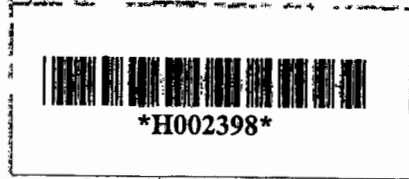


ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบแยกแยะเม็ดยาบ้า

Amphetamines Tablet Identification System



โดย



วัน เดือน ปี.....	23 ก.ค. 2558
เลขทะเบียน.....	02398
เลขเรียกหนังสือ.....	วทพ. พ46548 2548
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบแยกแยะเม็ดชาบ้ำ
นักศึกษา	นายพิเชษฐ์ อยู่ศรี
อาจารย์ที่ปรึกษา	ผศ.ดร. ธนารัตน์ ชลิตาพงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

ในปัจจุบันมีการแพร่ระบาดของชาบ้ำเป็นจำนวนมาก เมื่อมีการจับกุมผู้กระทำผิดได้ก็จะต้องมีการตรวจสอบเม็ดชาบ้ำของกลางว่าผลิตมาจากแหล่งผลิตใด ซึ่งระบบที่สร้างขึ้นนี้จะช่วยแยกแยะโดยดูจากลักษณะตัวอักษรที่ประทับอยู่บนเม็ดชา การสร้างระบบจะเริ่มจากการถ่ายรูปเม็ดชาที่มีการอัดผงแป้งลงในตัวอักษร เพื่อให้เห็นตัวอักษรชัดเจนขึ้น ทำการแปลงภาพให้เป็นรูปขาวดำ แล้วแปลงเป็นไฟล์ข้อความเพื่อป้อนเข้าโปรแกรมโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับเพื่อทำการฝึกสอนให้ระบบมีความสามารถในการรู้จำภาพตัวอย่าง ก่อนที่จะนำระบบไปใช้ช่วยในการตัดสินใจว่าเม็ดชานั้นอยู่ในกลุ่มรหัสใด

Title Amphetamines Tablet Identification System
Student Mr. Phichet Yoosri
Advisor Asst. Prof. Dr. Thanarat H. Chalidabhongse
Level of Study Master of Science in Information Technology
Major Information Science
Academic Year 2005

ABSTRACT

Nowadays, there is a high dispersion of drug. When the criminal was arrested the place of origin of drug will be examined. This system will use to identify the place of origin of drug base on attributes of alphabet that is stamped on the Amphetamines tablet. Technique that used in this system are filled white powder into the alphabet stamped on the Amphetamines tablet before taking the photography for more contrast, then convert its color image to be black and white image, then convert image's data to be text file. This text file will use for training the Pattern Recognition program based on the back propagation neuron networks. After training, the system will identify the group of drugs.

กิตติกรรมประกาศ

ในความสำเร็จของโครงการนี้ ผู้เขียนใคร่ขอแสดงความระลึกถึงบุคคลสำคัญผู้อยู่เบื้องหลังดังต่อไปนี้

คุณแม่และพี่สาวสำหรับกำลังใจที่เต็มเปี่ยมและทุกสิ่งทุกอย่างจนสามารถมีวันนี้ได้ อาจารย์ธรรมาภรณ์ ชลิตาพงศ์ อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษา ให้กำลังใจ และคำแนะนำต่าง ๆ ที่เป็นประโยชน์ยิ่งจนทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

คุณทิพวรรณ ปิยะเลิศกมล สำหรับกำลังใจและความช่วยเหลือที่มีให้เสมอ และเพื่อน ๆ IS17.2 ที่คอยให้ความช่วยเหลือ ข้อมูลต่างๆ ที่มีประโยชน์ตลอดมา

คุณชาติชาย สุทธิกมล ท่านรองเลขาธิการ ป.ป.ส. ที่ให้ความกรุณาในการตั้งการต่างๆเพื่อช่วยอำนวยความสะดวกในการขอข้อมูลต่างๆภายในหน่วยงาน

คุณวิชญา การัญญา สำหรับรูปภาพเม็ดยาและเอกสารต่างๆ เกี่ยวกับเรื่องยาบ้าและคอยประสานงานภายในสำนักงาน ป.ป.ส. ด้วยดีตลอดมา

คุณวิภาดา ปานอ่อน และคุณนุช สำหรับรูปภาพเม็ดยาเพิ่มเติม

บริษัทซิสโก้ ซิสเต็มส์ (ประเทศไทย) จำกัด ที่เอื้อเฟื้อสถานที่ในการทำงาน

พิเชษฐ์ อยู่ศรี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 แผนการดำเนินการ.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. ทฤษฎีและหลักการที่ใช้ในการพัฒนาระบบงาน.....	4
2.1 บทนำ.....	4
2.2 การศึกษาลักษณะเม็ดยา.....	5
2.3 การประมวลผลภาพ.....	7
2.4 นีวรอลเน็ตเวิร์ค.....	11
3. การวิเคราะห์ระบบปัจจุบัน.....	31
3.1 ความเป็นมาขององค์กร.....	31
3.2 การวิเคราะห์ระบบงานปัจจุบัน.....	33
3.3 ปัญหาที่พบจากการดำเนินงานในปัจจุบัน.....	39

สารบัญ (ต่อ)

บทที่	หน้า
3.4 แนวทางในการแก้ไขปัญหา.....	39
4. การวิเคราะห์และออกแบบระบบงานใหม่.....	40
4.1 ความต้องการของระบบงานใหม่.....	40
4.2 แบบจำลองเชิงแนวคิดของระบบงานใหม่.....	40
4.3 ขอบเขตของระบบงานใหม่.....	41
4.4 แบบจำลองเชิงกายภาพของระบบงานใหม่.....	42
4.4.1 ขั้นตอนการเตรียมรูปเม็ดยา.....	42
4.4.2 ขั้นตอนการประมวลผลภาพ.....	43
4.4.3 ขั้นตอนการฝึกสอนระบบรู้จำ.....	44
4.4.4 ขั้นตอนการทำงานของโปรแกรมแยกแยะยาบ้า.....	46
5. การสร้างและทดสอบระบบ.....	48
5.1 การถ่ายภาพเม็ดยา.....	48
5.2 โปรแกรมเตรียมข้อมูลภาพ.....	49
5.3 โปรแกรมฝึกสอนและทดสอบการรู้จำ.....	51
5.4 โปรแกรมแยกแยะเม็ดยาบ้า.....	54
5.5 ผลการทดสอบระบบการรู้จำ.....	55
6. บทสรุป.....	61
6.1 สรุปผลการพัฒนาระบบ.....	61
6.2 ข้อจำกัดของระบบที่พัฒนาขึ้น.....	61
6.3 ข้อเสนอแนะในการพัฒนาต่อ.....	62
บรรณานุกรม.....	63
ภาคผนวก.....	64
ประวัติผู้เขียน.....	98

สารบัญตาราง

หน้า

ตารางที่

5.1	จำนวนเม็ดยาที่ใช้ในการฝึกสอนและทดสอบ.....	56
5.2	ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ใช้ในการฝึกสอน แสดงด้วยค่าเฉลี่ย.....	58
5.3	ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ไม่ได้ผ่านการฝึกสอน แสดงด้วยค่าเฉลี่ย.....	58
5.4	ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ไม่ได้ผ่านการฝึกสอน.....	59



สารบัญรูป

	หน้า
รูปที่	
2.1 หัวดอกเม็ดยา.....	5
2.2 ตัวอย่างเม็ดยาบ้า.....	5
2.3 ความแตกต่างของสัญลักษณ์บนเม็ดยา.....	6
2.4 รูปแบบสี RGB.....	7
2.5 ฟังก์ชันการแปลงในแบบต่างๆที่ใช้สำหรับปรับปรุงความเปรียบต่างของภาพ.....	9
2.6 ภาพและฮิสโตแกรมหลังการปรับความเปรียบต่าง.....	10
2.7 แสดงโครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ.....	12
2.8 แสดง โคอะแกรมของนิวรอนที่สร้างขึ้น (Artificial Neuron).....	13
2.9 แสดง โมดูลนิวรอนที่สร้างขึ้นร่วมกับ Activation Function.....	15
2.10 แสดงกราฟที่ได้จากการชกมอดลลอจิสติกฟังก์ชัน.....	16
2.11 แสดง Hyperbolic Tangent Function.....	17
2.12 แสดงลักษณะ โครงข่ายประสาทเทียมแบบชั้นเดียว.....	18
2.13 โคอะแกรมของ Back Propagation Neural Networks แบบสองชั้น.....	19
2.14 โคอะแกรมของ Two Layer Feed Forward.....	23
2.15 แสดง โคอะแกรมของการปรับค่า Weight ในชั้น Output layer และ Hidden layer...	26
2.16 โคอะแกรมแสดงวิธีการหาค่าน้ำหนักที่เหมาะสมของการเทรนนิ่ง โดยใช้ อัลกอริทึมแบบแบคพรอบพาเกชัน.....	28
2.17 แสดง Flow Chart แสดงกระบวนการเทรนนิ่ง เพื่อหาค่าน้ำหนักที่เหมาะสมใน กระบวนการทดสอบ.....	29
3.1 สถานที่ตั้งของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด.....	32
3.2 แสดงขั้นตอนการดำเนินงานต่อยาเสพติดให้โทษของกลาง.....	33
4.1 แสดงแบบจำลองเชิงแนวคิดของระบบงานใหม่.....	40
4.2 ขั้นตอนการเตรียมรูปเม็ดยา.....	42

สารบัญรูป (ต่อ)

	หน้า
รูปที่	
4.3 ขั้นตอนการประมวลผลภาพ.....	43
4.4 ขั้นตอนการฝึกสอนระบบรู้จำ.....	45
4.5 ขั้นตอนการทำงานของโปรแกรมแยกแยะชาบี๋.....	47
5.1 แสดงความแตกต่างระหว่างภาพที่ถ่ายแบบปกติกับแบบอัดผงบั้ง.....	49
5.2 ภาพเมล็ดชาหลังจากแปลงเป็นภาพขาวดำ.....	49
5.3 โปรแกรมเตรียมข้อมูลภาพ.....	50
5.4 แสดงโปรแกรมฝึกสอน.....	52
5.5 แสดงการสร้างไฟล์ Target.....	53
5.6 แสดงโปรแกรมทดสอบการรู้จำ.....	54
5.7 โปรแกรมแยกแยะเมล็ดชาบี๋.....	55
5.8 แสดงผลการทดสอบการแยกแยะกลุ่มเมล็ดชาที่ได้จากโปรแกรม Neural Network....	57
5.9 แสดงค่า Sum Square Error ที่ได้จากการฝึกสอน 500 รอบ.....	60

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันการแพร่ระบาดของยาเสพติดในประเทศไทย ได้เปลี่ยนรูปแบบไปจากเดิมมาก ทั้งในเรื่องของชนิดยาเสพติดและวิธีการในการเสพ ตลอดจนถึงรูปแบบในการค้า และการแพร่ระบาด หนึ่งในยาเสพติดที่มีการแพร่ระบาดมากในปัจจุบันคือ ยาบ้า ซึ่งมีการพัฒนารูปแบบของตัวยาและวิธีการในการค้า และการเสพซึ่งแตกต่างกันไปในแต่ละพื้นที่ที่มีการแพร่ระบาด อันเป็นปัจจัยหลักที่ทำให้ปัญหาของยาบ้าเป็นความสำคัญในอันดับแรกๆ คือความง่ายของการเสพและผลกระทบที่เกิดขึ้นจากการเสพในรูปแบบต่างๆ ทั้งนี้ สิ่งที่เกิดขึ้นอาจเป็นผลโดยตรงมาจากองค์ประกอบของสารเคมีที่มีอยู่ในยาบ้า ที่มีการแพร่ระบาดในที่ต่าง ๆ นั้นเอง

จากการแพร่ระบาดของยาบ้าที่มีอยู่ในที่ต่าง ๆ นั้นพบว่ามีความแตกต่างกันในส่วนที่เห็นได้ชัดเจนคือ รูปแบบตัวอักษรหรือเครื่องหมายที่ปรากฏอยู่บนเม็ดยา ซึ่งมีลักษณะคล้ายกับเป็นเครื่องหมายทางการค้าที่ปรากฏอยู่บนเม็ดยาทั่วไป ซึ่งเมื่อทำการศึกษาถึงลักษณะของเครื่องหมายที่ปรากฏอยู่บนเม็ดยานั้น อาจจะไปสู่การอธิบายถึงวิธีในการลักลอบผลิต เครื่องมือที่ใช้ในการผลิต ตลอดจนอาจจะโยงไปถึงแหล่งลักลอบผลิตของยาบ้าที่ต่าง ๆ และจากการสังเกตจากตัวอย่างของเม็ดยาที่นำมาตรวจพิสูจน์นั้น พบในขั้นต้นว่าตัวอักษรที่ปรากฏอยู่บนเม็ดยา เช่นอักษร WY มีความแตกต่างอย่างชัดเจนเพียงพอที่จะเป็นองค์ประกอบสำคัญประการหนึ่งในหลายประการ ที่จะนำไปสู่การวิเคราะห์ถึงที่มาของแหล่งลักลอบผลิตซึ่งมีอยู่ในพื้นที่ต่าง ๆ ได้ และเพื่อให้การศึกษาข้อมูลในเรื่องดังกล่าว มีความชัดเจนมากขึ้น จึงได้มุ่งที่จะศึกษาถึงข้อมูลที่มีความแตกต่างในเรื่องของตัวอักษรหรือเครื่องหมายดังกล่าว เพื่อให้เกิดเป็นมาตรฐาน อันจะนำไปสู่ความเข้าใจร่วมกันถึงสิ่งที่เกิดขึ้นในพื้นที่ต่าง ๆ ได้ดีต่อไป แต่ทั้งนี้ความชัดเจนในการวิเคราะห์ข้อมูลจากตัวอย่างเม็ดยาที่ได้จากการตรวจพิสูจน์นั้น จำเป็นต้องมีการวิเคราะห์ข้อมูลที่เกี่ยวข้องทั้งหมด คือคุณลักษณะทางด้านกายภาพและคุณลักษณะทางเคมีของยาบ้าร่วมกันจึงจะสามารถระบุได้อย่างชัดเจนถึงแหล่งที่มาของยาบ้าที่ได้มานั้น และจากการศึกษาลักษณะของตัวอักษร WY บนเม็ดยาและภาชนะบรรจุยาบ้า พบว่าของกลางยาบ้าที่มาจากแหล่งลักลอบผลิตบางแห่งจะมีลักษณะตัวอักษรบนเม็ดยาและภาชนะบรรจุยาบ้าเป็นเอกลักษณ์เฉพาะแห่ง ที่สามารถชี้แจงกลุ่มยาที่แพร่ระบาดในประเทศ

ไทย ซึ่งเป็นข้อมูลให้เจ้าหน้าที่ที่ปฏิบัติงานเกี่ยวกับการปราบปรามยาเสพติดศึกษาสถานการณ์และกำหนดมาตรการในการปราบปรามยาเสพติดต่อไป (วิภาดา ปานอ่อน, 2547:1)

1.2 วัตถุประสงค์ของโครงการ

การศึกษาและพัฒนาโปรแกรมในครั้งนี้เพื่อนำไปใช้ประโยชน์ในการแยกแยะเม็ดยาบ้าของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด โดยมีการรวบรวมรูปภาพเม็ดยาบ้าแบบต่าง ๆ เพื่อศึกษาลักษณะการประทับตัวอักษรบนเม็ดยา ซึ่งจะเห็นได้ว่ามีเม็ดยาลักษณะต่าง ๆ มากมายกว่าร้อยแบบ ในปัจจุบันการแยกกลุ่มเม็ดยาเหล่านี้จะใช้เจ้าหน้าที่แยกแยะด้วยตาเปล่า ซึ่งเม็ดยาเหล่านี้มีขนาดเล็กและจำนวนมากที่ต้องทำการแยกแยะในแต่ละวัน ซึ่งเป็นปัญหาในการทำงานในปัจจุบัน ดังนั้นจึงได้ทำการพัฒนาระบบเพื่อช่วยให้มีประสิทธิภาพในการทำงานมากยิ่งขึ้น ซึ่งมีวัตถุประสงค์ดังต่อไปนี้

1. รวบรวมและศึกษาข้อมูลรูปภาพเม็ดยาบ้าที่มีอยู่ในปัจจุบันเพื่อดูความเป็นไปได้ในการแยกแยะเม็ดยาด้วยลักษณะทางกายภาพ
2. ศึกษากระบวนการประมวลผลภาพที่เหมาะสมเพื่อนำมาใช้ในการเตรียมรูปภาพให้เหมาะสมสำหรับการแยกแยะเม็ดยา
3. พัฒนาโปรแกรมที่ช่วยตัดสินใจในการแยกแยะรูปภาพเม็ดยา โดยดูจากลักษณะตัวอักษรที่ประทับบนเม็ดยา โดยใช้วิธีการของนิเวรอล เน็คเวิร์ค
4. นำระบบที่พัฒนาได้ไปใช้งานแทนการทำงานแบบเดิม เพื่อช่วยให้ในการทำงานมีประสิทธิภาพมากขึ้น

1.3 ขอบเขตของโครงการ

ระบบที่ทำการพัฒนาขึ้นนี้ จะเป็นการพัฒนาระบบขึ้นมาใหม่เพื่อทดแทนการทำงานที่มีอยู่เดิม โดยทำการศึกษาขั้นตอนการแยกแยะเม็ดยาบ้าของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด โดยมีขอบเขตการศึกษาดังต่อไปนี้

1. สามารถแยกแยะเม็ดยาจากลักษณะตัวอักษรบนเม็ดยา ไม่รวมถึงการแยกแยะสีของเม็ดยา
2. สามารถแยกแยะเม็ดยาที่ถ่ายภาพตัวอักษรในแนวระนาบเท่านั้น
3. สามารถเพิ่มและเปลี่ยนแปลง แก้วข้อมูลเม็ดยาได้
4. สามารถแสดงภาพผลการแยกแยะเม็ดยาเปรียบเทียบกับภาพในฐานข้อมูล โดยเรียงตามลำดับความเหมือน 3 อันดับแรกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 แผนการดำเนินการ

ขั้นตอนในการการพัฒนาระบบจะประกอบด้วยขั้นตอนดังต่อไปนี้

1. ศึกษาและรวบรวมข้อมูลรูปภาพเม็ดยา
2. ศึกษาวิธีการจัดกลุ่มเม็ดยา
3. ศึกษาวิธีที่เหมาะสมในการจัดเก็บรูปภาพเม็ดยา
4. ศึกษารายละเอียดของพัฒนาระบบแยกแยะเม็ดยาด้วยวิธีการของนิเวศน์เดวิร์ค
5. พัฒนาระบบแยกแยะเม็ดยาจากตัวอักษรที่ประทับบนเม็ดยา
6. ทดสอบการใช้งานและปรับปรุงแก้ไขระบบที่พัฒนาแล้ว
7. สรุปผลการทดสอบจากการใช้งานที่เกิดขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์ต่อผู้ทำการพัฒนาระบบ
 - เป็นการนำความรู้ที่ได้จากการศึกษามาประยุกต์ใช้ในการวิเคราะห์ ออกแบบ และพัฒนาระบบ เพื่อใช้งานจริง
 - เป็นการศึกษาเรียนรู้และประยุกต์ใช้เทคโนโลยีที่มีอยู่ในปัจจุบันมาใช้ในการสร้างระบบให้มีประสิทธิภาพ
 - รู้จักวิธีการวางแผนการพัฒนา การแก้ปัญหาที่เกิดขึ้นในการพัฒนาระบบ
2. ประโยชน์ต่อองค์กรในการนำระบบไปใช้งาน
 - การทำงานรวดเร็วและมีประสิทธิภาพมากขึ้น
 - เป็นการตัดสินใจแยกแยะด้วยคอมพิวเตอร์ ไม่ขึ้นกับผู้ใดผู้หนึ่ง
 - การแยกแยะมีมาตรฐานสูงขึ้น
 - แบ่งเบาระงานของเจ้าหน้าที่ตรวจสอบพิสูจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่ใช้ในการพัฒนาระบบแยกแยะยาบ้า

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการต่างๆ ที่ใช้ในการพัฒนาระบบงาน ซึ่งการพัฒนานั้นจะอยู่บนพื้นฐานของการประมวลผลภาพและหลักการนิเวศน์เคเวิร์ค โดยทฤษฎีต่างๆ ที่เกี่ยวข้องในการพัฒนาระบบงานมีดังต่อไปนี้

2.1 บทนำ

การผลิตยาบ้าในภูมิภาคนี้ โดยทั่วไปมี 3 ขั้นตอนหลัก ได้แก่ ขั้นตอนการผลิตหัวเชื้อเมทแอมเฟตามีน ขั้นตอนการผสม และขั้นตอนการตอกเม็ดคยา ซึ่งรวมถึงการบรรจุเม็ดคยาในภาชนะต่างๆ สำหรับขั้นตอนการตอกเม็ดคยานั้นจำเป็นต้องใช้เครื่องมือเฉพาะคือเครื่องตอกเม็ดคยา ซึ่งมีหลายประเภท ทั้งเครื่องมือที่ทำงานง่ายๆ สามารถตอกด้วยมือหรือเครื่องที่มีประสิทธิภาพสูง ผลิตเม็ดคยามีคุณภาพดีและกำลังผลิตสูง โดยเครื่องตอกเม็ดคยาต้องประกอบด้วยหัวตอก ซึ่งเป็นส่วนสำคัญของเครื่อง และส่วนใหญ่บนหัวตอกจะมีตัวอักษรหรือเครื่องหมายต่างๆ ประทับอยู่ หัวตอกนี้สามารถถอดเปลี่ยนได้เมื่อหมดอายุการใช้งาน ซึ่งขึ้นอยู่กับปริมาณและระยะเวลาการผลิต ซึ่งจะทำให้เกิดการสึกหรอของหัวตอก ดังนั้นตัวอักษรบนเม็ดคยานอกจากจะบอกถึงแหล่งผลิตแล้ว ลักษณะของเม็ดคยาก็สามารถบ่งบอกสภาพของชุดหัวตอกว่าใหม่หรือเก่า เช่น เม็ดคยาที่ผลิตจากหัวตอกชุดใหม่จะมีลักษณะผิวเรียบ ตัวอักษรและร่องของตัวอักษรบนเม็ดคยาชัดเจน ร่องของตัวอักษรไม่มีรอยแหวน ส่วนเม็ดคยาที่ผลิตจากหัวตอกชุดเก่า จะมีลักษณะผิวไม่เรียบ ตัวอักษรไม่ชัดเจน ร่องของตัวอักษรแหวน แต่ในขณะที่ผิวลักษณะของเม็ดคยายังขึ้นกับปัจจัยอื่นๆ ได้แก่ อัตราส่วนผสมขนาดของผงและปริมาณความชื้น เป็นต้น สำหรับตัวอักษรบนเม็ดคยาที่พบในภูมิภาคนี้ในปัจจุบันมีหลายรูปแบบได้แก่ WY, R, SY, 555, 888 เป็นต้น และแต่ละรูปแบบตัวอักษรก็อาจมีลักษณะเหมือนกันหรือแตกต่างกันแล้วแต่รูปแบบหัวตอกของผู้ผลิต ในปัจจุบันหน่วยงานที่มีหน้าที่ตรวจสอบจะต้องทำการเปรียบเทียบเม็ดคยาที่จับกุม ได้กับข้อมูลที่มีอยู่ด้วยสายตา ซึ่งอาจเกิดความผิดพลาดและเป็นอันตรายต่อสุขภาพได้ จึงมีแนวคิดที่จะศึกษาการนำวิธีการประมวลผลภาพและรู้จำรูปแบบมาใช้ในการแยกแยะเม็ดคยาบ้ามาใช้เพื่อเพิ่มประสิทธิภาพ ความถูกต้อง ความปลอดภัยและความรวดเร็วในการทำงาน

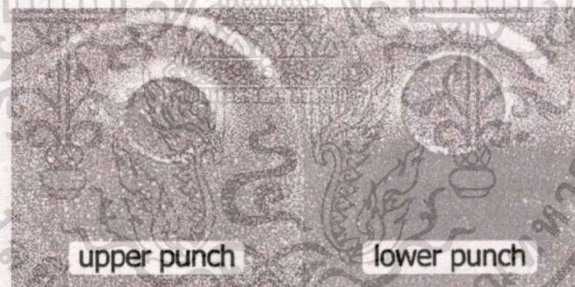
2.2 การศึกษาลักษณะเม็ดยา

จากการแพร่ระบาดของยาบ้าในพื้นที่ต่าง ๆ นั้นพบว่ามีความแตกต่างกันในส่วนที่เห็นได้ชัดเจนคือรูปแบบตัวอักษรหรือเครื่องหมายที่ปรากฏอยู่บนเม็ดยา ซึ่งมีลักษณะคล้ายกับเป็นเครื่องหมายทางการค้าที่ปรากฏอยู่บนเม็ดยาทั่วไป ตัวอย่างของตัวอักษรที่พบบนเม็ดยามีดังนี้ WY, R, SY, 555, 888 จากการสังเกตตัวอย่างของเม็ดยาพบว่าอักษร WY มีมากที่สุด ซึ่งผลิตจากแหล่งผลิตต่างกัน ทำให้ลักษณะของตัวอักษรแตกต่างกันแล้วแต่หัวตอกของผู้ผลิตแต่ละราย ส่วนตัวอักษรอื่นๆที่พบมีจำนวนน้อยและสามารถแยกแยะด้วยตาได้ง่าย ดังนั้นในบทความนี้จะเน้นการศึกษาความแตกต่างของตัวอักษร WY บนเม็ดยา

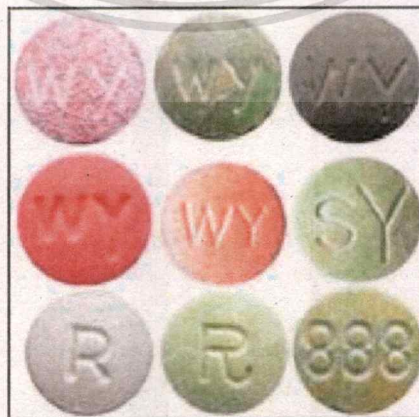
ลักษณะของแท่นปั๊มเม็ดยาของแหล่งผลิตจะประกอบด้วยหัวตอก 2 ส่วน ดังแสดงในรูปที่ 2.1 คือ

ส่วนที่ 1 หัวตอกส่วนบน (Upper punch) ส่วนมากจะมีตัวอักษรหรือเครื่องหมายเป็นตัวนูนตามที่ต้องการเช่น WY, R, SY, 555, 888 และอื่นๆ เป็นต้น

ส่วนที่ 2 หัวตอกส่วนล่าง (Lower punch) ส่วนมากจะเรียบหรือมีเส้นนูนแบ่งครึ่งหัวตอก



รูปที่ 2.1 หัวตอกเม็ดยา



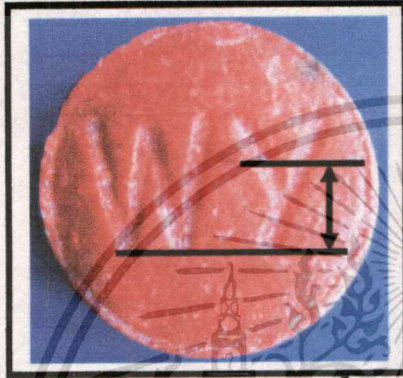
รูปที่ 2.2 ตัวอย่างเม็ดยาบ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

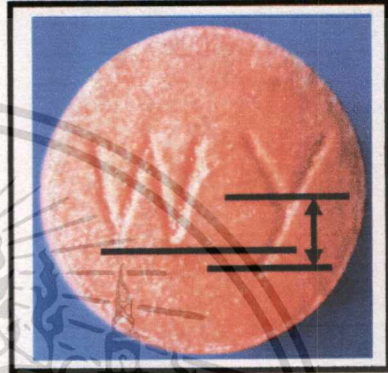
หลักเกณฑ์ในการพิจารณาความแตกต่างของสัญลักษณ์บนเม็ดยา

ในการพิจารณาเม็ดยาที่ด้านหนึ่งมีสัญลักษณ์เป็นตัวอักษร WY พิมพ์เป็นร่องลึก อีกด้านสัญลักษณ์นั้น ให้พิจารณาที่ตัวอักษร Y ก่อน จะมีความแตกต่างและขนาดของหางแบ่งได้เป็น 5 แบบดังนี้

แบบที่ 1 ตัววาย หางเฉียงสั้น



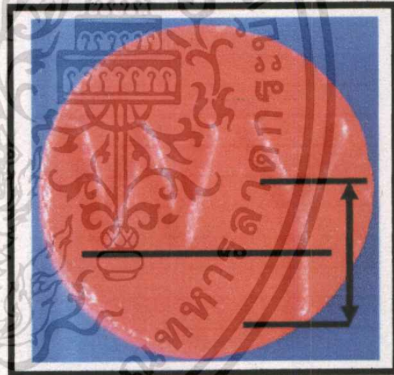
แบบที่ 2 ตัววาย หางเฉียงยาว



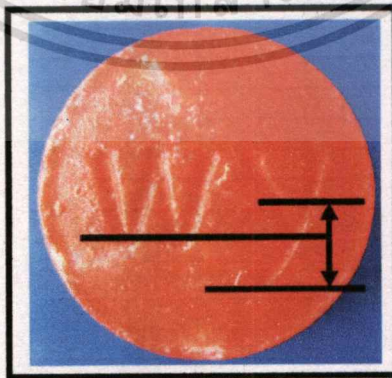
แบบที่ 3 ตัววาย หางตรงสั้น



แบบที่ 4 ตัววาย หางตรงยาว



แบบที่ 5 ตัววาย หางโค้งยาว



รูปที่ 2.3 ความแตกต่างของสัญลักษณ์บนเม็ดยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การประมวลผลภาพ (Image Processing)

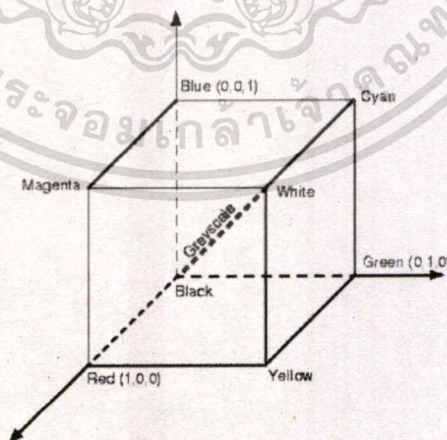
หลักการพื้นฐานที่นำมาใช้ในการประมวลผลภาพได้แก่ การแยกสี (Color) การปรับความเปรียบต่างของภาพ (Contrast) และการแปลงภาพระดับเทา (Grayscale)

รูปภาพสีและรูปแบบสี

รูปภาพดิจิทัลมีหลากหลายชนิด เช่นรูปภาพขาวดำ (Binary image), รูปภาพระดับเทา (Gray level image), รูปภาพสี (Color image) โดยรูปภาพแต่ละชนิดนี้จะได้มาด้วยวิธีการต่าง ๆ กัน เช่น จากกล้องดิจิทัล, สแกนเนอร์ หรือจากการประมวลผล

ระบบสีที่ใช้ในการประมวลผลภาพมีหลายรูปแบบ เช่นรูปแบบสี RGB (RGB color model), รูปแบบสี HSI (HSI color model), รูปแบบสี CMY (CMY color model) เป็นต้น ซึ่งแต่ละรูปแบบสีจะเหมาะกับลักษณะงานที่แตกต่างกัน เช่น รูปแบบสี RGB จะใช้สำหรับกล้องดิจิทัล หรือแสดงผลบนจอมอนิเตอร์, รูปแบบสี CMY จะใช้กับเครื่องพิมพ์ (Printer) ในเอกสารฉบับนี้จะกล่าวถึงรูปแบบสี RGB เนื่องจากภาพที่นำมาประมวลผลเป็นภาพที่ได้จากกล้องดิจิทัล

รูปแบบสี RGB เป็นรูปแบบที่เกิดจากการผสมกันของแม่สี 3 สีคือ สีแดง (Red), สีเขียว (Green) และสีน้ำเงิน (Blue) รวมกับค่าความเข้มแสง ซึ่งเขียนรูปแบบสีได้ดังรูป โดยแต่ละแกนจะแทนแม่สีแต่ละสีและมีค่าตั้งแต่ 0 ถึง 1 (หมายถึงค่าความเข้มแสงของสี) ซึ่งทำให้รูปแบบสีนี้สามารถสร้างสีได้ทั้งหมดจำนวน 16 ล้านสี นอกจากนี้จะยังสังเกตเห็นว่าเมื่อค่าความเข้มแสงของแม่สีทั้ง 3 สีเท่ากันแล้ว รูปแบบสีนี้จะสร้างรูปภาพเหมือนกับรูปภาพระดับเทา



