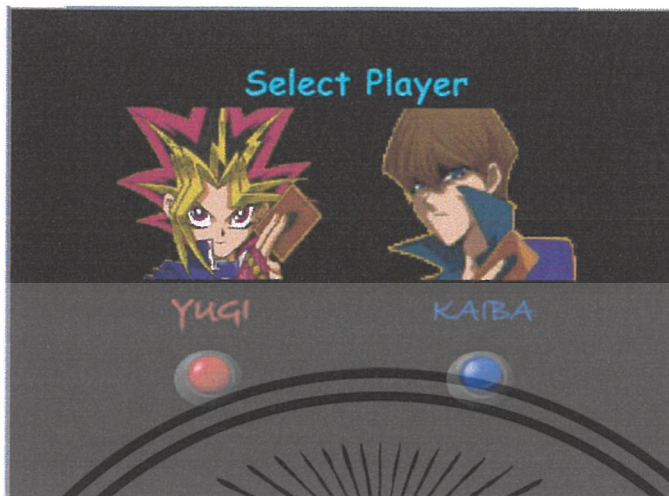


รูปที่ 4.11 แสดงผลที่ได้จากการรันโปรแกรมJavaที่ให้เครื่องนี้เป็นServer

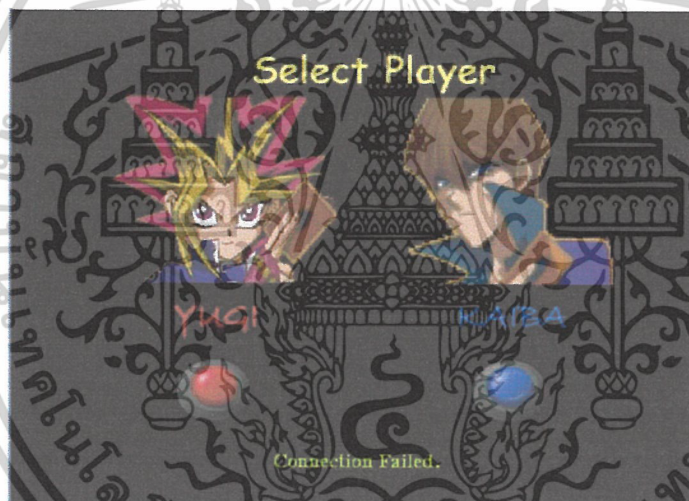


รูปที่ 4.12 ฉากที่เริ่มเข้าสู่เกม

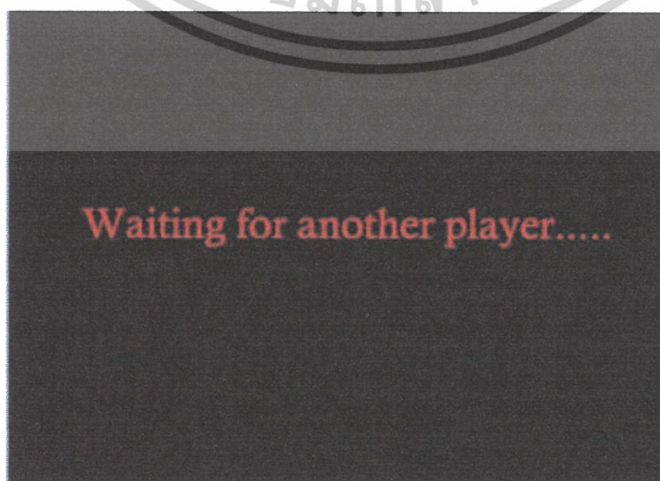
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ฉากสำหรับเลือกตัวละครที่จะเล่น



รูปที่ 4.14 กรณีที่การติดต่อล้มเหลว



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.15 รอคอยเชื่อมต่อจากอีกฝ่ายหนึ่ง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 จะมีข้อความบอกให้เลือกใหม่เมื่อเลือกตัวซ้ำกัน



รูปที่ 4.17 ฉากการโหลดไฟที่ส่งค่ามาจากส่วนรู้จำของแต่ละฝ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทดลองที่ 27

ทดสอบส่วนของกติกาและการส่งค่าภายในตัวโปรแกรมเกม

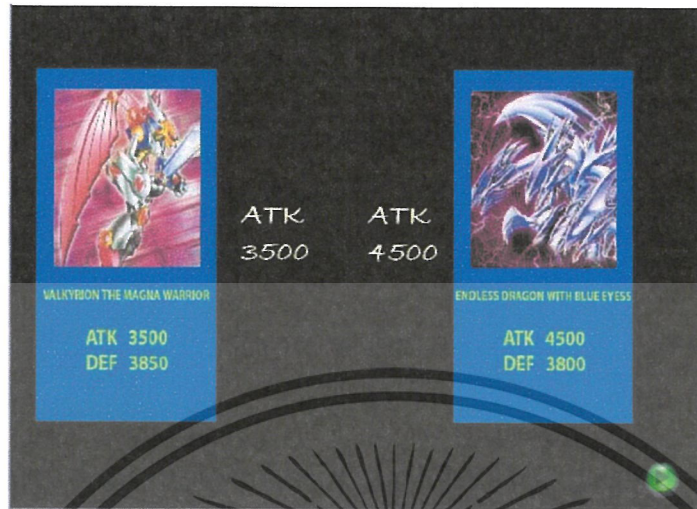
ขั้นตอนการทดลอง

1. ทดลองต่อจากการทดลองที่ 26 เมื่อทำการกดปุ่มOKในรูปที่ 4.17 แล้วโปรแกรมจะส่งค่าข้อมูลการ์ดที่เราเลือกไปให้ยังอีกฝ่ายหนึ่ง
2. จะขึ้นฉากในการรอส่งข้อมูลการ์ดจากอีกฝั่งหนึ่งดังรูปที่ 4.18
3. เมื่อมีข้อมูลส่งทั้งสองฝ่ายแล้วจะขึ้นฉากในการต่อสู้ขึ้นดังรูปที่ 4.19 และ 4.20
4. นำค่าของการ์ดและสถานะของการ์ดที่ได้รับได้มาคำนวณแต้มจนกว่า
5. กลับไปยังฉากที่โหดไฟไหม้ดังรูปที่ 4.21 โดยจะมีคะแนนเปลี่ยนไปตามกติกา ถ้าคะแนนรวมของฝ่ายที่มีคะแนนรวมน้อยกว่าหรือเท่ากับศูนย์จะเป็นฝ่ายแพ้และจะจบเกมสั้ไปยังฉากที่แสดงการแพ้ดังรูปที่ 4.22 และอีกฝ่ายจะเป็นผู้ชนะจะข้ามไปยังฉากที่แสดงการได้รับชัยชนะดังรูปที่ 4.23

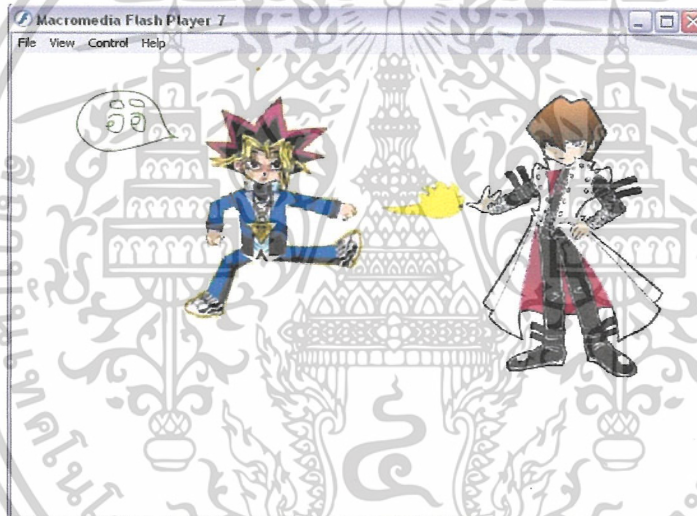


รูปที่ 4.18 ฉากรอการรับค่าจากอีกฝั่งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงไฟและสถานะที่ส่งค่ามาจากแต่ละฝ่าย



รูปที่ 4.20 ฉากของการต่อสู้



รูปที่ 4.21 ฉากที่กลับมาโหลดไฟและแสดงยอดคะแนนคงเหลือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของนักเขียนเพื่อใช้ในการศึกษาเท่านั้นซึ่งอนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 เป็นฉากที่บอกว่าผู้เล่นแพ้หรือชนะ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

จากการทดลองต่าง ๆ สรุปได้ว่า การรู้จำภาพ (Image Recognition) แสดงเป็นตัวการสำคัญในทุก ๆ ขั้นตอน ทำให้ภาพแปรปรวน ดังนั้นต้องควบคุมแสงให้ดี อีกเรื่องคือ ด้านตัวรับภาพได้ใช้ Web Camera ซึ่งไม่ได้ใช้กับงานนี้ โดยเฉพาะและ Card VGA เป็นแบบ onboard ภาพที่ได้ค่อนข้างมืด เวลา มองภาพขณะ Preview จะเห็นจุดสีขยับสลับไปสลับมา ซึ่งน่าจะเป็นตัวแปรสำคัญ ถัดจากแสง ที่ทำให้ การวิเคราะห์แบบจุดภาพได้ผลไม่ดี จึงควรแก้ปัญหาดังนี้เพื่อให้ได้ภาพ ต้นฉบับมีแปรปรวน น้อย ส่วนเรื่อง โครงข่ายประสาทเทียมสามารถสรุปได้ดังนี้

ควรทำความเข้าใจและศึกษาทฤษฎีของ โครงข่ายประสาทเทียม ( ANN ) แต่ละชนิดให้ดี และควรรู้หลายๆ ชนิดด้วย เพราะว่า ANN แต่ละตัวมีการใช้งานต่างกัน เหมาะแก่การปฏิบัติกับชนิด ของ input ที่ต่างชนิดกัน ดังนั้นการใช้ ANN ให้เหมาะกับงานจะช่วยให้ระบบมีความถูกต้องมากขึ้น ทนต่อความผิดพลาดได้มากขึ้น ( Fault Tolerance )

ผู้จัดทำได้เสียเวลาไปกับการฝึกฝน โครงข่ายประสาทเทียม แบบ Multilayer Feedforward Network ไปมาก เนื่องจากความไม่รู้ โดย Image Recognition นี้ ที่นิยมใช้จะเป็น Associative Memory

ด้าน ลักษณะเด่น (feature) ควรศึกษาการหา feature ให้ดีก่อนหลายๆ แบบแล้วเลือกวิธีที่ดี ที่สุดมา ถ้า feature ดีมาก ๆ อาจจะไม่ต้องใช้ ANN เลยก็ได้ จึงสรุปได้ว่า feature ที่ดีนั้นสำคัญกว่า ANN ที่ ซับซ้อน ข้อมูลใน real world นั้น ไม่แน่นอน , คลุมเครือ ( uncertain ) กว่าที่คาดไว้มาก

การ Normalisation เป็นสิ่งสำคัญ เพื่อที่จะทำให้ข้อมูลมีมาตรฐานเดียวกัน  
ข้อเสนอแนะ

จะเห็นว่าเมื่อเปลี่ยนสถานที่เล่น จะทำให้แสงเปลี่ยนไป ส่งผลให้ BAM ให้ output ผิด แม้ว่า จะใช้ ภาพในระบบ HSV แล้วก็ตาม ถ้าพิจารณาเป็นขั้นตอนแล้ว ก็คือเกิดจากค่า Threshold ที่คงที่ ไม่ เปลี่ยนแปลงตามแสงที่เปลี่ยน จึงมีแนวคิดที่จะทำให้ระบบสามารถปรับตัวได้โดย ปรับค่า Threshold ซึ่งอาจจะต้องใช้ตัวตรวจรับแสง เพื่อนำมาปรับค่า Threshold แสงนั้นค่อนข้างแปรปรวน จึงต้องใช้ Fuzzy Logic มาจัดการส่วนนี้ ในที่นี้ input จะเป็นกระแสไฟฟ้าจาก ตัวรับแสง output เป็น ค่า Threshold ที่จะนำมา Threshold ภาพ (เสนอโดย ฮัมดัน หะยีชะวัง)

โครงการนี้ไม่ได้เป็นไปตามกติกาของเกมยูกิ เช่นมี การ์ดเวทย์ กับซัค วางที่มากที่สุด 5 ใบ จึง สามารถพัฒนาต่อได้

ส่วนของ BAM ในโครงการไม่ได้ใช้ วิธี Dummy Augmentation ของ Wang (1990) ซึ่ง วิธีนี้ จะช่วยให้ การสืบค้น มีความถูกต้องมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านการติดต่อกับระบบเครือข่ายที่ทำให้ผู้เล่นทั้งสองสามารถที่จะเล่นเกมสั้โต้ตอบกันได้นั้น จะต้องใช้การเชื่อมต่อโดยมีการสร้าง Socket ขึ้นมาเพื่อช่วยในการเชื่อมต่อโดยตรงโดยใช้ XMLSocket ที่มีอยู่ในFlash และในส่วนของการเขียน ActionScript ในตัวโปรแกรมนั้นจะต้องตรวจเช็คการทำงานของเฟรมและปุ่มกดต่างๆให้ถูกต้องมิฉะนั้นจะไม่ได้ผลงานตามที่ต้องการ

สรุปได้ว่าการทดลองนี้เป็นไปตามจุดประสงค์คือ ได้ศึกษาทฤษฎี ประมวลผลภาพ ทฤษฎี การรู้จำรูปแบบ คอมพิวเตอร์กราฟฟิค และ การสื่อสารในระบบ LAN ในเบื้องต้นอย่างครบถ้วน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] กมลพรรณ ชีวพันธุ์ศรี และ โสภา เกริกไกรกุล. สมอกับการเรียนรู้. พิมพ์ครั้งที่ 3. กรุงเทพฯ : พรการพิมพ์ จำกัด, ม.ป.ป.
- [2] กำพล ตีลาภรณ์, **Advanced Flash ActionScript**. พิมพ์ครั้งที่ 1. กรุงเทพฯ. บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน).
- [3] ชูชาติ พิณทวิรุจน์. เอกสารประกอบการสอนวิชา **Digital Image Processing**. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ม.ป.ป.
- [4] ณรงค์ศักดิ์ ศรีสม. “เอกสารประกอบการสอนวิชา **Data Communication System**”. [ออนไลน์]. เข้าถึงได้จาก : <http://learning.ricr.ac.th/datacomm/Subjectnew/Less8.html> [14 ตุลาคม 2547]
- [5] นวภัค เอื้ออนันต์. “เอกสารประกอบการสอนวิชา ข่ายนิวรอนเทียม”. [ออนไลน์]. เข้าถึงได้จาก : <http://gear.kku.ac.th/~nawapak/neuralnet/neural4.ppt> [18 สิงหาคม 2547]
- [6] นิรุช อำนวยศิลป์. “**Image Processing Fundamentals**”. [ออนไลน์]. เข้าถึงได้จาก : <http://thaidev.com/html/devtutor.htm> [20 กรกฎาคม 2547]
- [7] มนัส สังวรศิลป์ และวรรรัตน์ ภัทรอมกุล. **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์**. กรุงเทพฯ : อินโฟเพรส, ๒๕๔๓.
- [8] สมเกียรติ อุดมพระยากุล. เอกสารประกอบการสอนวิชา **Selected topics in Information Engineering**. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ม.ป.ป.
- [9] สุขสถิต สุขใจ. การใช้โครงข่ายประสาทเทียมเพื่อจดจำรูปแบบลายมือเขียนตัวเลขอารบิก แสดงข้อมูลเป็นมุมสัมพันธ์. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ๒๕๔๐.
- [10] สุทธิชัย มณีรัตน์รุ่งโรจน์. การจำลองการทำงานโครงข่ายประสาทเทียม. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ๒๕๓๕.
- [11] อุกฤษณ์ มารังค์. การรู้จำตัวพิมพ์ภาษาไทยโดยใช้เจเนติก-นิวรอนเน็ตเวิร์ค. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ๒๕๔๔.
- [12] Cichocki, A and Unbehaven R. . **Neural Networks for Optimization and Signal Processing**. Grate Britain : John Wiley & Son Inc, 1993.
- [13] Eliasmith, Chris. “**Dictionary of Philosophy of Mind**”. [Online]. available : [www.artsci.wustl.edu/~philos/minddict/main.html](http://www.artsci.wustl.edu/~philos/minddict/main.html) [14 January 2005]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] Gonzalez, Rafael C. Woods, Richard E. and Eddins, Steven L. **Digital Image Processing Using MATLAB**. United State of America : Pearson Education Inc, 2004.
- [15] Haykin, Simon. **Neural Networks**. United State of America : Prentice Hall Inc, 1999.
- [16] McAndrew, Aladair. **Introduction to Digital Image Processing with MATLAB**. United State of America : Thomson Learning Inc, 2004.
- [17] Montri Karnjanadacha. **"Image Processing"**. [Online]. available : <http://fivedots.coe.psu.ac.th/~montri/Teaching/image> [2 July 2004]
- [18] Negnevitsky, Michael. **Artificial Intelligence**. 2<sup>nd</sup> ed. Grate Britain : Pearson Education Inc, 2005.
- [19] Nelson, McCord Marilyn and Illingworth, W. T. . **A Practical Guide to Neural Nets**. United State of America : Addison Wesley Publishing, 1991.
- [20] Patterson, Dan N. **Artificial Neural Networks**. Singapore : Simon & Schuster (Asia) Pte Ltd, 1996.
- [21] PhpBB Group. **"Cannibal Corpse"**. [Online]. available : [www.cannibalcorpse.org/picture](http://www.cannibalcorpse.org/picture) [10 December 2003]
- [22] Sangwine, S. J. and Horne, R. E. N. . **The Color Image Processing Handbook**. Grate Britain : Chapman & Hall, 1998.
- [23] Sankar, K. Pal and Sushmita, Mitra. **Neuro-Fuzzy Pattern Recognition**. United State of America : John Wiley & Son Inc, 1999.
- [24] Schalkoff, Robert J. . **Artificial Neural Networks**. Singapore : McGraw Hill, 1997.
- [25] Supot Nitsuwat. **"Image Processing System"**. [Online]. available : <http://ced.kmitnb.ac.th/sns/dip.html> [26 June 2004]
- [26] Zurada, Jacek M. . **Introduction to Artificial Neural Networks**. Singapore : West Publishing Company, 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Yugi Game จะแบ่งเป็น 2 ส่วนใหญ่ๆคือ ส่วน Image Analysis และ Graphical and LAN โดย Image Analysis จะใช้โครงข่ายประสาทเทียมแบบ Bidirectional Associative Memory (BAM) ส่วน Graphical จะใช้โปรแกรม Macromedia Flash MX, LAN จะใช้การสร้าง XML socket โดยใช้ JAVA