รูปที่ 2.4 รูปแบบสี RGB

การเปลี่ยนรูปภาพสี RGB เป็นรูปภาพระดับเทา กำหนดให้ I คือรูปภาพระดับเทา R, G, B คือ สีแดง, สีเขียว และสีน้ำเงินของรูปภาพสี RGB ตามลำดับ ดังนั้นสามารถหารูปภาพระดับเทาจากรูปภาพสี RGB ด้วยสมการที่ 1

$$I = \frac{R+G+B}{3} \dots(1)$$

ฮิสโตแกรม (Histogram)

ฮิสโตแกรมของภาพดิจิทัล คือ กราฟที่แสดงความถี่หรือความหนาแน่นของค่าระดับสีเทา (Gray level) ที่อยู่ในช่วง $[0, L-1]$ ของภาพนั้นๆ เมื่อ L คือค่าระดับสีเทาสูงสุดโดยที่กราฟดังกล่าวจะมีลักษณะเป็นกราฟแท่ง ที่มีแกน x แทนค่าระดับสีเทา ในช่วง $[0, L-1]$ ส่วนแกน y แทนค่าความถี่หรือความหนาแน่นของค่าระดับสีเทา

การกระจายตัวของกราฟฮิสโตแกรมจะทำให้เราทราบได้ว่าเป็นภาพที่มีลักษณะเป็นภาพที่มีมืดหรือเป็นภาพที่สว่าง โดยพิจารณาได้ดังนี้ ถ้าเป็นภาพที่มีลักษณะมืด ฮิสโตแกรมของภาพนั้นจะมีการกระจายตัวของแท่งกราฟอยู่ทางด้านขวาของกราฟเป็นส่วนใหญ่ แต่ถ้าเป็นภาพที่มีลักษณะสว่าง ฮิสโตแกรมก็จะมีลักษณะเบนไปทางด้านซ้ายของกราฟ

ฮิสโตแกรมของภาพนับเป็นเครื่องมือที่สำคัญในกระบวนการปรับปรุงภาพอย่างหนึ่ง ซึ่งในการปรับปรุงภาพที่จำเป็นต้องอาศัยการพิจารณาฮิสโตแกรมก็มีอยู่หลายวิธี เช่นการปรับความสว่างของภาพ (Brightness) การทำภาพให้เปรียบเทียบต่าง (Contrast) การแยกแยะภาพ (Image segmentation) ซึ่งวิธีหนึ่งที่จะนำมาใช้ในระบอบนี้คือ การปรับความเปรียบเทียบต่าง เพื่อปรับปรุงภาพให้ดีขึ้นก่อนจะนำไปประมวลผลต่อไป

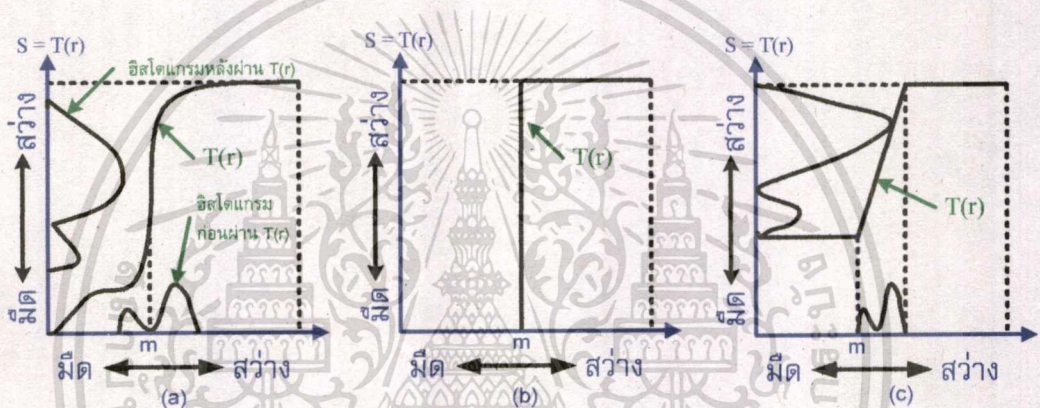
การปรับความเปรียบเทียบต่างของภาพ

ในภาพบางภาพที่เราจะนำมาประมวลผล บางครั้งอาจจะมี ความเปรียบเทียบต่าง (Contrast) ที่ต่ำคือภาพที่มีระดับสีเทาของจุดภาพอยู่ใกล้เคียงกันตลอดทั้งภาพ ถ้ามองจากฮิสโตแกรม จะพบว่าการกระจายตัวของค่าระดับสีเทาของจุดภาพจะรวมกันเป็นช่วงแคบๆ ทำให้ไม่สามารถมองเห็นรายละเอียดเล็กๆของภาพได้ ดังนั้นในการปรับปรุงความเปรียบเทียบต่างของภาพ คือการทำให้ฮิสโตแกรมมีการกระจายตัวของค่าระดับสีเทาใหม่ ทำให้มองเป็นรายละเอียดของภาพได้มากขึ้น โดยจะมีฟังก์ชันที่ใช้สำหรับทำการแปลง (Transformation function) หรือบางครั้งเรียกว่าการ mapping ดังสมการที่ 2

$$S = T(r) \dots (2)$$

เมื่อ	$T(r)$	คือ Transformation function
	r	คือ ค่าระดับสีเทาของภาพก่อนการแปลง
	S	คือ ค่าระดับสีเทาของภาพหลังการแปลง

จากฟังก์ชันการแปลงในสมการที่ 2 จะมีอยู่หลายรูปแบบเช่น ฟังก์ชันแบบเชิงเส้นตรง ฟังก์ชันที่ไม่เป็นเชิงเส้น ดังแสดงในรูปที่ 2.5

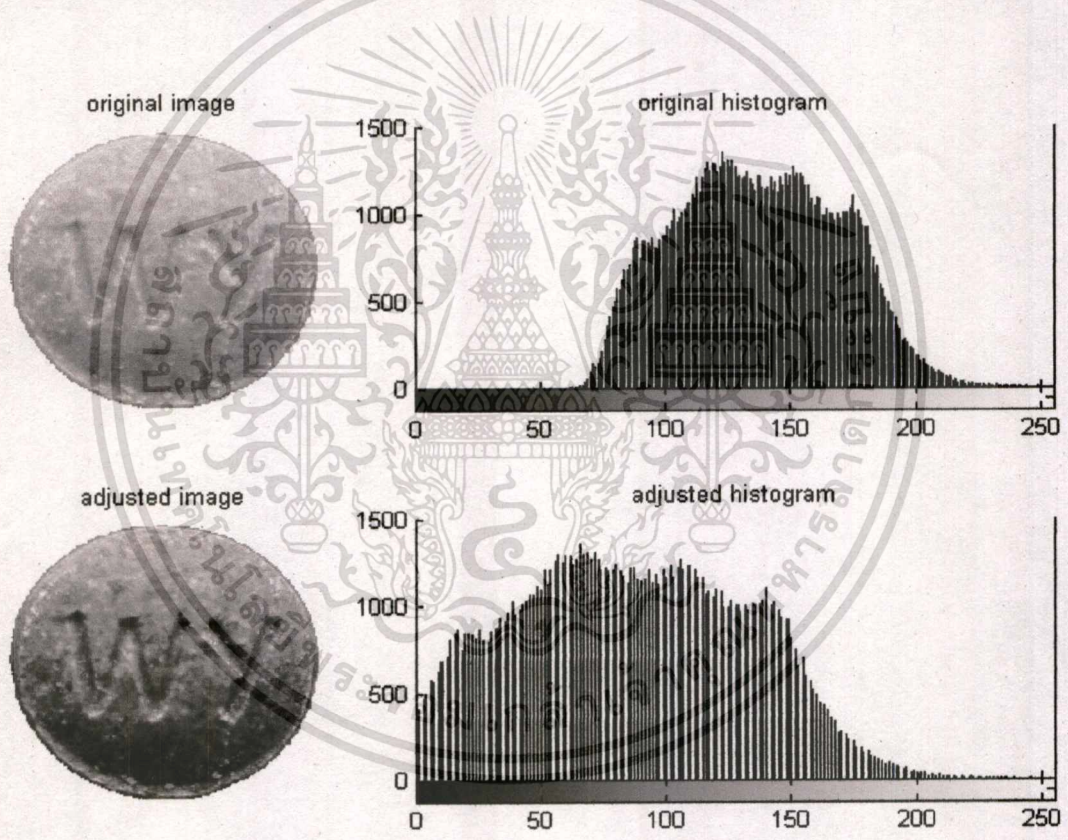


รูปที่ 2.5 ฟังก์ชันการแปลงในแบบต่างๆที่ใช้สำหรับปรับปรุงความเปรียบต่างของภาพ

จากรูปที่ 2.5 (a) เมื่อค่า $r < m$ (ค่าเทรสโฮลด์) ซึ่งในขณะที่นำค่า r มาผ่าน $T(r)$ จะทำให้ได้ค่า S มีการเปลี่ยนแปลงน้อยมาก เนื่องจากค่าความชันของ $T(r)$ มีค่าไม่มาก ดังนั้นค่าของ S ก็จะมีลักษณะเป็นโทนมืดคล้ายๆกับ r แต่เมื่อค่า r เข้าใกล้ค่า m มากขึ้นก็จะทำให้ค่า S มีการเปลี่ยนแปลงอย่างรวดเร็วเพราะความชันของ $T(r)$ ในช่วง m นั้นมีค่าสูงมาก ดังนั้นฮิสโตแกรมหลังการแปลงจะมีลักษณะขยายตัวอย่างมากส่วนในรูป (b) นั้น $T(r)$ จะทำให้ได้ค่าของ S มีค่าอยู่ 2 ระดับเท่านั้นคือดำกับขาว และส่วนในรูป (c) นั้นจะเป็นกราฟของฟังก์ชันเชิงเส้นตรงแบบหนึ่งซึ่งเราอาจจะกำหนดสมการของฟังก์ชันดังกล่าวได้ดังสมการที่ 3

$$S_{(i,j)} = \left\{ \begin{array}{l} S_{\max} \\ \frac{(T(r_{(i,j)}) - r_{\min})(S_{\max} - S_{\min})}{(r_{\max} - r_{\min})} \\ S_{\min} \end{array} \right\} + S_{\min} \begin{array}{l} \text{if } r_{\max} \leq T(r_{(i,j)}) \leq 255 \\ \text{if } r_{\min} \leq T(r_{(i,j)}) \leq r_{\max} \dots(3) \\ \text{if } 0 \leq T(r_{(i,j)}) \leq r_{\min} \end{array}$$

จากสมการที่ 3 จะเห็นว่าค่า r ที่อยู่ในช่วง r_{\min} กับ r_{\max} เท่านั้นที่จะทำให้ฮิสโตแกรมของภาพผลลัพธ์มีการกระจายตัว แต่ถ้าค่า r อยู่นอกช่วง r_{\min} กับ r_{\max} ก็จะทำให้ได้ S_{\min} และ S_{\max} ตามลำดับ จากที่กล่าวถึงการปรับความเปรียบต่างนี้สามารถแสดงภาพได้ดังรูปที่ 2.6



รูปที่ 2.6 ภาพและฮิสโตแกรมหลังการปรับความเปรียบต่าง

จากรูปที่ 2.6 จะเห็นว่าภาพที่ได้รับการปรับความเปรียบต่างนั้นมีรายละเอียดที่เพิ่มขึ้นจากภาพต้นแบบ และฮิสโตแกรมจะมีการกระจายตัวของค่าระดับสีเทาเต็มช่วง [0-255] ซึ่งต่างกับฮิสโตแกรมของภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 นิวรอลเน็ตเวิร์ค (Neural Networks)

ความรู้เบื้องต้นเกี่ยวกับนิวรอลเน็ตเวิร์ค

นิวรอลเน็ตเวิร์ค หมายถึง โครงข่ายประสาทที่เชื่อมต่อกันระหว่างเซลล์ประสาทจำนวนมากมาย ซึ่งมีความสามารถประมวลผลสูงบรรจุอยู่ในสมอง โดยปัจจุบันมีการประยุกต์นำนิวรอลเน็ตเวิร์คมาใช้ในอุปกรณ์เครื่องใช้ต่างๆ มากขึ้น ซึ่งเครื่องมือและอุปกรณ์ต่างๆ ที่นำเอามาใช้ช่วยวิเคราะห์นั้น จะมีความฉลาดมากขึ้นและมีระบบความคิดที่มีการทำงานในลักษณะคล้ายกับมนุษย์

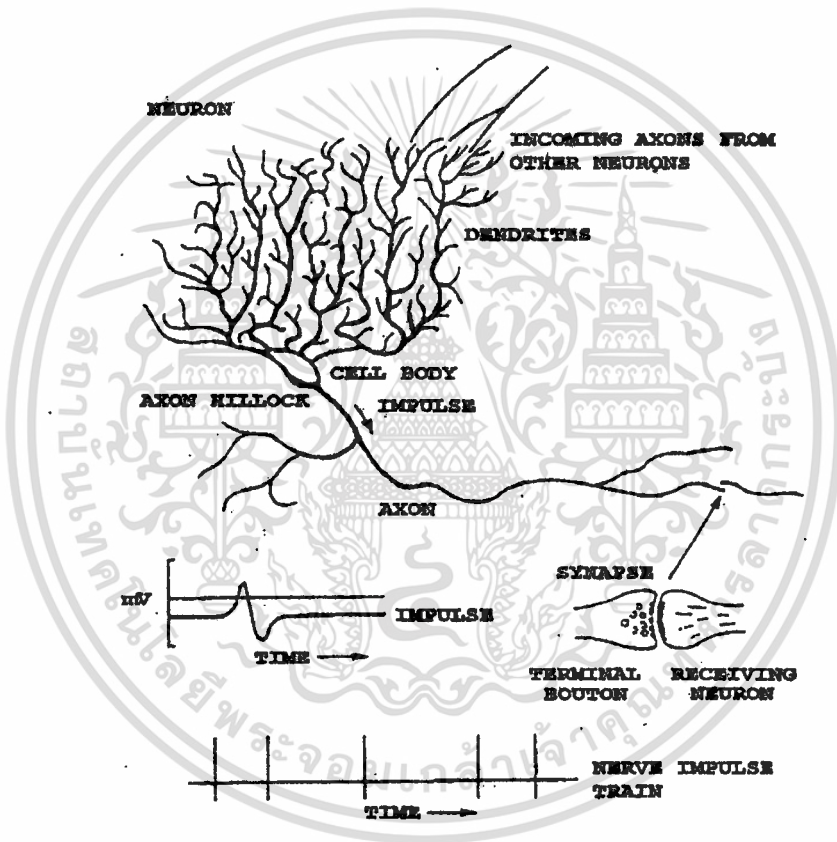
การสร้างโครงข่ายประสาทเทียมมีแนวคิดเลียนแบบการทำงานของสมองชีวภาพ โดยเรียนรู้และศึกษาการทำงานของสมองชีวภาพเพื่อกำหนดแนวทางสำหรับการสร้างแบบจำลองขึ้นมาแล้วพยายามสมมติฐานลักษณะการทำงานโดยจำลองเป็น โมเดลคณิตศาสตร์ที่มีลักษณะเดียวกันแล้วคำนวณโดยใช้คอมพิวเตอร์ (เสรี ปานชาง, 2540:20)

นิวรอลเน็ตเวิร์คชีวภาพ

ระบบคิดคำนึงของมนุษย์ มีโครงสร้างพื้นฐานจากเซลล์สมองที่เรียกว่านิวรอล (Neural) เรียงเป็นชั้นๆ อย่างซับซ้อนจำนวนมากประมาณหมื่นล้าน (10^{11}) นิวรอล และอาจมีจุดเชื่อมโยงส่งผ่านจุดเชื่อมโยงภายในถึงพันล้านล้าน (10^{15}) จุด แต่ละนิวรอลจะมีคุณลักษณะแตกต่างกันไป โดยมีการทำงานคล้ายกัน คือ รับเข้า, ประมวลผล, ส่งออกสัญญาณไฟฟ้าเคมีผ่านไปยังนิวรอล ซึ่งจะส่งสื่อสาร ไปตามระบบของสมอง

จากรูปที่ 2.7 ส่วนแขนงที่ขยายแยกออกไปจากตัวเซลล์คือ ไปยังเซลล์อื่นๆ เพื่อรับสัญญาณ เรียกว่าเดนไดรท์ (Dendrites) จุดรับสัญญาณจากเซลล์อื่นเข้ามายังตัวเซลล์จะผ่านมาทางจุดเชื่อมต่อที่เรียกว่าซินแนปส์ (Synapse) ซึ่งแอกซอน (Axon) จะเป็นตัวส่งสัญญาณเอาท์พุท ออกไปยังนิวรอลอื่น จากผลการวิจัยพบว่า แต่ละนิวรอลจะเชื่อมต่อออกไปยังนิวรอลอื่นๆ ซึ่งแต่ละนิวรอลจะมีคุณสมบัติในการเพิ่ม ขยาย หรือลดทอนความเข้มของสัญญาณบางสัญญาณที่เข้ามาทางเดนไดรท์ของเซลล์ (ซึ่งมีแขนงมากมาย) อาจสามารถกระตุ้นตัวเซลล์ แต่บางสัญญาณก็อาจจะยับยั้งตัวเซลล์ เนื่องจากเซลล์ประสาทหนึ่งเซลล์ มีเดนไดรท์มาก ฉะนั้น สัญญาณกระตุ้นจากเดนไดรท์ ที่รับเข้ามาจากเซลล์ประสาทอื่นๆ จะถูกนำมารวมกันที่ตัวเซลล์ประสาท ที่เซลล์ประสาทจะมีค่าเทรชโฮลด์ (Threshold) ค่าหนึ่งหากผลรวมของสัญญาณไฟฟ้าเคมี (Electrochemical) มีค่ามากกว่าเทรชโฮลด์ เซลล์ประสาทก็จะส่งสัญญาณขนาดหนึ่งผ่านทางแอกซอน ไปยังนิวรอลอื่นๆ การจัดเรียงชั้น (Layer) และลักษณะการเชื่อมโยงระหว่างนิวรอลในสมองนั้น มีการจัดเรียงที่ซับซ้อน

สอดคล้องกับหน้าที่การทำงานเฉพาะส่วน มีการเจริญเติบโตสัมพันธ์กับสิ่งแวดล้อม และมีการเรียนรู้ตลอดเวลา ซึ่งใช้เวลานานนับปี ดังนั้นจึงยากที่จะสร้างโมเดลขึ้นมา เพื่อเลียนแบบให้มีความลักษณะคล้ายสมองชีวภาพได้ทั้งหมด ผลงานที่ได้จากการทำวิจัยในปัจจุบันเป็นเพียงการจำลอง การเลียนแบบ และการทำงานเฉพาะบางส่วนของโครงข่ายประสาทมาใช้เฉพาะกับงานใดงานหนึ่ง ซึ่งมีการวิจัยลักษณะของโครงข่ายแบบต่างๆ ขึ้นมา โดยแต่ละแบบจะเหมาะกับงานประเภทหนึ่งๆ เท่านั้น



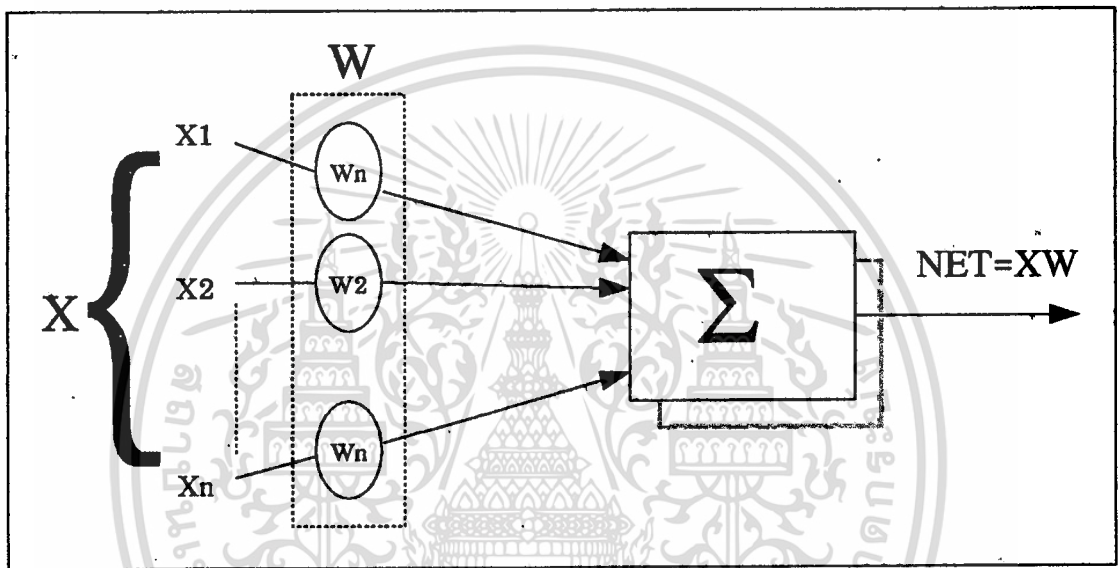
รูปที่ 2.7 แสดงโครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ

โครงข่ายประสาทเทียม (Artificial Neural Networks)

การออกแบบสร้างประสาทเทียมนั้น มีสมมติฐานขั้นแรกจากคุณสมบัติ ของระบบประสาทชีวภาพ ดังที่กล่าวมาคือ ชุดรับสัญญาณข้อมูล อินพุทของเซลล์ประสาทหนึ่ง ได้จากสัญญาณเอาต์พุทของเซลล์ประสาทอื่นๆ ผ่านทางซินแนปส์และเดนไดรต์ ข้อมูลแต่ละค่าที่รับมา จะถูกกลั่นกรองด้วย ซินแนปติกส์ ซึ่งภายในประกอบด้วยสารเคมีประเภท K^+ , Ca^{++} , Na^+ , Cl^- ซึ่งจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะทางความนำพัลส์ (Pulse) สัญญาณไฟฟ้าเคมีที่แตกต่างกัน (James A. Freeman and David M. Skapura, 1991:8-9) ด้วยเหตุนี้ โมเดลประสาทเทียมที่สร้างขึ้น จะต้องมีการถ่วงน้ำหนักให้กับ โมเดล ก่อนที่จะนำเข้าสู่โมเดลประสาทเทียม จุดนี้เรียกว่า น้ำหนักซินแนปติก ปริมาณของข้อมูลที่ เข้าสู่นิวรอน จะถูกนำมารวมกัน และตัดสินใจด้วยระดับความสนใจของนิวรอน (Activation level) แล้วจะส่งเป็นเอาต์พุตที่แยกออกไปยังนิวรอนอื่น ๆ



รูปที่ 2.8 แสดงไดอะแกรมของนิวรอนที่สร้างขึ้น (Artificial Neuron)

จากรูปที่ 2.8 แสดงถึงโมเดลที่สร้างขึ้นโดยแนวความคิดจากเซลล์สมองชีวภาพ สัญญาณอินพุต คือ X_1, X_2, \dots, X_n จะถูกป้อนเข้าไปยังนิวรอนที่สร้างขึ้น ซึ่งเปรียบเทียบกับสัญญาณที่ป้อนเข้ายังซินแนปส์ของนิวรอนชีวภาพ สัญญาณอินพุตนี้จะนำไปคูณกับค่าน้ำหนักซินแนปติกที่มีค่าตั้งแต่ 0.00-1 (Weight: ค่าที่ใช้ถ่วงน้ำหนัก) W_1, W_2, \dots, W_n ก่อนที่จะเข้าสู่บล็อกซัมเมชัน (Σ : Summation) ซึ่งค่าถ่วงน้ำหนักนี้ จะสอดคล้องกับค่าสเตรงท์ (Strength) ของจุดต่อซินแนปส์ชีวภาพแต่ละจุด (Single biological synaptic connection) บล็อกซัมเมชันนี้ ก็จะทำหน้าที่สอดคล้องคล้ายกับตัวเซลล์สมองชีวภาพ ผลรวมทางคณิตศาสตร์ของอินพุต และน้ำหนักจะได้เป็นเอาต์พุต เราเรียกว่า เน็ต (NET) ซึ่งเราจะรวมกันในรูปของเวกเตอร์ได้ดังนี้

$$NET = X_1W_1 + X_2W_2 + \dots + X_nW_n \dots(4)$$

จะได้

$$NET = XW \dots(5)$$

ฟังก์ชันกระตุ้นความสนใจ (Activation Function)

เมื่อได้สัญญาณ NET แล้ว กระบวนการต่อมาที่นิวรอนต้องทำการตัดสินใจ เราจึงต้องกำหนดฟังก์ชันการตัดสินใจ เพื่อใช้เป็นระดับของการตัดสินใจให้กับนิวรอน เพื่อให้ได้สัญญาณเอาต์พุตของนิวรอนออกมา ซึ่งเชื่อมต่อไปยังนิวรอนตัวอื่น ๆ เป็นโครงข่าย OUT ที่ได้อาจเป็น Simple linear function โดย

$$OUT = K[NET] \dots(6)$$

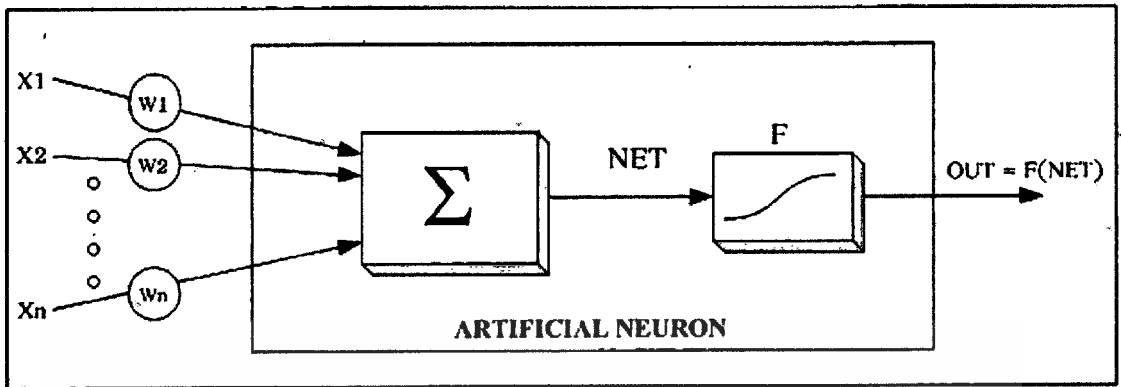
โดย K เป็นค่าคงที่ ที่เรียกว่า Threshold function

ตัวอย่างเช่น

$$OUT = 1 \text{ ถ้า } NET > T$$

$$OUT = 0 \text{ เมื่อเป็นกรณีอื่น} \dots(7)$$

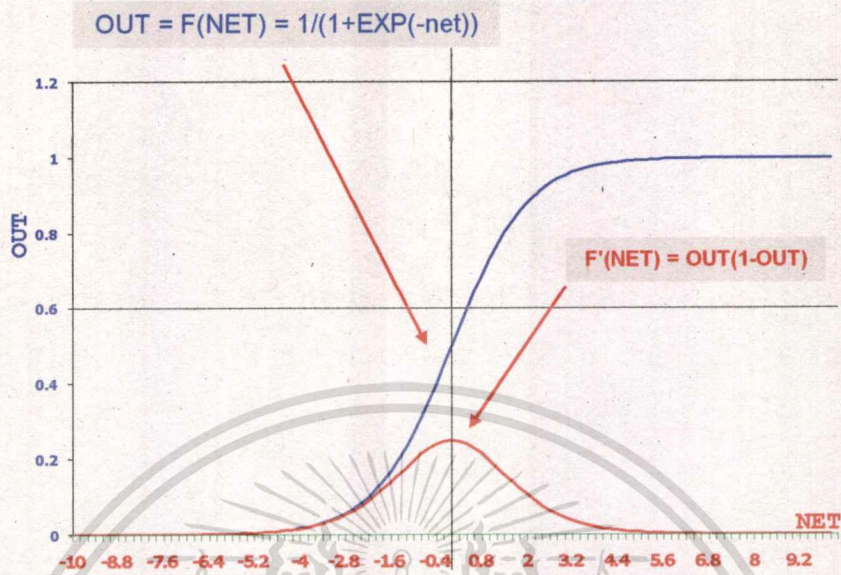
และ T เป็นค่าเทรชโฮลด์คงที่ หรืออาจเป็น Function อื่นๆ ที่เลียนแบบคุณสมบัติไม่เป็นเชิงเส้นของเซลล์ประสาทชีวภาพได้อย่างใกล้เคียงกว่า และใช้เป็นฟังก์ชันให้กับโครงข่ายทั่วไปได้



รูปที่ 2.9 แสดงโมเดลนิวรอนที่สร้างขึ้นร่วมกับ Activation Function

ในรูปที่ 2.9 บล็อก F จะรับผลที่ได้จาก NET มาสร้างเป็นสัญญาณเอาต์พุตที่ OUT โดยกระบวนการภายในบล็อก F จะบีบช่วงของ OUT ให้อยู่ในขอบเขตจำกัด ตามต้องการ ดังนั้นค่า OUT จะมีค่าไม่ต่ำกว่าช่วงที่กำหนดโดยค่าของ NET เราเรียกบล็อก F นี้ว่าสแควซิ่งฟังก์ชัน (Squashing function) และโดยทั่วไป สแควซิ่งฟังก์ชันที่ใช้เป็นแบบลอจิสติกฟังก์ชัน หรือซิกมอยด์ (Logistic function or “Sigmoid”) ซึ่งมีรูปร่างคล้ายตัว S โดยเขียนเป็นสมการคณิตศาสตร์ได้ ดังนี้คือ

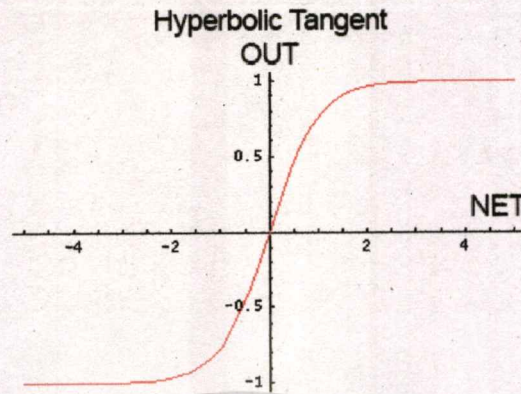
$$F(x) = \frac{1}{(1 + e^{-x})} \dots (8)$$



รูปที่ 2.10 แสดงกราฟที่ได้จากสมการซิกมอยด์ลอจิสติกฟังก์ชัน (Sigmoidal logistic function)

ลักษณะของเทรซโพลด์ฟังก์ชันมีลักษณะเป็น Non-linear function เช่น S-Curve เราจะได้ค่าเอาต์พุตที่มีความไวต่อสัญญาณอินพุตที่มีขนาดเล็กๆ และเฉื่อยต่อสัญญาณแรงๆ ซึ่งสัญญาณอ่อนๆ ไปทางบวกเพียงเล็กน้อยก็จะให้ OUT ใกล้เคียง "1" กระตุ้นหรือสัญญาณอ่อนๆ ทางลบเพียงเล็กน้อย ก็จะทำให้ Output ใกล้เคียง "0" (ขยับขึ้น) ขณะที่สัญญาณแรงๆ ทางบวกก็ยังคงให้ Output ใกล้เคียง "1" และสัญญาณทางลบแรงๆ ก็คงให้ Output ใกล้เคียง "1" และสัญญาณทางลบแรงๆ ก็คงให้ Output ใกล้เคียง "0" เช่นกัน คุณลักษณะแบบนี้ เป็นแบบ Non-linear gain ซึ่งครอสเบ็ก (Grossberg, 1973) พบว่า คุณลักษณะที่เป็น Non-linear gain นี้สามารถแก้ปัญหา Noise-saturation dilemma ได้ และทำให้นิเวศเทียมที่สร้างขึ้นสามารถทำงานกับขนาดของอินพุตได้กว้างมากขึ้น

ยังมีฟังก์ชันอื่นๆ อีกคือ ไฮเปอร์โบลิค แทนเจนต์ (Hyperbolic tangent) มันจะมีลักษณะคล้ายกับ Logistic function และนิยมใช้บ่อยๆ ในการสร้างโมเดลคณิตศาสตร์ การกระตุ้นเร้าความสนใจของเซลล์สมองเทียม ซึ่งมีคุณสมบัติคล้ายชีวภาพของเซลล์สมองคือ $OUT = \text{Tanh}(X)$



รูปที่ 2.11 แสดง Hyperbolic Tangent Function

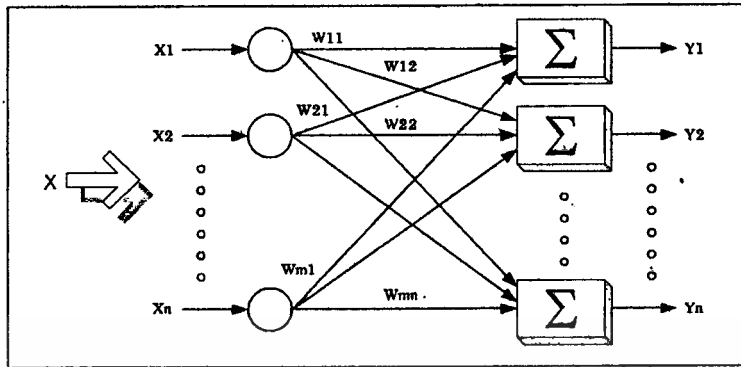
จากรูปที่ 2.11 ส่วนที่เหมือนกับ ซิกมอยด์ ลอจิสติก ฟังก์ชัน คือมีลักษณะเป็น S แต่เนื่องจากมันจะมีความสมมาตร จึงให้ OUTPUT อยู่ระหว่าง “-1” ถึง “1” OUTPUT จะเป็น “0” เมื่อ NET เป็น “0” OUTPUT เข้าใกล้ “1” เมื่ออินพุตไปทางบวก และเข้าใกล้ “-1” เมื่อ อินพุตมีทิศทางไปทางลบ