ส่วนที่ยังไม่เสถียรก็คือ ส่วน Image Analysis ถึงแม้จะใช้ระบบภาพแบบ HSV ซึ่งทนต่อความแปรปรวนของแสงในระดับหนึ่งแต่ก็ช่วยได้ไม่มาก ดังนั้นเกมนี้จะใช้ได้ดี (วิเคราะห์ไฟลูกต้อง) จะต้องรับภาพตัวอย่างและสร้าง weight ของ BAM ใหม่ และสร้าง กฎ IF-THEN สำหรับ Decision tree ใหม่ ของสภาพแวดล้อมในห้องที่เล่น (ห้องต้องมีการควบคุมแสงด้วย)

### โปรแกรมที่สำคัญในการใช้งาน

1. MatLab เวอร์ชัน 6.5
2. โปรแกรม Video for MatLab ของ Farzad Pezeshkpour
3. J2SDK ตั้งแต่ version 1.4.2 (เฉพาะเครื่องที่เป็นServer)
4. Edit Plus 2 (เฉพาะเครื่องที่เป็นServer)
5. Macromedia Flash MX 2004 หรือ Macromedia Flash MX (ลงทั้งเครื่อง Server และ Cleint)

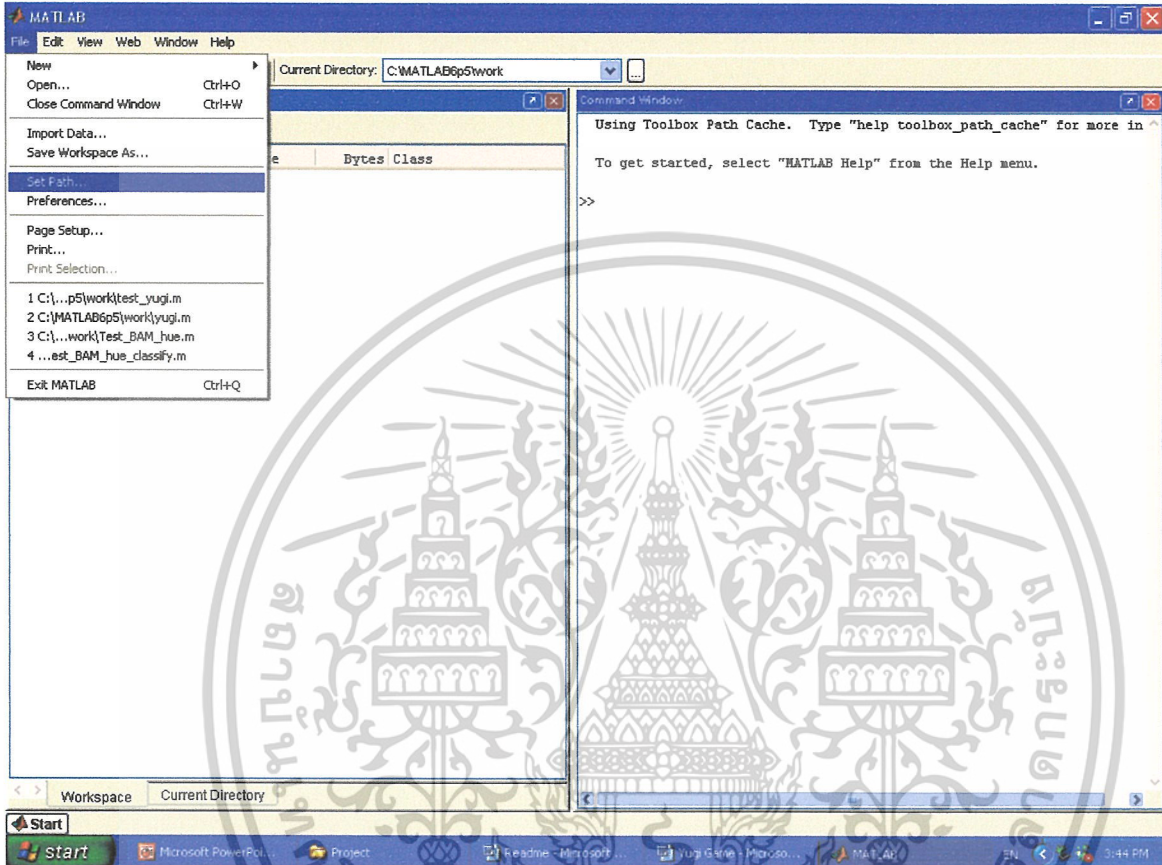
### อุปกรณ์ที่ใช้ในการเล่น

1. เครื่องคอมพิวเตอร์ 2 เครื่อง (มี card จอที่มีประสิทธิภาพใกล้เคียงกัน) พร้อมสาย LAN แบบ Cross
2. กล้อง web camera 2 ตัว (เป็นกล้องชนิดเดียวกัน)
3. แท่นวางกล้อง (สูงจากไฟประมาณ 1 ฟุต)
4. ไฟยูทึ 30 โย
5. กระดาษขาว A4 1 แผ่น

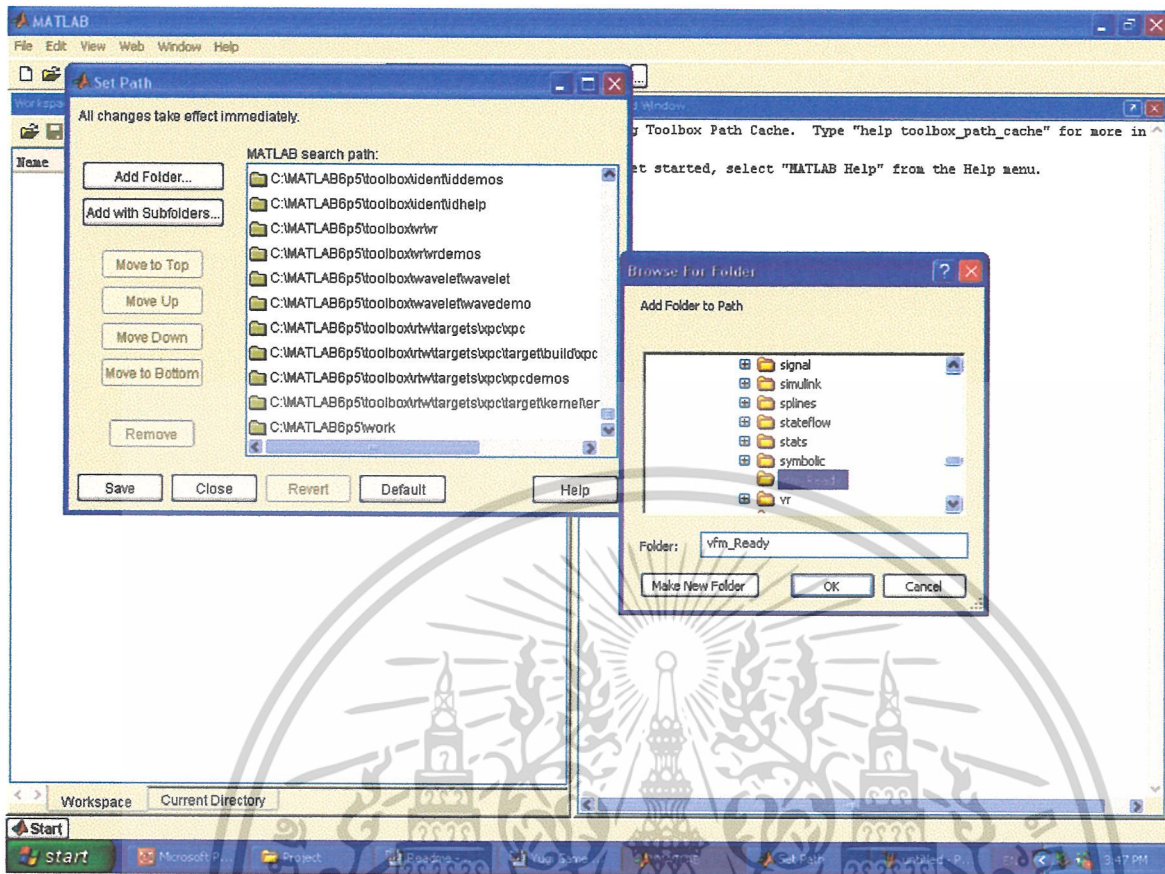
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วิธีการ Set โปรแกรม ส่วน Image Analysis

1. Copy folder vfm-Ready ไว้ใน C:/MATLAB 6p5/toolbox/
2. เปิด MatLab ที่ File → Set Path



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



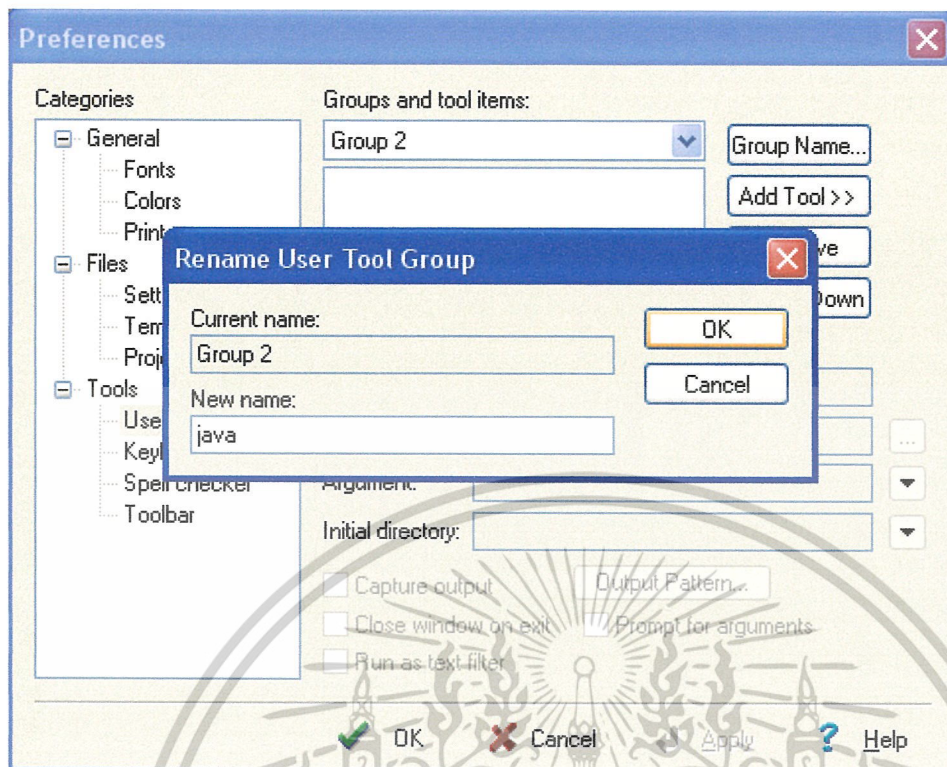
แล้ว click Add Folder จากนั้นเข้าไปใน C: MatLab6p5/toolbox/vfm-Ready โดยการ Double click แล้ว click OK

3. click save
4. copy folder yugi และ yugi 2 ไว้ที่ Desktop
5. copy file ทั้งหมด ใน Folder Final-Yugi ไว้ใน C: /MATLAB 6p5/work

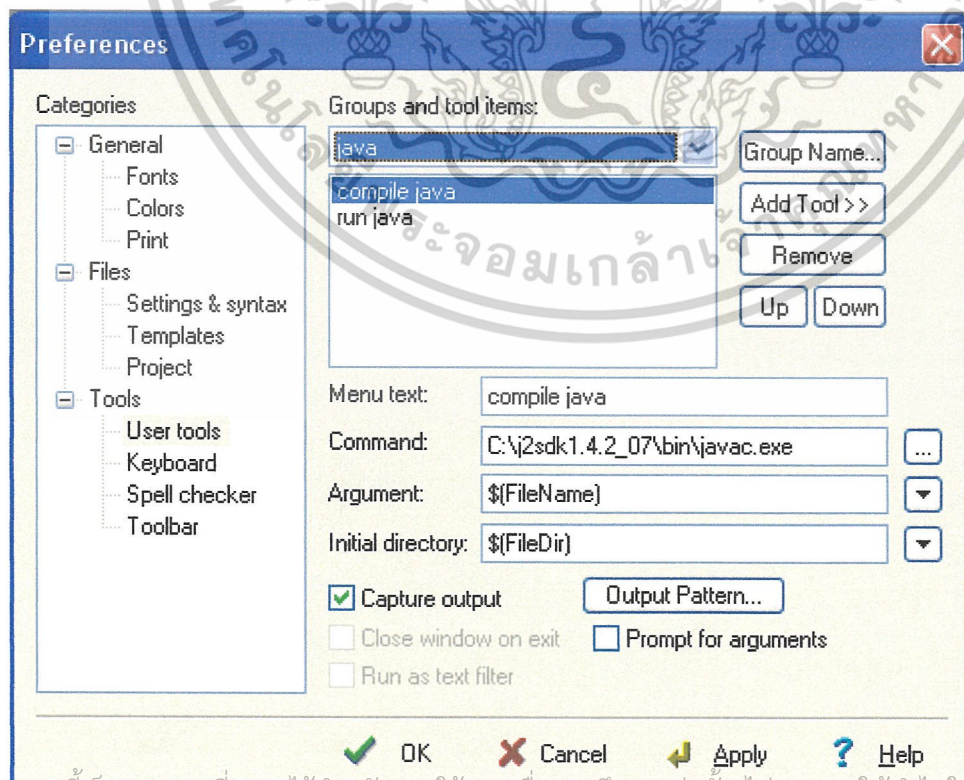
### วิธีการ Set โปรแกรม ส่วน Graphical and LAN

1. ตั้งหมายเลข IP Address ที่เครื่องServer เป็น 192.168.0.2
2. ทำการเซตค่าเริ่มต้นในEdit Plus 2 เพื่อรัน โปรแกรม Java ดังนี้
  - เปิดโปรแกรม Edit Plus 2 แล้วเข้า Menu Tools → Preferences
  - เข้าไปที่ Tools → User tools คลิกที่ปุ่ม Group Name เพื่อตั้งชื่อ Group ว่า java แล้วคลิกปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

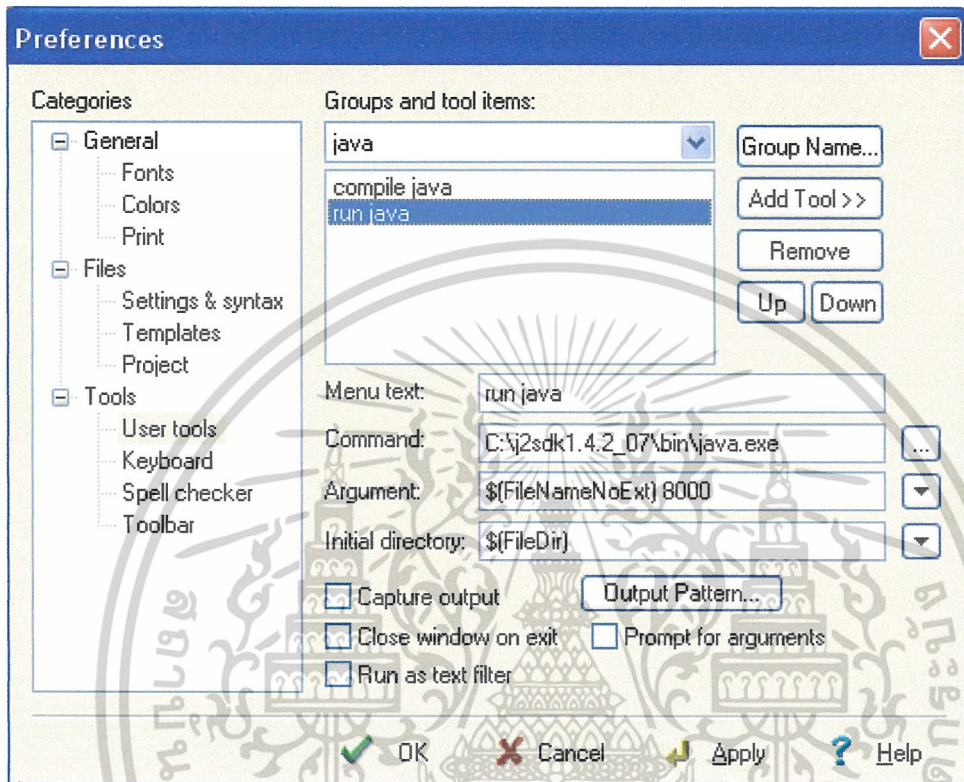


- คลิกปุ่ม Add Tool>> แล้วเลือก Programme แล้วตั้งชื่อตรง Menu text
- เซทค่าต่างๆ เพื่อใช้ในการคอมไพล์ java ดังรูป

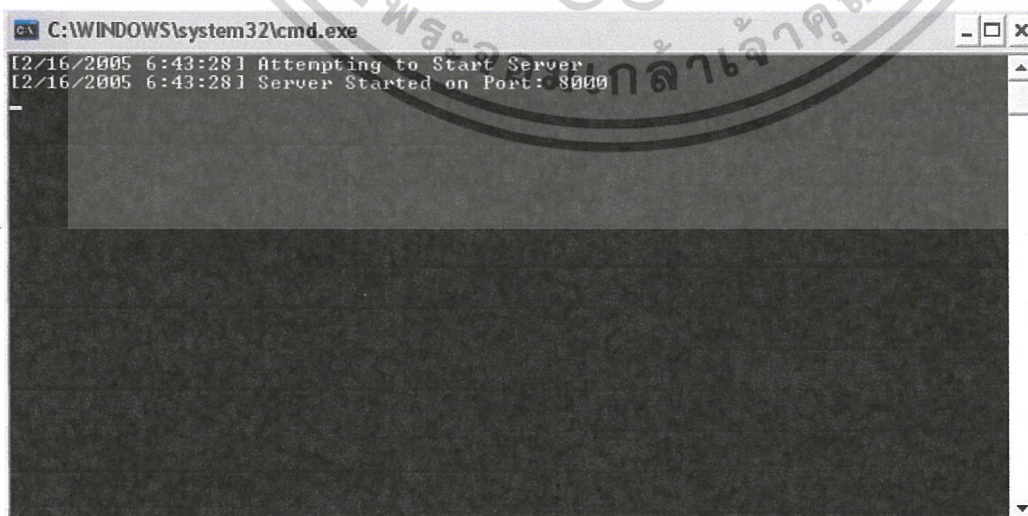


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิกปุ่ม Add Tool>> อีกครั้งแล้วเลือก Programe แล้วตั้งชื่อตรงMenu text
- เซทค่าต่างๆ เพื่อใช้ในการรัน โปรแกรม java ค้างรูป แล้วคลิกปุ่มOK



3. ทำการเปิดไฟล์ Server.java ที่มีมาให้เปิดแล้วรันโปรแกรมโดยคลิกที่ Tools → run java เครื่องจะรันโปรแกรมเพื่อใช้เป็น Server โดยดูจากผลที่ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

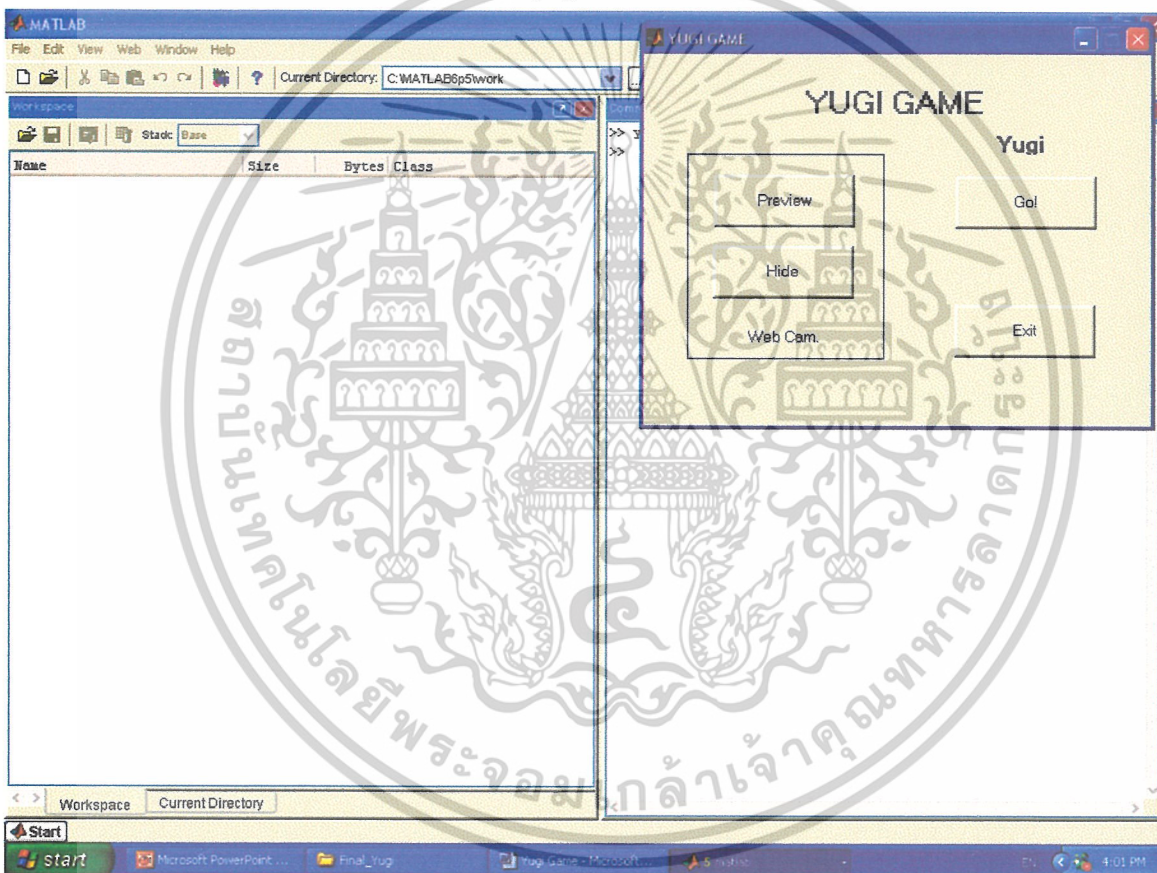
4. เปิดโปรแกรม YUGIOHfor1card.swf (ไฟล์Flashที่เป็น iconสีเทา) ขึ้นมาเพื่อใช้เล่นเกมส์

หมายเหตุ

- จะต้องมีเครื่องที่รันโปรแกรม Server.java อยู่ถึงจะเล่นเกมนี้ได้
- ต้องต่อสาย Lan (ใช้แบบCross) ต่อไว้กับเครื่องคอมพิวเตอร์อีกเครื่องด้วย
- ถ้าใช้ Macromedia Flash MX ให้เปิดไฟล์ YUGIOHfor1cardMX.swf

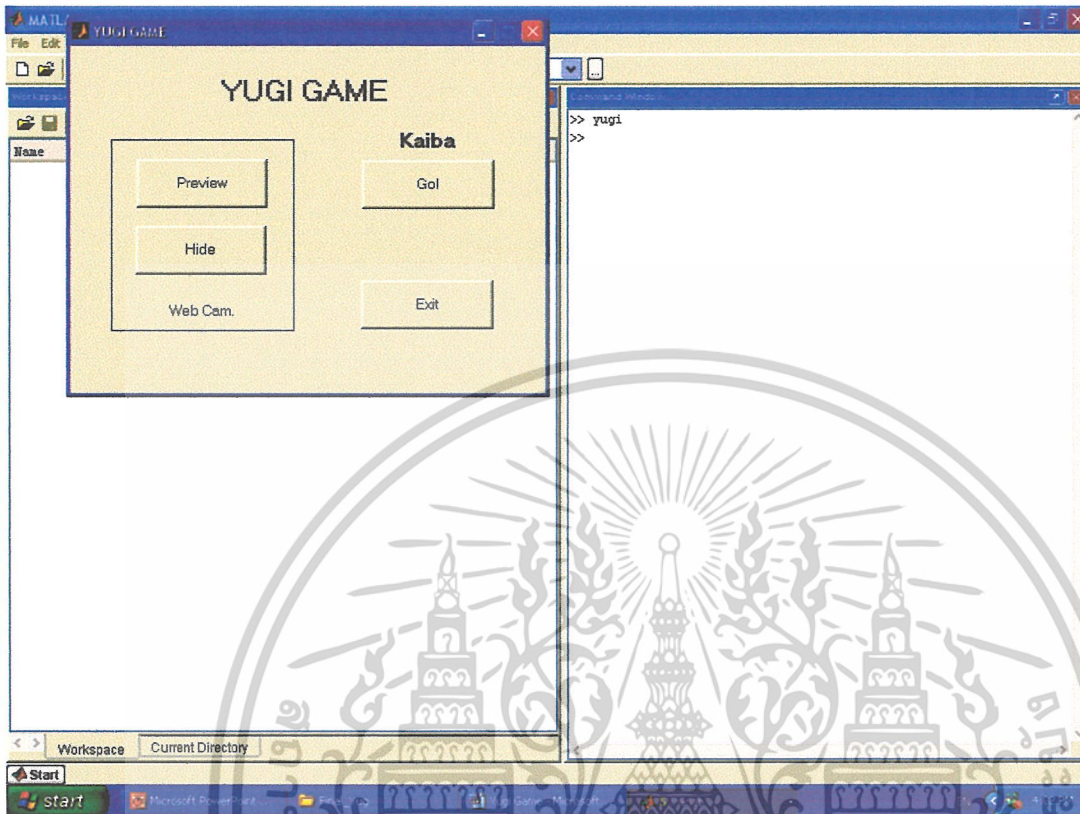
วิธีเล่นเกม

1. ที่ Command Window ของ MatLab พิมพ์ yugi จะได้ผลดังนี้

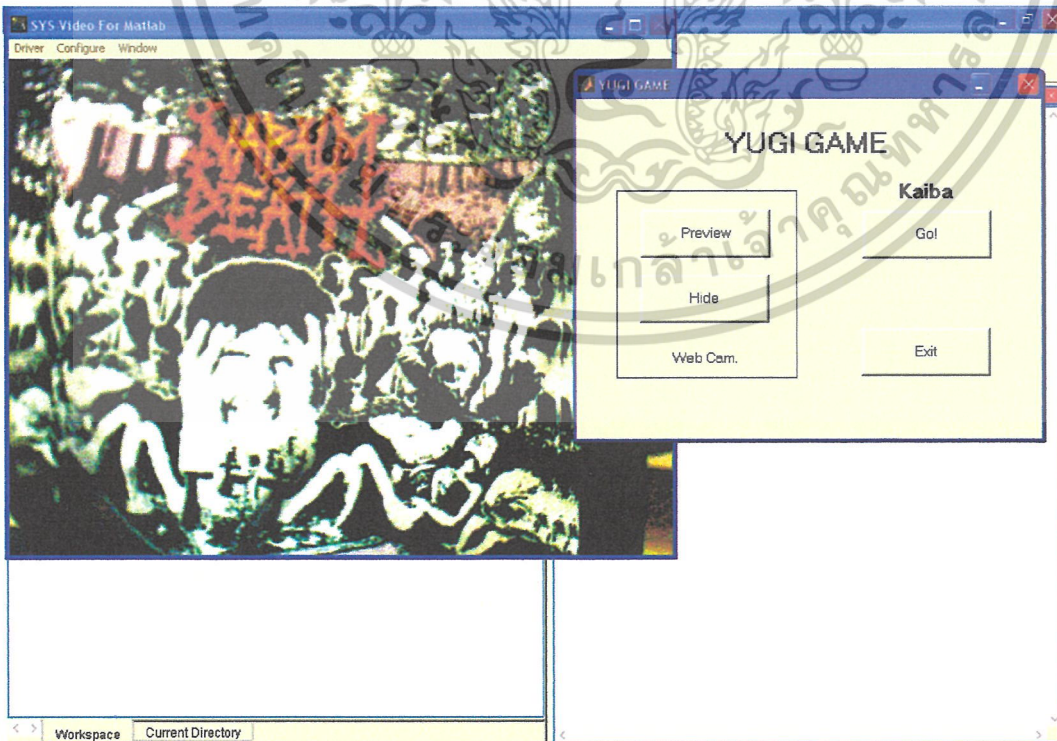


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ที่ Command Window พิมพ์ kaiba จะได้ผลดังนี้