ฟังก์ชันกระตุ้นความสนใจแบบไม่เป็นเชิงเส้น (The Nonlinear Activation Function)

การนำเอาฟังก์ชันของเลเยอร์หนึ่ง มาเชื่อมกับอินพุตของเลเยอร์ชั้นถัดไป โดยผ่านฟังก์ชันกระตุ้นความสนใจแบบไม่เป็นเชิงเส้น จะทำให้โครงข่ายมีความสามารถในการคำนวณเพิ่มขึ้น (หากไม่ผ่านฟังก์ชันดังกล่าวความสามารถการคำนวณจะไม่เพิ่มขึ้น และจะมีความสามารถไม่แตกต่างไปจาก Single layer networks) และสามารถกำหนดขอบเขตของเอาต์พุตให้อยู่ในช่วงที่ต้องการได้

โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Artificial Neural Networks)

การจะนำเซลล์ประสาทเทียมมาใช้งานได้นั้น ต้องใช้เซลล์ประสาทเทียมที่มีคุณลักษณะต่างๆ กัน (ค่า Weight จะทำให้คุณสมบัติของเซลล์ประสาทเทียม แต่ละเซลล์มีคุณสมบัติแตกต่างกันไป) มาเชื่อมโยงเป็นโครงข่ายในลักษณะเดียวกับเซลล์สมองชีวภาพเสียก่อนซึ่งลักษณะการเชื่อมโยงมีหลายชนิด แต่ละชนิดก็มีคุณลักษณะเด่นที่แตกต่างกันไป



รูปที่ 2.12 แสดงลักษณะ โครงข่ายประสาทเทียมแบบชั้นเดียว

จากรูปที่ 2.12 เป็นโครงข่ายประสาทเทียมแบบชั้นเดียว ที่ประกอบด้วยเซลล์ประสาทเทียม ง่าย ๆ หลาย ๆ ชุด ความสามารถในการคำนวณของโครงข่ายประสาทเทียม ได้มาจากลักษณะการ เชื่อมต่อ เป็นโครงข่ายประสาทเทียมโครงข่ายง่าย ๆ เป็นกลุ่มโมดูลประสาทเทียมที่เชื่อมต่อกันเป็น ชั้น ๆ (Layer) ที่ประกอบด้วยเอาต์พุทเลเยอร์ (กลุ่มของบล็อกข้ามเมชันที่อยู่ทางขวามือ) และ อินพุทเลเยอร์ (วงกลมทางซ้ายมือ) โดยไม่พิจารณาอินพุทเลเยอร์ว่าเป็นนิวรอลเลเยอร์ เนื่องจาก อินพุทเลเยอร์จะทำหน้าที่เชื่อมต่อกับอินพุทที่รับมา และส่งออกไปให้ยังแต่ละอินพุทนิวรอลเลเยอร์ (ในที่นี้คือ Output layer) ในชั้นถัดไป โดยแต่ละอินพุทจะถูกคูณโดยค่าน้ำหนักเฉพาะแต่ละอินพุท โครงข่ายประสาทเทียมที่สร้างขึ้นในขั้นแรกไม่ซับซ้อน โดยแต่ละนิวรอลจะได้เอาต์พุทจากสมการ ที่ 9 หรือเขียนให้สั้น ได้ดังสมการที่ 10

$$\text{OUT} = \text{Logistic Function} * (\text{ผลรวมของ Input X กับ Weight}) \dots (9)$$

หรือ

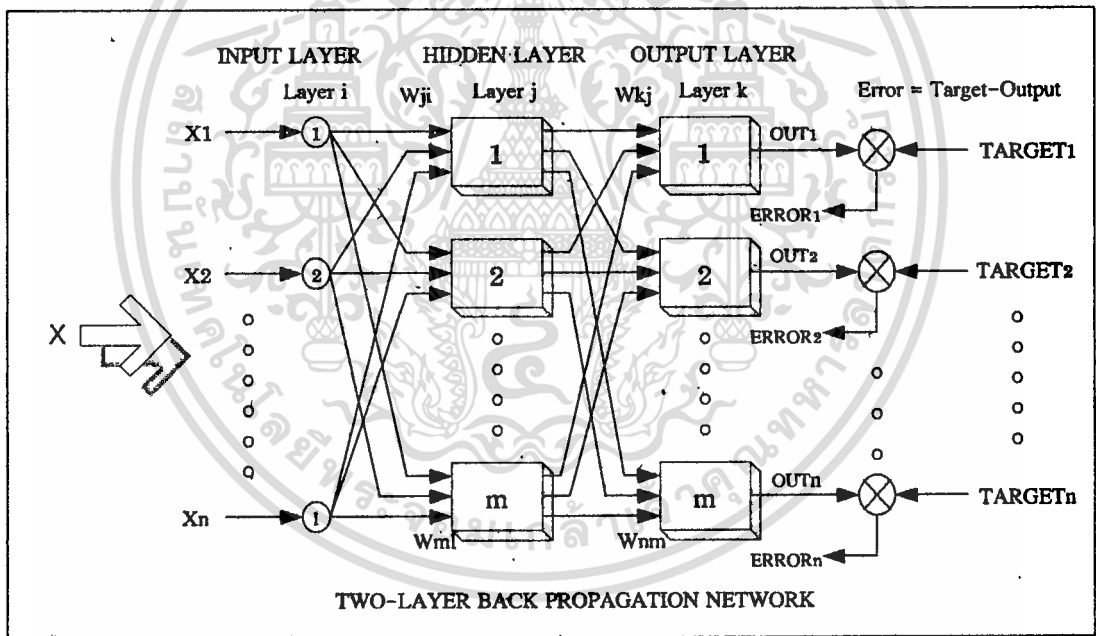
$$\text{OUT} = F(\text{NET}) \dots (10)$$

อย่างไรก็ดีลักษณะการเชื่อมโยงระหว่างโครงข่ายไม่ได้มีแบบเดียว การเชื่อมโยงระหว่าง เลเยอร์อาจมีการเชื่อมโยงย้อนกลับไปที่อินพุทเลเยอร์อีก ซึ่งโครงข่ายประสาทชีวภาพก็มีลักษณะ ดังกล่าวเช่นกัน สำหรับค่า Weight ในรูปที่ 2.12 มีวิธีการพิจารณาในรูปของ เมตริกน้ำหนัก (Weight matrix) ซึ่งหากโครงข่ายมีหลายชั้น จะช่วยให้ระบุค่าน้ำหนักได้ง่ายขึ้น และเพื่อหลีกเลี่ยง ความสับสน จะกำหนดเป็นไดเมนชัน (Dimensions) ของเมตริก โดยให้ m แทนจำนวนแถว หรือ

จำนวนของอินพุต และ n แทนจำนวนของนิวรอน ที่สร้างขึ้น ตัวอย่างเช่น น้ำหนักที่เชื่อมระหว่างอินพุต ตัวที่ 4 กับนิวรอนตัวที่ 2 คือ $W_{4,2}$

โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Artificial Neural Networks)

โครงข่ายที่ซับซ้อนจะมีความสามารถในการคำนวณที่ซับซ้อนมันจะเป็นโครงข่ายที่มีโครงสร้างเป็นจินตนาการที่น่าเป็นไปได้โดยการจัดการเชื่อมโยงนิวรอน มีโครงสร้างเป็นชั้นๆ คล้ายส่วนหนึ่งของสมอง และมีการพัฒนาอัลกอริทึม เกี่ยวกับการฝึกสอนให้โครงข่ายแบบหลายชั้นทำงานได้ตามความต้องการ โครงข่ายแบบหลายชั้นอาจจะสร้างจากกลุ่มของโครงข่ายแบบชั้นเดียวเอาที่ทุกของ Layer หนึ่ง จะใช้เป็นอินพุตของ Layer ถัดไป ในรูปที่ 2.13 แสดงเน็ตเวิร์คที่มีการเชื่อมต่อแบบสองชั้น



รูปที่ 2.13 ไคอะแกรมของ Back Propagation Neural Networks แบบสองชั้น

รูปที่ 2.13 แสดงโครงข่ายประสาทเทียมแบบสองชั้นที่ต่อเชื่อมโยงแบบเต็มชั้น ในโครงข่ายแบบสองชั้น มีการเรียกชื่อชั้นต่างๆ ดังนี้ คือ ชั้นที่ต่อโดยตรงกับอินพุต เรียกว่า อินพุตเลเยอร์ (Input layer) ชั้นนี้จะไม่มีการคำนวณ แต่จะทำหน้าที่ต่อเชื่อมข้อมูลไปยังชั้นถัดไป ชั้นที่อยู่ท้ายสุดทางขวามือเรียกว่า เอาท์พุทเลเยอร์ (Output layer) เป็นชั้นที่โครงข่ายจะให้ผลลัพธ์ ส่วนชั้น

ที่อยู่ระหว่างอินพุทเลเยอร์ และเอาต์พุทเลเยอร์ จะมีที่ชั้นก็ตามจะเรียกว่า ฮิดเดนเลเยอร์ (Hidden layer) หากฮิดเดนเลเยอร์ มีหลายๆ ชั้น ก็จะมีการตั้งชื่อเฉพาะลงไปให้กับแต่ละชั้น

การฝึกสอนให้กับโครงข่ายประสาทเทียม (Training of Artificial Neural Networks)

ค่าน้ำหนัก มีความสัมพันธ์กับอะไร เปลี่ยนแปลงอย่างไร นั่นก็เช่นเดียวกับเด็กที่คลอดออกมา ก็มีสมองแล้ว แต่สมองยังไม่เจริญเติบโตเพียงพอ และยังไม่ได้รับการฝึกสอน และเรียนรู้ เด็กจึงไม่สามารถทำกิจกรรมใดๆ ด้วยตนเอง เด็กจะพัฒนาการเรียนรู้ไปตามขั้นตอน หลังจากนั้นสมองของเขาจะได้รับการฝึกสอน และเจริญเติบโตไปพร้อมๆ กัน เซลล์สมองจะได้รับการปรับคุณลักษณะสอดคล้องกับการฝึกสอน และจะเจริญเป็นโครงข่ายสอดคล้องกัน

โครงข่ายประสาทเทียมที่สร้างขึ้นมีลักษณะเช่นเดียว กัน คือ เมื่อสร้างเสร็จ แต่ละเซลล์ประสาทที่สร้างขึ้นมานั้น จะยังไม่มีคุณลักษณะใดเลย เนื่องจากยังไม่มีกำหนดค่าวงที่เหมาะสมกับงานที่ต้องการให้กับมัน จึงต้องมีกรฝึกสอนเพื่อให้เน็ตเวิร์ค ที่สร้างขึ้นมีคุณลักษณะตามที่ต้องการ การฝึกสอนของโครงข่ายประสาทเทียม จะกระทำโดยการปรับ เปลี่ยนค่าน้ำหนัก เพื่อให้โครงข่ายจดจำแพตเทิร์นความสัมพันธ์ระหว่างอินพุทกับเอาต์พุทได้ โดยในขั้นแรกอาจกำหนดเป็นค่าสุ่มใดๆ (Random weight) ก่อนแล้วถึงปรับเปลี่ยนน้ำหนักไปตามอัลกอริทึมสมมติฐาน หลายๆ รอบจนกว่าจะได้เอาต์พุทของเน็ตเวิร์ค เหมือนกับเอาต์พุทที่ต้องการ ในเงื่อนไขความผิดพลาดที่ยอมรับได้

วัตถุประสงค์ของการเทรนนิ่ง (Objective of Training)

เนื่องจากค่าน้ำหนักที่ให้เป็นค่าสุ่มใดๆ โครงข่ายจึงไม่แสดงคุณลักษณะใดออกมา การฝึกสอน (Training) ให้โครงข่ายก็คือ การปรับค่าน้ำหนักทุกๆ จุด ให้สอดคล้องกับอินพุทหลายๆ แบบ เพื่อให้ได้เอาต์พุทตามต้องการนั่นเอง การฝึกสอนโครงข่าย จะต้องบรรลุถึงกระบวนการเข้าใจพื้นฐานเสียก่อน คือ การเรียนรู้ในโครงข่ายประสาทเทียมนั้นมีขีดจำกัด ปัญหาต่างๆ ผู้ใช้คงต้องแก้ให้มันก่อน แล้วนำผลนั้น ไปอ้างอิงสำหรับการปรับปรุงค่าน้ำหนัก หลังจากปรับน้ำหนักจนได้ค่าผิดพลาดที่เอาต์พุทเทียบกับเป้าหมายน้อยลงเป็นที่พอใจแล้ว โครงข่ายประสาทเทียมนั้นก็พร้อมที่จะวิเคราะห์อินพุท และให้เอาต์พุทตามลักษณะตัวอย่างที่มันเคยเรียนรู้มา การเรียนรู้จะมีการปรับน้ำหนักหลายๆ รอบ จนค่าน้ำหนักสอดคล้องกับตัวอย่างหลายๆ ตัวอย่าง และให้เอาต์พุทตามต้องการ พบว่าโครงข่ายได้ตัวอย่างสำหรับการเทรนนิ่งมากๆ โครงข่ายก็จะมีความแม่นยำสูงขึ้น แต่ก็ใช้เวลาในการเทรนนิ่งเพิ่มขึ้นเช่นกัน หากพิจารณาต่อไปจะพบว่า โครงข่ายประสาท

เทียมที่สร้างขึ้นจะมีพฤติกรรมคล้ายกับระบบการเรียนรู้ของมนุษย์มาก เป็นเพราะมีต้นแบบมาจาก ระบบประสาทชีวภาพนั่นเอง

การเทรนนิ่งแบบควบคุม (Supervised Training)

เทรนนิ่งอัลกอริทึมถูกจัดเป็น 2 ประเภท คือ แบบควบคุม(Supervised training) และแบบ อิสระ (Unsupervised training) โดย การเทรนนิ่งแบบควบคุม จะต้องการคู่ของเทรนนิ่งระหว่าง อินพุตกับเป้าหมายที่ต้องการ ที่เรียกว่า เทรนนิ่งแพร์ (Training pairs) โครงข่ายจะถูกเทรนไปตาม จำนวนของคู่ที่เทรนนิ่ง (จำนวนคู่ของ Input กับ Output ที่ต้องการให้โครงข่ายรู้จัก) เอาท์พุทที่ คำนวณได้จากโครงข่ายจะถูกเปรียบเทียบกับความสอดคล้องกับเป้าหมาย ค่าผิดพลาดที่เกิดขึ้นจะ ถูกป้อนกลับไปยัง โครงข่ายและเปลี่ยนแปลงค่าน้ำหนักให้สอดคล้องกับอัลกอริทึมที่ทำให้ แนวโน้มของค่าผิดพลาดที่เกิดขึ้นระหว่างเอาท์พุทกับเป้าหมายโดยเฉลี่ยมีค่าลดต่ำลง ตัวอย่างการ เทรนนิ่งแบบนี้ ได้แก่ การเทรนนิ่งแบบแพร่กลับ (Back propagation)

การเทรนนิ่งแบบอิสระ (Unsupervised Training)

ถึงแม้ว่า อัลกอริทึมแบบควบคุม (Supervised training) สามารถจะประยุกต์ใช้เพื่อปรับ คุณลักษณะของโครงข่ายได้สำเร็จ แต่ก็ยังมีข้อวิจารณ์อยู่ คือ มันเป็น ไปอย่างแบบชีวภาพไม่ได้ และ ยากที่จะเชื่อได้ว่า กลไกการเทรนนิ่งของสมองจะต้องการการเปรียบเทียบระหว่างคุณที่ต้องการกับ เอาท์พุทจริง โดยกระบวนการป้อนกลับไปแก้ไขคุณลักษณะของโครงข่าย และถ้าสมมติว่า ถ้า สมองมีกลไกเช่นนี้ ต้องมีผู้หาเอาท์พุทที่ต้องการเพื่อนำมาเป็นเป้าหมายตลอดเวลา และจะเอามา จากที่ใด สรุปคือ ต้องมีผู้คิดเป้าหมายให้กับโครงข่ายก่อน โครงข่ายไม่สามารถคิดและปรับ คุณลักษณะได้ก่อนด้วยตนเอง ในทางตรงกันข้ามหากพิจารณาทารกแรกเกิดสมองของเขาสามารถ จัดระบบเองได้อย่างไร การเทรนนิ่งแบบระบบของสมอง จนกระทั่งมีการพัฒนาการเทรนนิ่งแบบ อิสระนี้ขึ้นราวปี 1984 โดยโคโฮเนน (Kohonen) และบุคคลอื่นๆ โดยได้เสนอแนวคิดที่เป็นการ เทรนนิ่งแบบไม่ต้องการเป้าหมาย ไม่มีการตัดสินใจด้วยเหตุผลในอุดมคติมาก่อน ชุดของการ เทรนนิ่ง จะมีเพียงอินพุทเวกเตอร์เท่านั้น เทรนนิ่งอัลกอริทึม จะเปลี่ยนแปลงค่าน้ำหนักของ โครงข่าย เพื่อสร้างเอาท์พุทที่มีความมั่นคง ยกตัวอย่าง เช่น หากให้โครงข่ายรู้จำภาพหน้าคนหนึ่ง หากภาพหน้าคนคนนั้น เปลี่ยนแปลง ไปเล็กน้อย (Image อาจมี Noise ร่วมอยู่บ้าง) โครงข่ายนั้นก็ยังสามารถบอกได้ว่า คนคนนั้นเป็นคนเดิมเป็นต้น การเทรนนิ่งจะไม่มี การตัดสินใจมาก่อน ไม่มีการ กำหนดแบบเอาท์พุทมาก่อน (อาจกล่าวได้ว่าแบบเอาท์พุทจะถูกกำหนดโดยอินพุทเวกเตอร์นั่นเอง)

ดังนั้น เอ้าท์พุทของโครงข่ายก็เช่นกัน ส่วนใหญ่จะถูกแปรรูปซึ่งจะเข้าใจได้ภายหลังกระบวนการเทรนนิ่ง ดังนั้นจึงไม่สามารถแก้ปัญหาที่เคร่งครัดสำคัญได้ แต่มักนิยมใช้โครงข่ายแบบนี้กับงานง่าย ๆ ประเภทการเปรียบเทียบเอกลักษณ์, รูปแบบที่สัมพันธ์กันระหว่างอินพุท-เอ้าท์พุท ที่ถูกกำหนดโดยโครงข่าย

วิธีการแก้ปัญหาคณิตศาสตร์ (Training Algorithm)

ส่วนใหญ่แล้วทุกวันนี้ การแก้ปัญหาฝึกสอนของโครงข่ายค่อยๆ พัฒนาก้าวหน้าขึ้นจากแนวความคิดของ คี โอ เฮบบ์ (ปี 1961) เขาได้เสนอโมเดลของ การเทรนนิ่งแบบอิสระ (Unsupervised training) ในแบบซินแนปติกส์ตรงๆ หรือน้ำหนัก ซึ่งจะเพิ่มขึ้น ถ้าทั้งแหล่งกำเนิด (Input Source) และจุดหมายปลายทาง (Destination) ของนิวรอนได้รับการสนใจ กรณีนี้ถ้ามีการใช้งานทางเส้นนี้บ่อยๆ ก็จะทำให้ซินแนปติกส์ตรงๆ (หรือน้ำหนัก) แข็งแรงขึ้น (เซลล์สมองที่ใช้งานมากบ่อยๆ ก็จะทำให้ ซินแนปส์ใหญ่ขึ้น การส่งผ่านข้อมูลพัลส์ไฟฟ้าทำได้ดีขึ้น ทำให้มีประสิทธิภาพดีขึ้น เช่นสามารถคิด หรือจดจำได้เร็ว และดีขึ้น) โครงข่ายประสาทเทียมนี้ ใช้การเรียนรู้แบบเฮบบ์ (Hebbian learning) จะเพิ่มค่าน้ำหนักของโครงข่ายอย่างสอดคล้องกับผลคูณของระดับความสนใจของแหล่งกำเนิด และจุดหมายของนิวรอนตามสมการดังนี้

$$W_{ij}(n+1) = W_{ij}(n) + \alpha \text{OUT}_i \text{OUT}_j \dots (11)$$

โดย $W_{ij}(n)$ คือค่าน้ำหนัก จากนิวรอน i ไปยังนิวรอน j ก่อนปรับปรุงค่า
 $W_{ij}(n+1)$ คือค่าน้ำหนัก จากนิวรอน i ไปยังนิวรอน j หลังปรับปรุงค่า
 α คือค่าคงที่ของการเรียนรู้ (Learning rate coefficient)
 OUT_i คือเอ้าท์พุทของนิวรอน i และเป็นอินพุทของนิวรอน j
 OUT_j คือเอ้าท์พุทของนิวรอน j

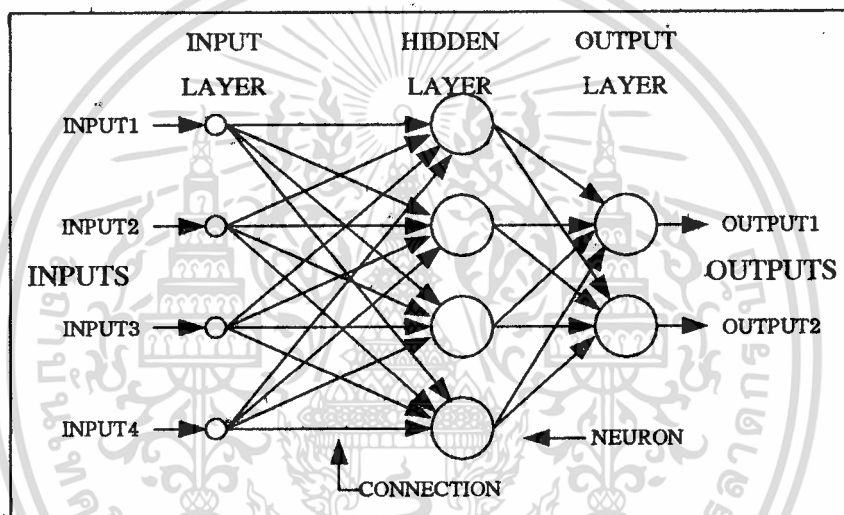
โครงข่ายที่มีลักษณะการเทรนนิ่งแบบเฮบบ์นั้น เป็นผลมาจากการพัฒนามาแล้วกว่า 20-30 ปี โดยเฉพาะงานของโรเซนเบลทท์ (Rosenblatt:1962), วิโดว์ (Widrow:1959), วิโดว์ และโฮลฟ์ (Widrow&Hoff:1960) และอีกหลายๆ คนที่พยายามพัฒนาระบบการเทรนนิ่งแบบควบคุม ที่สร้างโครงข่ายที่สามารถเรียนรู้แบบของอินพุทได้อย่างกว้างขวาง และมีอัตราการเรียนรู้

สูงที่บรรลุผลได้จากหลักการพื้นฐานของการเรียนรู้ หรือเทรนนิ่งให้กับโครงข่ายเช่น Perceptrons, Hopfieldnets, Backpropagation Networks และ Counter propagation,

การประยุกต์ใช้นิวรอลเน็ตเวิร์ค

จากการศึกษารูปแบบของเม็ดยาพบว่าสอดคล้องกับอัลกอริทึม การเทรนนิ่งแบบควบคุม (Supervised training) จึงเลือกใช้วิธี Back Propagation มีกลิตเลเซอร์เป็นแบบ 2 เลเยอร์ แสดงดังรูปที่

2.14



รูปที่ 2.14 โคอะแกรมของ Two Layer Feed Forward

พิจารณารูปที่ 2.14 ประกอบ จะเห็นว่าประกอบด้วย Input array, Hidden layer และ Output layer ในส่วนของ Input array จะไม่ถือเป็น layer เพราะไม่มีการคำนวณใดๆ แต่มันจะทำหน้าที่เชื่อมโยงข้อมูลให้กับ Hidden layer ทุกๆ นิวรอล Output ของ Hidden layer ก็จะถูกเชื่อมโยงไปเป็น Input ให้กับ Output layer ทุกอินพุทของนิวรอลทุกตัว จะมีค่า Weight ประจำตัว ค่า Weight นี้ทำให้ระดับของอินพุทผ่านเข้าไปภายในนิวรอลแต่ละตัวไม่เท่ากัน ซึ่งเป็นตัวกำหนดคุณสมบัติของนิวรอลแต่ละตัวในเน็ตเวิร์ค วิธีการกำหนดคุณสมบัติดังกล่าว ทำได้โดยการเทรนนิ่งโดยในขั้นแรก จะสมมุติว่า ค่าแบบสุ่มน้อยๆ ให้(ช่วงที่ใช้คือตั้งแต่ -0.5 ถึง $+0.5$) แล้วจึงค่อยๆ ปรับเปลี่ยนค่าน้ำหนักในแต่ละ layer ให้สอดคล้องกับคุณสมบัติที่ต้องการ

กระบวนการเทรนนิ่ง

กระบวนการเทรนนิ่ง คือ กระบวนการปรับคุณลักษณะของนิวรอนเน็ตเวิร์ค ให้มีคุณสมบัติตามที่ต้องการ การเทรนนิ่งของอัลกอริทึม Back Propagation เป็นแบบ Supervised Training แบ่งเป็น 2 ส่วนคือ Forward pass และ Reversed pass การเทรนนิ่งจะกระทำทั้ง Forward pass และ Reverse pass สลับกันไป จนกว่าค่าผิดพลาดระหว่าง Output กับ Target จะมีค่าต่ำกว่าจุดที่ต้องการ

Forward Pass

จากภาพ 2.13 อินพุต X ตั้งแต่ X_1 ถึง X_I จะถูกคูณด้วย Weight matrix ผลรวมของอินพุตเมื่อคูณกับค่า Weight ของแต่ละนิวรอนเรียกว่า เน็ต (NET) เขียนเป็นสมการได้คือ

$$NET_j = X_1W_{j1} + X_2W_{j2} + \dots + X_IW_{jI} \dots (12)$$

หรือ

$$NET_j = \sum_{i=1}^I X_i W_{ji} \dots (13)$$

โดย	X	คือ ค่าที่รับเข้ามาของ Input node
	W	คือ ค่าน้ำหนักประจำตัวของแต่ละอินพุต โหนด
	I	คือ หมายเลขโหนด โดยเริ่มที่โหนด 1 ถึง โหนด I
	J	คือ หมายเลขนิวรอนใน Hidden layer ตั้งแต่ นิวรอนที่ 1 ถึง M
	K	คือ หมายเลขนิวรอนใน Output layer ตั้งแต่ นิวรอนที่ 1 ถึง N

และทำนองเดียวกัน

$$NET_k = \sum_{j=1}^M OP_j W_{kj} \dots (14)$$

เมื่อได้ค่า NET ของแต่ละนิวรอนแล้ว จะปรับให้ได้ Output อยู่ในช่วงที่ต้องการเป็นค่าที่จะนำไปยับยั้ง หรือกระตุ้นนิวรอนตัวถัดไป โดย Squashing Function ที่ใช้เป็น Logistic Function หรือ Sigmoid จะได้ตามสมการ 15 และ 16

$$OP_j = F_j(NEI_j) = \frac{1}{1 + e^{-(net\theta_j)}} \dots (15)$$

$$OP_k = F_k(NEI_k) = \frac{1}{1 + e^{-(net\theta_k)}} \dots (16)$$

กำหนด $\theta_j = 0.05$ และ $\theta_k = 0.05$

θ_j และ θ_k คือ ค่าที่กำหนดความชันของ Sigmoid curve

OP_j คือ ค่า Output ที่ Neuron j ของ Hidden Layer (เป็น Input ของ Output Layer)

OP_k คือ ค่า Output ที่ Neuron k ของ Output Layer (เป็น Output ของ Output Layer)

การคำนวณหาค่า Output จะเริ่มที่ Hidden layer ที่ชั้น j ก่อน เมื่อได้ค่า output ของ Hidden layer ครบทุกตัวแล้ว จึงคำนวณหาค่า Output ที่ Output layer โดย Input ของ Output layer ก็คือ Output ของ Hidden layer นั้นเอง โดยใช้การคำนวณทำนองเดียวกันกับ Hidden layer Output ที่ได้ จะนำไปเปรียบเทียบกับ Target ที่ต้องการเกิดเป็น Error ขึ้น ค่า Error นี้ จะถูกนำไปใช้ร่วมกับ ค่าดิฟเฟอเรนเชียล ของ $F'(F')$ ใน Output layer เพื่อปรับค่า Weight ให้กับ Output layer ดังจะกล่าวต่อไปในส่วนของ Reverse pass

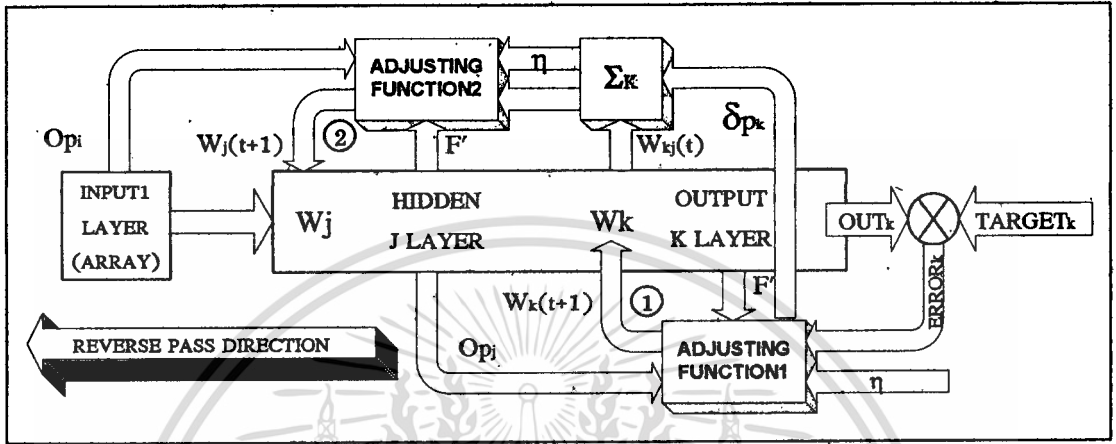
Reverse Pass

เป็นส่วนที่ป้อนกลับค่า Error (Target-Output) จากเอาต์พุตเลเยอร์ย้อนกลับมาปรับปรุงค่าน้ำหนักที่เลเยอร์ก่อนหน้า สำหรับนิเวศน์ตเวร์คที่มีมากกว่า 1 เลเยอร์ การปรับค่าน้ำหนักชั้นถัดออกมา จะใช้ค่าน้ำหนักเดิมของเลเยอร์ที่สูงกว่า ร่วมกับตัวร่วมทางด้าน Output ของเลเยอร์มาคำนวณ (เนื่องจากไม่มีค่า Error มาใช้เหมือน Output layer) ดังจะกล่าวในรายละเอียดดังต่อไปนี้

จากรูปที่ 2.15 การปรับเปลี่ยนค่าน้ำหนักใหม่ จะเป็นไปตามทิศทางของลูกศร และได้ข้อมูลจากค่า Error ที่ Output layer เป็นต้นกำเนิดเพื่อกำหนดทิศทางปรับค่าน้ำหนักให้นิวรอนตเวร์ค มีคุณลักษณะสมบัติตามต้องการ ลำดับขั้นตอนของ Reverse pass พอสรุปได้ดังนี้คือ

1. นำ Output ที่ได้จาก Output Layer ไปลบออกจาก Target ได้เป็น error
2. นำค่า Error มาคูณกับ F' ที่ได้จาก Output layer เรียกว่า δp_k

3. นำค่า δ_{pk} คูณกับ Op_j แล้วคูณกับ η (อัตราการเรียนรู้ 0.01-1)
4. ค่า Weight ค่าใหม่ที่ได้ คือ ค่า Weight เดิมบวกกับผลลัพธ์ ที่ได้จากข้อ 3



รูปที่ 2.15 แสดงไดอะแกรมของการปรับค่า Weight ในชั้น Output layer และ Hidden layer

พอเขียนเป็นสมการใหม่ได้คือ

$$W_{kj}(t+1) = W_{kj}(t) + \eta \delta_{pk} Op_j \dots (17)$$

เพราะว่า

$$\eta \delta_{pk} Op_j = \Delta W_{kj} \dots (18)$$

ฉะนั้น

$$W_{kj}(t+1) = W_{kj}(t) + \Delta W_{kj} \dots (19)$$

- เมื่อ $W_{kj}(t)$ คือ ค่าน้ำหนักเดิมก่อนปรับ
- $W_{kj}(t+1)$ คือ ค่าน้ำหนักหลังปรับ
- η คือ อัตราการเรียนรู้
- Op_j คือ Output ของ Layer j
- δ_{pk} คือ ค่าที่ได้จาก Error คูณกับดิฟเฟอเรนเชียลของฟังก์ชัน F

การปรับค่าของ Layer ถัดมา (Hidden layer) มีลักษณะคล้ายคลึงกับ Output layer คือ

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_{pj} Op_i \dots (20)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่า

$$\eta \delta_{p_j} O_{p_j} = \Delta W_{ji} \dots (21)$$

ฉะนั้น

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \dots (22)$$

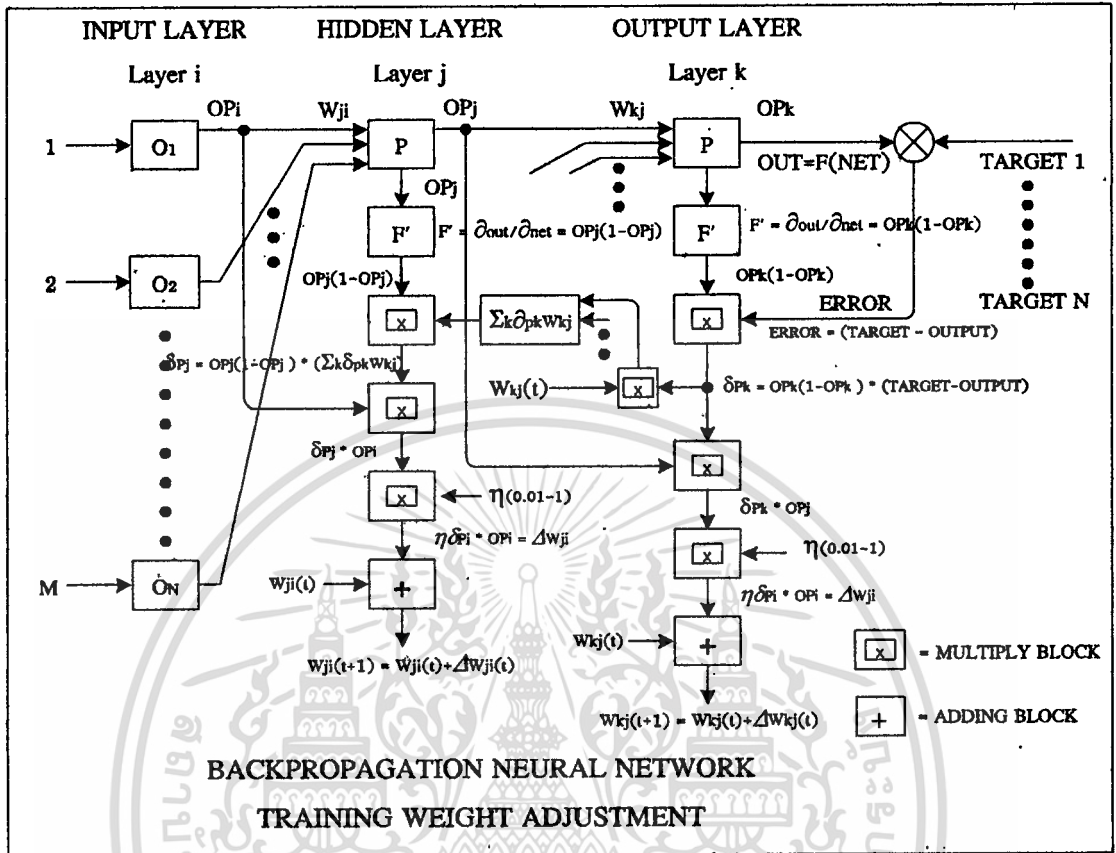
สมการที่ 21 แตกต่างกับสมการ 18 ตรงการได้มาซึ่ง δ_{p_j} เพราะไม่มีค่า Output กับ Target ที่ Hidden layer ค่าดังกล่าวสามารถทดแทนได้ด้วย $\sum_k (\delta_{p_k} W_{kj})$

โดย

$$\delta_{p_j} = O_{p_j}(1-O_{p_j})(\sum_k (\delta_{p_k} W_{kj})) \dots (23)$$

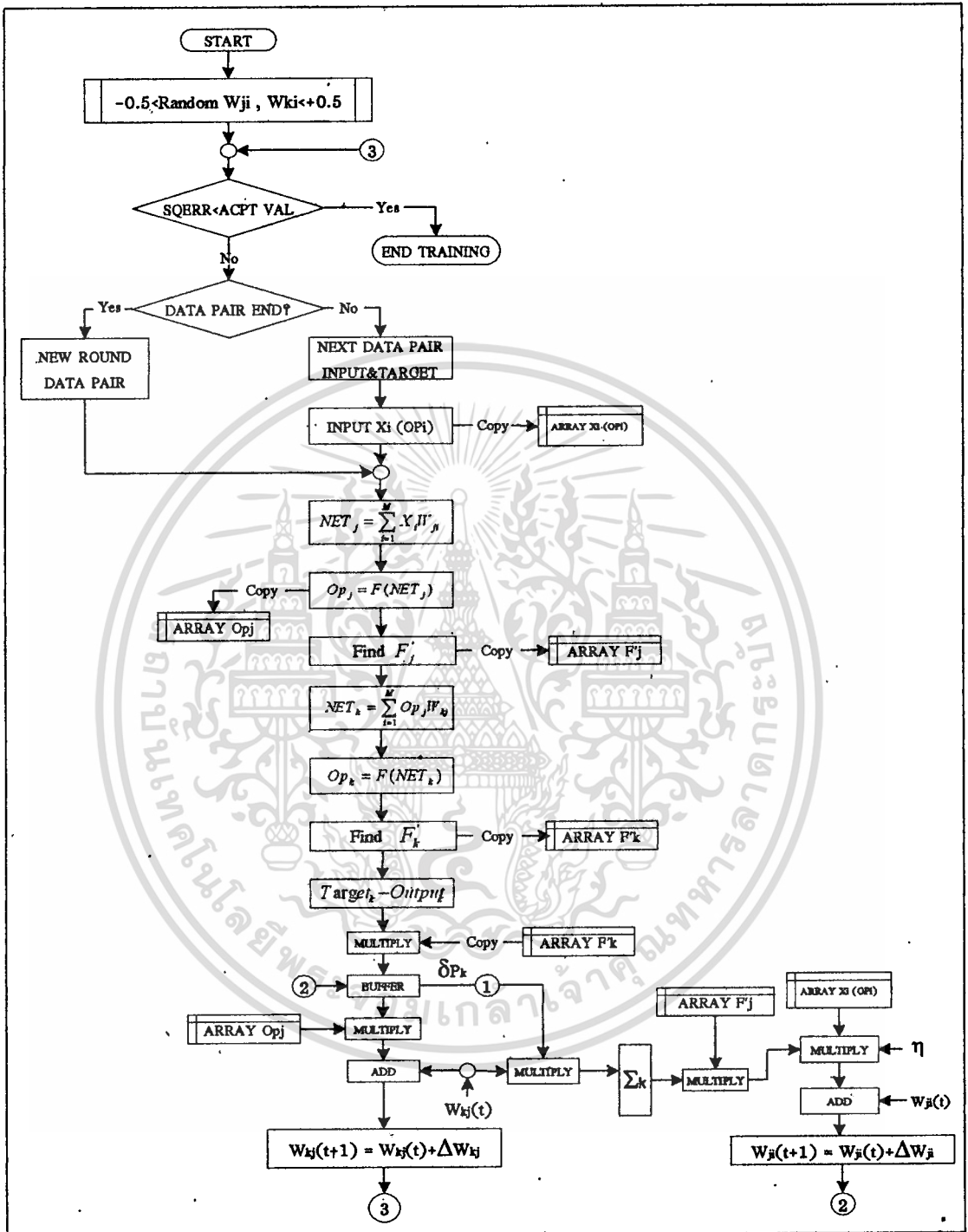
เมื่อ	$W_{ji}(t)$	คือ ค่าน้ำหนักเดิมก่อนปรับ
	$W_{ji}(t+1)$	คือ ค่าน้ำหนักหลังปรับ
	η	คือ อัตราการเรียนรู้
	O_{p_j}	คือ ข้อมูลที่ได้จาก Input Layer
	δ_{p_j}	คือ ค่าควบคุมให้น้ำหนักเปลี่ยนแปลงเข้าสู่จุดที่ดีที่สุด

การปรับค่าน้ำหนักจะปรับทีละเลเยอร์จาก Output layer กลับไปยัง Input layer เมื่อเสร็จสิ้นก็จะกลับสู่ส่วนของ Forward pass อีกครั้ง เพื่อหา Output มาเปรียบเทียบกับ Target และดำเนินการปรับค่าน้ำหนักในส่วนของ Reverse pass เช่นนี้สลับกันไปจนกว่าค่า Error ที่ได้จะลดลงต่ำกว่าค่าที่ต้องการจึงหยุดกระบวนการ ค่าน้ำหนักที่ได้จะเป็นค่าที่เหมาะสมสำหรับทุกคู่ของ Input และ Target ซึ่งพร้อมที่จะนำไปใช้ในกระบวนการตรวจสอบ การปรับค่าน้ำหนักเทรนนิ่งของ Backpropagation Networks พออธิบายได้ด้วยไดอะแกรมดังนี้



รูปที่ 2.16 โค้ดแกรมแสดงวิธีการหาค่าน้ำหนักที่เหมาะสมของการเทรนนิ่ง โดยใช้อัลกอริทึมแบบแบคพรอบพากชัน

และเขียนเป็น Flow Chart ได้ดังรูปที่ 2.17



รูปที่ 2.17 แสดง Flow Chart แสดงกระบวนการเทรนนิ่ง เพื่อหาค่าน้ำหนัก
ที่เหมาะสมในกระบวนการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขการหยุดการฝึกสอน

การกำหนดเงื่อนไขการหยุดฝึกสอนทำได้สองกรณีคือ

1. เมื่อผลรวมของค่าผิดพลาดเฉลี่ยระหว่างเข้าที่พู่กับเป้าหมายทั้งหมด (SSQERR) ลดลงน้อยกว่าค่าที่กำหนด โดยกำหนดสมการหาค่าผิดพลาดเฉลี่ยของเข้าที่พู่ทั้งหมดดังนี้

$$SSQERR = \frac{1}{2} \sum (Target - Output)^2 \dots (24)$$

SSQERR คือ ค่า Sum Square error ที่ได้จาก Output neurons เทียบกับ Target ใช้สำหรับบ่งชี้ผลของการฝึกสอนว่าเพียงพอกับการใช้งาน หรือเงื่อนไขที่กำหนดหรือไม่

2. จำนวนรอบของการฝึกสอนดำเนินไปถึงค่าที่ตั้งไว้ ซึ่งจำนวนรอบที่เหมาะสมจะได้ออกการทดลอง (ในการทดลองจำนวนรอบที่เหมาะสมอยู่ในช่วง 500 รอบ)



บทที่ 3

ระบบงานปัจจุบัน

ในการวิเคราะห์ระบบงานถึงขั้นตอนการทำงานต่าง ๆ ของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติดนั้น ผู้เขียนได้นำเอาระบบแยกแยะเม็ดยา มาเป็นกรณีศึกษา เพื่อทำการวิเคราะห์ถึงขั้นตอนการทำงานและปัญหาต่าง ๆ ที่มีอยู่ในการทำงานของระบบปัจจุบัน ซึ่งรายละเอียดต่าง ๆ ที่เกี่ยวข้องและขั้นตอนการทำงานของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด มีดังนี้

3.1 ความเป็นมาขององค์กร

องค์กรที่จะนำมาเป็นกรณีศึกษาคือ สำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด ซึ่งมีรายละเอียดต่าง ๆ ดังต่อไปนี้

บทบาทขององค์กร

1. เป็นหน่วยประสานนโยบายและแผน
2. เป็นหน่วยประสานการปฏิบัติงาน
3. เป็นหน่วยสนับสนุนข้อมูล ข่าวสาร และวิชาการ
4. เป็นหน่วยปฏิบัติสำหรับงานด้านประชาสัมพันธ์ต่อต้านยาเสพติด งานบริหารโครงการที่มีลักษณะเป็น Pilot Project และงานตามภารกิจที่กฎหมายระบุให้สำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติดเป็นหน่วยงานรับผิดชอบ
5. เป็นหน่วยงานกลางในการประสานความร่วมมือระหว่างประเทศ
6. เป็นหน่วยงานพัฒนาบุคลากรที่เกี่ยวข้องกับการป้องกันและปราบปรามยาเสพติดของประเทศและภูมิภาค
7. เป็นหน่วยงานในการควบคุม ตรวจสอบและกำกับ ตลอดจนติดตามและประเมินผลการดำเนินงานป้องกันและแก้ไขปัญหายาเสพติด
8. เป็นหน่วยงานกลางในการประสาน กำกับ ดูแล และให้ความเห็น ตลอดจนข้อเสนอแนะด้านงบประมาณป้องกันและปราบปรามยาเสพติดของประเทศ

สถานที่ตั้งและหมายเลขติดต่อ

สำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด

เลขที่ 5 ถนนดินแดง แขวงสามเสนใน เขตพญาไท กรุงเทพมหานคร 10400

โทรศัพท์ 0-2247-0901-19

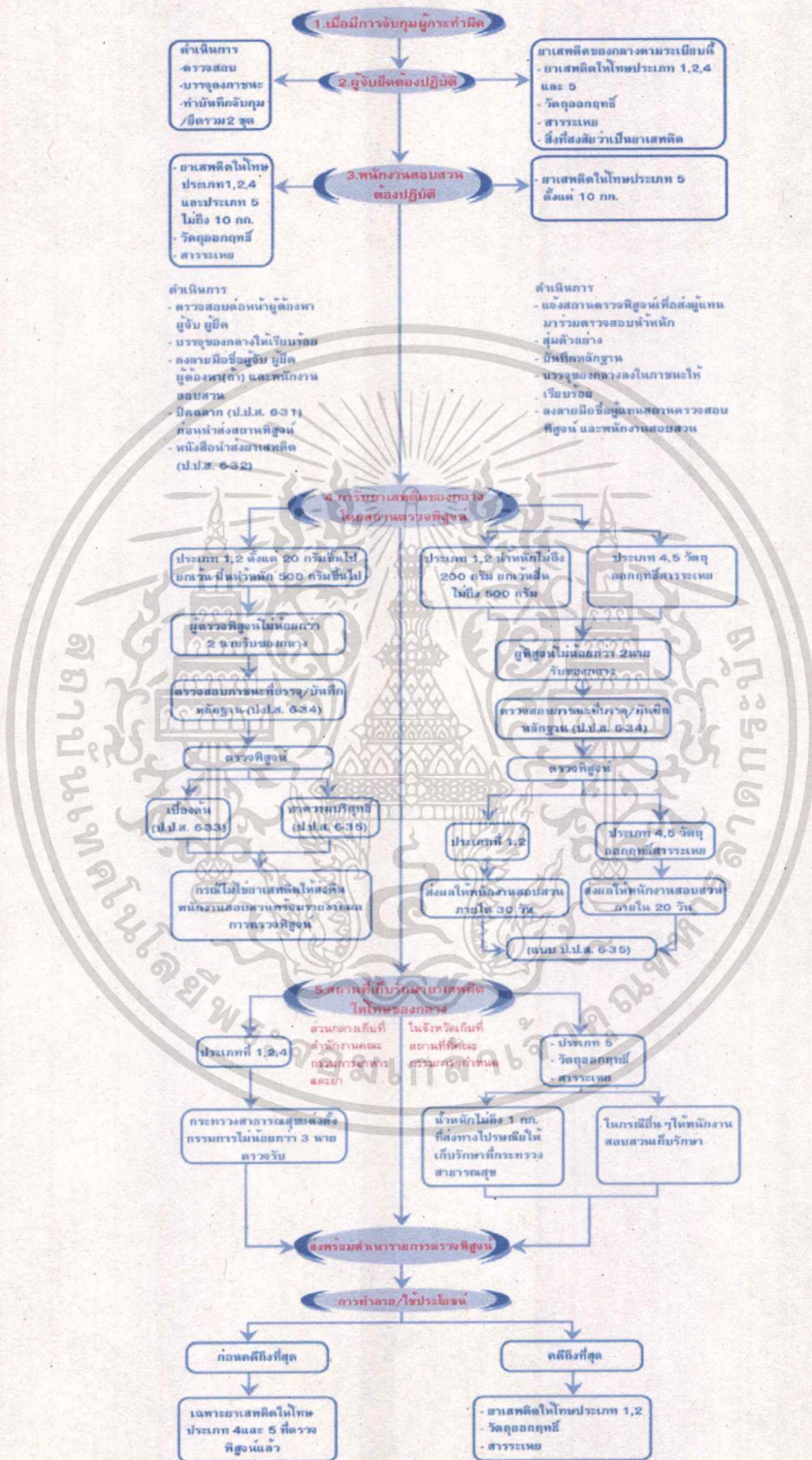
โทรสาร 0-2246-8526, 0-2247-7217

E-mail: oncb@oncb.go.th



รูปที่ 3.1 สถานที่ตั้งของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด

3.2 การวิเคราะห์ระบบงานปัจจุบัน



รูปที่ 3.2 แสดงขั้นตอนการดำเนินงานต่อยาเสพติดให้โทษของกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับยาเสพติดของกลางโดยสถานตรวจพิสูจน์

1. พนักงานสอบสวนเจ้าของคดีหรือผู้ที่ได้รับมอบหมาย ที่มียศตั้งแต่ร้อยตำรวจตรีขึ้นไป นำของกลางมาส่งยังสถานตรวจพิสูจน์ยาเสพติด พร้อมแสดงบัตรประจำตัว เว้นแต่ยาเสพติดให้โทษ ประเภท 1 หรือ 2 ที่มีน้ำหนักน้อยกว่า 20 กรัม และฝิ่นน้อยกว่า 500 กรัม ยาเสพติดให้โทษประเภท 5 น้อยกว่า 1 กิโลกรัม วัตถุออกฤทธิ์น้อยกว่า 5 กรัม หรือน้อยกว่า 200 เม็ด สารระเหยน้อยกว่า 1 กิโลกรัม ยาเสพติดให้โทษประเภท 4 พนักงานสอบสวนเจ้าของคดีจะมอบหมายให้เจ้าหน้าที่ตำรวจเป็นผู้ส่งมอบ หรือจะส่งกับการสื่อสารแห่งประเทศไทยในจดหมายไปรษณีย์ภัณฑ์ลงทะเบียนได้
2. ของกลางที่นำมาส่ง ต้องบรรจุในภาชนะที่เรียบร้อย แข็งแรง ปิดผนึก ปิดทับด้วยแบบ ป.ป.ส. 6-31
3. คณะกรรมการตรวจรับประกอบด้วย นักวิทยาศาสตร์ไม่น้อยกว่า 2 คน ในกรณีที่เป็นผู้ตรวจพิสูจน์หนึ่งคน อาจดำเนินการตรวจรับร่วมกับเจ้าหน้าที่อื่นที่เป็นนายตำรวจชั้นสัญญาบัตร หรือข้าราชการตั้งแต่ระดับ 3 ขึ้นไป
4. การตรวจสอบของกลาง ตรวจสอบต่อหน้าเจ้าหน้าที่ตำรวจผู้นำส่ง แล้วบันทึกรายละเอียดต่างๆทั้งหมด ในใบตรวจรับของกลางยาเสพติด พร้อมทั้งลงลายมือชื่อไว้เป็นหลักฐาน

การเก็บรักษาของกลาง

อนึ่ง ยาเสพติดให้โทษของกลางที่ได้ตรวจพิสูจน์แล้วว่า เป็นยาเสพติดให้โทษในประเภทที่ 1 หรือประเภทที่ 2 สถานตรวจพิสูจน์จะนำของกลางส่วนที่เหลือจากการตรวจพิสูจน์ มาส่งมอบเพื่อเก็บรักษาที่กระทรวงสาธารณสุข (ตามระเบียบสำนักนายกรัฐมนตรี ว่าด้วยการจับ ยึด และตรวจพิสูจน์ยาเสพติดพ.ศ.2537) จากนั้น สำนักงานคณะกรรมการอาหารและยา จะเป็นผู้รับมอบและเก็บรักษาไว้ที่คลังยาเสพติดให้โทษของกลาง

การทำลายของกลางยาเสพติด

ของกลางยาเสพติดที่เก็บรักษาไว้ที่สำนักงานคณะกรรมการอาหารและยา เดิมต้องเก็บรักษาไว้จนกว่าคดีถึงที่สุดแล้วจึงนำไปเผาทำลายได้ แต่ปัจจุบันยาเสพติดที่เก็บในคลังแต่ละปีมีปริมาณมากขึ้น รัฐบาล จึงได้มีการปรับกฎหมายให้เอื้อต่อการปฏิบัติ คือ ให้สามารถเผาทำลายยาเสพติดของกลางได้เมื่อศาลชั้นต้นมีคำพิพากษาคดีโดยไม่ต้องรอคดีถึงที่สุด จึงจะทำให้เผาทำลายยาเสพติดได้เร็วขึ้นอย่างน้อยปีละ 1 ครั้ง ดังนั้นต่อไปนี้ สำหรับคดียาเสพติดที่ผ่านการพิจารณาตาม

กระบวนการยุติธรรม และศาลชั้นต้นมีคำพิพากษาหรือคำสั่งให้ริบยาเสพติดให้โทษของกลางในคดี กระทรวงสาธารณสุขสามารถนำยาเสพติดดังกล่าวไปเผาทำลายต่อหน้าสาธารณชน มีสื่อมวลชน เป็นสักขีพยานในการเผาทำลายและเพื่อเป็นการป้องกันและรักษาสิ่งแวดล้อมจึงกำหนดให้เผาใน ระบบเตาเผาขยะ ณ นิคมอุตสาหกรรมบางปะอิน ซึ่งมีระบบควบคุมอากาศและการกำจัดอย่างมี ประสิทธิภาพ

การตรวจพิสูจน์ของกลางยาเสพติด

การดำเนินคดีผู้กระทำความผิดเกี่ยวกับยาเสพติดนั้น ของกลางต้องสงสัยว่าเป็นยาเสพติดที่ จับยึด ได้จากผู้ต้องหา จะต้องได้รับการยืนยันจากหน่วยงานบางหน่วยงานก่อนว่าของกลางดังกล่าว เป็นยาเสพติดผิดกฎหมายจริงตามข้อกล่าวหา จึงสามารถดำเนินการตามกระบวนการยุติธรรมได้ อย่างสมบูรณ์ หน่วยงานที่มีหน้าที่ตามกฎหมายในการบ่งชี้ว่า ของกลางนั้นเป็นยาเสพติดจริง และ สามารถออกรายงานผลการตรวจพิสูจน์ เพื่อเป็นพยานหลักฐานในการดำเนินคดีผู้กระทำผิดในชั้น ศาลได้นั้น เรียกว่า “สถานตรวจพิสูจน์” ซึ่งเป็นหน่วยงานของทางราชการ ทำหน้าที่ในการตรวจ พิสูจน์ของกลางต้องสงสัย ที่ส่งมาจากพนักงานสอบสวน โดยใช้วิธีการทางวิทยาศาสตร์ในการ ตรวจพิสูจน์

สถานตรวจพิสูจน์

หมายถึงสถานตรวจพิสูจน์ ที่กำหนดโดยคณะกรรมการป้องกันและปราบปรามยาเสพติด ตามกฎหมายว่าด้วยการป้องกันและปราบปรามยาเสพติด โดยประกาศในราชกิจจานุเบกษา ให้มี หน้าที่ตรวจพิสูจน์ของกลางต้องสงสัยว่าเป็นยาเสพติด ที่ส่งมาจากพนักงานสอบสวน และออก รายงานผลการตรวจพิสูจน์เพื่อใช้เป็นพยานหลักฐาน

ภาระหน้าที่

ตรวจพิสูจน์สารที่สงสัยว่าเป็นยาเสพติด หรือยาเสพติดตามพระราชบัญญัติป้องกันและ ปราบปรามยาเสพติด พ.ศ. 2519, พระราชบัญญัติยาเสพติดให้โทษ พ.ศ. 2522 ที่พนักงานสอบสวน ส่งให้ตรวจพิสูจน์ เพื่อต้องการรู้ว่า สารส่งสัยนั้นเป็นยาเสพติดหรือไม่ ถ้าเป็นยาเสพติดเป็น ยาเสพติดประเภท หรือชนิดใด มีจำนวน ปริมาณ และน้ำหนักเท่าใด ตลอดจนการตรวจหาปริมาณ ความบริสุทธิ์ ของสารเสพติดด้วย เพื่อนำผลที่ได้จากการตรวจพิสูจน์ไปเป็นพยานหลักฐาน

ประกอบการพิจารณาของไทยผู้กระทำได้ในชั้นศาล สถานตรวจพิสูจน์จะยึดถือระเบียบสำนักนายกรัฐมนตรีว่าด้วยการจับ ยึด และตรวจพิสูจน์ยาเสพติด พ.ศ.2537 เป็นแนวทางปฏิบัติ

จำนวนสถานตรวจพิสูจน์

ปัจจุบันมีสถานตรวจพิสูจน์ทั้งสิ้น 88 แห่ง สังกัดหน่วยราชการ 3 หน่วยงาน ได้แก่

1. สำนักงานวิทยาการตำรวจ (สวท) 73 แห่ง

- กองพิสูจน์หลักฐาน 1 แห่ง
- กองกำกับการวิทยาการเขต 12 แห่ง
- วิทยาการจังหวัด 60 แห่ง

2. กรมวิทยาศาสตร์การแพทย์ 14 แห่ง

- สำนักยาและวัตถุเสพติด 1 แห่ง
- ศูนย์วิทยาศาสตร์การแพทย์ 13 แห่ง

3. สำนักงาน ป.ป.ส. 1 แห่ง

- ส่วนวิชาการทางวิทยาศาสตร์และการตรวจพิสูจน์ยาเสพติด

การดำเนินงานของสถานตรวจพิสูจน์ สำนักงาน ป.ป.ส.

1. การตรวจรับของกลางต้องสงสัย
2. การตรวจพิสูจน์ยาเสพติด
3. การรายงานผลการตรวจพิสูจน์
4. การจัดส่งของกลาง ไปเก็บรักษา

การตรวจรับของกลางต้องสงสัย

1. สถานตรวจพิสูจน์มอบหมายให้ผู้ตรวจพิสูจน์ไม่น้อยกว่าสองคน ทำหน้าที่เวรตรวจรับของกลางในแต่ละวัน

2. เวรตรวจรับฯ จะทำการตรวจดูบัตรประจำตัวผู้นำส่ง ซึ่งเป็นพนักงานสอบสวน หัวหน้าของคดี หรือผู้ที่ได้รับมอบหมายจากพนักงานสอบสวนพร้อมตรวจดู รายละเอียดหนังสือนำส่งยาเสพติดไปตรวจพิสูจน์ (ป.ป.ส. 6-32) ว่าตรงกับ ของกลางที่นำส่งหรือไม่

3. ตรวจสอบคุณภาพของกลางต้องสงสัย นับจำนวน ชั่งน้ำหนัก โดยปฏิบัติตาม รายละเอียดที่กำหนดไว้ในระเบียบสำนักงานกฤษฎีกาว่าด้วยการ จับ ยึด และตรวจพิสูจน์ ยาเสพติด พ.ศ.2537

4. ทำรายงานการตรวจรับตามแบบ ป.ป.ส. 6-33 หรือ ป.ป.ส. 6-34 แล้วแต่กรณี จำนวนของ กลางตามที่ระเบียบกำหนด ลงลายมือชื่อผู้รับมอบ ผู้ส่งมอบ และ พยาน ไว้เป็นหลักฐาน

5. ของกลางนำไปเก็บรักษาไว้ในห้องมั่นคง เพื่อรอการตรวจพิสูจน์

การตรวจพิสูจน์ยาเสพติด

การตรวจพิสูจน์ยาเสพติด สถานตรวจพิสูจน์จะทำการตรวจพิสูจน์ของกลางยาเสพติดตาม หลักวิธีการทางวิทยาศาสตร์แบ่งการดำเนินการออกได้เป็น

1. การตรวจพิสูจน์ทางกายภาพ (Physical Examination)

เพื่อแยกแยะจัดกลุ่ม ตามรูปร่าง ลักษณะภายนอกของของกลางที่ต้องสงสัยว่า เป็นพืช, ยาง หนึบ, ผง, ผลึก, เม็ดยา หรือของเหลว มีสี, กลิ่น, ขนาด, จำนวนหรือน้ำหนักเป็นอย่างใด มีตราประทับ, เครื่องหมาย, ลักษณะการบรรจุ และภาชนะที่ใช้บรรจุแบบใด

2. การตรวจพิสูจน์ทางเคมี (Chemical Examination)

ตรวจพิสูจน์เบื้องต้นเพื่อแบ่งกลุ่มชนิดของยาเสพติด โดยทำการตรวจพิสูจน์ด้วย

- วิธี Color Test ดูการเปลี่ยนแปลงสีของน้ำยาเคมีที่เกิดปฏิกิริยากับของกลาง

- วิธี Microcrystal Test ดูลักษณะเฉพาะของพืชเสพติดบางชนิด และดูลักษณะรูปร่าง ผลึก ของตัวยาสเสพติด

3. ตรวจพิสูจน์ทางคุณภาพวิเคราะห์ (Qualitative Analysis) เพื่อให้รู้ว่าเป็นยาเสพติดหรือไม่ ถ้าใช่เป็นยาเสพติดชนิดหรือประเภทใด โดยทำการตรวจ พิสูจน์ด้วย

- วิธี Thin Layer Chromatography (TLC)

- Gas Liquid Chromatography (GC)

4. ตรวจพิสูจน์ทางปริมาณวิเคราะห์ (Quantitative Analysis) เพื่อให้รู้ว่าเป็นของกลางยาเสพติด นั้นมีปริมาณความบริสุทธิ์ของตัวยาสเสพติดเท่าใด โดยทำการตรวจพิสูจน์ด้วย

- เครื่อง Gas Liquid Chromatograph (GC)

- เครื่อง High Performance Liquid Chromatograph (HPLC)

- เครื่อง Gas Liquid Chromatograph/ Mass Spectrometer (GC/MS)

การรายงานผลการตรวจพิสูจน์

ผู้ตรวจพิสูจน์จะทำการตรวจพิสูจน์ของกลางคดีใด ต้องได้รับมอบหมายจากหัวหน้าสถานตรวจพิสูจน์นั้น เมื่อทำการตรวจพิสูจน์เสร็จแล้ว ผลการตรวจพิสูจน์จะเป็นยาเสพติด หรือไม่ใช่ยาเสพติด ผู้ตรวจพิสูจน์จะต้องทำรายงานผลการตรวจพิสูจน์ ตามแบบ ป.ป.ส. 6-35 ในรายงานผลการตรวจจะบอกให้รู้ว่า ผลการตรวจของกลางต้องสงสัยนั้น เป็นยาเสพติดหรือไม่ ถ้าเป็นยาเสพติดเป็นยาเสพติดอะไร มีจำนวน น้ำหนักสุทธิ และมีปริมาณความบริสุทธิ์ของตัวยาเสพติดเท่าใดแล้วส่งรายงานกลับไปยังพนักงานสอบสวนเจ้าของคดีเพื่อกำหนดการในขั้นต่อไป

การจัดส่งของกลางไปเก็บรักษา

เมื่อส่งรายงานผลการตรวจพิสูจน์กลับไปยังพนักงานสอบสวนแล้ว ของกลางที่เหลือจากการตรวจพิสูจน์ทั้งหมด ผู้ตรวจพิสูจน์จะทำการผนึกของกลางให้อยู่ในสภาพที่ เรียบร้อย แข็งแรง และปลอดภัย แล้วทำการส่งไปเก็บรักษาที่ สำนักงานคณะกรรมการอาหารและยา กระทรวงสาธารณสุข เพื่อรอการเผาทำลายต่อไป หากของกลางไม่ใช่ยาเสพติดให้ผู้ตรวจพิสูจน์ส่งคืนพนักงานสอบสวน พร้อมทั้งรายงานผลการตรวจพิสูจน์

หมายเหตุ:

ผู้ตรวจพิสูจน์ หมายถึงผู้ปฏิบัติหน้าที่ในสถานตรวจพิสูจน์ ที่มีความรู้ ความสามารถทำการตรวจพิสูจน์ยาเสพติด และสามารถรายงานผลการตรวจพิสูจน์ได้

น้ำหนักสุทธิ หมายถึงน้ำหนักของของกลางยาเสพติดที่ไม่รวมน้ำหนักของภาชนะบรรจุ

น้ำหนักสารบริสุทธิ์ หมายถึงน้ำหนักของตัวยาเสพติดอย่างเดี่ยว ไม่รวมสิ่งเจือปนอื่นๆ