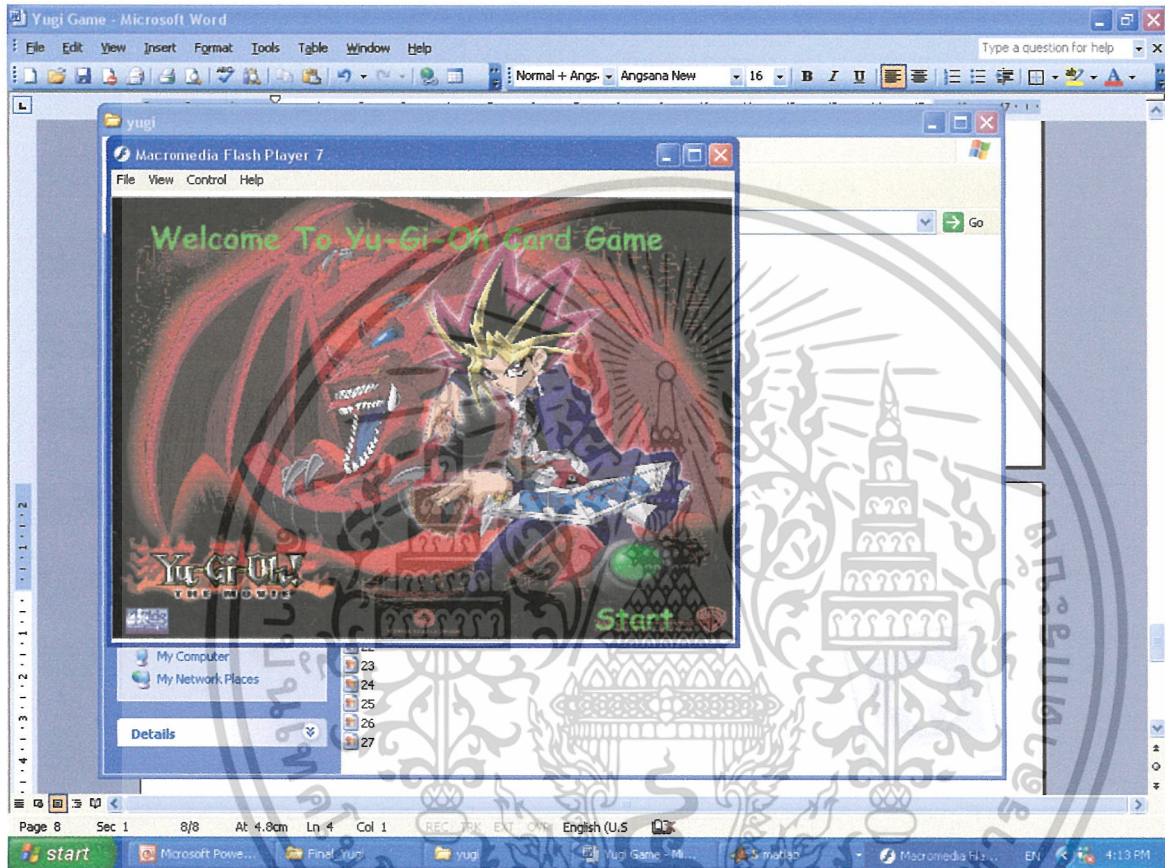


3. Click Preview จะได้ผลดังนี้



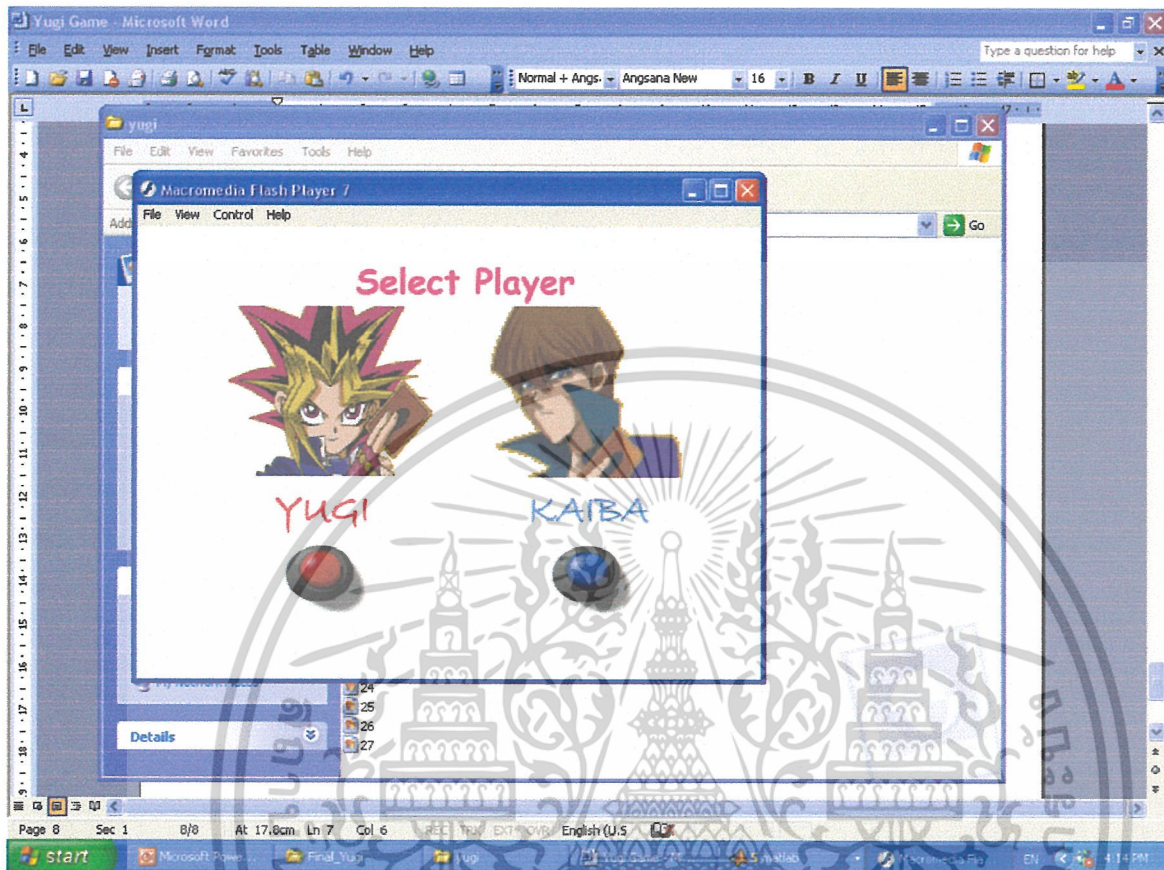
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. จัดไฟให้ตรงแล้ว Click ปุ่มGo!
5. ในส่วนของ Yugi Click Preview จะได้ผลตามข้อ 3 แล้วทำข้อ 4 ต่อ
6. เข้าไปใน folder Yugi และ Yugi2 แล้ว double click ที่ File YUGIOHGAME ทั้ง 2 folder ทั้งสองหน้าจอ จะได้ผลดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Click Start จะได้ผลตามนี้ เลือกผู้เล่น แล้ว ลองเล่นตามปกติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ลักษณะแสดงการติดกล้อง Web Camera กับ แท่น และการวางไฟตั้งรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**ภาคผนวก ข.  
Action Script โปรแกรม Macromedia Flash MX 2004**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Scene 1

Layer : actionDeck

Frame : 53

```
loadVariablesNum("buffercard.txt", 0);
```

Frame : 60

```
scored.text = amountscore1;
if (card1_1==0){
    CardNo_1 = card1;
}else {CardNo_1=card1_1;}
if (card2_1==0){
    CardNo_2 = card2;
}else {CardNo_2=card2_1;}
Atk1 = status1;
Atk2 = status2;
loadMovie(CardNo_1+".jpg", "_root.card1");
loadMovie(CardNo_2+".jpg", "_root.card2");
cardScore = new Array(["3000", "2500"], ["1800", "2000"], ["2800", "2600"], ["4000", "4000"],
["1300", "1400"], ["3500", "3850"], ["2000", "1700"], ["2800", "2600"], ["2300", "2100"],
["2200", "1900"], ["3200", "1800"], ["4500", "3800"], ["3000", "2500"], ["4000", "4000"],
["2300", "2000"], ["2600", "2300"], ["2500", "2100"], ["4500", "3800"], ["3000", "2500"],
["3500", "3850"], ["3200", "2500"], ["1600", "1000"], ["5000", "5000"], ["5000", "5000"],
["2600", "2500"], ["2900", "2400"], ["5000", "5000"], ["2650", "2250"], ["3000", "2500"],
["2800", "2600"]);
if (Atk1 == 1) {
    power1.text = (cardScore[CardNo_1-1][0]);
    action1.text = "ATK";
} else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
power1.text = (cardScore[CardNo_1-1][1]);
action1.text = "DEF";
}
stop();
```

### ปุ่ม Reload

```
on (release) {
    play1 = 1;
    _root.sendMessage(CardNo_1, Atk1, CardNo_2, Atk2, CardNo_3, Atk3, CardNo_4,
    Atk4, CardNo_5, Atk5, "move");
    gotoAndPlay(65);
}
```

### ปุ่ม OK

```
on (release) {
    play1 = 1;
    _root.sendMessage(CardNo_1, Atk1, CardNo_2, Atk2, CardNo_3, Atk3, CardNo_4,
    Atk4, CardNo_5, Atk5, "move");
    gotoAndPlay(65);
}
```

### Frame : 65

```
// waiting for another card//
stop();
```

### ปุ่ม GO

```
on (release) {
    gotoAndPlay(69);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Frame : 69**

```
c=1;
if (card1_1==0){
    CardNo_1 = card1;
} else {CardNo_1=card1_1;}
if (card2_1==0){
    CardNo_2 = card2;
} else {CardNo_2=card2_1;}
loadMovie(CardNo_1+".jpg", "_root.card1");
loadMovie(p2CardNo_1+".jpg", "_root.p2card1");
if (Atk1 == 1) {
    power1.text = (cardScore[CardNo_1-1][0]);
    action1.text = "ATK";
    myscore = Number(power1.text);
} else {
    power1.text = (cardScore[CardNo_1-1][1]);
    action1.text = "DEF";
    myscore = Number(power1.text);
}
if (p2Atk1 == 1) {
    p2power1.text = (cardScore[p2CardNo_1-1][0]);
    p2action1.text = "ATK";
    p2score = Number(p2power1.text);
} else {
    p2power1.text = (cardScore[p2CardNo_1-1][1]);
    p2action1.text = "DEF";
    p2score = Number(p2power1.text);
}
stop();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม back

```
on(release){  
    gotoAndPlay(65);  
}
```

ปุ่ม fight

```
on (release) {  
    gotoAndPlay(70);  
}
```

Frame : 70

//กรณีผู้เล่นเป็นYugi

```
if(player==1){  
    //กรณีที่เป็น ATK ทั้งคู่  
    if(Atk1==1&p2Atk1==1){  
        gotoAndPlay("Scene 2",1);  
    }  
    //กรณีที่เราATK แต่อีกฝั่งDEF  
    else if(Atk1==1&p2Atk1==0){  
        gotoAndPlay("Scene 3",1);  
    }  
    //กรณีที่เราเป็นฝ่ายDEF แล้วฝ่ายตรงข้าม ATK  
    else if(Atk1==0&p2Atk1==1){  
        gotoAndPlay("Scene 4",1);  
    }  
    //กรณีที่เป็นDEFทั้งคู่  
    else {  
        gotoAndPlay("Scene 5",1);  
    }  
}  
//กรณีผู้เล่นเป็นKaiba
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {
    //กรณีที่เป็น ATK ทั้งคู่
    if(Atk1==1&p2Atk1==1){
        gotoAndPlay("Scene 6",1);
    }
    //กรณีที่เราATK แต่อีกฝั่งDEF//
    else if(Atk1==1&p2Atk1==0){
        gotoAndPlay("Scene 7",1);
    }
    //กรณีที่เราเป็นฝ่ายDEF แล้วฝ่ายตรงข้าม ATK//
    else if(Atk1==0&p2Atk1==1){
        gotoAndPlay("Scene 8",1);
    }
    //กรณีที่เป็นDEFทั้งคู่//
    else {
        gotoAndPlay("Scene 5",1);
    }
}

```

Frame : 75

```

//กรณีที่เป็น ATK ทั้งคู่//
if(Atk1==1&p2Atk1==1){
    amountscore1=amountscore1-p2score;
    score1.text=amountscore1;
    amountscore2=amountscore2-myscore;
    score2.text=amountscore2;
}

```

```

//กรณีที่เราATK แต่อีกฝั่งDEF//
else if(Atk1==1&p2Atk1==0){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

score1.text=amountscore1;
if (myscore>=p2score){
    amountscore2=amountscore2-(myscore-p2score);
    score2.text=amountscore2;
}else {
    score2.text=amountscore2;
}
}
//กรณีที่เราเป็นฝ่ายDEF แล้วฝ่ายตรงข้าม ATK//
else if(Atk1==0&p2Atk1==1){
    score2.text=amountscore2;
    if(myscore<=p2score){
        amountscore1=amountscore1-(p2score-myscore);
        score1.text=amountscore1;
    }else{
        score1.text=amountscore1;
    }
}
//กรณีที่เป็นDEFทั้งคู่//
else {
    score1.text=amountscore1;
    score2.text=amountscore2;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//กรณีที่เป็น ATK ทั้งคู่
if(Atk2==1&p2Atk2==1){
gotoAndPlay("Scene 6",1);
}
//กรณีที่เราATK แต่อีกฝั่งDEF//
else if(Atk2==1&p2Atk2==0){
gotoAndPlay("Scene 7",1);
}
//กรณีที่เราเป็นฝ่ายDEF แล้วฝ่ายตรงข้าม ATK//
else if(Atk2==0&p2Atk2==1){
gotoAndPlay("Scene 8",1);
}
//กรณีที่ป็นDEFทั้งคู่//
else {
gotoAndPlay("Scene 5",1);
}
}

```

Frame : 105

```

//กรณีที่เป็น ATK ทั้งคู่//
if(Atk2==1&p2Atk2==1){
amountscore1=amountscore1-p2score;
score1.text=amountscore1;
amountscore2=amountscore2-myscore;
score2.text=amountscore2;
}
//กรณีที่เราATK แต่อีกฝั่งDEF//
else if(Atk2==1&p2Atk2==0){
score1.text=amountscore1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (myscore>=p2score){
    amountscore2=amountscore2-(myscore-p2score);
    score2.text=amountscore2;
} else {
    score2.text=amountscore2;
}
}
//กรณีที่เราเป็นฝ่ายDEF แล้วฝ่ายตรงข้าม ATK//
else if(Atk2==0&p2Atk2==1){
    score2.text=amountscore2;
    if(myscore<=p2score){
        amountscore1=amountscore1-(p2score-myscore);
        score1.text=amountscore1;
    } else {
        score1.text=amountscore1;
    }
}
//กรณีที่เป็นDEFทั้งคู่//
else {
    score1.text=amountscore1;
    score2.text=amountscore2;
}
}

```

**Frame : 124**

```

if (amountscore1 == 0 | amountscore1<0) {
    gotoAndPlay(125);
} else if (amountscore2<=0) {
    gotoAndPlay(126);
} else gotoAndPlay(57);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Layer : Layer 14**

**Frame : 75**

```
picPlayer1 = new Array(["Yugi", "yugi1.jpg"], ["Kaiba", "kaiba1.jpg"]);  
if(player==1){  
    loadMovie(picPlayer1[player-1][1], "_root.loadPlayer1");  
    loadMovie(picPlayer1[player][1], "_root.loadPlayer2");  
}else {  
    loadMovie(picPlayer1[player-1][1], "_root.loadPlayer1");  
    loadMovie(picPlayer1[player-2][1], "_root.loadPlayer2");  
}
```

**Frame : 105**

```
picPlayer1 = new Array(["Yugi", "yugi1.jpg"], ["Kaiba", "kaiba1.jpg"]);  
if(player==1){  
    loadMovie(picPlayer1[player-1][1], "_root.loadPlayer1");  
    loadMovie(picPlayer1[player][1], "_root.loadPlayer2");  
}else {  
    loadMovie(picPlayer1[player-1][1], "_root.loadPlayer1");  
    loadMovie(picPlayer1[player-2][1], "_root.loadPlayer2");  
}
```

**Frame : 125**

```
stop();
```

**Frame : 126**

```
stop();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Frame : 127**

```
stop();
```