แบบ ป.ป.ส. 6-31 แบบฉลากปิดภาชนะยาเสพติด

แบบ ป.ป.ส. 6-32 แบบหนังสือนำส่งยาเสพติดไปตรวจพิสูจน์

แบบ ป.ป.ส. 6-33 แบบการตรวจรับของกลางยาเสพติดและ

รายงานผลการตรวจพิสูจน์เบื้องต้น

แบบ ป.ป.ส. 6-34 แบบการตรวจรับของกลางยาเสพติด

แบบ ป.ป.ส. 6-35 แบบรายงานผลการตรวจพิสูจน์

ขั้นตอนการตรวจพิสูจน์ยาเสพติดทางกายภาพจากตัวอักษรบนเม็ดยา

1. เจ้าหน้าที่ตรวจพิสูจน์ทำการตรวจรับของกลางเม็ดยาบ้าที่จับกุมได้จากตำรวจท้องที่ต่างๆ
2. บันทึกข้อมูลเกี่ยวกับรายละเอียดต่าง ๆ ของคดีลงในเอกสาร
3. นำตัวอย่างเม็ดยาที่ได้รับมา ทำการถ่ายรูปภาพด้วยกล้องดิจิทัล
4. นำภาพถ่ายมาทำการตัดภาพด้วยโปรแกรม Photoshop
5. รวบรวมและวิเคราะห์รูปแบบตัวอักษรหรือเครื่องหมายต่าง ๆ บนเม็ดยา จากนั้นแบ่งกลุ่มตามรายละเอียดของตัวอักษรบนเม็ดยา แล้วกำหนดรหัสของแต่ละรูปแบบ
6. เมื่อได้รับเม็ดยาของกลางมาใหม่ ก็จะนำมาถ่ายภาพตามขั้นตอนเดิม และนำมาเปิดเทียบกับภาพฐานข้อมูลที่เก็บไว้โดยใช้การเปรียบเทียบด้วยตาเปล่าเพื่อจัดกลุ่มประเภทของเม็ดยา
7. เมื่อทราบว่าเม็ดยาที่ได้รับมาอยู่ในกลุ่มใด ก็จะบันทึกลงในเอกสารเพื่อใช้เป็นหลักฐานในการดำเนินคดีต่อไป

3.3 ปัญหาที่พบจากการดำเนินงานในปัจจุบัน

1. การถ่ายรูปภาพเม็ดยา และการตัดภาพไม่มีการกำหนดมาตรฐานที่แน่นอน
2. การจัดเก็บข้อมูลรูปภาพ ไม่มีแบบแผนในการจัดเก็บ ทำให้ยากต่อการค้นหาข้อมูล
3. การเปรียบเทียบรูปภาพด้วยตาเปล่า อาจทำให้เกิดความผิดพลาดในการแบ่งกลุ่ม
4. สิ้นเปลืองเวลาของเจ้าหน้าที่ และทรัพยากรในการทำงานซ้ำๆ กระบวนการ
5. การคัดแยกด้วยเจ้าหน้าที่แต่ละคน อาจจะใช้มาตรฐานต่างกัน ขึ้นอยู่กับการตัดสินใจ

3.4 แนวทางในการแก้ไขปัญหา

1. กำหนดแนวทางในการจัดเก็บข้อมูลให้เป็นระบบระเบียบมากขึ้น เพื่อให้ข้อมูลมีความน่าเชื่อถือและเรียกใช้งานได้ง่าย
2. วางแบบแผนในการถ่ายรูปเม็ดยาให้ไปในเป็นรูปแบบเดียวกัน เพื่อที่จะใช้เปรียบเทียบกันได้ด้วยระบบคอมพิวเตอร์
3. นำโปรแกรมคอมพิวเตอร์เข้ามาช่วยในการแยกแยะกลุ่มเม็ดยาแทนการใช้ตาเปล่า เพื่อเพิ่มความถูกต้องแม่นยำ และลดเวลาในการทำงาน

บทที่ 4

การวิเคราะห์และออกแบบระบบงานใหม่

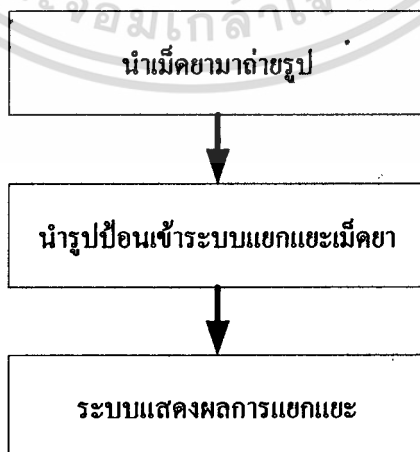
หลังจากที่ได้ทำการศึกษาและเก็บข้อมูลการทำงานของระบบงานปัจจุบันอย่างละเอียดแล้ว ทำให้ได้รายละเอียดต่างๆดังนี้

4.1 ความต้องการของระบบงานใหม่

ระบบแยกแยะเมล็ดชาที่ทำ การพัฒนานั้น จะมีการออกแบบให้ตรงกับความต้องการของผู้ใช้ ซึ่งมีรายละเอียดดังต่อไปนี้

- ระบบต้องสามารถแยกแยะกลุ่มชาบ้ำได้ถูกต้องไม่น้อยกว่า 85 เปอร์เซ็นต์
- ผู้ใช้งานต้องสามารถดูผลลัพธ์การแยกแยะภาพตัวอย่าง โดยเปรียบเทียบกับภาพในฐานข้อมูล เรียงตามลำดับความเหมือนได้
- ผลลัพธ์การแยกแยะต้องอยู่ในรูปแบบที่ง่ายต่อการทำความเข้าใจ
- ผู้ใช้งานต้องสามารถเพิ่มและแก้ไขภาพในฐานข้อมูลได้
- ระบบต้องใช้งานง่าย ไม่ซับซ้อน และไม่ขึ้นกับผู้ใดผู้หนึ่งเท่านั้น
- ระบบต้องใช้เวลาประมวลผลรวดเร็ว

4.2 แบบจำลองเชิงแนวคิดของระบบงานใหม่ (Conceptual Models)



รูปที่ 4.1 แสดงแบบจำลองเชิงแนวคิดของระบบงานใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 มีรายละเอียดในแต่ละขั้นตอนดังนี้

1. นำเม็ดยาของกลางมาถ่ายรูปด้วยกล้องดิจิทัล โดยการถ่ายภาพเม็ดยาด้วยวิธีอัดผ่งแป็งฝุ่น สีขาวลงในร่องบนเม็ดยา ถ่ายด้วยกล้องดิจิทัลที่ความละเอียด 1600x1200 จุด บนพื้นหลัง สีดำ ใช้แสงธรรมชาติ แล้วบันทึกไฟล์เพื่อเตรียมนำเข้าสู่ระบบแยกแยะเม็ดยา
2. นำรูปเม็ดยาป้อนเข้าระบบแยกแยะเม็ดยา ด้วยการใส่โปรแกรมแยกแยะเม็ดยาเข้า โดยการนำไฟล์ค่านำหนักที่ได้จากโปรแกรมฝึกสอนมาใช้ในการตัดสินใจแยกแยะกลุ่มเม็ดยา
3. ระบบแยกแยะเม็ดยาแจ้งผลการแยกแยะ โดยแสดงผลเรียงลำดับความเหมือน ผลลัพธ์จะแสดงออกมาในรูปแบบของชื่อกลุ่มเม็ดยาและแสดงด้วยรูปภาพที่ใกล้เคียงมากที่สุด 3 ลำดับ

4.3 ขอบเขตของระบบงานใหม่

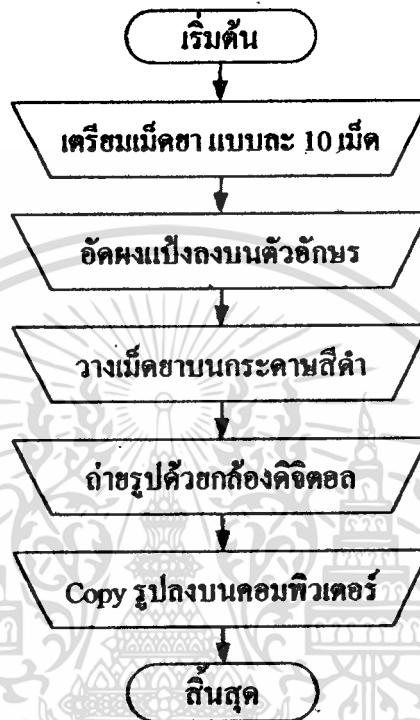
ระบบงานใหม่ที่จะทำการพัฒนานั้น เป็นการพัฒนาระบบในส่วนของการตรวจพิสูจน์เม็ดยาขาวของกลางทางด้วยภาพ โดยใช้การแยกแยะตัวอักษรที่ประทับบนเม็ดยาจากภาพถ่ายด้วยกล้องดิจิทัล โดยในขั้นแรกจะพัฒนาระบบให้สามารถแยกแยะตัวอักษร WY ก่อน เนื่องจากเป็นลักษณะเม็ดยาที่พบมากที่สุดในประเทศไทย หลังจากระบบในขั้นตอนแรกเสร็จสิ้นแล้ว จึงจะค่อยพัฒนาระบบในส่วนถัดไปให้ครอบคลุมลักษณะเม็ดยาทั้งหมด ดังนั้น ระบบงานใหม่จึงมีขอบเขตงานดังนี้

- ระบบต้องสามารถแยกแยะความแตกต่างของตัวอักษรบนเม็ดยาได้ โดยไม่ขึ้นกับสีและขนาด
- ระบบสามารถทำงานได้บนระบบปฏิบัติการวินโดวส์เท่านั้น
- ในขั้นแรกจะใช้ตัวอย่างเม็ดยาที่ประทับตัวอักษร WY เท่านั้น เนื่องจากเป็นลักษณะที่พบมากที่สุด และมีความคล้ายคลึงกันมาก ทำให้แยกแยะด้วยตาไม่สะดวก
- รูปที่นำมาแยกแยะต้องเป็นรูปภาพดิจิทัลเท่านั้น โดยอาจใช้กล้องดิจิทัลหรือสแกนเนอร์ก็ได้
- รูปภาพเม็ดยาจะต้องถ่ายในสภาพแสงที่มีความสว่างเพียงพอ เพื่อให้เกิดความแม่นยำในการแยกแยะ

หลังจากที่ได้วิเคราะห์ระบบงานใหม่แล้ว จึงทำการออกแบบระบบงานใหม่ดังหัวข้อที่ 4.4

4.4 แบบจำลองเชิงกายภาพของระบบงานใหม่ (Physical Models)

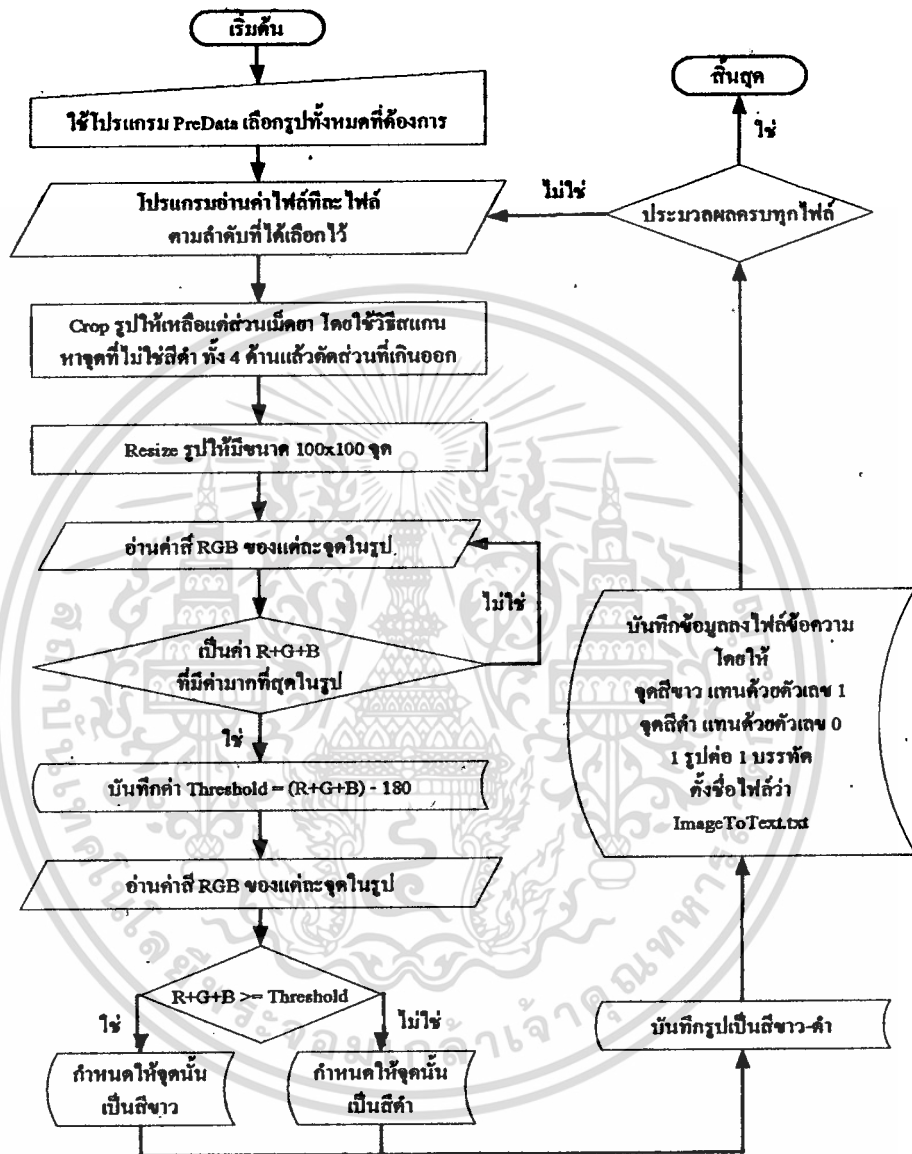
4.4.1 ขั้นตอนการเตรียมรูปเม็ดยา



รูปที่ 4.2 ขั้นตอนการเตรียมรูปเม็ดยา

จากรูปที่ 4.2 ในขั้นตอนนี้จะเป็นการเตรียมรูปเม็ดยาตัวอย่างเพื่อใช้เป็นตัวอย่างในการฝึกสอนและทดสอบระบบ โดยเตรียมรูปเม็ดยาที่ประทับตัวอักษร WY มาแบบละประมาณ 10 เม็ด ทั้งทั้งสีเขียวและสีส้ม เพื่อมาถ่ายรูป จากการทดลองพบว่าเนื่องจากเม็ดยามีขนาดเล็ก และความแตกต่างของมุมแสงจะมีผลต่อการมองเห็นร่องของตัวอักษรบนเม็ดยา ทำให้ยากต่อการแยกแยะ จึงได้ใช้วิธีอัดผงแป้งสีขาวลงในร่องบนเม็ดยา ทำให้การมองเห็นตัวอักษรชัดเจน หลังจากนั้น นำเม็ดยาวางบนกระดาษสีดำเพื่อให้่ายในการตัดขอบภาพ ในการวางเม็ดยาจะต้องวางตัวอักษรในแนวระนาบ โดยดูจากฐานของตัว W เป็นหลัก แล้วถ่ายรูปเม็ดยาด้วยกล้องดิจิทัลที่ความละเอียดภาพ 1600x1200 จุด โดยใช้แสงธรรมชาติ เมื่อถ่ายรูปครบแล้วให้ทำการบันทึกไฟล์ลงบนคอมพิวเตอร์ โดยตั้งชื่อไฟล์รูปให้ตรงกับเลขกลุ่มแล้วค่อยถ่ายด้วยลำดับของเม็ดยา เช่น 1-1-01_01, 1-1-01_02 เป็นต้น

4.4.2 ขั้นตอนการประมวลผลภาพ



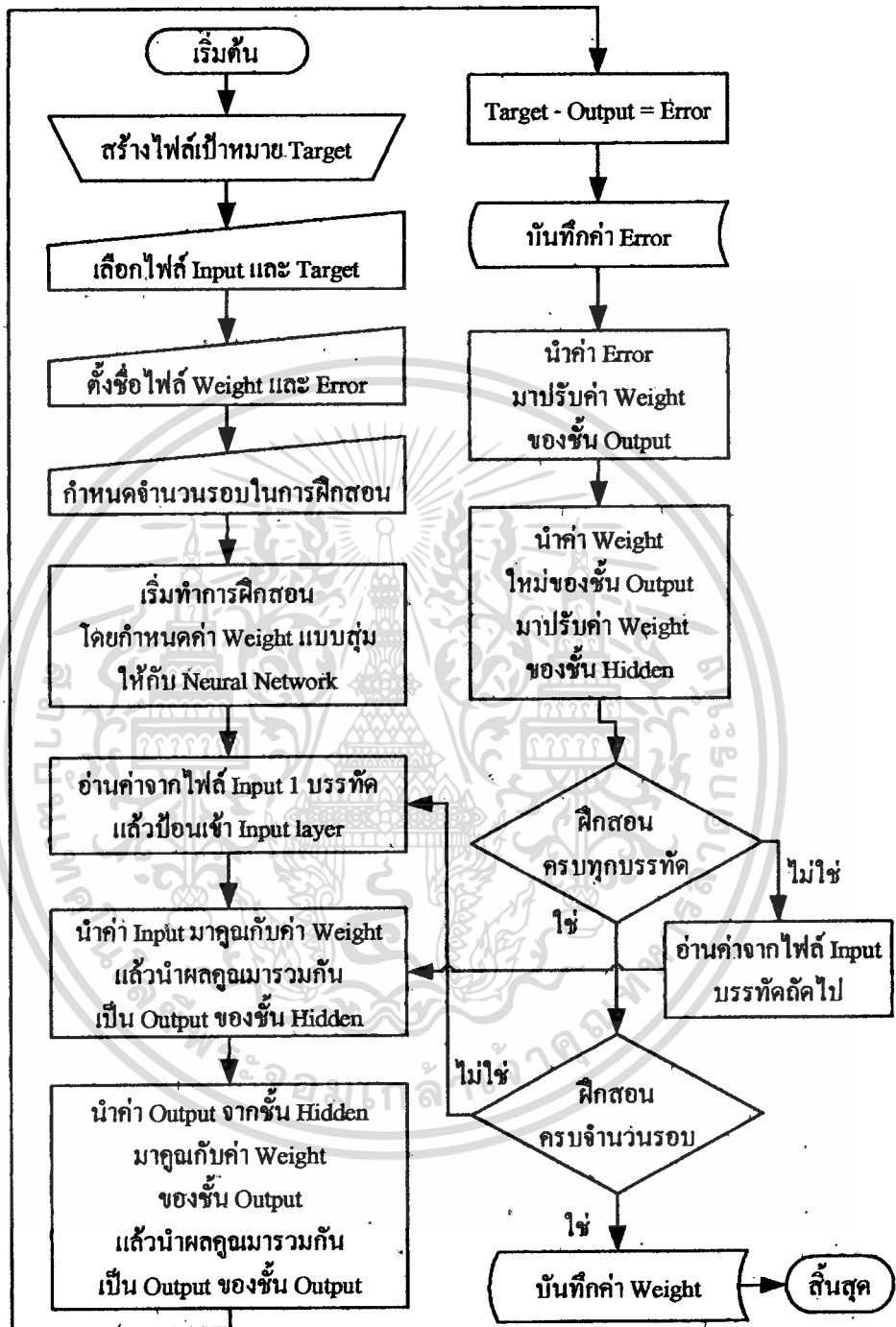
รูปที่ 4.3 ขั้นตอนการประมวลผลภาพ

จากรูปที่ 4.3 จะเป็นขั้นตอนการทำงานของโปรแกรม PreData ที่ใช้ในการประมวลผลภาพให้อยู่ในรูปแบบที่เหมาะสมกับโปรแกรมนิรโรคเน็ทเวิร์ค โดยขั้นตอนการทำงานของโปรแกรมจะเริ่มจากการเลือกรูปทั้งหมดที่ต้องการประมวลผลขึ้นมา ซึ่งโปรแกรมจะสามารถทำงานได้ครั้งละหลายๆไฟล์ในคราวเดียว เมื่อเลือกไฟล์เสร็จแล้ว โปรแกรมจะทำการเรียงลำดับชื่อไฟล์ให้โดย

อัตโนมัติ แล้วจะเริ่มทำการอ่านค่าจุดภาพใน ไฟล์แรกเข้ามาในหน่วยความจำ หลังจากนั้นจะทำการสแกนหาขอบของเม็ดชาทีละด้าน จากด้านนอกเข้ามาด้านใน แล้วคูสีของจุดภาพ ถ้าพบจุดที่ไม่ใช่สีดำก็จะทำการกำหนดว่าจุดนั้นเป็นขอบของเม็ดชา ซึ่งเม็ดชาอาจจะเป็นสีเขียวหรือสีส้ม เมื่อได้จุดขอบของเม็ดชาทั้ง 4 ด้านคือ ด้านบน ด้านล่าง ด้านซ้ายและด้านขวาแล้วก็จะทำการตัดภาพให้พอดีกับขอบเม็ดชา แล้วจะทำการปรับขนาดรูปให้มีขนาด 100x100 จุดเท่ากันทุกภาพ เพื่อให้ข้อมูลมีขนาดเท่ากันทุกภาพ เป็นการแก้ปัญหาจากการถ่ายรูปลำเอียงหรือมุมไม่เท่ากัน ไฟล์รูปที่ได้ตอนนี้จะเป็นรูปสี แต่ระบบนี้ต้องการเพียงข้อมูลของลักษณะตัวอักษรเท่านั้น ไม่ต้องการข้อมูลของสี จึงทำการแปลงรูปเป็นสีขาว-ดำ โดยให้ส่วนของตัวอักษรเป็นสีขาวและส่วนที่เหลือเป็นสีดำ การแปลงจะเริ่มจากการอ่านค่าสีแต่ละจุดจนครบทั้งรูป แล้วดูว่าจุดที่สว่างมากที่สุดมีค่าเป็นเท่าใด โดยคำนวณจากผลรวมของค่าสีแดง สีเขียวและสีน้ำเงิน ถ้าผลรวมมีค่ามากที่สุด แสดงว่าจุดนั้นเป็นจุดที่มีความสว่างมากที่สุดหรือเป็นสีขาวมากที่สุดนั่นเอง หลังจากนั้นจะกำหนดค่าเทรสโสด์ให้เท่ากับค่าผลรวมที่มีค่ามากที่สุด ลบด้วยค่า 180 ซึ่งตัวเลขนี้ได้มาจากการทดลอง ว่าเป็นค่าที่เหมาะสมที่สุดในการแยกตัวอักษรออกจากพื้นหลัง ต่อมาก็จะทำการอ่านค่าสีแต่ละจุดอีกครั้ง แล้วเปรียบเทียบกับค่าเทรสโสด์ ถ้าค่าสี R+G+B ที่จุดนั้นมีค่ามากกว่าหรือเท่ากับค่าเทรสโสด์ ก็จะกำหนดจุดนั้นให้เป็นสีขาว (R=255, G=255, B=255) ถ้ามีค่าน้อยกว่าก็จะกำหนดให้เป็นสีดำ (R=0, G=0, B=0) และบันทึกรูปเป็นไฟล์ JPEG ในขณะที่เคียวกันก็จะทำการบันทึกข้อมูลในไฟล์ข้อความด้วย โดยกำหนดให้จุดสีขาวแทนด้วยตัวเลข 1 จุดสีดำแทนด้วยตัวเลข 0 เรียงต่อกันไปจนครบทุกจุดในรูป ซึ่งมีจำนวนจุดทั้งหมด 10,000 จุด แต่จากการสังเกตรูปเม็ดชา พบว่าส่วนบนและส่วนล่างของเม็ดชาจะไม่มีข้อมูลของตัวอักษร จึงเลือกบันทึกเฉพาะช่วงกลางของตัวอักษรเพียง 6,000 จุดเท่านั้น หนึ่งรูปจะบันทึกเป็นหนึ่งบรรทัด เมื่อเสร็จทุกรูปแล้วตั้งชื่อไฟล์ว่า ImageToText.txt ไฟล์นี้จะนำไปใช้ฝึกสอนให้โปรแกรมนิเวศน์เทวีร์คต่อไป

4.4.3 ขั้นตอนการฝึกสอนระบบรู้จำ

จากรูปที่ 4.4 จะเป็นขั้นตอนการทำงานของโปรแกรม Neural ที่ใช้ในการเรียนรู้รูปเม็ดชา ตัวอย่าง การทำงานจะเริ่มจากการสร้างไฟล์เป้าหมายชื่อ Target.txt โดยใช้โปรแกรม Excel โดยให้ข้อมูลสอดคล้องกับข้อมูลรูปในไฟล์ ImageToText.txt สามารถดูตัวอย่างการสร้างไฟล์ได้จากบทที่ 5 ต่อมาจะทำการเลือกไฟล์ Input (ImageToText.txt) และ Target ให้กับโปรแกรม ตั้งชื่อไฟล์ Weight และ Error เพื่อบันทึกผลลัพธ์ที่ได้จากการฝึกสอน กำหนดจำนวนรอบในการฝึกสอน



รูปที่ 4.4 ขั้นตอนการฝึกสอนระบบรู้จำ

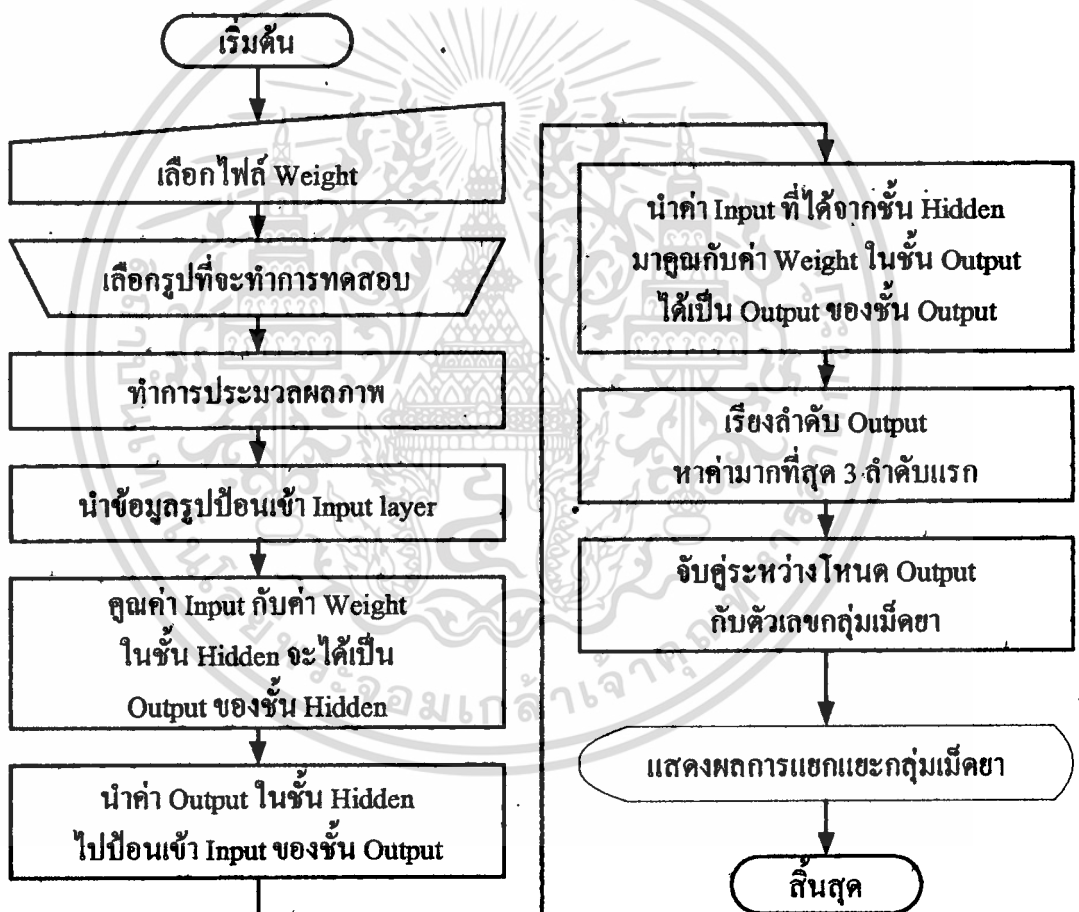
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้จะกำหนดให้เท่ากับ 500 รอบ เมื่อเตรียมข้อมูลต่างๆเสร็จแล้วก็จะเริ่มทำการฝึกสอน โดยโปรแกรมจะสร้างค่า Weight ขึ้นมาแบบสุ่ม มีค่าอยู่ในช่วง -0.5 ถึง 0.5 ให้กับแต่ละโหนดของ Neural Network ซึ่งค่า Weight ของ Hidden layer เท่ากับ Array[6000, 50] และ Output layer เท่ากับ Array[50, 13] แล้วอ่านค่าจากไฟล์ Input ครั้งละ 1 บรรทัด ตั้งแต่บรรทัดแรก ทีละตัวอักษร แล้วป้อนเข้าแต่ละโหนดของ Input layer ซึ่งข้อมูล Input ของแต่ละรูปจะมี 6,000 ตัวอักษร จึงต้องมี 6,000 Input node เพื่อรับข้อมูล แล้วนำค่าแต่ละ Input มาคูณกับค่า Weight ของแต่ละโหนด ในชั้น Hidden แล้วนำผลคูณมารวมกัน เป็น Output ของชั้น Hidden ถัดมาจะนำค่าแต่ละ Output ของแต่ละโหนดในชั้น Hidden ไปป้อนเข้าแต่ละ Input ของชั้น Output layer แล้วนำค่าแต่ละ Input จากชั้น Hidden มาคูณกับค่า Weight ของแต่ละโหนด ในชั้น Output layer แล้วนำผลคูณมารวมกันเป็น Output ของชั้น Output layer ค่า Output นี้จะนำมาลบกับค่า Target ของชั้น Output layer จะได้เป็นค่า Error ออกมา ค่า Error นี้จะบันทึกลงไฟล์และนำไปปรับค่า Weight ของชั้น Output layer ก่อนค่า Weight ใหม่ที่ได้ในชั้น Output จะนำไปใช้ในการปรับค่า Weight ของชั้น Hidden อีกทอดหนึ่ง เนื่องจากชั้น Hidden ไม่มีค่า Error การปรับค่า Weight มีเป้าหมายเพื่อให้ค่า Error ลดลง ที่จุดนี้ถือว่าการฝึกสอนให้กับรูป 1 รูป จำนวน 1 รอบ เพื่อให้โปรแกรมเรียนรู้รูปทั้งหมด ก็จะเริ่มทำการอ่านค่าในบรรทัดถัดไปแล้วผ่านขั้นตอนดังกล่าวจนครบทุกบรรทัดในไฟล์ Input ก็จะถือว่ารูปทุกรูป จำนวน 1 รอบ แต่การเรียนรู้เพียงรอบเดียวยังไม่เพียงพอ เนื่องจากการปรับค่า Weight จะยังไม่ใช่ค่าที่เหมาะสมในการแยกแยะ ดังนั้นจึงต้องฝึกสอนซ้ำหลายรอบ เพื่อให้ได้ค่า Output ใกล้เคียงกับ Target มากที่สุด เมื่อฝึกสอนครบจำนวนรอบที่ตั้งไว้แล้วก็บันทึกค่า Weight ลงใน Text file เพื่อนำไปใช้งาน

4.4.4 ขั้นตอนการทำงานของโปรแกรมแยกแยะยาบ้า

จากรูปที่ 4.5 จะเป็นขั้นตอนการทำงานของโปรแกรม Yaba ที่ใช้ในการแยกแยะกลุ่มเม็ดยา การทำงานจะเริ่มจากการเลือกไฟล์ Weight ที่ได้จากการฝึกสอนให้กับโปรแกรม เลือกรูปที่จะทำการทดสอบขึ้นมา 1 รูป โปรแกรมจะทำการประมวลผลภาพ Crop, Resize, Convert and Transform ด้วยวิธีเดียวกับข้างต้น จะได้ผลลัพธ์เป็นตัวเลข 1 หรือ 0 จำนวน 6,000 ตัว มาป้อนเข้าแต่ละโหนดของ Input layer โดย 1 โหนดจะรับค่าตัวเลข 1 ตัว ค่า Input จะถูกคูณกับค่า Weight ของแต่ละโหนด ในชั้น Hidden แล้วนำผลคูณที่ได้มาเป็น Output ของชั้น Hidden ค่า Output นี้จะนำไปป้อนเข้าแต่ละ Input ของชั้น Output แล้วนำมาคูณกับค่า Weight ในชั้น Output แล้วนำผลคูณที่

ได้มาเป็น Output ของชั้น Output ค่า Output ในแต่ละโหนดนี้จะเป็นตัวเลขทศนิยม มีค่าระหว่าง 0 ถึง 1 ตัวเลขนี้จะนำมาเรียงลำดับเพื่อหาค่าที่มากที่สุด โหนดที่มีค่าที่มากที่สุดนี้ก็คือคำตอบในการแยกแยะกลุ่มนั่นเอง ในการแสดงผล จะต้องทำการจับคู่ระหว่างโหนด Output กับตัวเลขกลุ่มเม็ดยา เพื่อให้ผู้ใช้สามารถเข้าใจผลลัพธ์ได้ การแสดงผลจะเป็นการแสดงผลเรียงลำดับตามค่า Output ที่มีค่ามากที่สุด โดยแสดงเป็นเปอร์เซ็นต์และแสดงภาพของ 3 ลำดับแรก เพื่อให้ผู้ใช้สามารถตรวจสอบรูปทดสอบกับต้นแบบในฐานะข้อมูลได้