**Layer : loadPlayer**

**Frame : 40**

```
amountscore1 = 10000;
```

```
amountscore2 = 10000;
```

```
var validate = false;
```

```
var host = "192.168.0.2";
```

```
var port = 8000;
```

```
var numClients;
```

```
function connect() {
```

```
    mySocket = new XMLSocket();
```

```
    mySocket.onConnect = handleConnect;
```

```
    mySocket.onClose = handleClose;
```

```
    mySocket.onXML = handleIncoming;
```

```
    if (!mySocket.connect(host, port)) {
```

```
        errorMsg.text = "Connection Failed.\nPlease contact administrator.";
```

```
    }
```

```
}
```

```
function handleConnect(success) {
```

```
    if (success) {
```

```
        mySocket.connected = true;
```

```
        mySocket.send(userID);
```

```
    } else {
```

```
        _root.errorMsg.text = "Connection Failed.";
```

```
    }
```

```
}
```

```
function handleIncoming(incomingObj) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

trace("-----start data received-----");
trace(incomingObj.toString());
trace("-----end data received-----");

if (incomingObj.firstChild.nodeName == "VALIDATE") {
    if (int(incomingObj.firstChild.firstChild.nodeValue) == 1) {
        validate = true;
    } else {
        errorMsg.text = "Your player is duplicate, please select a new one.";
        quit();
    }
} else if (incomingObj.firstChild.nodeName == "NUMCLIENTS") {
    if (validate) {
        numClients = incomingObj.firstChild.firstChild.nodeValue;
        if (numClients == 1) {
            gotoAndStop(55);
        } else if (numClients == 2 && !played) {
            gotoAndPlay(56);
        } else if (numClients > 2 && !played) {
            errorMsg.text = "Players exceeded now.\nPlease wait for
anyone left.";
            quit();
        }
    }
} else {
    var user = unescape(incomingObj.firstChild.firstChild.nodeValue);
    if (incomingObj.childNodes[1].nodeName == "X1") {
        mode = "move";
        var x1 = unescape(incomingObj.childNodes[1].firstChild.nodeValue);
        var shoot1 =
unescape(incomingObj.childNodes[2].firstChild.nodeValue);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        var x2 = unescape(incomingObj.childNodes[3].firstChild.nodeValue);
        var shoot2 =
unescape(incomingObj.childNodes[4].firstChild.nodeValue);
    } else if (incomingObj.childNodes[1].nodeName == "MESSAGE") {
        mode = "chat";
        var msg = unescape(incomingObj.childNodes[1].firstChild.nodeValue);
    }
}
if (mode == "chat") {
    _root.text2 = msg;
} else if (mode == "move" & user!= userID ) {
    _root.p2CardNo_1 = x1;
    _root.p2Atk1 = shoot1;
    _root.p2CardNo_2 = x2;
    _root.p2Atk2 = shoot2;
}
}
function sendMessage(x1, shoot1, x2, shoot2, x3, shoot3, x4, shoot4, x5, shoot5, mode) {
    if (mode == "move") {
        var message =
"<USER>"+escape(userID)+"</USER><X1>"+escape(x1)+"</X1><SHOOT1>"+escape(shoot1)
+"</SHOOT1><X2>"+escape(x2)+"</X2><SHOOT2>"+escape(shoot2)+"</SHOOT2><X3>"+
escape(x3)+"</X3><SHOOT3>"+escape(shoot3)+"</SHOOT3><X4>"+escape(x4)+"</X4><SH
OOT4>"+escape(shoot4)+"</SHOOT4><X5>"+escape(x5)+"</X5><SHOOT5>"+escape(shoot5
)+"</SHOOT5>";
    } else if (mode == "chat") {
        var message =
"<USER>"+escape(userID)+"</USER><MESSAGE>"+escape(x1)+"</MESSAGE>";
    }
    outgoingObj = new XML();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outgoingObj.parseXML(message);
trace("Sending: "+outgoingObj);
if (mySocket && mySocket.connected) {
    mySocket.send(outgoingObj);
} else {
    quit();
}
}
function handleClose() {
    mySocket.connected = false;
    numClients = 0;
}
function quit() {
    if (mySocket.connected) {
        mySocket.close();
        mySocket.connected = false;
        gotoAndStop(40);
    }
}
}

```

**Frame : 55**

```

// waiting frame//
amountscore1 = 10000;
amountscore2 = 10000;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Frame : 57**

```
// playinggame//  
stopAllSounds();  
picPlayer = new Array(["Yugi", "yugi.jpg"], ["Kaiba", "kaiba.jpg"]);  
loadMovie(picPlayer[player-1][1], "_root.loadPlayer");  
loadMovie(picPlayer[player][1], "_root.loadPlayer2");
```

**Layer : selectplayer**

**Frame : 40-54**

**ปุ่มเลือกยูกิ**

```
on (press) {  
    userID = "Yugi";  
    player = 1;  
    connect();  
}
```

**ปุ่มเลือกไคบะ**

```
on (press) {  
    userID = "Kaiba";  
    player = 2;  
    connect();  
}
```

**Frame : 50**

```
gotoAndPlay(40);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Layer : start**

**Frame : 34**

stop();

**ที่ปุ่ม**

```
on (release) {  
    gotoAndPlay(40);  
}
```

**Scene 2**

**Layer : y4**

**Frame : 122**

```
if (c==1){  
    gotoAndPlay("Scene 1",75);  
}else {  
    gotoAndPlay("Scene 1",105);  
}
```

**Scene 3**

**Layer : y1**

**Frame : 60**

```
if (c==1){  
    gotoAndPlay("Scene 1",75);  
}else {  
    gotoAndPlay("Scene 1",105);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Scene 4**

**Layer : y4**

**Frame : 65**

```
if(c==1){
    gotoAndPlay("Scene 1",75);
}else {
    gotoAndPlay("Scene 1",105);
}
```

**Scene 5**

**Layer : Layer1**

**Frame : 23**

```
if(c==1){
    gotoAndPlay("Scene 1",75);
}else {
    gotoAndPlay("Scene 1",105);
}
```

**Scene 6**

**Layer : Layer1**

**Frame : 120**

```
if(c==1){
    gotoAndPlay("Scene 1",75);
}else {
    gotoAndPlay("Scene 1",105);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scene 7

Layer : Layer4

Frame : 61

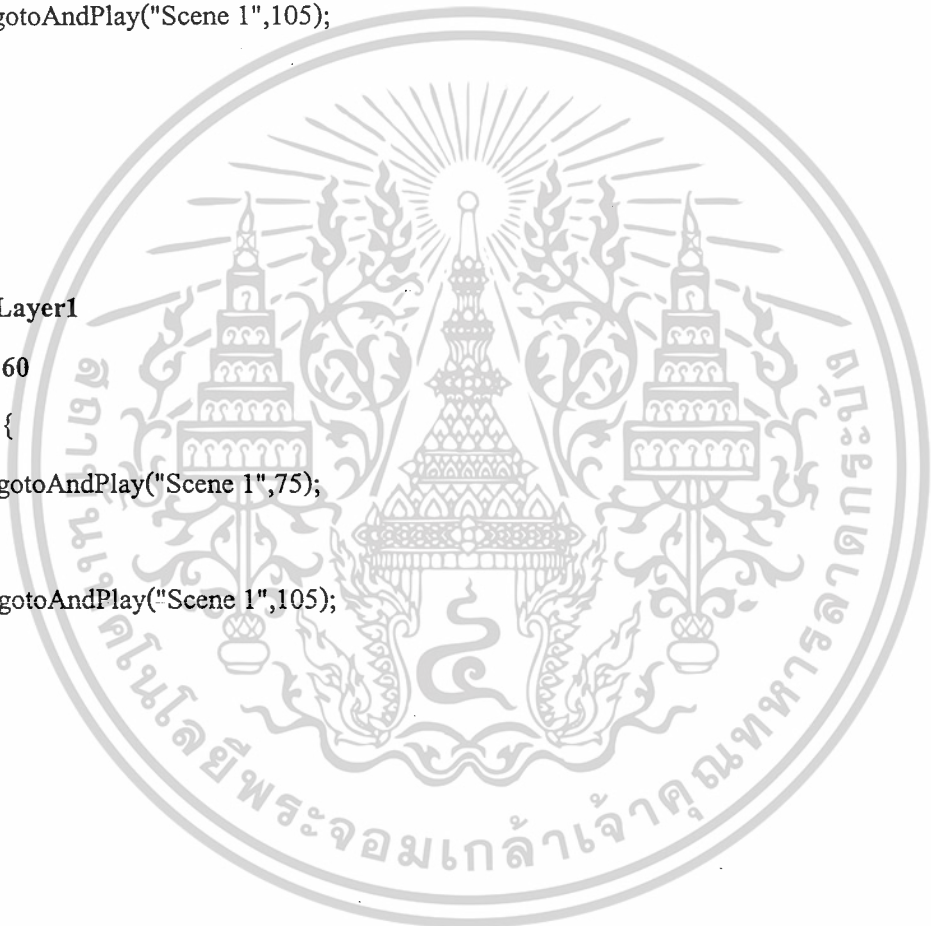
```
if(c==1){  
    gotoAndPlay("Scene 1",75);  
}else {  
    gotoAndPlay("Scene 1",105);  
}
```

Scene 8

Layer : Layer1

Frame : 60

```
if(c==1){  
    gotoAndPlay("Scene 1",75);  
}else {  
    gotoAndPlay("Scene 1",105);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Server.java

```
import java.awt.event.*;
import java.util.*;
import java.awt.*;
import java.io.*;
import java.net.*;

public class Server {
    private Vector players = new Vector();
    // Vector is a list of all connected clients
    ServerSocket server; // the server

    // constructor
    public Server(int port) {
        startServer(port);
    }

    public static void main(String[] args) {
        if(args.length == 1) {
            Server mySvr = new Server(Integer.parseInt(args[0]));
        } else {
            // otherwise give correct usage
            System.out.println("Usage: Server [port]");
        }
    }

    private void startServer(int port) {
        showStatus("Attempting to Start Server");

        try {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// create a new server
server = new ServerSocket(port);
showStatus("Server Started on Port: " + port);
// while the server is active...
while(true) {
    // ...listen for new client connections
    Socket socket = server.accept();
    Client player = new Client(this, socket);
    showStatus(player.getIP() + " connected.");

    // get user name
    player.initName();

    if (isNameDuplicate(player.getMyName()) != 0) {
        player.send("<VALIDATE>0</VALIDATE>0");
    }
    else {
        player.send("<VALIDATE>1</VALIDATE>0");

        // add the new client to our client list
        players.addElement(player);

        // start the client thread
        player.start();

        // broadcast the new number of clients
        broadcastMessage("<NUMCLIENTS>" + players.size()
            + "</NUMCLIENTS>");
    }
} //end of while

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} catch(IOException e) {
    showStatus("Server Error... Stopping Server");
    // kill this server
    killServer();
}
}

```

```

public synchronized void broadcastMessage(String message) {
    // add the null character to the message
    message += '\0';

    // send message to each client
    Enumeration enum = players.elements();
    while (enum.hasMoreElements()) {
        Client player = (Client)enum.nextElement();
        player.send(message);
    }
}

```

```

private int isNameDuplicate(String newName) {
    // Check for the duplicate of the users' name
    // return 0 --> name not duplicate
    // return -1 --> name is duplicate
    Enumeration enum = players.elements();
    while (enum.hasMoreElements()) {
        Client player = (Client)enum.nextElement();
        if (newName.compareTo(player.getMyName()) == 0) {
            return -1;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return 0;
}

public void removeClient(Client player) {
    showStatus(player.getIP() + " has left the server.");

    // remove the client from the list
    players.removeElement(player);

    // broadcast the new number of clients
    broadcastMessage("<NUMCLIENTS>" + players.size() + "</NUMCLIENTS>");
}

public void showStatus(String str){
    // get the current date and time
    Calendar cal = Calendar.getInstance();
    str = "[" + cal.get(Calendar.MONTH) + "/"
        + cal.get(Calendar.DAY_OF_MONTH)
        + "/" + cal.get(Calendar.YEAR)
        + " " + cal.get(Calendar.HOUR_OF_DAY)
        + ":" + cal.get(Calendar.MINUTE)
        + ":" + cal.get(Calendar.SECOND)
        + "]" + str + "\n";

    // display the activity
    System.out.print(str);
}

private void killServer() {
    try {
        // stop the server

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

server.close();
setStatus("Server Stopped");
} catch (IOException e) {
setStatus("Error!");
}
}
} // end of server class

```



#### Client.java

```

import java.net.*;
import java.util.*;
import java.io.*;

```

```

public class Client extends Thread {
private Thread myThread; // thread
private Socket socket;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private Server server;

private String ip;

private String myName;      // Player's name
protected BufferedReader inMsg; // in messages
protected PrintWriter outMsg; // out messages

public Client(Server server, Socket socket) {
    this.server = server;
    this.socket = socket;
    this.ip = socket.getInetAddress().getHostAddress();

    // initial incoming and outgoing message
    try {
        inMsg = new BufferedReader(new InputStreamReader (socket.getInputStream()));
        outMsg = new PrintWriter(socket.getOutputStream(), true);
        // enable autoflush
    } catch(IOException e) {
        server.showStatus("Error! ");
        killClient();
    }
}

// Thread run method
public void run() {
    try {

        char charBuffer[] = new char[1];

        // declare char array buffer size=1

        while(inMsg.read(charBuffer,0,1) != -1) {

            // read by setting offset=0 and max char read=1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// create a string buffer to hold the incoming stream
StringBuffer stringBuffer = new StringBuffer(1024);

// while not end of string
while(charBuffer[0] != '\0') {
// append the char read to charBuffer
stringBuffer.append(charBuffer[0]);
inMsg.read(charBuffer, 0,1);
}
// broadcast the message
server.broadcastMessage(stringBuffer.toString());
}
} catch(IOException e) {
server.showStatus("Error! ");
} finally {
killClient();
}
}

public String getIP() {
return ip;
}

public String getMyName() {
return myName;
}

// send message to this client by calling server's broadcast method
public void send(String message) {
// put the message in the buffer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outMsg.print(message);
if(outMsg.checkError()) {
    server.showStatus("Error! ");
    killClient();
}
}

```

```

private void killClient() {
    server.removeClient(this);
    try {
        socket.close();
        inMsg.close();
        outMsg.close();
        myThread = null;
    } catch (IOException e) {
        server.showStatus("Error!");
    }
}

```

// initName has the same method of operation as run thread ,

// but it just once call at the first time user logging in

```

public void initName() {

```

```

    try {

```

```

        char charBuffer[] = new char[1];

```

```

        // declare char array buffer size=1

```

```

            if(inMsg.read(charBuffer,0,1) != -1) {

```

```

                // read by setting offset=0 and max char read=1

```

```

                // create a string buffer to hold the incoming stream

```

```

                StringBuffer stringBuffer = new StringBuffer(1024);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// while not end of string
while(charBuffer[0] != '\0') {
    // append the char read to charBuffer
    stringBuffer.append(charBuffer[0]);
    inMsg.read(charBuffer, 0,1);
}

myName = stringBuffer.toString();
}
} catch(IOException e) {
server.showStatus("Error! ");
}
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ ยูกิ และ ไคบา จะเหมือนกัน ต่างกันตรงตรงที่ yugi จะเป็น kaiba

```
function varargout = yugi(varargin)
```

```
% YUGI M-file for yugi.fig
```

```
% YUGI, by itself, creates a new YUGI or raises the existing
```

```
% singleton*.
```

```
%
```

```
% H = YUGI returns the handle to a new YUGI or the handle to
```

```
% the existing singleton*.
```

```
%
```

```
% YUGI('CALLBACK', hObject, eventData, handles,...) calls the local
```

```
% function named CALLBACK in YUGI.M with the given input arguments.
```

```
%
```

```
% YUGI('Property','Value',...) creates a new YUGI or raises the
```

```
% existing singleton*. Starting from the left, property value pairs are
```

```
% applied to the GUI before yugi_OpeningFunction gets called. An
```

```
% unrecognized property name or invalid value makes property application
```

```
% stop. All inputs are passed to yugi_OpeningFcn via varargin.
```

```
%
```

```
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
```

```
% instance to run (singleton)".
```

```
%
```

```
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help yugi
```

```
% Last Modified by GUIDE v2.5 11-Mar-2005 15:58:30
```

```
% Begin initialization code - DO NOT EDIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @yugi_OpeningFcn, ...
                  'gui_OutputFcn', @yugi_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before yugi is made visible.
function yugi_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to yugi (see VARARGIN)

% Choose default command line output for yugi
handles.output = hObject;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**% Update handles structure**

**guidata(hObject, handles);**

**% UIWAIT makes yugi wait for user response (see UIRESUME)**

**% uiwait(handles.figure1);**

**% — Outputs from this function are returned to the command line.**

**function varargout = yugi\_OutputFcn(hObject, eventdata, handles)**

**% varargout cell array for returning output args (see VARARGOUT);**

**% hObject handle to figure**

**% eventdata reserved - to be defined in a future version of MATLAB**

**% handles structure with handles and user data (see GUIDATA)**

**% Get default command line output from handles structure**

**varargout{1} = handles.output;**

**% — YUGI -----**

**% — Executes on button press in Cancel.**

**function Cancel\_Callback(hObject, eventdata, handles)**

**% hObject handle to pushbutton3 (see GCBO)**

**% eventdata reserved - to be defined in a future version of MATLAB**

**% handles structure with handles and user data (see GUIDATA)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**% — Executes on button press in Hide.**

**function Hide\_Callback(hObject, eventdata, handles)**

**VFM('show', 0);**

**% — Executes on button press in Exit.**

**function Exit\_Callback(hObject, eventdata, handles)**

**% hObject handle to Exit (see GCBO)**

**% eventdata reserved - to be defined in a future version of MATLAB**

**% handles structure with handles and user data (see GUIDATA)**

**quit force;**

**% — Executes on button press in Preview.**

**function Preview\_Callback(hObject, eventdata, handles)**

**% hObject handle to pushbutton1 (see GCBO)**

**% eventdata reserved - to be defined in a future version of MATLAB**

**% handles structure with handles and user data (see GUIDATA)**

**VFM('show', 1);**

**vf('preview',1);**

**% — Executes on button press in Go.**

**function Go\_Callback(hObject, eventdata, handles)**

**% hObject handle to pushbutton2 (see GCBO)**

**% eventdata reserved - to be defined in a future version of MATLAB**

**% handles structure with handles and user data (see GUIDATA)**

**%clear;**

**%close all;**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%fid = fopen('C:\windows\Desktop\buffercard.txt', 'w');
fid = fopen('C:\Documents and Settings\ACER\Desktop\yugi\buffercard.txt', 'w');
status = fclose(fid);
load Total_w_for_BAM_30_Classify_E121105;

vfm('grab',1);           % receive image from webcam with "Video for MATLAB"
program
vfm('preview',1);

ans_gray = rgb2gray(ans);
ans = rgb2hsv(ans);
ans_HSV = ans;
ans_gray_double = double(ans_gray); % convert matrix to double for enable calculate
buffer_dimension = size(ans_gray);
dimensionx = buffer_dimension(1,1);
dimensiony = buffer_dimension(1,2);
imshow(hsv2rgb(ans));
% find min max intensity of image
buffer_min = ans_gray_double(1,1); % set minimum intensity of matrix
buffer_max = ans_gray_double(1,1); % set maximum intensity of matrix

for i=1:1:dimensionx % fine min/max intensity for create theshold
    for j=1:1:dimensiony
        if ans_gray(i,j) < buffer_min
            buffer_min = ans_gray_double(i,j);
        end
        if ans_gray(i,j) > buffer_max
            buffer_max = ans_gray_double(i,j);
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
```

```
end
```

```
% Enhancement image by Linear min/max scaling -----
```

```
b = -buffer_min;
```

```
a = 255/(buffer_max - buffer_min);
```

```
for i=1:1:dimensionx
```

```
    for j=1:1:dimensiony
```

```
        ans_gray_double(i,j) = (ans_gray_double(i,j) + b)*a;
```

```
    end
```

```
end
```

```
%-----
```

```
% Threshold image by mid-range -----
```

```
Threshold = (buffer_min + buffer_max)/2; % create mid-range threshold value
```

```
ans_binary_double = zeros(dimensionx,dimensiony); % initial binary matrix
```

```
for i=1:1:dimensionx % threshold
```

```
    for j=1:1:dimensiony
```

```
        if ans_gray_double(i,j) < Threshold
```

```
            ans_binary_double(i,j) = 0;
```

```
        end
```

```
        if ans_gray_double(i,j) >= Threshold
```

```
            ans_binary_double(i,j) = 1;
```

```
        end
```

```
    end
```

```
end
```

```
figure(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
imshow(logical(uint8(ans_binary_double)));
```

```
%-----
```

```
% Segmentation card-----
```

```
% Counter card in scene-----
```

```
i = 1;
```

```
j = 1;
```

```
quantity_of_card_LR = 0;
```

```
while j < dimensionx % if column over dimensiony, it is end of picture
```

```
    %-fine Left black pixel-----
```

```
    while ans_binary_double(i,j) == 1 & i ~= dimensionx % lock repeat statement don't  
over dimensionx rows
```

```
        if i ~= dimensionx
```

```
            i = i+1; % go to next row if found white pixel if a row not end
```

```
        end
```

```
        if i == dimensionx & j < dimensiony % if row = dimensionx(end row)
```

```
            j = j+1; % next to column
```

```
            i = 1; % return to first row
```

```
        end
```

```
    end
```

```
    if i ~= dimensionx & j ~= dimensiony % lock save position of card don't over  
dimensionx rows and don't over dimensiony column
```

```
        switch quantity_of_card_LR
```

```
            case 0
```

```
                save_positionL0 = zeros(1,2); %initial buffer for keep position of Left cards 1
```

```
                save_positionL0(1,1) = i; % save first left black pixel in row of card 1
```

```
                save_positionL0(1,2) = j; % save first left black pixel in column of card 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1
    save_positionL1 = zeros(1,2); %initial buffer for keep position of Left cards 2
    save_positionL1(1,1) = i; % save first left black pixel in row of card 2
    save_positionL1(1,2) = j; % save first left black pixel in column of card 2
    otherwise
end
end
%-----
%-fine Right black pixel -----
i = 1; % return to first row
if j < dimensiony
    j = j + 1; % next to column if column don't over dimensiony
end
while ans_binary_double(i,j) == 1 & i ~= dimensionx % if found white pixel and not
end row do statement
    while ans_binary_double(i,j) == 1 & i ~= dimensionx % if found white pixel and
not end row do go to next row
        i = i+1;
    end
    if i ~= dimensionx & j ~= dimensiony % if row is not end, save i,j and
go to first row and go to next column
        switch quantity_of_card_LR
            case 0
                save_positionR0 = zeros(1,2);
                save_positionR0(1,1) = i;
                save_positionR0(1,2) = j;
            case 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        save_positionR1(1,1) = i;
        save_positionR1(1,2) = j;
        otherwise
        end
        i = 1;
        j = j + 1;
    end
end
if j < dimensiony % lock increase quantity of card don't over
dimensiony column
    quantity_of_card_LR = quantity_of_card_LR + 1; % increase 1 card if found first and
end black pixel
    i = 1; % go to first row and go to next column
    if j < dimensiony
        j = j + 1;
    end
end
end
end
%-----

% Find Bottom black pixel of cards-----
counter = 0; % counter for count column to compare previous black pixel
quantity_of_card_B = 0;

while quantity_of_card_B ~= quantity_of_card_LR
    switch quantity_of_card_B
        case 0
            i = save_positionL0(1,1); % move to left black pixel of card 1
            j = save_positionL0(1,2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1
    i = save_positionL1(1,1);          % move to left black pixel of card 2
    j = save_positionL1(1,2);
otherwise
end

while i ~= dimensionx                % repeat statement = dimensionx (end row)
    i = i+1;                          % increase row for find bottom black pixel
    counter = 0;
    if ans_binary_double(i,j) == 0    % save position row and column if found black
pixel
        switch quantity_of_card_B
            case 0
                save_positionB0 = zeros(1,2);    % initial buffer for keep Bottom black
pixel
                save_positionB0(1,1) = i;
                save_positionB0(1,2) = j;
            case 1
                save_positionB1 = zeros(1,2);
                save_positionB1(1,1) = i;
                save_positionB1(1,2) = j;
            otherwise
        end
    end
end

if ans_binary_double(i,j) == 1
    while ans_binary_double(i,j) == 1 & j < dimensiony    % increase column if found
white pixel and not end column
        j = j+1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        counter = counter+1;           % increase counter to threshold boundary of
card
    end
    if counter > 30                   % 50 is threshold value for segment card , it must
vary when changing resolution
        i = dimensionx;
    else
        switch quantity_of_card_B
        case 0
            save_positionB0(1,1) = i;
            save_positionB0(1,2) = j;
        case 1
            save_positionB1(1,1) = i;
            save_positionB1(1,2) = j;
        otherwise
        end
    end
end
end
end
quantity_of_card_B = quantity_of_card_B + 1;
end
%-----

```

**% Find Top black pixel of cards-----**

```

counter = 0;           % counter for count column to compare previous black pixel
quantity_of_card_T = 0;

```

**while quantity\_of\_card\_T ~= quantity\_of\_card\_LR**

**switch quantity\_of\_card\_T**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0
    save_positionT0 = zeros(1,2);           % initial buffer for keep Top black pixel
    i = save_positionL0(1,1);              % move to left black pixel of card 1
    j = save_positionL0(1,2);
case 1
    save_positionT1 = zeros(1,2);
    i = save_positionL1(1,1);              % move to left black pixel of card 2
    j = save_positionL1(1,2);
otherwise
end

while i ~= 1
    % repeat statement = 1 (first row)
    i = i-1;                               % decrease row for find top black pixel
    counter = 0;
    if ans_binary_double(i,j) == 0         % save position row and column if found black
pixel
        switch quantity_of_card_T
            case 0
                save_positionT0(1,1) = i;
                save_positionT0(1,2) = j;
            case 1
                save_positionT1(1,1) = i;
                save_positionT1(1,2) = j;
            otherwise
        end
    end

    if ans_binary_double(i,j) == 1
        while ans_binary_double(i,j) == 1 & j < dimensiony % increase column if found

```

**white pixel and not end column**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

j = j+1;
counter = counter+1;          % increase counter to threshold boundary of
card
end
if counter > 30                % 50 is threshold value for segment card , it must
vary when changing resolution
i = 1;
else
switch quantity_of_card_T
case 0
save_positionT0(1,1) = i;
save_positionT0(1,2) = j;
case 1
save_positionT1(1,1) = i;
save_positionT1(1,2) = j;
otherwise
end
end
end
end
end
quantity_of_card_T = quantity_of_card_T + 1;
end
%-----

```

**% Segmentation card from group of card -----**

**quantity\_of\_card = 0;**

**north0 = zeros(1,2);north1 = zeros(1,2);**

**south0 = zeros(1,2);south1 = zeros(1,2);**

**east0 = zeros(1,2);east1 = zeros(1,2);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

west0 = zeros(1,2);west1 = zeros(1,2);
northwest0 = zeros(1,2);northwest1 = zeros(1,2);
southwest1 = zeros(1,2);
northeast0 = zeros(1,2);northeast1 = zeros(1,2);
new_row0 = 0;new_row1 = 0;new_row2 = 0;
new_column0 = 0;new_column1 = 0;

```

```

% Add boundary picture -----

```

```

while quantity_of_card ~= quantity_of_card_LR

```

```

    switch quantity_of_card

```

```

        case 0

```

```

            i = save_positionL0(1,1); % move to left black pixel of card 1

```

```

            j = save_positionL0(1,2);

```

```

            j = j-1;

```

```

            west0(1,1) = i;

```

```

            west0(1,2) = j;

```

```

            i = save_positionR0(1,1); % move to Right black pixel of card 1

```

```

            j = save_positionR0(1,2);

```

```

            j = j+1;

```

```

            east0(1,1) = i;

```

```

            east0(1,2) = j;

```

```

            i = save_positionT0(1,1); % move to Top black pixel of card 1

```

```

            j = save_positionT0(1,2);

```

```

            i = i-1;

```

```

            north0(1,1) = i;

```

```

            north0(1,2) = j;

```

```

            i = save_positionB0(1,1); % move to Bottom black pixel of card 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
j = save_positionB0(1,2);
```

```
i = i+1;
```

```
south0(1,1) = i;
```

```
south0(1,2) = j;
```

```
case 1
```

```
i = save_positionL1(1,1);           % move to left black pixel of card 2
```

```
j = save_positionL1(1,2);
```

```
j = j-1;
```

```
west1(1,1) = i;
```

```
west1(1,2) = j;
```

```
i = save_positionR1(1,1);           % move to Right black pixel of card 2
```

```
j = save_positionR1(1,2);
```

```
j = j+1;
```

```
east1(1,1) = i;
```

```
east1(1,2) = j;
```

```
i = save_positionT1(1,1);           % move to Top black pixel of card 2
```

```
j = save_positionT1(1,2);
```

```
i = i-1;
```

```
north1(1,1) = i;
```

```
north1(1,2) = j;
```

```
i = save_positionB1(1,1);           % move to Bottom black pixel of card 2
```

```
j = save_positionB1(1,2);
```

```
i = i+1;
```

```
south1(1,1) = i;
```

```
south1(1,2) = j;
```

```
otherwise
```

```
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**% copy any pixel-----**

**switch quantity\_of\_card**

**case 0**

**northwest0(1,1) = west0(1,1);**

**northwest0(1,2) = west0(1,2);**

**while northwest0(1,1) > north0(1,1)**

**northwest0(1,1) = northwest0(1,1) - 1;**

**end**

**southwest0(1,1) = west0(1,1);**

**southwest0(1,2) = west0(1,2);**

**while southwest0(1,1) < south0(1,1)**

**southwest0(1,1) = southwest0(1,1) + 1;**

**end**

**northeast0(1,1) = east0(1,1);**

**northeast0(1,2) = east0(1,2);**

**while northeast0(1,1) > north0(1,1)**

**northeast0(1,1) = northeast0(1,1) - 1;**

**end**

**new\_row0 = southwest0(1,1) - northwest0(1,1) + 1;      % create array to keep**

**segmented picture in binary and gray scale**

**new\_column0 = northeast0(1,2) - northwest0(1,2) + 1;**

**segmented\_card0\_binary = zeros(new\_row0,new\_column0);**

**segmented\_card0\_gray = zeros(new\_row0,new\_column0);**

**segmented\_card0\_hue = zeros(new\_row0,new\_column0);**

**segmented\_card0\_saturate = zeros(new\_row0,new\_column0);**

**segmented\_card0\_value = zeros(new\_row0,new\_column0);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i1 = 0;
j1 = 0;
for i = northwest0(1,1):1:southwest0(1,1)
    i1 = i1+1;
    j1 = 0;
    for j = northwest0(1,2):1:northeast0(1,2)
        j1 = j1+1;
        segmented_card0_binary(i1,j1) = ans_binary_double(i,j);
        segmented_card0_gray(i1,j1) = ans_gray_double(i,j);
        segmented_card0_hue(i1,j1) = ans_HSV(i,j,1);
        segmented_card0_saturnate(i1,j1) = ans_HSV(i,j,2);
        segmented_card0_value(i1,j1) = ans_HSV(i,j,3);
    end
end
% figure(2);
% imshow(logical(uint8(segmented_card0_binary)));

```

case 1

```

northwest1(1,1) = west1(1,1);
northwest1(1,2) = west1(1,2);
while northwest1(1,1) > north1(1,1)
    northwest1(1,1) = northwest1(1,1) - 1;
end
southwest1(1,1) = west1(1,1);
southwest1(1,2) = west1(1,2);
while southwest1(1,1) < south1(1,1)

```

```

    southwest1(1,1) = southwest1(1,1) + 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end

**northeast1(1,1) = east1(1,1);**

**northeast1(1,2) = east1(1,2);**

**while northeast1(1,1) > north1(1,1)**

**northeast1(1,1) = northeast1(1,1) - 1;**

end

**new\_row1 = southwest1(1,1) - northwest1(1,1);      % create array to keep**

**segmented picture in binary and gray scale**

**new\_column1 = northeast1(1,2) - northwest1(1,2);**

**segmented\_card1\_binary = zeros(new\_row1,new\_column1);**

**segmented\_card1\_gray = zeros(new\_row1,new\_column1);**

**segmented\_card1\_hue = zeros(new\_row1,new\_column1);**

**segmented\_card1\_saturate = zeros(new\_row1,new\_column1);**

**segmented\_card1\_value = zeros(new\_row1,new\_column1);**

**i1 = 0;**

**j1 = 0;**

**for i = northwest1(1,1):1:southwest1(1,1)**

**i1 = i1+1;**

**j1 = 0;**

**for j = northwest1(1,2):1:northeast1(1,2)**

**j1 = j1+1;**

**segmented\_card1\_binary(i1,j1) = ans\_binary\_double(i,j);**

**segmented\_card1\_gray(i1,j1) = ans\_gray\_double(i,j);**

**segmented\_card1\_hue(i1,j1) = ans\_HSV(i,j,1);**

**segmented\_card1\_saturate(i1,j1) = ans\_HSV(i,j,2);**

**segmented\_card1\_value(i1,j1) = ans\_HSV(i,j,3);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    end
end
% figure(3);
% imshow(logical(uint8(segmented_card1_binary)));

otherwise
end
quantity_of_card = quantity_of_card + 1;
end
%-----

% Decision status of card -----
quantity_of_card = 0;
buffer_status_card1 = 0; % Buffer_status is a buffer for keep status and sent it to Detect
lean part
buffer_status_card2 = 0;
buffer_status_card3 = 0;
buffer_status_card4 = 0;
buffer_status_card5 = 0;

while quantity_of_card ~= quantity_of_card_LR
    switch quantity_of_card
        case 0
            % decision status of card 1 -----

            buffer_status0 = size(segmented_card0_binary);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    fprintf('card 1 is attack \n');
    buffer_status_card1 = 1;
else
    fprintf('card 1 is defend \n');
    buffer_status_card1 = 2;
end
%-----
case 1
    % decision status of card 2 -----
    buffer_status1 = size(segmented_card1_binary);
    if buffer_status1(1,1) > buffer_status1(1,2)
        fprintf('card 2 is attack \n');
        buffer_status_card2 = 1;
    else
        fprintf('card 2 is defend \n');
        buffer_status_card2 = 2;
    end
    %-----
otherwise
end
quantity_of_card = quantity_of_card + 1;
end
%-----