รูปที่ 4.5 ขั้นตอนการทำงานของโปรแกรมแยกแยะยาบ้า

บทที่ 5

การสร้างและทดสอบระบบ

การสร้างและทดสอบระบบ ประกอบด้วย 4 ส่วนคือ

1. การถ่ายภาพเม็ดยา
2. โปรแกรมเตรียมข้อมูลภาพ
3. โปรแกรมฝึกสอนและทดสอบการรู้จำ
4. โปรแกรมแยกแยะเม็ดยา
5. ผลการทดสอบระบบการรู้จำ

ภาพรวมการทำงานของระบบ จะเริ่มจากการรวบรวมตัวอย่างเม็ดยาเพื่อใช้เป็นต้นแบบในการฝึกสอนให้กับ โปรแกรมรู้จำ หลังจากนั้นจะนำตัวอย่างเม็ดยามาถ่ายรูปด้วยวิธีที่จะทำให้ตัวอักษรได้ชัดเจน เสร็จแล้วจะนำรูปมาประมวลผล แล้วแปลงให้เป็นตัวเลข 1 หรือ 0 เพื่อให้สามารถป้อนเข้าโปรแกรมรู้จำแบบ Back Propagation Neural Network ได้ ก่อนการนำโปรแกรมรู้จำมาใช้งานจะต้องมีการฝึกสอนให้ระบบรู้จักกับตัวอย่างเม็ดยา เพื่อให้มีความสามารถในการแยกแยะกลุ่มเม็ดยาได้ เมื่อสิ้นสุดกระบวนการฝึกสอน จะได้ไฟล์ค่าน้ำหนักที่เหมาะสมกับกลุ่มเม็ดยาตัวอย่างออกมา ไฟล์นี้จะนำไปใช้ในการทำงานจริงต่อไป โปรแกรมทั้งหมดที่สร้างขึ้นเขียนด้วยภาษา C# บน Microsoft Visual C# 2005 Express Edition ซึ่งในแต่ละส่วนจะมีรายละเอียดดังนี้

5.1 การถ่ายภาพเม็ดยา

การถ่ายภาพเม็ดยานั้น จะถ่ายด้วยกล้องดิจิทัลที่ความละเอียดภาพ 1600x1200 จุด บนพื้นหลังสีดำ ใช้แสงธรรมชาติ จากการทดลองพบว่าเนื่องจากเม็ดยามีขนาดเล็ก และความแตกต่างของมุมแสงจะมีผลต่อการมองเห็นร่องของตัวอักษรบนเม็ดยา ทำให้ยากต่อการแยกแยะ จึงได้ใช้วิธีอัดผงแป้งฝุ่นสีขาวลงในร่องบนเม็ดยา ทำให้การมองเห็นตัวอักษรชัดขึ้น ซึ่งความแตกต่างของภาพที่ถ่ายแบบปกติกับแบบอัดผงแป้งแสดงดังรูปที่ 5.1



รูปที่ 5.1 แสดงความแตกต่างระหว่างภาพที่ถ่ายแบบปกติกับแบบอัดผงบั้ง

การฝึกสอนการรู้จำให้กับโปรแกรม จะใช้ภาพตัวอย่างของเม็ดยาแต่ละแบบประมาณ 8 ชุด และเม็ดยาที่ใช้ทดสอบในแต่ละแบบ จะใช้ประมาณ 2 ชุด ซึ่งเม็ดยาตัวอย่างที่ได้มาทั้งหมดมี 13 แบบ รวมภาพที่ถ่ายมาได้ทั้งหมด 125 ภาพ จึงนำมาแบ่งเพื่อใช้ในการฝึกสอน 100 ภาพ และใช้ทดสอบ 25 ภาพ

5.2 โปรแกรมเตรียมข้อมูลภาพ

ใช้ในการเตรียมข้อมูลภาพ โดยเริ่มจากการตัดภาพเม็ดยาที่ได้จากกล้องดิจิทัล ให้เหลือแต่ส่วนของเม็ดยา โดยใช้วิธีสแกนหาจุดที่ไม่ใช่สีดำไปที่ละด้าน เมื่อพบจุดที่ไม่ใช่สีดำจุดแรก จะทำการเช็ดจุดถัดเข้าไปด้านในเม็ดยาอีก 10 จุด เพื่อให้แน่ใจว่าจุดที่พบนั้นไม่ใช่จุดเลอะบนพื้นหลัง เมื่อได้จุดตัดทั้ง 4 ด้านคือ บน, ด้าน, ซ้าย และขวาแล้ว ก็จะทำการตัดส่วนที่เป็นพื้นหลังออกและทำการย่อภาพลงเป็นขนาด 100x100 จุด หลังจากนั้นจะทำการแปลงภาพให้เป็นสีขาวดำ โดยให้ส่วนที่เป็นตัวอักษรเป็นสีขาว และส่วนอื่นเป็นสีดำ การแปลงจะเริ่มจากการสแกนทั้งภาพเพื่อหาจุดที่มีสีขาวมากที่สุด แล้วกำหนดค่าเทรชโวลด์ของสีขาวให้น้อยกว่าจุดที่มีสีขาวมากที่สุดเท่ากับ 60 แล้วเริ่มทำการแปลงจุดสี โดยดูว่าถ้าค่า RGB ของจุดสีนั้น มีค่ามากกว่ากำหนดค่าเทรชโวลด์ของสีขาว จะกำหนดให้เป็นสีขาว ถ้ามีค่าน้อยกว่า จะกำหนดให้เป็นสีดำ ภาพขาวดำที่ได้นี้ จะเหลือเป็นรูปตัวอักษรสีขาว โดยที่ส่วนของเม็ดยาและพื้นหลัง จะกลายเป็นสีดำไปทั้งหมด อาจจะมี Noise ปนมาบ้างเล็กน้อย ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 ภาพเม็ดยาหลังจากแปลงเป็นภาพขาวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลภาพเม็ดยาที่แปลงเป็นภาพขาวดำแล้ว จะบันทึกลงไฟล์ข้อความในแฟ้มเดียวกับที่ตั้งโปรแกรม ให้ชื่อว่า ImageToText.txt 1 ภาพต่อ 1 แถวข้อความ โดยจุดสีขาวจะบันทึกเป็นเลข 1 และจุดสีดำบันทึกเป็นเลข 0 เขียนเรียงต่อกันไปเรื่อยๆจนครบทุกจุด ดังนั้นจะมีจำนวนจุดทั้งหมด 10,000 จุด เนื่องจากภาพมีขนาด 100x100 แต่จากสังเกตตัวอย่างภาพเม็ดยาทั้งหมด พบว่าภาพตัวอักษรบนเม็ดยาจะเริ่มคันที่ประมาณแถวที่ 28 และสิ้นสุดที่แถวที่ 87 ดังนั้นการทดลองนี้จึงใช้ข้อมูลตั้งแต่แถวที่ 28 ถึง 87 เท่านั้น จะเหลือจำนวนจุดทั้งหมด 6,000 จุด ทั้งนี้เพื่อเป็นการลดเวลาที่ใช้ในการประมวลผล

โปรแกรมที่ใช้ในการเตรียมข้อมูลภาพนี้คือ โปรแกรม PreData.exe ดังแสดงในรูปที่ 5.3



รูปที่ 5.3 โปรแกรมเตรียมข้อมูลภาพ

หน้าที่การทำงานของแต่ละปุ่ม มีดังนี้

1. Select File

ใช้เลือกไฟล์ที่ต้องการแปลงข้อมูล สามารถเลือกได้ครั้งละหลายๆไฟล์ ไฟล์ที่เลือกไว้จะมาแสดงอยู่ในหน้าต่าง File list ทางด้านขวา โปรแกรมจะทำการเรียงชื่อไฟล์ให้โดยอัตโนมัติ

2. Process

ใช้เริ่มกระบวนการแปลงข้อมูล โดยเริ่มจากอ่านไฟล์ภาพถ่ายเข้ามา ทำการตัดขอบภาพให้เหลือเฉพาะส่วนของเม็ดยา แล้วทำการแปลงเป็นภาพขาวดำ หลังจากนั้นจะบันทึกข้อมูลของแต่ละภาพรวมอยู่ในไฟล์ข้อความไฟล์เดียวกัน ในขณะที่โปรแกรมกำลังทำงาน ก็จะแสดงภาพการทำงานของแต่ละไฟล์ไปพร้อมกัน เพื่อช่วยในการตรวจสอบความถูกต้องของแต่ละภาพ

3. Show Picture

ใช้ในกรณีที่ต้องการดูภาพของไฟล์ที่เลือกมา เพื่อตรวจสอบความถูกต้อง โดยจะแสดงภาพที่ช่อง Original ไปเรื่อยๆ ตั้งแต่ไฟล์แรกจนถึงไฟล์สุดท้าย โดยชื่อไฟล์ที่กำลังแสดงภาพอยู่ จะปรากฏในช่อง Filename ด้านบน

4. Clear

ใช้ลบชื่อไฟล์ที่อยู่ใน File list

5. Close

ใช้ออกจากโปรแกรม

5.3 โปรแกรมฝึกสอนและทดสอบการรู้จำ

จะใช้นิวรอลเน็ตเวิร์คแบบแบ็คพรอพพาเกชันในการตัดสินใจแยกแยะกลุ่มเมล็ดยา โดยวิเคราะห์จากตัวอักษรที่ประทับบนเมล็ดยา ใช้เมล็ดยาในการทดสอบทั้งหมด 13 กลุ่ม กลุ่มละประมาณ 10 เม็ด โดยรหัสของเมล็ดยาแต่ละกลุ่มมีดังนี้คือ 1-1-05, 1-1-06, 1-1-08, 1-1-10, 1-1-15, 1-2-01, 1-2-05, 1-2-06, 1-2-07, 2-1-01, 2-1-16, 2-1-59 และ 2-2-00

ในขั้นตอนของการฝึกสอน จะใช้เมล็ดยาดัวอย่างแบบละประมาณ 8 เม็ด รวมทั้งหมด 100 ภาพ และในขั้นตอนของการทดสอบจะใช้เมล็ดยาดัวอย่างที่เหลือแบบละประมาณ 2 เม็ด รวมทั้งหมด 25 ภาพ

รายละเอียดโมดูลของนิวรอลเน็ตเวิร์คที่สร้างขึ้นมีดังนี้

Input Neurons	=	6,000
Hidden Neurons	=	50
Output Neurons	=	13
Learning Rate	=	0.1

ในการฝึกสอนและทดสอบจะใช้โปรแกรม NeuralTrain.exe ดังแสดงในรูปที่ 5.4



รูปที่ 5.4 แสดง โปรแกรมฝึกสอน

เมื่อได้ไฟล์ข้อมูลอินพุท ImageToText.txt จากโปรแกรมเตรียมข้อมูลภาพมาแล้ว ก็จะสามารถเริ่มกระบวนการฝึกสอนและทดสอบการรู้จำได้ โดยการฝึกสอนนั้น เราจะต้องกำหนดเป้าหมายให้กับโปรแกรม โดยจะเก็บไว้ในไฟล์ข้อความ ไฟล์นี้จะสร้างโดยใช้โปรแกรม Excel ตามรูปที่ 5.5 ภายในไฟล์จะประกอบด้วยตัวเลข 0 และ 1 ขนาด $M \times N$ โดยที่ M คือ จำนวนแถว ซึ่งจะเท่ากับจำนวนภาพที่นำมาฝึกสอนและ N คือจำนวนหลัก ซึ่งจะเท่ากับจำนวนเอาต์พุทของนิเวศน์ตวิร์คหรือกลุ่มของเม็ดยา ถ้าอินพุทเป็นรูปของกลุ่มไหน ค่า 1 จะอยู่ที่หลักของกลุ่มนั้น ส่วนหลักอื่น ๆ จะเป็น 0 จากการทดลองนี้ ค่าเป้าหมายจะมีจำนวน 100×13 ตัว

Microsoft Excel - Target

เพิ่ม แก้ไข มุมมอง แทรก รูปแบบ เครื่องมือ ข้อมูล หน้าต่าง วิธีใช้ Adobe PDF

A1 1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	0	0	0	0	0	0	0	0	0	0	0	0			
2	1	0	0	0	0	0	0	0	0	0	0	0	0			
3	1	0	0	0	0	0	0	0	0	0	0	0	0			
4	1	0	0	0	0	0	0	0	0	0	0	0	0			
5	1	0	0	0	0	0	0	0	0	0	0	0	0			
6	1	0	0	0	0	0	0	0	0	0	0	0	0			
7	1	0	0	0	0	0	0	0	0	0	0	0	0			
8	1	0	0	0	0	0	0	0	0	0	0	0	0			
9	1	0	0	0	0	0	0	0	0	0	0	0	0			
10	0	1	0	0	0	0	0	0	0	0	0	0	0			
11	0	1	0	0	0	0	0	0	0	0	0	0	0			
12	0	1	0	0	0	0	0	0	0	0	0	0	0			
13	0	1	0	0	0	0	0	0	0	0	0	0	0			
14	0	1	0	0	0	0	0	0	0	0	0	0	0			
15	0	1	0	0	0	0	0	0	0	0	0	0	0			
16	0	1	0	0	0	0	0	0	0	0	0	0	0			
17	0	0	1	0	0	0	0	0	0	0	0	0	0			
18	0	0	1	0	0	0	0	0	0	0	0	0	0			
19	0	0	1	0	0	0	0	0	0	0	0	0	0			
20	0	0	1	0	0	0	0	0	0	0	0	0	0			

Target / NUM

รูปที่ 5.5 แสดงการสร้างไฟล์ Target

ในการฝึกสอน จะกำหนดรอบของการฝึกสอนเท่ากับ 500 รอบ จะใช้เวลาประมาณ 45 นาที สาเหตุที่กำหนดให้หยุดสอนที่ 500 รอบ เพราะค่า Sum square error จะลดลงอย่างช้า ๆ หลังจาก 500 รอบ เมื่อฝึกสอนเสร็จแล้วจะได้ไฟล์เอ๊าท์พุท 2 ไฟล์ คือ ไฟล์ค่าน้ำหนัก และไฟล์ค่าผิดพลาด ไฟล์น้ำหนักที่ได้นี้จะเก็บไว้ใช้ในกระบวนการทดสอบต่อไป

ในขั้นตอนของการทดสอบ จะนำข้อมูลภาพเม็ดยาที่ผ่านการประมวลผลแล้ว ป้อนให้นิวรอลเน็ตเวิร์คที่ผ่านการฝึกสอนมาแล้ว จะได้ค่าออกมาจากทั้ง 13 เอ๊าท์พุทของนิวรอลเน็ตเวิร์ค เป็นตัวเลขทศนิยมมีค่าอยู่ในช่วง 0-1 ซึ่งเอ๊าท์พุทของภาพที่นิวรอลเน็ตเวิร์คจำได้จะมีค่ามากกว่าเอ๊าท์พุทตัวอื่น เอ๊าท์พุทที่ได้ออกมานี้จะถูกนำไปจับคู่เข้ากับรหัสเม็ดยาเพื่อแสดงผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบการรู้จำจะใช้โปรแกรม NeuralTrain.exe ดังแสดงในรูปที่ 5.6

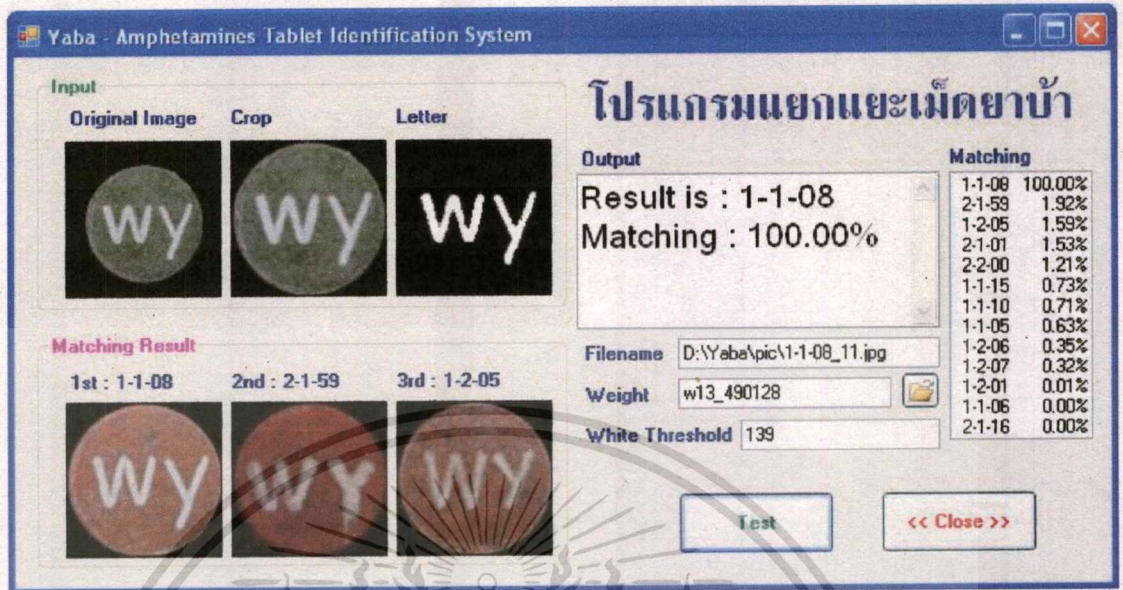
100.00	1.06	1.14	2.08	2.41	0.00	0.00	0.19	0.64	0.00	0.54	1.25	0.55	>>100.00	1-1-85
100.00	0.01	75.69	29.96	100.00	0.00	0.00	0.01	13.46	0.00	0.00	99.81	0.00	>>100.00	1-1-86
0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	99.98	99.82	0.00	>>100.00	1-1-86
0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	1.71	0.20	0.00	>>100.00	1-1-86
0.00	0.00	100.00	23.99	0.01	2.19	57.59	0.00	0.00	95.12	0.00	96.41	0.01	>>100.00	1-1-88
0.63	0.00	100.00	0.71	0.73	0.01	1.59	0.25	0.32	1.53	0.00	1.82	1.21	>>100.00	1-1-88
0.28	0.02	0.00	100.00	3.97	0.00	3.33	0.00	0.85	0.00	100.00	0.00	0.01	>>100.00	1-1-10
99.66	100.00	0.00	31.88	48.52	0.00	0.15	0.00	0.00	0.02	1.84	0.75	2.50	>>100.00	1-1-06
0.00	0.22	0.57	99.91	100.00	69.59	50.59	0.00	0.00	100.00	0.02	0.00	0.00	>>100.00	1-1-15
2.00	0.00	0.10	7.29	100.00	0.12	0.18	0.00	0.73	26.65	21.46	0.00	0.06	>>100.00	1-1-15
0.00	0.06	0.00	0.00	0.16	100.00	0.08	97.98	0.00	100.00	100.00	0.57	99.94	>>100.00	1-2-01
0.00	0.00	0.00	0.04	0.00	100.00	37.48	3.76	76.70	0.00	99.63	4.10	100.00	>>100.00	1-2-01
0.00	0.00	99.99	0.00	0.00	100.00	100.00	0.00	1.57	100.00	0.00	0.00	38.27	>>100.00	1-2-05
0.00	0.12	0.04	0.61	0.00	0.09	100.00	0.00	13.17	5.44	0.00	0.00	10.21	>>100.00	1-2-05
0.00	0.00	0.02	0.00	0.00	100.00	0.00	100.00	94.80	99.94	100.00	0.06	0.00	>>100.00	1-2-01
95.68	82.60	0.56	0.00	95.97	94.20	0.00	100.00	100.00	0.00	0.00	0.17	100.00	>>100.00	1-2-07
0.00	0.00	0.85	0.01	0.00	39.61	7.83	96.80	100.00	0.00	3.99	0.00	0.00	>>100.00	1-2-07
0.00	0.00	74.92	0.17	0.00	99.39	81.34	0.08	0.00	100.00	95.59	80.76	0.00	>>100.00	1-2-07
0.00	0.08	2.07	0.54	0.32	0.69	0.11	0.20	0.00	100.00	0.25	2.28	0.00	>>100.00	2-1-01
0.02	0.00	0.08	14.13	100.00	0.00	97.40	0.01	11.45	0.01	100.00	49.66	100.00	>>100.00	2-1-01
0.72	0.85	0.30	0.00	2.85	0.15	97.42	0.00	0.05	0.00	100.00	0.00	32.15	>>100.00	2-1-16
0.47	0.12	1.29	0.01	0.00	1.46	1.19	1.01	0.57	0.12	0.00	100.00	0.00	>>100.00	2-1-59
0.00	1.28	99.85	0.00	0.00	0.15	5.62	9.61	52.24	0.01	0.00	100.00	0.00	>>100.00	2-1-59
76.14	0.00	93.34	14.79	99.96	0.00	100.00	1.07	47.39	0.00	0.00	0.02	100.00	>>100.00	2-2-00
0.00	0.00	9.82	0.71	0.02	0.00	10.49	0.09	54.50	0.00	0.00	0.00	100.00	>>100.00	2-2-00

รูปที่ 5.6 แสดง โปรแกรมทดสอบการรู้จำ

5.4 โปรแกรมแยกแยะเมล็ดยาบ้า

โปรแกรมนี้จะเป็น โปรแกรมที่นำไปใช้ในการทำงานจริง โดยนำไฟล์ค่าน้ำหนักที่ได้จากโปรแกรมฝึกสอนมาใช้ในการตัดสินใจแยกแยะกลุ่มเมล็ดยา ผลลัพธ์การแยกแยะจะแสดงออกมาในรูปแบบของชื่อกลุ่มเมล็ดยาและแสดงด้วยรูปภาพที่ใกล้เคียงมากที่สุด 3 ลำดับแรก ดังแสดงในรูปที่

5.7



รูปที่ 5.7 โปรแกรมแยกแยะเม็ดยาบ้า

การทำงานของโปรแกรม เริ่มจากเลือกไฟล์ Weight โดยกดปุ่มทางขวาของช่อง Weight เมื่อผู้ใช้เลือกไฟล์ Weight แล้ว ชื่อไฟล์จะแสดงในช่อง Weight หลังจากนั้นให้กดปุ่ม Test โปรแกรมจะถามหาไฟล์ที่ต้องการจะทดสอบ เมื่อผู้ใช้เลือกไฟล์ได้แล้ว ชื่อไฟล์จะแสดงในช่อง Filename และโปรแกรมจะเริ่มทำงาน โดยเริ่มจากตัดขอบภาพ แปลงภาพเป็นสีขาวดำ และป้อนเข้าโปรแกรมรู้จำ เพื่อทำการวิเคราะห์ แล้วโปรแกรมจะแสดงผลลัพธ์ออกมา โดยเรียงตามลำดับความเหมือนในช่อง Matching พร้อมกับแสดงค่าที่ถูกต้องมากที่สุด ในช่อง Output และสุดท้าย จะนำภาพจากฐานข้อมูลที่ตรงกับผลลัพธ์ 3 อันดับแรกมาแสดงในช่อง Matching Result

5.5 ผลการทดสอบระบบการรู้จำ

ในการทดสอบ จะใช้ไฟล์ Weight ที่ได้จากการฝึกสอนจำนวน 500 รอบ ของตัวอย่างเม็ดยา จำนวน 13 กลุ่ม กลุ่มละประมาณ 10 รูป รวมทั้งหมด 100 รูป

การทดสอบจะแยกเป็น 2 กลุ่มคือ

1. ทดสอบด้วยรูปเดิมที่ใช้ในการฝึกสอน จำนวน 100 รูป
2. ทดสอบด้วยรูปใหม่ที่ไม่ได้ใช้ในการฝึกสอน จำนวน 25 รูป

ตารางที่ 5.1 จำนวนเม็ดยาที่ใช้ในการฝึกสอนและทดสอบ

รหัสเม็ดยา ตัวอย่าง	จำนวนเม็ดยาที่ใช้ในการ ฝึกสอน	จำนวนเม็ดยาที่ใช้ในการ ทดสอบ
1-1-05	9	2
1-1-06	7	2
1-1-08	9	2
1-1-10	7	2
1-1-15	6	2
1-2-01	8	2
1-2-05	6	2
1-2-06	6	1
1-2-07	8	2
2-1-01	8	2
2-1-16	10	2
2-1-59	8	2
2-2-00	8	2
ผลรวม	100	25

ในการทดสอบเม็ดยาครั้งละจำนวนมาก จะใช้โปรแกรมชื่อ Neural Network ซึ่งผลการทดสอบจะบันทึกในไฟล์ข้อความ ดังรูปที่ 5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File	Edit	Format	View	Help											
1.00	0.01	0.01	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	100.00	1-1-05	
1.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.00	100.00	1-1-05	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.00	100.00	1-1-06	
0.03	1.00	0.00	0.07	0.00	0.02	0.05	0.00	0.00	0.04	0.00	0.02	0.00	100.00	1-1-06	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	100.00	1-1-06	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	100.00	1-1-06	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.01	0.00	100.00	1-1-06	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	100.00	1-1-06	
0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.01	0.00	100.00	1-1-06	
0.01	0.00	1.00	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.02	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.01	100.00	1-1-08	
0.01	0.00	1.00	0.01	0.01	0.00	0.02	0.00	0.00	0.02	0.00	0.02	0.01	100.00	1-1-08	

รูปที่ 5.8 แสดงผลการทดสอบการแยกแยะกลุ่มเม็ดยาที่ได้จากโปรแกรม Neural Network

จากรูปที่ 5.8 จะเห็นว่าผลการทดสอบจะออกมาในรูปแบบของตัวเลขทศนิยม ที่มีค่าระหว่าง 0 ถึง 1 ค่าตัวเลขในแต่ละแถว ของ Column ที่ 1-13 จะบอกถึงค่าความ Matching หรือค่าความมั่นใจ ในการแยกแยะกลุ่มเม็ดยา ถ้าค่าที่มากที่สุดอยู่ใน Column ไหน จะแสดงว่าเม็ดยาที่ทำการทดสอบ มีความเหมือนกับรหัสของกลุ่มเม็ดยานั้น โดยรหัสของแต่ละกลุ่มเม็ดยาที่เป็นค่าของแต่ละ Column เรียงตามลำดับดังนี้คือ 1-1-05, 1-1-06, 1-1-08, 1-1-10, 1-1-15, 1-2-01, 1-2-05, 1-2-06, 1-2-07, 2-1-01, 2-1-16, 2-1-59 และ 2-2-00

ข้อมูลใน Column ที่ 14 เป็นการนำค่าตัวเลขความมั่นใจในการแยกแยะกลุ่มเม็ดยาที่มีค่ามากที่สุดมาแสดงในรูปแบบของเปอร์เซ็นต์ ส่วนข้อมูลใน Column ที่ 15 เป็นการแสดงผลการแยกแยะกลุ่มเม็ดยา ว่าเม็ดยาที่ทำการทดสอบมีความเหมือนกับกลุ่มตัวอย่างแบบใดมากที่สุด ซึ่งผลการทดสอบทั้งหมดมีดังนี้

ตารางที่ 5.2 ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ใช้ในการฝึกสอน แสดงด้วยค่าเฉลี่ย

ภาพ ทดสอบ	เอาร์ทูท (%)												
	1-1-05	1-1-06	1-1-08	1-1-10	1-1-15	1-2-01	1-2-05	1-2-06	1-2-07	2-1-01	2-1-16	2-1-59	2-2-00
1-1-05	100.00	1.04	1.00	1.44	1.58	0.00	0.00	0.13	0.98	0.00	1.48	1.16	0.32
1-1-06	1.33	100.00	0.00	1.12	0.07	0.57	0.70	0.44	0.00	1.84	0.47	1.10	0.03
1-1-08	0.68	0.00	100.00	0.78	0.69	0.01	1.54	0.35	0.33	1.46	0.00	1.84	1.16
1-1-10	1.43	1.72	1.08	100.00	1.04	0.00	0.82	0.00	0.75	0.71	1.41	0.50	0.26
1-1-15	2.25	0.00	1.06	1.13	100.00	2.02	0.03	0.00	0.65	1.75	2.86	0.00	1.20
1-2-01	0.00	0.46	0.51	0.03	1.44	100.00	0.93	0.78	1.40	0.48	0.34	1.26	0.51
1-2-05	0.00	0.98	2.64	1.14	0.49	0.63	100.00	0.00	1.16	0.26	0.02	1.18	2.21
1-2-06	0.00	0.13	1.11	0.00	0.40	1.64	0.01	100.00	1.08	1.32	1.31	0.81	1.29
1-2-07	0.95	0.00	1.03	0.68	0.31	0.87	1.00	1.31	100.00	0.88	0.95	0.19	0.97
2-1-01	0.00	0.84	0.83	0.78	0.72	0.89	0.08	0.87	0.00	100.00	1.56	0.97	0.00
2-1-16	1.35	0.59	0.00	1.14	1.15	0.51	0.43	0.65	0.31	0.41	100.00	0.00	0.00
2-1-59	0.62	0.14	0.95	0.00	0.00	1.48	1.28	1.09	0.50	0.12	0.00	100.00	0.00
2-2-00	0.09	0.62	0.64	0.03	0.06	0.59	0.50	1.04	0.59	0.00	0.55	0.97	100.00

ตารางที่ 5.3 ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ไม่ได้ผ่านการฝึกสอน แสดงด้วยค่าเฉลี่ย

ภาพ ทดสอบ	เอาร์ทูท (%)												
	1-1-05	1-1-06	1-1-08	1-1-10	1-1-15	1-2-01	1-2-05	1-2-06	1-2-07	2-1-01	2-1-16	2-1-59	2-2-00
1-1-05	100.00	0.54	38.42	15.97	51.21	0.00	0.00	0.10	7.05	0.00	0.27	50.03	0.28
1-1-06	0.03	100.00	0.00	97.85	0.00	0.00	0.00	0.00	0.00	50.85	49.56	0.00	87.22
1-1-08	0.32	0.00	100.00	11.90	0.37	1.10	29.59	0.18	0.16	48.33	0.00	50.17	0.61
1-1-10	49.46	50.01	0.00	69.94	26.30	0.00	1.74	0.00	0.43	0.01	50.52	0.38	1.46
1-1-15	1.00	0.11	0.34	53.60	100.00	35.05	25.38	0.00	0.37	63.33	10.75	0.00	0.03
1-2-01	0.00	0.03	0.00	0.02	0.08	100.00	18.78	50.87	38.35	50.00	99.82	2.34	99.97
1-2-05	0.00	0.06	50.01	0.31	0.00	50.05	100.00	0.00	7.37	52.72	0.00	0.00	54.24
1-2-06	0.00	0.00	0.02	0.00	0.00	100.00	0.00	100.00	94.80	99.84	100.00	0.06	0.00
1-2-07	47.84	41.30	0.71	0.01	47.99	61.96	3.92	68.40	100.00	0.00	1.55	0.09	50.00
2-1-01	0.00	0.04	38.50	0.36	0.16	50.34	45.73	0.19	0.00	100.00	47.93	41.53	0.00
2-1-16	0.37	0.33	0.04	7.07	51.48	0.07	92.41	0.01	5.75	0.01	100.00	24.83	96.08
2-1-59	0.24	0.75	50.57	0.01	0.00	0.82	3.40	5.31	26.41	0.07	0.00	100.00	0.00
2-2-00	38.07	0.00	51.58	7.75	50.01	0.00	55.22	0.58	51.22	0.00	0.00	0.01	100.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.4 ผลการทดสอบการรู้จำด้วยภาพทดสอบที่ไม่ได้ผ่านการฝึกสอน

ภาพทดสอบ	เซตทุก													% ความถูกต้อง
	1-1-05	1-1-06	1-1-08	1-1-10	1-1-15	1-2-01	1-2-05	1-2-06	1-2-07	2-1-01	2-1-16	2-1-59	2-2-00	
1-1-05	2													100
1-1-06		2												100
1-1-08			2											100
1-1-10		1		1										50
1-1-15					2									100
1-2-01						2								100
1-2-05							2							100
1-2-06	1							0						0
1-2-07									2					100
2-1-01										2				100
2-1-16											2			100
2-1-59												2		100
2-2-00													2	100
													เฉลี่ย	88.46

จากผลการทดสอบจะเห็นว่าระบบสามารถแยกแยะกลุ่มเม็ดยาจำนวน 13 กลุ่ม เป็นจำนวนรวม 25 เม็ดได้ค่าความถูกต้องเฉลี่ย 88.46% ผิดพลาด 11.54% โดยมีรายละเอียดดังนี้

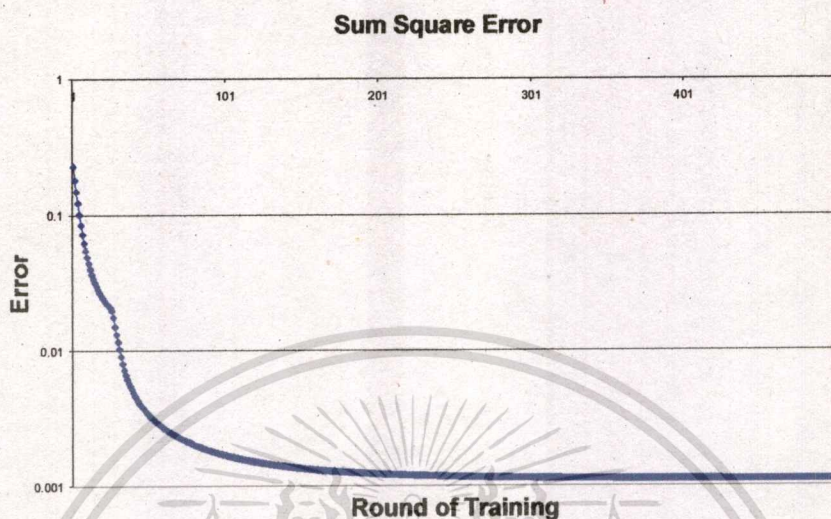
- เม็ดยาจำนวน 11 กลุ่ม จำนวน 22 เม็ด มีผลการแยกแยะถูกต้อง 100% ผิดพลาด 0%
- เม็ดยาจำนวน 1 กลุ่ม จำนวน 2 เม็ด มีผลการแยกแยะถูกต้อง 50% ผิดพลาด 50%
- เม็ดยาจำนวน 1 กลุ่ม จำนวน 2 เม็ด มีผลการแยกแยะถูกต้อง 0% ผิดพลาด 100%

สาเหตุในการแยกแยะผิดพลาดอาจเกิดจากหลายสาเหตุดังนี้

- การถ่ายรูปเม็ดยาที่อยู่ในสภาพแวดล้อมแตกต่างจากกลุ่มตัวอย่างมากเกินไป เช่นมืดเกินไป หรือสว่างเกินไป เป็นต้น
- กลุ่มตัวอย่างที่ใช้ในการฝึกสอนมีจำนวนน้อยเกินไป
- เม็ดยามีตำหนิมากเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sum Square Error



รูปที่ 5.9 แสดงค่า Sum Square Error ที่ได้จากการฝึกสอน 500 รอบ

ค่า Sum Square Error ที่ได้จากการฝึกสอน

รอบที่ 1 = 0.18403200

รอบที่ 100 = 0.00168467

รอบที่ 300 = 0.00114645

รอบที่ 500 = 0.00112210

จากรูปที่ 5.9 จะเห็นว่าค่า Sum Square Error จะลดลงอย่างรวดเร็วในช่วงแรกของการฝึกสอน และจะลดลงอย่างช้าๆ หลังจากผ่านการฝึกสอนประมาณ 200 รอบ

ค่า Error = Target - Output ดังนั้น ค่า Error ที่น้อยเป็นการแสดงว่าค่า Weight ได้ถูกปรับจนมีค่าที่เหมาะสมแล้ว นั่นคือเมื่อนำค่า Input มาคูณกับค่า Weight แล้ว ผลรวมของ Output จะได้ออกมาใกล้เคียงกับค่า Target ที่ตั้งไว้ แสดงว่าระบบพร้อมที่จะนำไปใช้งานได้แล้ว

บทที่ 6

บทสรุป

การพัฒนาาระบบแยกแยะเมล็ดชาบ้า มีวัตถุประสงค์เพื่อช่วยอำนวยความสะดวกในการทำงานของเจ้าหน้าที่ตรวจพิสูจน์ของสำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด ในการแยกแยะเมล็ดชาบ้า ช่วยลดเวลาในการทำงาน โดยนำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการวิเคราะห์แทนการใช้สายตา เป็นการเพิ่มประสิทธิภาพในการทำงานและยังช่วยรักษาสุขภาพของเจ้าหน้าที่อีกด้วย เนื่องจากหากต้องทำงานกับเมล็ดชาเป็นเวลานาน อาจทำให้ได้รับสารเคมีที่เป็นอันตราย ทำให้เกิดผลเสียต่อร่างกายในระยะยาวได้

6.1 สรุปผลการพัฒนาระบบ

จากผลการพัฒนาระบบ พบว่าระบบสามารถแยกแยะเมล็ดชาที่ประทับตัวอักษร WY ที่แตกต่างกัน 13 แบบ ได้ประมาณ 88% สามารถแยกเมล็ดชาที่มีสีแตกต่างกันได้ ทั้งสีส้มและสีเขียว โดยไม่ขึ้นกับขนาดของภาพที่ถ่ายได้ และไม่ขึ้นกับความละเอียดของภาพ ทั้งนี้ความถูกต้องในการแยกแยะ จะขึ้นอยู่กับ 2 ปัจจัยคือภาพตัวอย่างที่นำมาฝึกสอนให้โปรแกรม ควรจะมีจำนวนมากพอที่จะสอนให้โปรแกรมรู้ถึงความแตกต่างของภาพในแบบต่างๆ อีกปัจจัยคือภาพที่นำมาทำการทดสอบ จะต้องถ่ายในลักษณะที่เหมือนกันภาพตัวอย่างหรือใกล้เคียงกัน จึงจะทำให้ระบบสามารถแยกแยะได้อย่างชัดเจน

6.2 ข้อจำกัดของระบบที่พัฒนาขึ้น

- เมล็ดชาที่ระบบสามารถแยกแยะได้ ต้องนำมาอัดผงแป้งฝุ่นลงบนร่องตัวอักษรก่อนการถ่ายภาพ เพื่อให้ให้เห็นตัวอักษรชัดเจนขึ้น
- ต้องถ่ายภาพเมล็ดชาบนพื้นหลังสีดำหรือสีทึบเท่านั้น
- ระบบไม่สามารถแยกแยะภาพเมล็ดชาที่ภาพตัวอักษรอยู่ในมุมอื่นๆ ได้ เนื่องจากภาพตัวอย่างที่ใช้จะเป็นตัวอักษรที่อยู่ในแนวระนาบเท่านั้น
- ระบบไม่สามารถแยกแยะภาพเมล็ดชาที่มีคดเคี้ยวไปได้

- แบบของเม็ดยาบ้าในการทำงานจริงมีมากกว่า 13 แบบ แต่เนื่องจากยาบ้าเป็นของผิดกฎหมาย จึงไม่สะดวกในการนำเม็ดยาทั้งหมดมาถ่ายภาพเพื่อเป็นตัวอย่าง ระบบที่พัฒนาขึ้นมาจึงเป็นเพียงต้นแบบเท่านั้น หากจะนำไปใช้งานจริงจะต้องมีการฝึกสอนระบบใหม่ด้วยเม็ดยาทุกๆแบบเสียก่อน

6.3 ข้อเสนอแนะในการพัฒนาต่อ

ระบบที่พัฒนาขึ้นมาถือได้ว่ามีความสามารถในการแยกแยะระดับที่น่าพอใจ แต่ก็ยังมีบางส่วนที่สามารถพัฒนาต่อเพื่อให้ระบบมีประสิทธิภาพมากขึ้นได้ ตัวอย่างเช่น

- ปรับปรุงวิธีการหาขอบเม็ดยาให้มีความทนทานต่อ Noise บนพื้นหลังมากขึ้น เพราะหากบนพื้นหลังเอะแวม ระบบอาจจะเข้าใจผิดว่าเป็นสีของเม็ดยา อาจจะใช้วิธีตรวจหาวงกลมเข้ามาช่วย
- ปรับปรุงวิธีการแยกตัวอักษรออกจากเม็ดยา โดยวิธีที่ใช้ในระบบนี้เป็นแบบง่ายๆคือหาพื้นที่ที่เป็นสีขาว ดังนั้นหากบนเม็ดยามีผงเอะแวมอยู่ในร่องที่ไม่ใช่ส่วนของตัวอักษร ก็จะทำให้เกิดเป็น Noise ได้
- ปรับปรุงวิธีการรู้จำให้สามารถแยกแยะภาพในมุมที่แตกต่างกันได้

บรรณานุกรม

- ยุทธนา ทิลาสวัฒนกุล. 2546. คู่มือการเขียนโปรแกรมวินโดวส์ขั้นสูงด้วย Visual C++ .NET Episode One. กรุงเทพฯ: ดวงกมลสมัย.
- วิภาดา ปานอ่อน. 2547. การศึกษาลักษณะทางกายภาพของตัวอักษรบนเม็ดยาและการบรรจุของเม็ดยาบ้า. กรุงเทพฯ: สถาบันวิชาการและตรวจพิสูจน์ยาเสพติด สำนักงานคณะกรรมการป้องกันและปราบปรามยาเสพติด กระทรวงยุติธรรม.
- เสรี ปานซาง. 2540. “การรู้จำเสียงพูดคำไทยแบบไม่ขึ้นกับผู้พูดด้วยนิเวศน์เน็ตเวิร์ค.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- Zhang, C. and Wang, P. 2000. **A New Method of Color Image Segmentation Based on Intensity and Hue Clustering.** [Online]
Available: <http://www.cs.fiu.edu/~czhang/publication/icpr2000final.pdf>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code

โปรแกรมทั้งหมดเขียนด้วยภาษา C# บน Microsoft Visual C# 2005 Express Edition

โปรแกรม Neural Network

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using System.Globalization;
using System.ComponentModel;
using System.Collections.Generic;

namespace Neural
{
    public partial class Form1 : Form
    {
        const int INLAYER = 6000; // Number of Input (Pixel)
        const int OUTNEURAL = 13; // Number of Output (Type of Yaba)
        const int HIDLAYER = 1; // Number of Hidden layer
        const int HIDNEURAL = 50; // Number of Hidden neural for hidden layer1
        const double LR = 0.1; // Learning rate value
        const double THETAJ = 0.05;
        const double THETAK = 0.05;
        const double ALPHA = 0.90;
        int INRECORD = 0;

        Double[] Opi = new Double[INLAYER];
        Double[] Opj = new Double[HIDNEURAL];
        Double[] Opk = new Double[OUTNEURAL];
        Double[] Tpk = new Double[OUTNEURAL];
        Double[,] Wji = new Double[HIDNEURAL, INLAYER];
        Double[,] Wkj = new Double[OUTNEURAL, HIDNEURAL];
        Double[] Error = new Double[OUTNEURAL];
        Double[] Deltapk = new Double[OUTNEURAL];
        Double[] Deltapj = new Double[HIDNEURAL];
        Double[] SummationK = new Double[HIDNEURAL];
        Double netpj = 0, netpk = 0;
        Double[] difnetpj = new Double[HIDNEURAL];
        Double[] difnetpk = new Double[OUTNEURAL];
        Double[,] Deltabuff1 = new Double[OUTNEURAL, HIDNEURAL];
        Double[,] Deltabuff2 = new Double[OUTNEURAL, HIDNEURAL];
        Double sssqerr, ssqerr, sqerr;
        uint round = 0; uint sequence;
        int p = 0, q = 0, x = 0, y = 0;
        OpenFileDialog openFileDialog1;
        StreamReader inputfp, target;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

StreamWriter fpo, fpx;

public Form1()
{
    InitializeComponent();
    openFileDialog1.Filter = "Text files (*.txt)|*.txt";
}

private void btTrain_Click(object sender, EventArgs e)
{
    uint ep = 0;
    string strline = "";
    round = 0;
    string ProcessStart = DateTime.Now.ToString();
    //ขยายขนาดตัวอักษร
    tbData.Font = new Font(tbData.Font.Name, 30.0F, tbData.Font.Style,
    tbData.Font.Unit);

    //string directoryName = Path.GetDirectoryName(openFileDialog1.FileName);
    string directoryName = @"D:\Yaba\code\Neural";

    fpo = new System.IO.StreamWriter(directoryName + "\\ " + tbFError.Text + ".txt");
    fpo.Close();

    tbData.Clear();
    rndweight();
    INRECORD = RecordCount(directoryName + "\\ " + tbFInput.Text + ".txt");

    int EndRound = Convert.ToUInt16(tbRound.Text);
    double sssqerr_result;

    progressBar1.Maximum = EndRound + 1;
    progressBar1.Minimum = 0;
    progressBar1.Value = 0;
    progressBar1.Step = 1;

    for (round = 0; round <= EndRound; round++)
    {
        sssqerr = 0;
        inputfp = new System.IO.StreamReader(directoryName + "\\ " + tbFInput.Text +
        ".txt");
        target = new System.IO.StreamReader(directoryName + "\\ " + tbFTarget.Text +
        ".txt");

        for (sequence = 0; sequence < INRECORD; sequence++)
        {
            read_input_file();
            read_target_file();
            propagate_forward_train();
            sssqerr += sssqerr;
            compute_output_error();
            backpropagate_error();
            adjust_weightji();
        }
        sssqerr_result = sssqerr / 200;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        tbData.Text = "Training...\r\nRound : \t" + ep.ToString() + "\r\nError : \t" +
        sssqerr_result.ToString("0.00000000");
        strline = ep.ToString() + "\t" + sssqerr_result.ToString("0.00000000");
        using (StreamWriter w = File.AppendText(directoryName + "\\ " + tbFError.Text +
        ".txt"))
        {
            w.WriteLine(strline);
            w.Close();
        }
        Application.DoEvents();
        ep++;
        target.Close();
        inputfp.Close();
        progressBar1.PerformStep();
    }
    // Save Weight to file
    fpx = new System.IO.StreamWriter(directoryName + "\\ " + tbFWeight.Text + ".txt");
    keepweight();
    fpx.Close();

    tbFilename.Text = tbFInput.Text;
    tbRecord.Text = Convert.ToString(INRECORD);

    using (StreamWriter w = File.AppendText(directoryName + "\\ " + tbFError.Text +
    ".txt"))
    {
        w.WriteLine("Process Start: " + ProcessStart);
        w.WriteLine("Process Finished: " + DateTime.Now.ToString());
        w.Close();
    }
    MessageBox.Show(this, "Training Complete", "Neural Network",
    MessageBoxButtons.OK);
}

private void btTest_Click(object sender, EventArgs e)
{
    openFileDialog1.Title = "Select a Input Text File";
    tbData.Clear();

    //ขนาดขนาดตัวอักษร
    tbData.Font = new Font(tbData.Font.Name, 7F, tbData.Font.Style, tbData.Font.Unif);
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        inputfp = new System.IO.StreamReader(openFileDialog1.FileName);
        //tbData.Text = inputfp.ReadToEnd();
        tbFTest.Text = ShowFilenameFromPath(openFileDialog1.FileName);
        string directoryName = Path.GetDirectoryName(openFileDialog1.FileName);
        string OutputFilename = directoryName + "\\ " + tbFOutput.Text + ".txt";
        fpo = new System.IO.StreamWriter(OutputFilename);

        inputweight();
        INRECORD = RecordCount(directoryName + "\\ " + tbFTest.Text + ".txt");
        tbFilename.Text = ShowFilenameFromPath(openFileDialog1.FileName);
        tbRecord.Text = Convert.ToString(INRECORD);
    }
}

```

```

progressBar1.Maximum = INRECORD;
progressBar1.Minimum = 0;
progressBar1.Value = 0;
progressBar1.Step = 1;

for (int sequence = 0; sequence < INRECORD; sequence++)
{
    read_input_file();
    propagate_forward();
    fpo.Write("\r\n"); // new line for output.txt
    progressBar1.PerformStep();
}
fpo.Close();
inputfp.Close();
MessageBox.Show("Test Completed");
}

private void btRecordCount_Click(object sender, EventArgs e)
{
    tbData.Clear();
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        tbFilename.Text = ShowFilenameFromPath(openFileDialog1.FileName);
        tbRecord.Text = Convert.ToString(RecordCount(openFileDialog1.FileName));
    }
}

private void btClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void btSelInput_Click(object sender, EventArgs e)
{
    openFileDialog1.Title = "Select a File Input";
    //openFileDialog1.Filter = "Text files (*.txt)|*.txt";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        tbFInput.Text = ShowFilenameFromPath(openFileDialog1.FileName);
}

private void btSelTarget_Click(object sender, EventArgs e)
{
    openFileDialog1.Title = "Select a File Target";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        tbFTarget.Text = ShowFilenameFromPath(openFileDialog1.FileName);
}

private void btSelWeight_Click(object sender, EventArgs e)
{
    openFileDialog1.Title = "Select a File Weight";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        tbFWeight.Text = ShowFilenameFromPath(openFileDialog1.FileName);
}

```

```

public void inputweight() // Read input weight from file
{
    p = 0; q = 0; x = 0; y = 0;
    Double weightvalue;
    char[] charSeparators = new char[] { '\t' };
    String inputline;
    string[] result;
    string FILE_NAME = @"D:\Yaba\code\Neural\" + tbFWeight.Text + ".txt";
    if (!File.Exists(FILE_NAME)) { MessageBox.Show(FILE_NAME + " does not exist.");
return; }
    using (StreamReader sr = File.OpenText(FILE_NAME))
    {
        while ((inputline = sr.ReadLine()) != null)
        {
            result = inputline.Split(charSeparators, StringSplitOptions.None);

            foreach (string inputdata in result)
            {
                if (inputdata.Trim().Length > 0)
                {
                    weightvalue = Convert.ToDouble(inputdata);
                    if (q < HIDNEURAL)
                    {
                        if (x >= INLAYER)
                        {
                            x = 0; q++;
                        }
                        if (q != HIDNEURAL)
                        {
                            Wji[q, x] = weightvalue;
                            x++;
                        }
                    }
                    if (p < OUTNEURAL && q >= HIDNEURAL)
                    {
                        if (y >= HIDNEURAL)
                        {
                            y = 0; p++;
                        }
                        if (p != OUTNEURAL)
                        {
                            Wkj[p, y] = weightvalue;
                            y++;
                        }
                    }
                }
            }
        }
        sr.Close();
    }
}

public void read_input_file() // Read input data file อ่านข้อมูลมีคต
{
    String inputline;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double ival;
x = 0;
char[] charSeparators = new char[] {','};
if ((inputline = inputfp.ReadLine()) != null)
{
    char[] c = new char[inputline.Length];
    StringReader sr = new StringReader(inputline.Trim());
    sr.Read(c, 0, inputline.Length);
    foreach (char OneChar in c)
    {
        ival = Convert.ToDouble(OneChar.ToString());
        if (x >= INLAYER - 1)
        {
            Opi[x] = ival;
            x = 0;
            break;
        }
        Opi[x] = ival;
        x++;
    }
}

public void propagate_forward()
{
    int i, j, x = 0;
    String recog = "", strline = ""; ;
    Double Value, Maximum = 0, dTemp; //find out

    for (j = 0; j < HIDNEURAL; j++) // j is the number of hidden neural
    {
        netpj = 0;
        for (i = 0; i < INLAYER; i++)
            netpj += Opi[i] * Wji[j, i]; // update point
        Opj[j] = 1.0 / (1.0 + Math.Exp(-netpj + THETAJ)); // OUT=F(NET)
    }
    for (j = 0; j < OUTNEURAL; j++) // j is the number of output neural
    {
        netpk = 0;
        for (i = 0; i < HIDNEURAL; i++)
            netpk += Opj[i] * Wkj[j, i]; // NET=Opj*Wkj
        Opk[j] = 1.0 / (1.0 + Math.Exp(-netpk + THETAJ)); // OUT=F(NET)

        dTemp = Opk[j] * 100;
        strline += dTemp.ToString("#0.00") + "\t";

        fpo.Write(Opk[j].ToString("#0.00") + "\t"); // keep data to output file
        Value = Opk[j]; //find out
        if (Maximum < Value) x = j; //find out
        Maximum = (Maximum > Value) ? Maximum : Value; //find out
        switch (x)
        {
            case 0 : recog = "1-1-05"; break;
            case 1 : recog = "1-1-06"; break;
            case 2 : recog = "1-1-08"; break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case 3 : recog = "1-1-10"; break;
        case 4 : recog = "1-1-15"; break;
        case 5 : recog = "1-2-01"; break;
        case 6 : recog = "1-2-05"; break;
        case 7 : recog = "1-2-06"; break;
        case 8 : recog = "1-2-07"; break;
        case 9 : recog = "2-1-01"; break;
        case 10 : recog = "2-1-16"; break;
        case 11 : recog = "2-1-59"; break;
        case 12 : recog = "2-2-00"; break;
    }
}
dTemp = Maximum * 100;
strline += ">> " + dTemp.ToString("0.00") + "\t" + recog;
tbData.AppendText(strline + "\n");
fpo.Write(dTemp.ToString("0.00") + "\t" + recog);
}

public void rndweight() // PUT IN RANDOM WEIGHT
{
    Random autoRand = new Random();
    int inputx, weightw;

    // For hidden neurals
    for(weightw=0; weightw<HIDNEURAL; weightw++)
        for (inputx = 0; inputx < INLAYER; inputx++)
            Wjj[weightw, inputx] = 0.5 - autoRand.NextDouble();
    // For output neurals
    for(weightw=0; weightw<OUTNEURAL; weightw++)
        for(inputx=0; inputx<HIDNEURAL; inputx++)
            Wkj[weightw, inputx] = 0.5 - autoRand.NextDouble();
    // For prepare & initial Wkjbuff2 in the first time
    Array.Clear(Deltabuff2, 0, Deltabuff2.Length);
}

public void read_target_file() // Read target file
{
    x = 0;
    double ival;
    String inputline;
    char[] charSeparators = new char[] { '\t' };
    string[] result;
    if ((inputline = target.ReadLine()) != null)
    {
        result = inputline.Split(charSeparators, StringSplitOptions.None);
        foreach (string inputdata in result)
        {
            ival = System.Convert.ToDouble(inputdata);
            if (x >= OUTNEURAL - 1)
            {
                Tpk[x] = ival;
                x = 0;
                break;
            }
            Tpk[x] = ival;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x++;
    }
}

public void propagate_forward_train()
{
    int i, j;
    for(j = 0; j < HIDNEURAL; j++)
    {
        netpj = 0;
        for(i=0;i<INLAYER;i++)
            netpj += Opi[i]*Wji[j,i]; // update point
        Opj[j] = 1.0 / (1.0 + Math.Exp(-netpj + THETAJ)); // OUT=F(NET)
    }
    ssqerr=sqerr=0;
    for(j=0;j<OUTNEURAL;j++)
    {
        netpk=0;
        for(i=0;i<HIDNEURAL;i++)
            netpk += (Opj[i]*Wkj[j,i]); // NET=Opj*Wkj
        Opk[j] = 1.0 / (1.0 + Math.Exp(-netpk + THETAJ)); // OUT=F(NET)
        sqerr += ((Tpk[j]-Opk[j])*(Tpk[j]-Opk[j]));
    }
    ssqerr=sqerr/2;
}

public void compute_output_error()
{
    for(int j=0;j<OUTNEURAL;j++)
        Deltapk[j]=Opk[j]*(1.0-Opk[j])*(Tpk[j]-Opk[j]);
}

public void backpropagate_error() //backpropagate error from an upper to lower layer
{
    int i, j;
    for(i=0;i<HIDNEURAL;i++)
    {
        SummationK[i]=0;
        for(j=0;j<OUTNEURAL;j++)
            SummationK[i] += (Deltapk[j]*Wkj[j,i]);
        SummationK[i] *= (Opj[i]*(1.0-Opj[i]));
    }
}

public void adjust_weightji()
{
    int j,k;
    // adjust Wji
    for(j=0;j<HIDNEURAL;j++)
        for(k=0;k<INLAYER;k++)
            Wji[j,k] += (Opi[k]*LR*SummationK[j]);

    // adjust Wkj with option momentum

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(j=0;j<OUTNEURAL;j++)
    for(k=0;k<HIDNEURAL;k++) // prepare for momentum
        Deltabuff1[j,k]=Opj[k]*LR*Deltap[k];

for(j=0;j<OUTNEURAL;j++) // adjust Wkj
    for(k=0;k<HIDNEURAL;k++){
        Wkj[j,k]+=(Deltabuff1[j,k]+(ALPHA*Deltabuff2[j,k]));
        Deltabuff2[j,k]=Deltabuff1[j,k]; // Keep old delta Wkj for
momentum
    }
}

public void keepweight()
{
    for (int j = 0; j < HIDNEURAL; j++) // KEEP THE LAST Wji
    {
        for(int i=0;i<INLAYER;i++)
            fpx.Write(Wji[j,i].ToString("#0.000000")+ "\t"); // keep data to output file
        fpx.Write("\n"); // new line
    }
    for (int k = 0; k < OUTNEURAL; k++) // KEEP THE LAST Wkj
    {
        for(int j=0;j<HIDNEURAL;j++)
            fpx.Write(Wkj[k,j].ToString("#0.000000") + "\t"); // keep data to output file
        fpx.Write("\n"); // new line
    }
}

public int RecordCount(string FILE_NAME) //นับจำนวนแถวข้อมูลในไฟล์
{
    string inputline = "";
    INRECORD = 0;
    StreamReader sr = File.OpenText(FILE_NAME);
    while ((inputline = sr.ReadLine()) != null)
    {
        INRECORD++;
    }
    return INRECORD;
}

private string ShowFilenameFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ '};
    string[] result, fn;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    char[] charSeparatorsDot = new char[] { '.'};
    fn = result[result.Length - 1].Split(charSeparatorsDot, StringSplitOptions.None);
    return fn[0];
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม PreData

```

using System;
using System.IO;
using System.Text;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using System.ComponentModel;
using System.Collections.Generic;

namespace PreData
{
    public partial class Form1 : Form
    {
        public Bitmap image1, image2;
        public string ImageToText, toText;
        int x, y, BlackThreshold = 80;

        public Form1()
        {
            InitializeComponent();
            InitializeOpenFileDialog();
        }

        private void InitializeOpenFileDialog()
        {
            OpenFileDialog1.Multiselect = true;
            OpenFileDialog1.RestoreDirectory = true;
            OpenFileDialog1.Title = "My Image Browser";
            OpenFileDialog1.InitialDirectory = @"D:\Yaba\pic";
            OpenFileDialog1.Filter = "JPEG, Bitmap|.jpg;*.bmp";
        }

        private void btClear_Click(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
        }

        private void btProcess_Click(object sender, EventArgs e)
        {
            //prepare file output for write
            ImageToText = "";
            StreamWriter fpo = new System.IO.StreamWriter("ImageToText.txt");
            fpo.Close();

            foreach (string file in listBox1.Items)
            {

```

```

tbFilename.Text = file;
picOrg.Image = new Bitmap(file, true);
crop(file);
LetterSegment(file);
SaveImageToText();
}
MessageBox.Show("Process Completed");
}

private void crop(string file)
{
    image1 = new Bitmap(picOrg.Image);
    int cutX1 = 0, cutX2 = 0, cutY1 = 0, cutY2 = 0;
    Color PixelColor, PixelColor2;

    for (x = 0; x < image1.Width; x++) //find left of circle
    {
        for (y = 0; y < image1.Height; y++)
        {
            PixelColor = image1.GetPixel(x, y);
            if (PixelColor.R + PixelColor.G + PixelColor.B > BlackThreshold)
            {
                PixelColor2 = image1.GetPixel(x + 10, y); //check next point to make sure it's
                not noise
                if (PixelColor2.R + PixelColor2.G + PixelColor2.B > BlackThreshold)
                {
                    cutX1 = x; //keep left of circle
                    x = image1.Width; //stop loop for
                    y = image1.Height;
                }
            }
        }
    }

    for (x = image1.Width - 1; x > 0; x--) //find right of circle
    {
        for (y = 0; y < image1.Height; y++)
        {
            PixelColor = image1.GetPixel(x, y);
            if (PixelColor.R + PixelColor.G + PixelColor.B > BlackThreshold)
            {
                PixelColor2 = image1.GetPixel(x - 10, y);
                if (PixelColor2.R + PixelColor2.G + PixelColor2.B > BlackThreshold)
                {
                    cutX2 = x; //keep right of circle
                    x = 0; //stop loop for
                    y = image1.Height;
                }
            }
        }
    }

    for (y = 0; y < image1.Height; y++) //find top of circle
    {
        for (x = 0; x < image1.Width; x++)
        {
            PixelColor = image1.GetPixel(x, y);

```

```

if (PixelColor.R + PixelColor.G + PixelColor.B > BlackThreshold)
{
    PixelColor2 = image1.GetPixel(x, y + 10);
    if (PixelColor2.R + PixelColor2.G + PixelColor2.B > BlackThreshold)
    {
        cutY1 = y; //keep top of circle
        x = image1.Width; //stop loop for
        y = image1.Height;
    }
}
}
}
for (y = image1.Height - 1; y > 0; y--) //find bottom of circle
{
    for (x = 0; x < image1.Width; x++)
    {
        PixelColor = image1.GetPixel(x, y);
        if (PixelColor.R + PixelColor.G + PixelColor.B > BlackThreshold)
        {
            PixelColor2 = image1.GetPixel(x, y - 10);
            if (PixelColor2.R + PixelColor2.G + PixelColor2.B > BlackThreshold)
            {
                cutY2 = y; //keep bottom of circle
                x = image1.Width; //stop loop for
                y = 0;
            }
        }
    }
}

//ImageResize
Bitmap bitmap = new Bitmap(100, 100, image1.PixelFormat);
Graphics new_g = Graphics.FromImage(bitmap);
new_g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;
new_g.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;

// Create parallelogram for drawing image.
Point ulCorner = new Point(0, 0);
Point urCorner = new Point(100, 0);
Point llCorner = new Point(0, 100);
Point[] destPara = { ulCorner, urCorner, llCorner };

// Create rectangle for source image.
Rectangle srcRect = new Rectangle(cutX1, cutY1, cutX2 - cutX1, cutY2 - cutY1);
GraphicsUnit units = GraphicsUnit.Pixel;
new_g.DrawImage(image1, destPara, srcRect, units);
bitmap.Save(ShowFilenameFromPath(file) + "_crop.jpg",
System.Drawing.Imaging.ImageFormat.Jpeg);
picCrop.Image = bitmap;
new_g.Dispose();
}

private void LetterSegment(string file)
{

```

```

image1 = new Bitmap(picCrop.Image);
image2 = new Bitmap(image1.Width, image1.Height, image1.PixelFormat);
Color WhiteColor = Color.White;
Color BlackColor = Color.Black;
int ColorSum, MaxWhite = 0;
Color MaxWhitePixel = Color.Black, PixelColor;

for (x = 0; x < image1.Width; x++)
{
    for (y = 0; y < image1.Height; y++)
    {
        PixelColor = image1.GetPixel(x, y);
        ColorSum = PixelColor.R + PixelColor.G + PixelColor.B;
        if (ColorSum > MaxWhite)
        {
            MaxWhitePixel = PixelColor;
            MaxWhite = ColorSum;
        }
    }
}
int WhiteThreshold = Convert.ToInt16(MaxWhite/3 - 60);
tbThreshold.Text = WhiteThreshold.ToString();

// Mark letter is white, other is black
for (x = 0; x < image1.Width; x++)
{
    for (y = 0; y < image1.Height; y++)
    {
        PixelColor = image1.GetPixel(x, y);
        //if RGB more than WhiteThreshold, mark it's white.
        if (PixelColor.R > WhiteThreshold && PixelColor.G > WhiteThreshold &&
PixelColor.B > WhiteThreshold)
        {
            image2.SetPixel(x, y, WhiteColor);
            if (y > 27 && y < 88)
                ImageToText += "1";
        }
        else
        {
            image2.SetPixel(x, y, BlackColor);
            if (y > 27 && y < 88)
                ImageToText += "0";
        }
    }
}
picLetter.Image = image2;
image2.Save(ShowFilenameFromPath(file) + "_letter.gif",
System.Drawing.Imaging.ImageFormat.Gif);
}

private void SaveImageToText()
{
    using (StreamWriter w = File.AppendText("ImageToText.txt"))
    {
        w.WriteLine(ImageToText);
    }
}

```

```

        w.Close();
    }
    ImageToText = "";
    Application.DoEvents();
}

private void btSelMulFile_Click(object sender, EventArgs e)
{
    if (OpenFileDialog1.ShowDialog() == DialogResult.OK)
    {
        this.Activate();
        string[] files = OpenFileDialog1.FileNames;

        Array.Sort(files, 0, files.Length);

        foreach (string file in files)
        {
            listBox1.Items.Add(file);
        }
    }
}

private void btShowPic_Click(object sender, EventArgs e)
{
    this.Activate();
    foreach (string file in listBox1.Items)
    {
        tbFilename.Text = file;
        System.IO.FileInfo fileInfo = new System.IO.FileInfo(file);
        System.IO.FileStream fileStream = fileInfo.OpenRead();
        picOrg.Image = System.Drawing.Image.FromStream(fileStream);
        Application.DoEvents();
        fileStream.Close();
        //Call Sleep so the picture is briefly displayed, which will create a slide-show effect.
        System.Threading.Thread.Sleep(1000);
    }
}

private string ShowFilenameFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ ' };
    string[] result, fn;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    char[] charSeparatorsDot = new char[] { '.' };
    fn = result[result.Length - 1].Split(charSeparatorsDot, StringSplitOptions.None);
    return fn[0];
}

private string ShowExtFileFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ ' };
    string[] result, ext;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    char[] charSeparatorsDot = new char[] { '.' };
    ext = result[result.Length - 1].Split(charSeparatorsDot, StringSplitOptions.None);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return "." + ext[1];
}

private string ShowDirectoryFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\' };
    string[] result;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    string Dir = "";
    for (int x = 0; x < result.Length - 1; x++)
    {
        Dir += result[x] + "\\";
    }
    return Dir;
}

private void btClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Yaba