% Color Classification by copy color from HUE -----
quantity_of_card = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while quantity_of_card ~= quantity_of_card_LR
```

```
    switch quantity_of_card
```

```
        case 0
```

```
            up1 = zeros(5,6);
```

```
            down1 = zeros(5,6);
```

```
            if buffer_status_card1 == 1
```

```
                for j = 16:1:21
```

```
                    for i = 69:1:73
```

```
                        up1(i-68,j-15) = segmented_card0_hue(i,j);
```

```
                    end
```

```
                end
```

```
                for j = 16:1:21
```

```
                    for i = 183:1:187
```

```
                        down1(i-182,j-15) = segmented_card0_hue(i,j);
```

```
                    end
```

```
                end
```

```
            else
```

```
                up1 = zeros(6,5);
```

```
                down1 = zeros(6,5);
```

```
                for j = 69:1:63
```

```
                    for i = 162:1:167
```

```
                        up1(i-161,j-68) = segmented_card0_hue(i,j);
```

```
                    end
```

```
                end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for j = 183:1:187
    for i = 162:1:167
        down1(i-161,j-182) = segmented_card0_hue(i,j);
    end
end

end

up1 = round(sum(sum(up1))); % keep summation of hue top pixel
down1 = round(sum(sum(down1))); % keep summation of hue bottom pixel

if (up1 == 20 | up1 == 19 | up1 == 18) & (down1 == 19 | down1 == 18) % blue
condition
    w_hue_1 = w_hue_blue; fprintf('blue ');
end
if (up1 == 22 | up1 == 21) & (down1 == 22 | down1 == 21) % blue condition
    w_hue_1 = w_hue_purple; fprintf('purple ');
end
if (up1 == 1 | up1 == 2) & (down1 == 1 | down1 == 2) % blue condition
    w_hue_1 = w_hue_brown; fprintf('brown ');
end
if (up1 == 4 | up1 == 3) & (down1 == 4 | down1 == 3) % blue condition
    w_hue_1 = w_hue_yellow; fprintf('yellow ');
end

case 1
    up2 = zeros(5,6);
    down2 = zeros(5,6);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if buffer_status_card2== 1
    for j = 16:1:21
        for i = 69:1:73
            up2(i-68,j-15) = segmented_card1_hue(i,j);
        end
    end

    for j = 16:1:21
        for i = 183:1:187
            down2(i-182,j-15) = segmented_card1_hue(i,j);
        end
    end
else
    up2 = zeros(6,5);
    down2 = zeros(6,5);

    for j = 69:1:73
        for i = 162:1:167
            up2(i-161,j-68) = segmented_card1_hue(i,j);
        end
    end

    for j = 183:1:187
        for i = 162:1:167
            down2(i-161,j-182) = segmented_card1_hue(i,j);
        end
    end
end

```

end

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

up2 = round(sum(sum(up2)));           % keep summation of hue top pixel
down2 = round(sum(sum(down2)));       % keep summation of hue bottom pixel

if (up2 == 20 | up2 == 19 | up2 == 18) &( down2 == 20 | down2 == 19 | down2 == 18)
% blue condition
    w_hue_2 = w_hue_blue; fprintf('blue ');
end
if (up2 == 22 | up2 == 21) &( down2 == 22 | down2 == 21) % blue condition
    w_hue_2 = w_hue_purple; fprintf('purple ');
end
if (up2 == 1 | up2 == 2) &( down2 == 1 | down2 == 2 | down2 == 3) % blue
condition
    w_hue_2 = w_hue_brown; fprintf('brown ');
end
if (up2 == 4 | up2 == 3) &( down2 == 4 | down2 == 3) % blue condition
    w_hue_2 = w_hue_yellow; fprintf('yellow ');
end

otherwise
end

quantity_of_card = quantity_of_card + 1;
end

% -----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**% Scaling a card to standard size -----**

**quantity\_of\_card = 0;**

**while quantity\_of\_card ~= quantity\_of\_card\_LR % lock repeat statement not over**

**quantity\_of\_card**

**switch quantity\_of\_card**

**case 0**

**if buffer\_status\_card1 == 1 % 220 165**

**segmented\_card0\_hue = imresize(segmented\_card0\_hue,[220 165],'bilinear');**

**% segmented\_card0\_hue = histeq(uint8segmented\_card0\_hue);**

**figure(17);**

**imshow(double(segmented\_card0\_hue));**

**segmented\_card0\_binary = im2bw(histeq(uint8(segmented\_card0\_gray)),0.66);**

**segmented\_card0\_binary = imresize(segmented\_card0\_binary,[220 165],'nearest');**

**segmented\_card0\_binary = double(segmented\_card0\_binary);**

**figure(18);**

**imshow(logical(segmented\_card0\_binary));**

**else**

**segmented\_card0\_hue = imresize(segmented\_card0\_hue,[165 220],'bilinear');**

**% segmented\_card0\_hue = histeq(uint8segmented\_card0\_hue);**

**figure(17);**

**imshow(double(segmented\_card0\_hue));**

**segmented\_card0\_binary = im2bw(histeq(uint8(segmented\_card0\_gray)),0.66);**

**segmented\_card0\_binary = imresize(segmented\_card0\_binary,[165 220],'nearest');**

**segmented\_card0\_binary = double(segmented\_card0\_binary);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
figure(18);  
imshow(logical(segmented_card0_binary));
```

```
end
```

```
case 1
```

```
if buffer_status_card2 == 1  
    segmented_card1_hue = imresize(segmented_card1_hue,[220 165],'bilinear');  
    % segmented_card1_hue = histeq(uint8segmented_card1_hue);  
    figure(19);  
    imshow(double(segmented_card1_hue));  
  
    segmented_card1_binary = im2bw(histeq(uint8(segmented_card1_gray)),0.66);  
    segmented_card1_binary = imresize(segmented_card1_binary,[220 165],'nearest');  
    segmented_card1_binary = double(segmented_card1_binary);  
    figure(20);  
    imshow(logical(segmented_card1_binary));
```

```
else
```

```
    segmented_card1_hue = imresize(segmented_card1_hue,[165 220],'bilinear');  
    % segmented_card1_hue = histeq(uint8segmented_card1_hue);  
    figure(19);  
    imshow(double(segmented_card1_hue));  
  
    segmented_card1_binary = im2bw(histeq(uint8(segmented_card1_gray)),0.66);  
    segmented_card1_binary = imresize(segmented_card1_binary,[165 220],'nearest');  
    segmented_card1_binary = double(segmented_card1_binary);  
    figure(20);  
    imshow(logical(segmented_card1_binary));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    end
    otherwise
end
quantity_of_card = quantity_of_card + 1;
end
%-----

% Crop picture and Threshold of each card -----
quantity_of_card = 0;
while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card
    switch quantity_of_card
        case 0
            if buffer_status_card1 == 1 % status is attack
                segmented_card0_hue = imcrop(segmented_card0_hue,[26 49 112 108]); %Crop
the image for cut edge because the edge is noise for moment invariant
                figure(32);
                imshow(double(segmented_card0_hue));

                segmented_card0_binary = imcrop(segmented_card0_binary,[26 49 112 108]);
                figure(33);
                imshow(logical(segmented_card0_binary));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**% Threshold image by mid-range% find min max intensity of image**

```
segmented_card0_hue1 = double(segmented_card0_hue);  
buffer_min = segmented_card0_hue(1,1);    % set minimum intensity of matrix  
buffer_max = segmented_card0_hue1(1,1);    % set maximum intensity of matrix  
dimension = size(segmented_card0_hue);
```

**% threshold hue**

```
for i=1:1:dimension(1,1)    % fine min/max intensity for create threshold
```

```
    for j=1:1:dimension(1,2)
```

```
        if segmented_card0_hue(i,j) < buffer_min
```

```
            buffer_min = segmented_card0_hue(i,j);
```

```
        end
```

```
        if segmented_card0_hue(i,j) > buffer_max
```

```
            buffer_max = segmented_card0_hue(i,j);
```

```
        end
```

```
    end
```

```
end
```

```
% -----
```

```
Threshold = (buffer_min + buffer_max)/2;    % create mid-range threshold value
```

```
for i=1:1:dimension(1,1)    % threshold
```

```
    for j=1:1:dimension(1,2)
```

```
        if segmented_card0_hue(i,j) < 0.66
```

```
            segmented_card0_hue1(i,j) = -1;
```

```
        end
```

```
        if segmented_card0_hue(i,j) >= 0.66
```

```
            segmented_card0_hue1(i,j) = 1;
```

```
        end
```

```
    end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end

for i=1:1:dimension(1,1) % translate binary to bipolar

for j=1:1:dimension(1,2)

if segmented\_card0\_binary(i,j) == 0

segmented\_card0\_binary(i,j) = -1;

end

end

end

figure(34);

imshow(logical(uint8(segmented\_card0\_hue1)));

else % status is defend

segmented\_card0\_hue = imcrop(segmented\_card0\_hue,[49 26 108 112]); %Crop

the image for cut edge because the edge is noise for moment invariant

figure(32);

imshow(double(segmented\_card0\_hue));

segmented\_card0\_binary = imcrop(segmented\_card0\_binary,[49 26 108 112]);

figure(33);

imshow(logical(segmented\_card0\_binary));

% Threshold image by mid-range% find min max intensity of image-----

segmented\_card0\_hue1 = double(segmented\_card0\_hue);

buffer\_min = segmented\_card0\_hue(1,1); % set minimum intensity of matrix

buffer\_max = segmented\_card0\_hue(1,1); % set maximum intensity of matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dimension = size(segmented_card0_hue);

% thershold hue
for i=1:1:dimension(1,1)    % fine min/max intensity for create theshold
    for j=1:1:dimension(1,2)
        if segmented_card0_hue(i,j) < buffer_min
            buffer_min = segmented_card0_hue(i,j);
        end
        if segmented_card0_hue(i,j) > buffer_max
            buffer_max = segmented_card0_hue(i,j);
        end
    end
end
end
% -----
Threshold = (buffer_min + buffer_max)/2;
for i=1:1:dimension(1,1)
    for j=1:1:dimension(1,2)
        if segmented_card0_hue(i,j) < 0.66
            segmented_card0_hue1(i,j) = -1;
        end
        if segmented_card0_hue(i,j) >= 0.66
            segmented_card0_hue1(i,j) = 1;
        end
    end
end
end
end

for i=1:1:dimension(1,1)    % translate binary to bipolar

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if segmented_card0_binary(i,j) == 0
            segmented_card0_binary(i,j) = -1;
        end
    end
end
end
figure(34);
imshow(logical(uint8(segmented_card0_hue1)));
end

case 1
    if buffer_status_card2 == 1 % status is attack
        segmented_card1_hue = imcrop(segmented_card1_hue,[26 49 112 108]); %Crop
the image for cut edge because the edge is noise for moment invariant
        figure(35);
        imshow(double(segmented_card1_hue));

        segmented_card1_binary = imcrop(segmented_card1_binary,[26 49 112 108]);
        figure(36);
        imshow(logical(segmented_card1_binary));

        % Threshold image by mid-range% find min max intensity of image-----
        -----
        segmented_card1_hue1 = double(segmented_card1_hue);
        buffer_min = segmented_card1_hue(1,1); % set minimum intensity of matrix
        buffer_max = segmented_card1_hue(1,1); % set maximum intensity of matrix
        dimension = size(segmented_card1_hue);
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% thershold hue
for i=1:1:dimension(1,1)    % fine min/max intensity for create theshold
    for j=1:1:dimension(1,2)
        if segmented_card1_hue(i,j) < buffer_min
            buffer_min = segmented_card1_hue(i,j);
        end
        if segmented_card1_hue(i,j) > buffer_max
            buffer_max = segmented_card1_hue(i,j);
        end
    end
end
end
% -----
Threshold = (buffer_min + buffer_max)/2; % create mid-range threshold value
for i=1:1:dimension(1,1)    % threshold
    for j=1:1:dimension(1,2)
        if segmented_card1_hue(i,j) < 0.66
            segmented_card1_hue1(i,j) = -1;
        end
        if segmented_card1_hue(i,j) >= 0.66
            segmented_card1_hue1(i,j) = 1;
        end
    end
end
end

for i=1:1:dimension(1,1)    % translate binary to bipolar
    for j=1:1:dimension(1,2)
        if segmented_card1_binary(i,j) == 0
            segmented_card1_binary(i,j) = -1;
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**end**

**end**

**figure(37);**

**imshow(logical(uint8(segmented\_card1\_hue1))));**

**else % status is defend**

**segmented\_card1\_hue = imcrop(segmented\_card1\_hue,[49 26 108 112]); %Crop**

**the image for cut edge because the edge is noise for moment invariant**

**figure(35);**

**imshow(double(segmented\_card1\_hue));**

**segmented\_card1\_binary = imcrop(segmented\_card1\_binary,[49 26 108 112]);**

**figure(36);**

**imshow(logical(segmented\_card1\_binary));**

**% Threshold image by mid-range% find min max intensity of image**

**segmented\_card1\_hue1 = double(segmented\_card1\_hue);**

**buffer\_min = segmented\_card1\_hue(1,1); % set minimum intensity of matrix**

**buffer\_max = segmented\_card1\_hue1(1,1); % set maximum intensity of matrix**

**dimension = size(segmented\_card1\_hue);**

**% thershold hue**

**for i=1:1:dimension(1,1) % fine min/max intensity for create thershold**

**for j=1:1:dimension(1,2)**

**if segmented\_card1\_hue(i,j) < buffer\_min**

**buffer\_min = segmented\_card1\_hue(i,j);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
if segmented_card1_hue(i,j) > buffer_max
    buffer_max = segmented_card1_hue(i,j);
end
end
end
end
% -----
Threshold = (buffer_min + buffer_max)/2; % create mid-range threshold value
for i=1:1:dimension(1,1) % threshold
    for j=1:1:dimension(1,2)
        if segmented_card1_hue(i,j) < 0.66
            segmented_card1_hue1(i,j) = -1;
        end
        if segmented_card1_hue(i,j) >= 0.66
            segmented_card1_hue1(i,j) = 1;
        end
    end
end
end
end

for i=1:1:dimension(1,1) % translate binary to bipolar
    for j=1:1:dimension(1,2)
        if segmented_card1_binary(i,j) == 0
            segmented_card1_binary(i,j) = -1;
        end
    end
end
end

figure(37);
imshow(logical(uint8(segmented_card1_hue1)));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end

otherwise
end
quantity_of_card = quantity_of_card + 1;
end
%-----

% BAM for hue

quantity_of_card = 0;

while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card
switch quantity_of_card
case 0
if buffer_status_card1 == 1 % status is attack
x = segmented_card0_hue1;
else % status is defend
x = fliplr(segmented_card0_hue1');
end
x = x'; % unknow pattern
x = reshape(x,109*113,1);
xold = x;
yold = ones(size(y1),1);
counter = 0;
terminate = 0;