```

using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Yaba
{
    public partial class Yaba : Form
    {
        public Bitmap image1, image2;
        public string inputtext, toText;
        int x, y, BlackThreshold = 80;

        const int INLAYER = 6000; // Number of Input (Pixel)
        const int OUTNEURAL = 13; // Number of Output (Type of Yaba)
        const int HIDLAYER = 1; // Number of Hidden layer
        const int HIDNEURAL = 50; // Number of Hidden neural for hidden layer1
        const double LR = 0.1; // Learning rate value
        const double THETAJ = 0.05;
        const double THETAJ = 0.05;
        const double ALPHA = 0.90;
        Double[] Opi = new Double[INLAYER];
        Double[] Opj = new Double[HIDNEURAL];
        Double[] Opk = new Double[OUTNEURAL];
        Double[] Tpk = new Double[OUTNEURAL];
        Double[,] Wji = new Double[HIDNEURAL, INLAYER];
        Double[,] Wkj = new Double[OUTNEURAL, HIDNEURAL];
        Double[] Error = new Double[OUTNEURAL];
        Double[] Deltapk = new Double[OUTNEURAL];
        Double[] Deltapj = new Double[HIDNEURAL];
        Double[] SummationK = new Double[HIDNEURAL];
        Double netpj = 0, netpk = 0;
        Double[] difnetpj = new Double[HIDNEURAL];
        Double[] difnetpk = new Double[OUTNEURAL];
        Double[,] Deltabuff1 = new Double[OUTNEURAL, HIDNEURAL];
        Double[,] Deltabuff2 = new Double[OUTNEURAL, HIDNEURAL];
        int p = 0, q = 0;

        public Yaba()
        {
            InitializeComponent();
            InitializeOpenFileDialog();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

private void InitializeOpenFileDialog()
{
    OpenFileDialog1.Multiselect = false;
    OpenFileDialog1.RestoreDirectory = true;
    OpenFileDialog1.Title = "My Image Browser";
    OpenFileDialog1.InitialDirectory = @"D:\Yaba\pic";
    OpenFileDialog1.Filter = "JPEG, Bitmap|.jpg;*.bmp";
}

private void btTest_Click(object sender, EventArgs e)
{
    tbData.Clear();
    showResult.Items.Clear();
    Application.DoEvents();
    InitializeOpenFileDialog();
    if (OpenFileDialog1.ShowDialog() == DialogResult.OK)
    {
        this.Activate();
        string file = OpenFileDialog1.FileName;
        tbFilename.Text = file;

        //prepare file output for write
        inputtext = "";
        string TestTextFile = ShowDirectoryFromPath(file) + ShowFilenameFromPath(file)
+ ".txt";

        picOrg.Image = new Bitmap(file, true);
        crop(file);
        LetterSegment(file);

        //NeuralNetwork
        inputweight();
        propagate_forward();
    }
}

private void crop(string file)
{
    image1 = new Bitmap(picOrg.Image);
    int cutX1 = 0, cutX2 = 0, cutY1 = 0, cutY2 = 0;
    Color PixelColor, PixelColor2;

    for (x = 0; x < image1.Width; x++) //find left of circle
    {
        for (y = 0; y < image1.Height; y++)
        {
            PixelColor = image1.GetPixel(x, y);
            if (PixelColor.R + PixelColor.G + PixelColor.B > BlackThreshold)
            {
                PixelColor2 = image1.GetPixel(x + 10, y); //check next point to make sure it's
not noise
                if (PixelColor2.R + PixelColor2.G + PixelColor2.B > BlackThreshold)
            {

```



```

    }
}

//ImageResize
Bitmap bitmap = new Bitmap(100, 100, image1.PixelFormat);
Graphics new_g = Graphics.FromImage(bitmap);
new_g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;
new_g.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;

// Create parallelogram for drawing image.
Point ulCorner = new Point(0, 0);
Point urCorner = new Point(100, 0);
Point llCorner = new Point(0, 100);
Point[] destPara = { ulCorner, urCorner, llCorner };

// Create rectangle for source image.
Rectangle srcRect = new Rectangle(cutX1, cutY1, cutX2 - cutX1, cutY2 - cutY1);
GraphicsUnit units = GraphicsUnit.Pixel;
new_g.DrawImage(image1, destPara, srcRect, units);
bitmap.Save(ShowFilenameFromPath(file) + "_crop.jpg",
System.Drawing.Imaging.ImageFormat.Jpeg);
picCrop.Image = bitmap;
new_g.Dispose();
}

private void LetterSegment(string file)
{
    image1 = new Bitmap(picCrop.Image);
    image2 = new Bitmap(image1.Width, image1.Height, image1.PixelFormat);
    Color WhiteColor = Color.White;
    Color BlackColor = Color.Black;
    int ColorSum, MaxWhite = 0;
    Color MaxWhitePixel = Color.Black, PixelColor;

    for (x = 0; x < image1.Width; x++)
    {
        for (y = 0; y < image1.Height; y++)
        {
            PixelColor = image1.GetPixel(x, y);
            ColorSum = PixelColor.R + PixelColor.G + PixelColor.B;
            if (ColorSum > MaxWhite)
            {
                MaxWhitePixel = PixelColor;
                MaxWhite = ColorSum;
            }
        }
    }

    int WhiteThreshold = Convert.ToInt16(MaxWhite / 3 - 60);
    tbThreshold.Text = WhiteThreshold.ToString();

    // Mark letter is white, other is black
    int i = 0;
    for (x = 0; x < image1.Width; x++)

```

```

{
    for (y = 0; y < image1.Height; y++)
    {
        PixelColor = image1.GetPixel(x, y);
        //if RGB more than WhiteThreshold, mark it's white.
        if (PixelColor.R > WhiteThreshold && PixelColor.G > WhiteThreshold &&
PixelColor.B > WhiteThreshold)
        {
            image2.SetPixel(x, y, WhiteColor);
            if (y > 27 && y < 88)
            {
                Opi[i] = 1.0;
                i++;
            }
        }
        else
        {
            image2.SetPixel(x, y, BlackColor);
            if (y > 27 && y < 88)
            {
                Opi[i] = 0.0;
                i++;
            }
        }
    }
}
picLetter.Image = image2;
image2.Save(ShowFilenameFromPath(file)+"_letter.gif",
System.Drawing.Imaging.ImageFormat.Gif);
}

private string ShowFilenameFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ };
    string[] result, fn;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    char[] charSeparatorsDot = new char[] { '.' };
    fn = result[result.Length - 1].Split(charSeparatorsDot, StringSplitOptions.None);
    return fn[0];
}

private string ShowExtFileFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ };
    string[] result, ext;
    result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
    char[] charSeparatorsDot = new char[] { '.' };
    ext = result[result.Length - 1].Split(charSeparatorsDot, StringSplitOptions.None);
    return "." + ext[1];
}

private string ShowDirectoryFromPath(string path)
{
    char[] charSeparatorsSlash = new char[] { '\\ };
    string[] result;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

result = path.Split(charSeparatorsSlash, StringSplitOptions.None);
string Dir = "";
for (int x = 0; x < result.Length - 1; x++)
{
    Dir += result[x] + "\\";
}
return Dir;
}

private void btClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}

public void inputweight() // Read input weight from file
{
    p = 0; q = 0; x = 0; y = 0;
    Double weightvalue;
    char[] charSeparators = new char[] { '\t' };
    String inputline;
    string[] result;
    string FILE_NAME = tbFWeight.Text + ".txt";

    if (!File.Exists(FILE_NAME)) { MessageBox.Show(FILE_NAME + " does not exist.");
return; }
    using (StreamReader sr = File.OpenText(FILE_NAME))
    {
        while ((inputline = sr.ReadLine()) != null)
        {
            result = inputline.Split(charSeparators, StringSplitOptions.None);
            foreach (string inputdata in result)
            {
                if (inputdata.Trim().Length > 0)
                {
                    weightvalue = Convert.ToDouble(inputdata);
                    if (q < HIDNEURAL)
                    {
                        if (x >= INLAYER)
                        {
                            x = 0; q++;
                        }
                        if (q != HIDNEURAL)
                        {
                            Wji[q, x] = weightvalue;
                            x++;
                        }
                    }
                    if (p < OUTNEURAL && q >= HIDNEURAL)
                    {
                        if (y >= HIDNEURAL)
                        {
                            y = 0; p++;
                        }
                        if (p != OUTNEURAL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

lbM1.Text = "1st : " + type[TopRank.Length - 1];
lbM2.Text = "2nd : " + type[TopRank.Length - 2];
lbM3.Text = "3rd : " + type[TopRank.Length - 3];
}

private void btSelWeight_Click(object sender, EventArgs e)
{
    OpenFileDialog1.Multiselect = false;
    OpenFileDialog1.RestoreDirectory = true;
    OpenFileDialog1.InitialDirectory = @"D:\Yaba\code\Yaba\Yaba\Yaba\bin\Debug";
    OpenFileDialog1.Filter = "Text files (*.txt)|*.txt";
    OpenFileDialog1.Title = "Select a File Weight";
    if (OpenFileDialog1.ShowDialog() == DialogResult.OK)
        tbFWeight.Text = ShowFileNameFromPath(OpenFileDialog1.FileName);
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานระบบ

การใช้งานระบบ ประกอบด้วย 4 ส่วนคือ

1. การเตรียมเม็ดยา
2. โปรแกรมเตรียมข้อมูลภาพ
3. โปรแกรมฝึกสอนและทดสอบการรู้จำ
4. โปรแกรมแยกแยะเม็ดยา

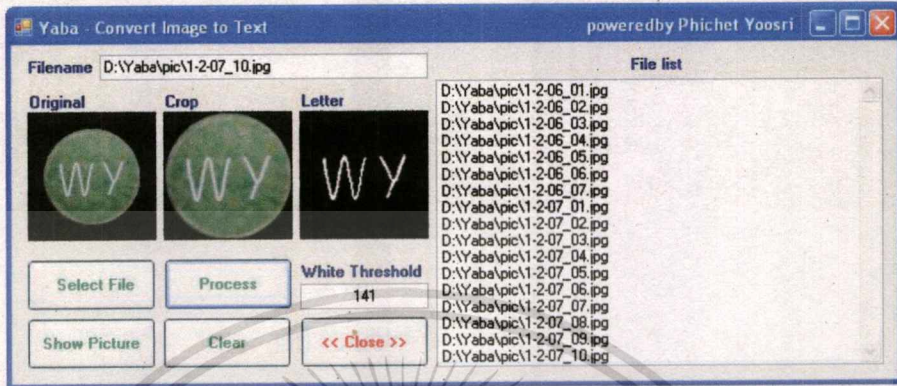
1. การเตรียมเม็ดยา

ขั้นตอนแรกในการใช้งานระบบ จะเริ่มจากการรวบรวมตัวอย่างเม็ดยาที่มีทุกแบบมาทำการถ่ายภาพ แบบละประมาณ 10 เม็ด โดยการถ่ายภาพเม็ดยานั้น จะต้องนำเม็ดยาทั้งหมดมาอัดผงแป้งฝุ่นสีขาวลงในร่องบนเม็ดยา เพื่อให้การมองเห็นตัวอักษรชัดเจนขึ้น ซึ่งความแตกต่างของภาพที่ถ่ายแบบปกติกับแบบอัดผงแป้งแสดงดังรูปที่ 1 แล้วทำการถ่ายภาพด้วยกล้องดิจิทัลที่ความละเอียดภาพ 1600x1200 จุด บนพื้นหลังสีดำ ใช้แสงธรรมชาติ รูปด้านล่างแสดงความแตกต่างระหว่างภาพที่ถ่ายแบบปกติกับแบบอัดผงแป้ง

เมื่อถ่ายรูปครบแล้ว ให้บันทึกภาพลงใน Folder เดียวกัน โดยตั้งชื่อไฟล์ตามกลุ่มเม็ดยาแล้วตามด้วยเลขลำดับ เพื่อให้ง่ายในการเรียงลำดับ เช่น ถ้ารหัสของเม็ดยาคือ 1-1-01 ให้ตั้งชื่อว่า 1-1-01_xx โดยที่ xx คือลำดับภาพ เช่น 01, 02, ..., 10

ถ้าภาพมีขนาดใหญ่เกินไป สามารถใช้โปรแกรม ACDsee หรือ Photoshop ทำการ Crop ภาพให้มีขนาดเล็กกลงได้ โดยตัดให้ภาพมีขนาดใหญ่กว่าเม็ดยาเล็กน้อย เพื่อประหยัดพื้นที่ในการจัดเก็บ

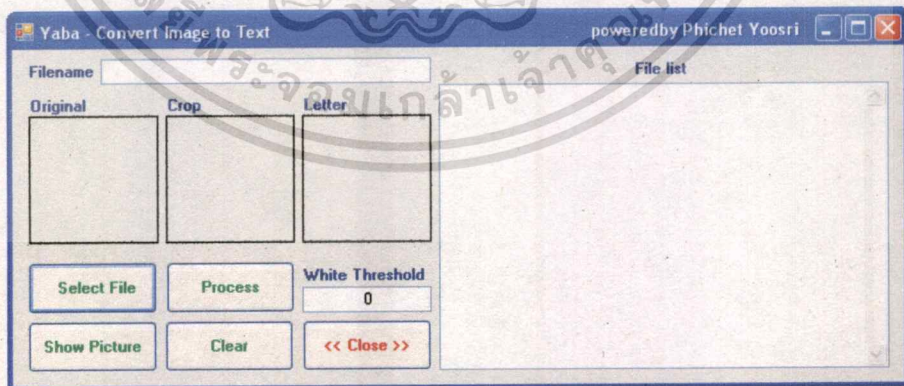
2. โปรแกรมเตรียมข้อมูลภาพ



เมื่อได้รูปทั้งหมดมาแล้ว ก็จะใช้โปรแกรม PreData.exe เพื่อเตรียมข้อมูลภาพ โดยโปรแกรมจะทำการตัดภาพเม็ดยาที่ได้จากกล้องดิจิทัลให้เหลือแต่ส่วนของเม็ดยา แล้วทำการย่อภาพลงเป็นขนาด 100x100 จุด หลังจากนั้นจะทำการแปลงภาพให้เป็นสีขาวดำ โดยให้ส่วนที่เป็นตัวอักษรเป็นสีขาว และส่วนอื่นเป็นสีดำ เสร็จแล้วจะทำการแปลงข้อมูลรูปภาพทั้งหมดให้เป็นตัวเลขแล้วบันทึกในไฟล์ข้อความ เพื่อนำไปใช้กับโปรแกรมแยกแยะภาพเม็ดยาต่อไป

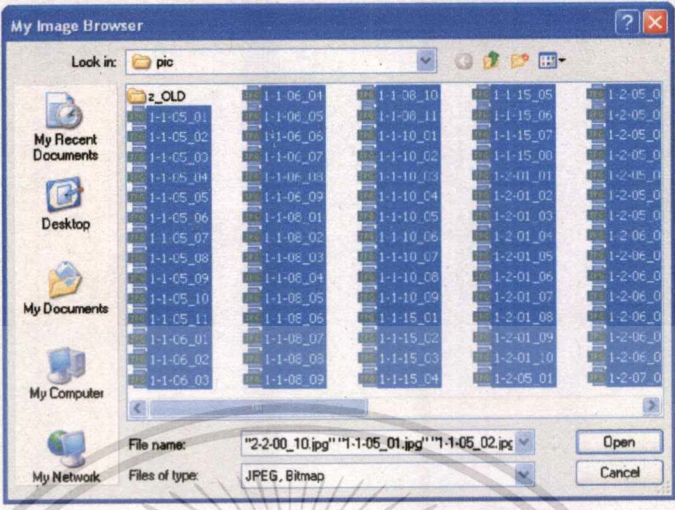
ขั้นตอนการใช้งานโปรแกรมเตรียมข้อมูลภาพ

1. เปิดโปรแกรม PreData.exe ขึ้นมา จะพบหน้าจอดังรูป

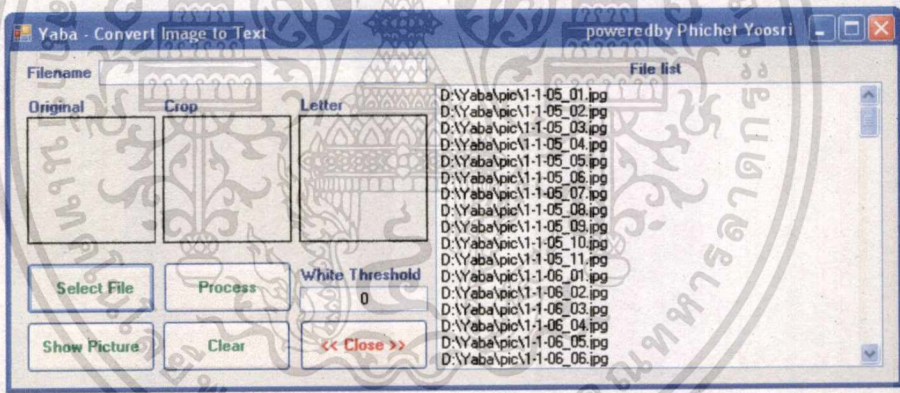


2. กดปุ่ม Select File เพื่อเลือกไฟล์ตัวอย่างทั้งหมดขึ้นมา สามารถเลือกได้ครั้งละหลายๆไฟล์ เมื่อเลือกไฟล์ทั้งหมดแล้ว กดปุ่ม Open

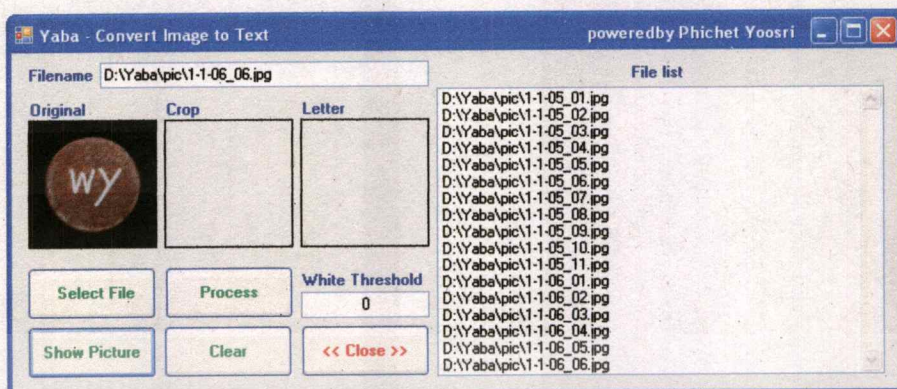
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



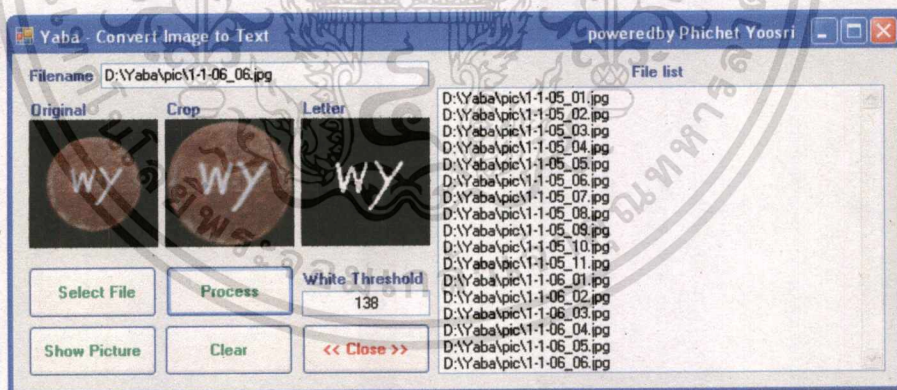
3. จะเห็นว่าชื่อไฟล์ทั้งหมดปรากฏอยู่ใน File list โดยจะเรียงลำดับตามชื่อไฟล์ให้โดยอัตโนมัติ



4. ถ้าต้องการดูรูปไฟล์ที่เลือกขึ้นมา เพื่อตรวจสอบความถูกต้อง ให้กดปุ่ม Show Picture โปรแกรมจะทำการแสดงรูปที่ช่อง Original ไปตามลำดับที่ระบุ ตั้งแต่ไฟล์แรกจนถึงไฟล์สุดท้าย โดยชื่อไฟล์ที่กำลังแสดงภาพอยู่ จะปรากฏในช่อง Filename ด้านบน



5. ถ้าต้องการลบชื่อไฟล์ที่อยู่ใน File list ให้กดปุ่ม Clear
6. กดปุ่ม Process เพื่อทำการประมวลผล โปรแกรมจะทำการประมวลผลไปที่ละรูปจนหมด โดยโปรแกรมจะแสดงรูปเมื่อยาค้นฉบับ เมื่อยาคัดขอบและ Resize แล้ว และรูปเมื่อยาคที่แปลงเป็นรูปขาว-ดำไปพร้อมกัน เพื่อให้ผู้ใช้งานสามารถตรวจสอบการทำงานของโปรแกรมได้ ส่วนค่า White Threshold คือค่าที่โปรแกรมใช้ในการแบ่งจุดภาพว่าจะให้เป็นสีดำหรือสีขาว เช่นถ้าจุดภาพมีค่าสี $(R+G+B)/3$ มากกว่า ค่า White Threshold จะกำหนดให้จุดนั้นเป็นสีขาว ถ้ามีค่าน้อยกว่าก็จะกำหนดให้เป็นสีดำ



7. หลังจากโปรแกรมทำการประมวลผลเสร็จแล้ว จะได้ Output เป็น Text file ชื่อว่า ImageToText.txt ไฟล์นี้จะเก็บข้อมูลทั้งหมดของภาพตัวอย่างไว้ ในรูปแบบตัวเลข 1 และ 0 ภาพแต่ละที่ผ่านการประมวลผลแล้วจะถูกบันทึกไว้ด้วย โดยจะตั้งชื่อไฟล์ตามชื่อภาพเดิมแต่จะต่อท้ายด้วย _crop และ _letter เพื่อนำไปใช้เป็นฐานข้อมูลต่อไป

3. โปรแกรมฝึกสอนและทดสอบการรู้จำ

โปรแกรมนี้จะใช้ในการฝึกสอนและทดสอบเพื่อให้ระบบมีความสามารถในการตัดสินใจแยกแยะกลุ่มเม็ดชา

ขั้นตอนการใช้งานโปรแกรมฝึกสอนและทดสอบการรู้จำ

1. เปิดโปรแกรม NeuralTrain.exe ขึ้นมา จะพบหน้าจอดังรูป






- ทำการสร้างไฟล์ Target.txt โดยใช้โปรแกรม Excel ภายในไฟล์จะประกอบด้วยตัวเลข 0 และ 1 ขนาด $M \times N$ โดยที่ M คือ จำนวนแถว ซึ่งจะเท่ากับจำนวนภาพที่นำมาฝึกสอนและ N คือจำนวนหลัก ซึ่งจะเท่ากับจำนวนเอาต์พุตของนิวรอลเน็ตเวิร์คหรือกลุ่มของเม็ดชา ถ้าอินพุตเป็นรูปของกลุ่มไหน ค่า 1 จะอยู่ที่หลักของกลุ่มนั้น ส่วนหลักอื่น ๆ จะเป็น 0 จากภาพตัวอย่างด้านล่างเป็นค่าเป้าหมายของกลุ่มเม็ดชา 13 แบบ จำนวน 100 รูป ดังนั้นค่า Target จะมีจำนวน 100×13 ตัว

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกไฟล์ Input, Target โดยกดปุ่มสี่เหลี่ยมคางหมู และตั้งชื่อไฟล์ Weight และ Error ในช่องสีขาว

File Input	ImageToText	
File Target	Target	
File Weight	Weight	
File Error	Error	

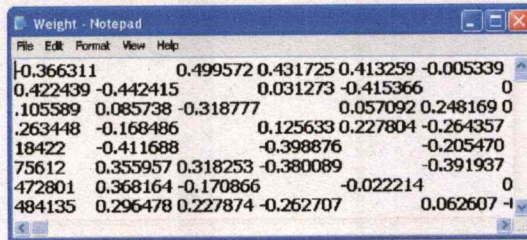
4. กำหนดจำนวนรอบในการฝึกสอน ในช่อง Round ในที่นี้กำหนดให้เท่ากับ 500 ถ้าตั้งค่าไว้มากเกินไปจะต้องใช้เวลาในการฝึกสอนนาน แต่ถ้าตั้งค่าน้อยเกินไป จะทำให้ระบบมีความถูกต้องในการแยกแยะลดลง

Round 500

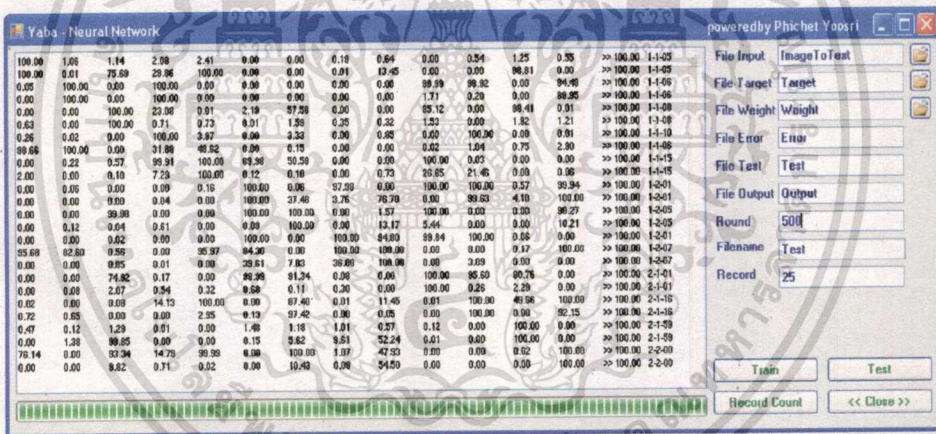
5. กดปุ่ม Train เพื่อเริ่มการฝึกสอน โปรแกรมจะแสดงความคืบหน้าและค่า Error จะเห็นว่าค่า Error ลดลงเรื่อยๆตามจำนวนรอบการฝึกสอนที่มากขึ้น



6. เมื่อฝึกสอนเสร็จแล้วจะได้ผลลัพธ์เป็นไฟล์ Weight และ Error ตามที่ได้ตั้งชื่อไว้ ข้อมูลในไฟล์ Weight จะเป็นตัวเลขทศนิยม มีค่าระหว่าง -0.5 ถึง 0.5 ดังรูป



- ขณะนี้ระบบมีความสามารถในการแยกแยะภาพเม็ดยาแล้ว โดยนำไฟล์ Weight ที่ได้จากการฝึกสอนมาใช้งาน ถ้าต้องการทดสอบการทำงานของระบบ ให้ทำการตั้งชื่อในช่อง File Output เพื่อใช้ในการบันทึกผลการทดสอบ แล้วทำการกดปุ่ม Test ระบบจะให้เลือกไฟล์ที่ต้องการทดสอบ หลังจากนั้นระบบก็จะทำการทดสอบและแสดงผลออกมาทางหน้าจอและบันทึกใน File Output ด้วย ผลลัพธ์ในการแยกแยะจะแสดงใน Column สุดท้ายในแต่ละแถว



- ถ้าต้องการนับจำนวน Record ที่อยู่ในไฟล์ Input สามารถกดปุ่ม Record Count แล้วโปรแกรมจะให้เลือกชื่อไฟล์ที่ต้องการนับ แล้วจะแสดงผลพร้อมออกมาในช่อง Filename และ Record
- กดปุ่ม Close เมื่อต้องการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรแกรมแยกแยะเม็ดยาบ้า

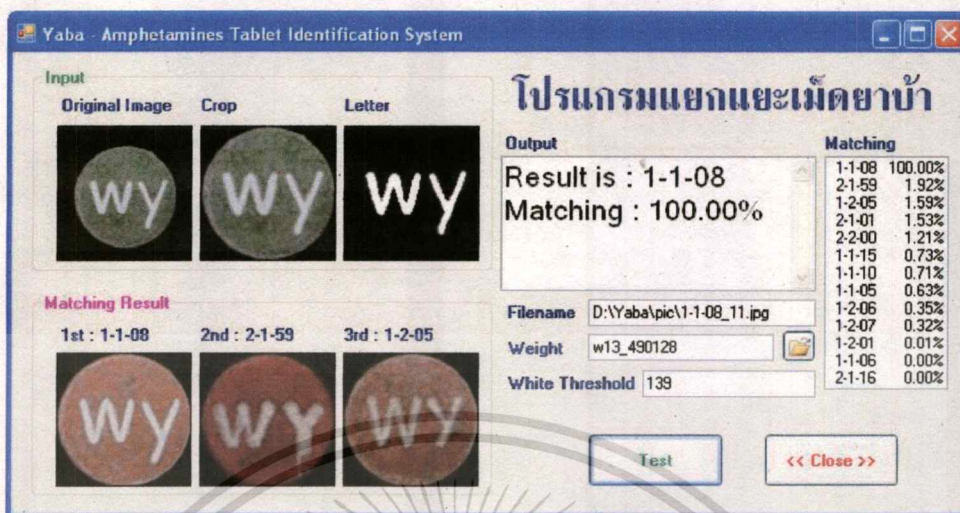
โปรแกรมนี้จะเป็นโปรแกรมที่นำไปใช้ในการทำงานจริง โดยนำไฟล์ Weight ที่ได้จากโปรแกรมฝึกสอนมาใช้ในการตัดสินใจแยกแยะกลุ่มเม็ดยา ผลลัพธ์การแยกแยะจะแสดงออกมาในรูปแบบของชื่อกลุ่มเม็ดยาและแสดงด้วยรูปภาพที่ใกล้เคียงมากที่สุด 3 ลำดับแรก

ขั้นตอนการใช้งานโปรแกรมโปรแกรมแยกแยะเม็ดยาบ้า

1. ถ่ายรูปเม็ดยาด้วยวิธีการเดียวกับข้างต้น
2. เปิดโปรแกรม Yaba.exe ขึ้นมา จะพบหน้าจอดังรูป



3. กดปุ่มสี่เหลี่ยมข้างช่อง Weight เพื่อเลือกไฟล์ Weight ที่ผ่านการฝึกสอนมาแล้วมาใช้งาน
4. กดปุ่ม Test เพื่อทำการทดสอบ โปรแกรมจะให้เลือกไฟล์ที่ต้องการทดสอบ
5. โปรแกรมจะทำการประมวลผลภาพและทำการแยกแยะ โดยแสดงผลเป็นตัวอย่างของกลุ่มเม็ดยาที่มีความใกล้เคียงกับตัวอย่างที่ได้เคยเรียนรู้มากที่สุดออกมาในช่อง Output และยังแสดงเป็นภาพเรียงตามลำดับความเหมือนใน 3 อันดับแรก ในช่อง Matching Result เพื่อช่วยในการตรวจสอบ ส่วนในช่อง Matching จะเป็นการแสดงผลการแยกแยะทั้งหมดออกมาเป็นค่าเปอร์เซ็นต์ โดยเรียงตามลำดับเช่นกัน



6. ผู้ใช้งานทำการบันทึกผลการแยกแยะ เพื่อนำไปใช้ประกอบการทำคดีต่อไป
7. ถ้าเม็ดยาที่นำมาทดสอบ เป็นกลุ่มเม็ดยาที่เป็นแบบใหม่ที่ยังไม่เคยใช้ฝึกสอนให้กับโปรแกรม โปรแกรมจะไม่สามารถทำการแยกแยะได้อย่างถูกต้อง จะต้องนำเม็ดยาแบบใหม่นั้นมาเพิ่มเข้าไปในกลุ่มตัวอย่างเม็ดยา แล้วทำการฝึกสอนให้กับโปรแกรมใหม่ จึงจะสามารถใช้งานได้
8. กดปุ่ม Close เมื่อต้องการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายพิเชษฐ์ อยู่ศรี
วันเกิด	10 พฤศจิกายน พ.ศ. 2520
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนนพคุณวิทยา จังหวัดสมุทรปราการ
ระดับมัธยมศึกษาตอนต้น	โรงเรียนสมุทรปราการ จังหวัดสมุทรปราการ
ระดับประกาศนียบัตรวิชาชีพ	คณะช่างอุตสาหกรรม สาขาวิชาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสมุทรปราการ
ระดับประกาศนียบัตรวิชาชีพชั้นสูง	คณะช่างอุตสาหกรรม สาขาวิชาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสมุทรปราการ
ระดับปริญญาตรี	คณะวิศวกรรมศาสตร์ ภาควิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าฯลาดกระบัง
ประสบการณ์การทำงาน	บริษัท ชิสโก้ ซิสเต็มส์ (ประเทศไทย) จำกัด บริษัท ทู คอร์ปอเรชั่น จำกัด (มหาชน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้