while terminate ~= 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

nety = (w_hue_1')*xold;          % feedforward part
check_size_y = size(nety);
for i = 1:1:check_size_y(1,1)   % signum function
    if nety(i,1) > 0
        nety(i,1) = 1;
    end
    if nety(i,1) < 0
        nety(i,1) = -1;
    end
end
ynew = nety;

netx = w_hue_1*ynew;           %backward part
check_size_x = size(netx);
for i = 1:1:check_size_x(1,1)  % signum function
    if netx(i,1) > 0
        netx(i,1) = 1;
    end
    if netx(i,1) < 0
        netx(i,1) = -1;
    end
end
xnew = netx;

if xnew == xold
    terminate = 1;
else
    xold = xnew;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    yold = ynew;
end
counter = counter + 1;
end

output_hue1 = 0;
if ynew(1,1) == 1
    output_hue1 = 16;
end
if ynew(2,1) == 1
    output_hue1 = output_hue1 + 8;
end
if ynew(3,1) == 1
    output_hue1 = output_hue1 + 4;
end
if ynew(4,1) == 1
    output_hue1 = output_hue1 + 2;
end
if ynew(5,1) == 1
    output_hue1 = output_hue1 + 1;
end
% if ynew(6,1) == 1
%     output = output + 2;
% end
% if ynew(7,1) == 1
%     output = output + 1;
% end

```

#### case 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if buffer_status_card1 == 1 % status is attack
    x = segmented_card1_hue1;
else % status is defend
    x = fliplr(segmented_card1_hue1');
end

x = x'; % unknow pattern
x = reshape(x,109*113,1);
xold = x;
yold = ones(size(y1),1);
counter = 0;
terminate = 0;

while terminate ~= 1
    nety = (w_hue_2')*xold; % feedforward part
    check_size_y = size(nety);
    for i = 1:1:check_size_y(1,1) % signum function
        if nety(i,1) > 0
            nety(i,1) = 1;
        end
        if nety(i,1) < 0
            nety(i,1) = -1;
        end
    end
end

ynew = nety;

netx = w_hue_2*ynew; %backward part
check_size_x = size(netx);

for i = 1:1:check_size_x(1,1) % signum function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if netx(i,1) > 0
    netx(i,1) = 1;
end
if netx(i,1) < 0
    netx(i,1) = -1;
end
end
xnew = netx;

if xnew == xold
    terminate = 1;
else
    xold = xnew;
    yold = ynew;
end
counter = counter + 1;
end

output_hue2 = 0;
if ynew(1,1) == 1
    output_hue2 = 16;
end
if ynew(2,1) == 1
    output_hue2 = output_hue2 + 8;
end
if ynew(3,1) == 1
    output_hue2 = output_hue2 + 4;
end
if ynew(4,1) == 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        output_hue2 = output_hue2 + 2;
    end
    if ynew(5,1) == 1
        output_hue2 = output_hue2 + 1;
    end
    % if ynew(6,1) == 1
    %     output = output + 2;
    % end
    % if ynew(7,1) == 1
    %     output = output + 1;
    % end
otherwise
end
quantity_of_card = quantity_of_card + 1;
end

% mapping when output don't stay in group of output vector
quantity_of_card = 0;
calcel_binary_BAM1 = 0;
calcel_binary_BAM2 = 0;
output_binary1 = 0;
output_binary2 = 0;

while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card
    switch quantity_of_card
        case 0

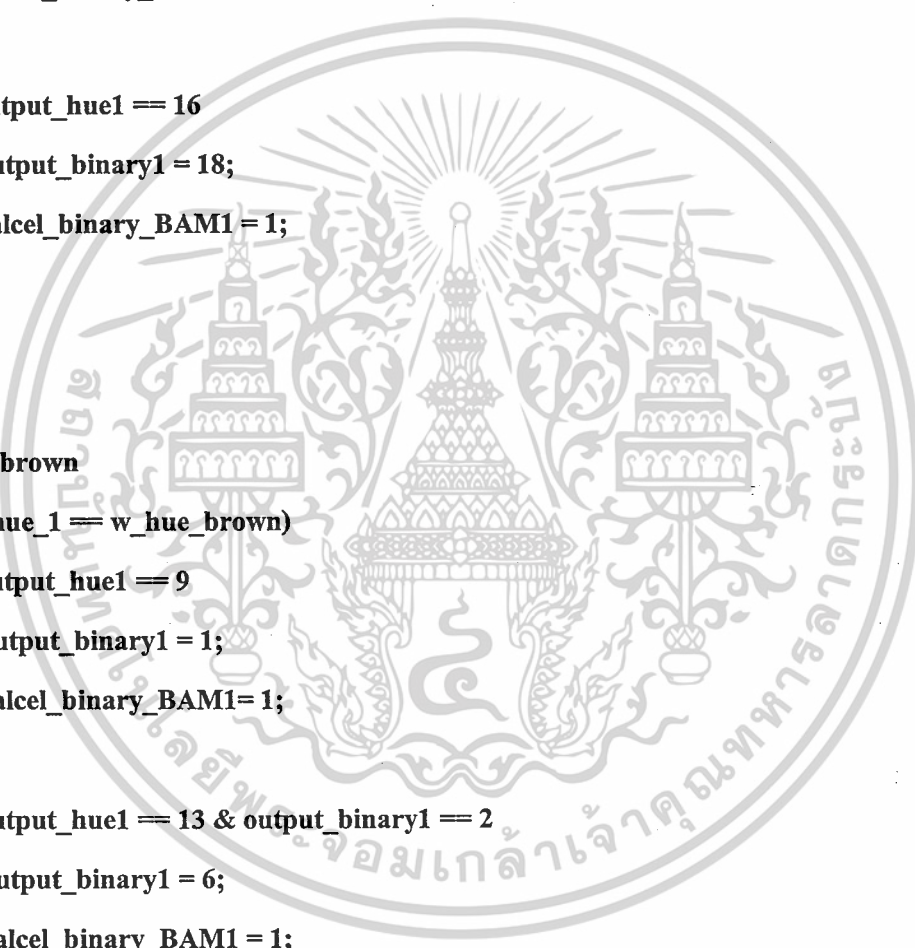
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% for purple
if(w_hue_1 == w_hue_purple)
    if output_hue1 == 10
        output_binary1 = 3;
        calcel_binary_BAM1 = 1;
    end
    if output_hue1 == 16
        output_binary1 = 18;
        calcel_binary_BAM1 = 1;
    end
end
% for brown
if(w_hue_1 == w_hue_brown)
    if output_hue1 == 9
        output_binary1 = 1;
        calcel_binary_BAM1 = 1;
    end
    if output_hue1 == 13 & output_binary1 == 2
        output_binary1 = 6;
        calcel_binary_BAM1 = 1;
    end
end
% for yellow
if(w_hue_1 == w_hue_yellow)
    if output_hue1 == 6
        output_binary1 = 2;
        calcel_binary_BAM1 = 1;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
if output_hue1 == 7
    output_binary1 = 5;
    calcel_binary_BAM1 = 1;
end
end

```

case 1

```

% for purple
if (w_hue_2 == w_hue_purple)
    if output_hue2 == 10
        output_binary2 = 3;
        calcel_binary_BAM1 = 1;
    end
    if output_hue2 == 16
        output_binary2 = 18;
        calcel_binary_BAM1 = 1;
    end
end

```

end

% for brown

```

if (w_hue_2 == w_hue_brown)
    if output_hue2 == 9
        output_binary2 = 1;
        calcel_binary_BAM1 = 1;
    end
    if output_hue2 == 13 & output_binary1 == 2

```

```

        output_binary2 = 6;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        calcel_binary_BAM1 = 1;
    end
end
% for yellow
if (w_hue_2 == w_hue_yellow)
    if output_hue2 == 6
        output_binary2 = 2;
        calcel_binary_BAM1 = 1;
    end
    if output_hue2 == 7
        output_binary2 = 5;
        calcel_binary_BAM1 = 1;
    end
end
end
otherwise
end
quantity_of_card = quantity_of_card + 1;
end

% END BAM
%
% BAM Dicision Tree BAM
% card 1
if((output_hue1 == 11 & (w_hue_1 ~= w_hue_blue)) | output_hue1 == 15 | output_hue1 ==
28 | output_hue1 == 25 |...
    (output_hue1 == 14 & (w_hue_1 ~= w_hue_blue)))& calcel_binary_BAM1 ~= 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if w_hue_1 == w_hue_purple
    w_binary = w_binary_purple; % w consist 3 10 15 26 28
end

if w_hue_1 == w_hue_yellow
    w_binary = w_binary_yellow; % w consist 2 5
end

if w_hue_1 == w_hue_brown
    w_binary = w_binary_brown; % w consist 2 5
end

% BAM for Binary

if buffer_status_card1 == 1 % status is attack
    x = segmented_card0_binary;
else % status is defend
    x = fliplr(segmented_card0_binary');
end

x = x'; % unknow pattern
x = reshape(x,109*113,1);
xold = x;
yold = ones(size(y1),1);
counter = 0;
terminate = 0;

while terminate ~= 1
    nety = (w_binary')*xold; % feedforword part
    check_size_y = size(nety);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for i = 1:1:check_size_y(1,1)      % signum function
    if nety(i,1) > 0
        nety(i,1) = 1;
    end
    if nety(i,1) < 0
        nety(i,1) = -1;
    end
end
ynew = nety;

netx = w_binary*ynew;             %backword part
check_size_x = size(netx);
for i = 1:1:check_size_x(1,1)    % signum function
    if netx(i,1) > 0
        netx(i,1) = 1;
    end
    if netx(i,1) < 0
        netx(i,1) = -1;
    end
end
xnew = netx;

if xnew == xold
    terminate = 1;
else
    xold = xnew;
    yold = ynew;
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        counter = counter + 1;
    end

    output_binary1 = 0;
    if ynew(1,1) == 1
        output_binary1 = 16;
    end
    if ynew(2,1) == 1
        output_binary1 = output_binary1 + 8;
    end
    if ynew(3,1) == 1
        output_binary1 = output_binary1 + 4;
    end
    if ynew(4,1) == 1
        output_binary1 = output_binary1 + 2;
    end
    if ynew(5,1) == 1
        output_binary1 = output_binary1 + 1;
    end
    % if ynew(6,1) == 1
    %     output = output + 2;
    % end
    % if ynew(7,1) == 1
    %     output = output + 1;
    % end

end

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% END BAM
```

```
% card 2
```

```
if((output_hue2 == 11 & (w_hue_2 ~= w_hue_blue))| output_hue2 == 15 | output_hue2 ==  
28 | output_hue2 == 25 | ...
```

```
output_hue2 == 14 & (w_hue_2 ~= w_hue_blue)) & cancel_binary_BAM1 ~= 1 &  
output_hue1 ~= 24
```

```
if w_hue_2 == w_hue_purple
```

```
w_binary = w_binary_purple; % w consist 3 10 15 26 28
```

```
end
```

```
if w_hue_2 == w_hue_yellow
```

```
w_binary = w_binary_yellow; % w consist 2 5
```

```
end
```

```
if w_hue_2 == w_hue_brown
```

```
w_binary = w_binary_brown; % w consist 2 5
```

```
end
```

```
% BAM for Binary
```

```
if buffer_status_card1 == 1 % status is attack
```

```
x = segmented_card1_binary;
```

```
else % status is defend
```

```
x = fliplr(segmented_card1_binary');
```

```
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x = x';          % unknow pattern
x = reshape(x,109*113,1);
xold = x;
yold = ones(size(y1),1);
counter = 0;
terminate = 0;

while terminate ~= 1
    nety = (w_binary)*xold;          % feedforward part
    check_size_y = size(nety);
    for i = 1:1:check_size_y(1,1)    % signum function
        if nety(i,1) > 0
            nety(i,1) = 1;
        end
        if nety(i,1) < 0
            nety(i,1) = -1;
        end
    end
    ynew = nety;

    netx = w_binary*ynew;          %backward part
    check_size_x = size(netx);
    for i = 1:1:check_size_x(1,1)    % signum function
        if netx(i,1) > 0
            netx(i,1) = 1;
        end
        if netx(i,1) < 0
            netx(i,1) = -1;
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
end
xnew = netx;

if xnew == xold
    terminate = 1;
else
    xold = xnew;
    yold = ynew;
end
counter = counter + 1;
end

if ynew(1,1) == 1
    output_binary2 = 16;
end
if ynew(2,1) == 1
    output_binary2 = output_binary2 + 8;
end
if ynew(3,1) == 1
    output_binary2 = output_binary2 + 4;
end
if ynew(4,1) == 1
    output_binary2 = output_binary2 + 2;
end
if ynew(5,1) == 1
    output_binary2 = output_binary2 + 1;
end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% if ynew(6,1) == 1
%   output = output + 2;
% end
% if ynew(7,1) == 1
%   output = output + 1;
% end
end
end

% END BAM

% mapping
quantity_of_card = 0;
while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card
switch quantity_of_card
case 0

%for purple
if (w_hue_1 == w_hue_purple)
if output_binary1 == 11
output_binary1 = 3;
calcel_binary_BAM1 = 1;
end
if output_binary1 == 13
output_binary1 = 21;
calcel_binary_BAM1 = 1;
end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
%for yellow
if (w_hue_1 == w_hue_yellow)
    if output_binary1 == 15
        output_binary1 = 23;
        calcel_binary_BAM1 = 1;
    end
end
end

```

case 1

```

%for purple
if (w_hue_1 == w_hue_purple)
    if output_binary2 == 11
        output_binary2 = 3;
        calcel_binary_BAM1 = 1;
    end
    if output_binary2 == 13
        output_binary2 = 21;
        calcel_binary_BAM1 = 1;
    end
end

```

end

```

%for yellow
if (w_hue_1 == w_hue_yellow)
    if output_binary2 == 15
        output_binary2 = 23;
        calcel_binary_BAM1 = 1;
    end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end

otherwise
end

quantity_of_card = quantity_of_card + 1;
end

```

```

output_hue1
output_hue2
output_binary1
output_binary2

```

```

% Send all output to text file

```

```

quantity_of_card = 0;

```

```

while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card

```

```

switch quantity_of_card

```

```

case 0

```

```

    fid = fopen('C:\Documents and Settings\ACER\Desktop\yugi\buffercard.txt', 'w');

```

```

    %fid = fopen('C:\windows\Desktop\buffercard.txt', 'w');

```

```

    fprintf(fid, '&card1=%d', output_hue1);

```

```

    fprintf(fid, '&card1_1=%d', output_binary1);

```

```

    fprintf(fid, '&status1=%d', buffer_status_card1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

status = fclose(fid);

case 1

%fid = fopen('C:\windows\Desktop\buffercard.txt', 'w');

fid = fopen('C:\Documents and Settings\ACER\Desktop\yugi\buffercard.txt', 'w');
fprintf(fid, '&card1=%d', output_hue1);
fprintf(fid, '&card1_1=%d', output_binary1);
fprintf(fid, '&status1=%d', buffer_status_card1);

fprintf(fid, '&card2=%d', output_hue2);
fprintf(fid, '&card2_1=%d', output_binary2);
fprintf(fid, '&status2=%d', buffer_status_card2);

status = fclose(fid);

otherwise
end

quantity_of_card = quantity_of_card + 1;
end

```

```
quantity_of_card = 0;
```

```

while quantity_of_card ~= quantity_of_card_LR % lock repeat statement not over
quantity_of_card
switch quantity_of_card
case 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if w_hue_1 == w_hue_blue
    fprintf('blue ');
end

if w_hue_1 == w_hue_purple
    fprintf('purple ');
end

if w_hue_1 == w_hue_brown
    fprintf('brown ');
end

if w_hue_1 == w_hue_yellow
    fprintf('yellow ');
end
output_hue1
output_binary1
case 1

if w_hue_2 == w_hue_blue
    fprintf('blue ');
end

if w_hue_2 == w_hue_purple
    fprintf('purple ');
end

if w_hue_2 == w_hue_brown
    fprintf('brown ');
end

if w_hue_2 == w_hue_yellow
    fprintf('yellow ');
end

output_hue2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
output_binary2
otherwise
end
quantity_of_card = quantity_of_card + 1;
end
end
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้