

บทที่ 4

การออกแบบโปรแกรมการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์ค

การออกแบบเพื่อใช้งานโปรแกรมคอมพิวเตอร์ สามารถใช้ภาษาคอมพิวเตอร์ ที่ใกล้เคียงกับภาษามนุษย์ ซึ่งมีอยู่มากมาย เช่น ภาษาเคลไฟ , ภาษาซี หรืออื่นๆ ที่จัดอยู่ในกลุ่มภาษาคอมพิวเตอร์ระดับสูง (High Level Language) ในงานวิจัยการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์คนี้ ได้เลือกใช้โปรแกรม ภาษา MatLab เพื่อสร้างโปรแกรมจำลอง ของงานวิจัยชิ้นนี้

MatLab เป็นคำย่อมาจาก Matrix Laboratory เริ่มแรกถูกสร้างขึ้นมาเพื่อช่วยในการคำนวณเฉพาะข้อมูลที่เกี่ยวข้องกับ เมตริกซ์ (Matrix) เท่านั้น ต่อมามีการพัฒนาขึ้นมาเพื่อช่วยในแก้ปัญหาทางวิศวกรรมศาสตร์ทั่วไป เนื่องมาจากข้อมูลที่ใช้ในการคำนวณ เช่น สมการเชิงอนุพันธ์ (Differential Equation) , กลุ่มข้อมูล (Array of Data) , กราฟ (Graph) ซึ่งล้วนแต่อยู่ในรูปของสมการเมตริกซ์

4.1 ศึกษาโปรแกรมภาษา Matlab เพื่อสร้างโปรแกรมใช้ในงานวิจัย

4.1.1 การประมวลผลภาพ (Image Processing)

ในงานวิจัยชิ้นนี้เป็นการวิเคราะห์การรู้จำโครงร่างภาพวัตถุ จึงได้มีโปรแกรมในส่วนของ การประมวลผลภาพ (Image Processing Toolbox) [13] เพื่อจัดการภาพที่ได้มา ซึ่งอาจจะไม่มีความสมบูรณ์ และข้อมูลภาพก็อาจมีจำนวนมากเกินไป ไม่เหมาะสมที่จะนำไปฝึกสอน ให้เข้าๆ จดจำไว้เนื่องจากจะใช้เวลานานเกินไป และมีข้อมูลบางส่วนที่ไม่จำเป็นที่จะต้องจดจำ ดังนั้น ข้อมูลภาพที่ได้มานี้ ก็จะต้องนำไปผ่านขั้นตอนการประมวลผลต่างๆที่เหมาะสมเพื่อให้เป็นข้อมูลของโครงร่างภาพวัตถุที่กระชับ และมีรายละเอียดเพียงพอสำหรับการรู้จำ โดยขั้นตอนการประมวลผลภาพที่จำเป็นต่างๆ มีดังนี้

- ชนิดของภาพ (Types of Image)
- การอ่าน การเขียน และการแสดงผล ภาพ (Reading Writing and Displaying Images)
- การเตรียมภาพ (Image Preprocessing)

- การแบ่งส่วนภาพ (Image Segmentation)
- การจัดเก็บภาพ (Storage Images Files)

1) ชนิดของภาพ (Types of Image)

ก) Indexed Images

แสดงในลักษณะของภาพสี เป็นภาพ ที่ประกอบด้วย สอง แถวลำดับ (Array) คือเมตริกซ์ภาพ (Image Matrix) และ คัลเลอร์แมป (Color Map) ใน 1 พิกเซล (Pixel) จะแทนด้วย 1 ครรชนี (Indexed) แล้วแต่ละครรชนี จึงแทนค่า คัลเลอร์แมป

ข) Intensity Images

ภาพสีขาว - เทา - ดำ โดยแต่ละตำแหน่งของเมตริกซ์ จะแทนด้วย จำนวน 1 พิกเซลของภาพ โดยมีค่าเปลี่ยนแปลงอยู่ในช่วง 0-1 (0 คือ ดำ; 1 คือ ขาว)

ค) Binary Images

ภาพสีขาว-ดำ มีค่าอยู่ 2 ค่า คือ ขาว (1) และ ดำ (0) เท่านั้น

ง) RGB Images

แสดงด้วยภาพสี แต่ละ 1 พิกเซล เก็บข้อมูลเป็นกลุ่ม 3 ค่า มีขนาด $m \times n \times 3$ แถวลำดับ โดย $m \times n$ เป็นขนาด (Row x Colume) ของภาพพิกเซล ซึ่งเป็นการเก็บค่าโดยตรงไม่ต้องผ่านคัลเลอร์แมป เหมือน Indexed Images

2) การอ่าน การเขียน และ การแสดงผล ภาพ (Reading Writing and Displaying Images)

ในการอ่าน (Reading) ข้อมูลภาพจากแฟ้มกราฟฟิก มาเก็บไว้ในภาพเมตริกซ์ โดยใช้รูปแบบคำสั่ง

`imread (I);`

ตัวอย่างรูปแบบ คำสั่ง : `>> imread`

```
>>I=('animal25'.tif');           %Rem I = filenameth.tif
>>imread(I);
```

ซึ่งชนิดของข้อมูลที่ทำงานภายใต้คำสั่ง จะเป็นชนิด `Uint 8` เท่านั้น สำหรับชนิดของแฟ้มกราฟฟิกที่อ่านได้แสดงในตารางที่ 6 ภาคผนวก ก

ในการเขียนข้อมูล (Writing) ภาพมาเก็บไว้ในแฟ้ม จะจัดเก็บข้อมูล เป็นชนิดเดียวกันกับที่ได้อ่านมาไว้ และ เป็นข้อมูลชนิด Uint 8 เช่นเดียวกัน โดยมีรูปแบบคำสั่ง

```
imwrite (I,'Filename');
```

ภาพที่ต้องการแสดงให้เห็น สามารถใช้คำสั่งได้หลายรูปแบบ เช่น image , imagesc , imshow โดยที่การทำงานภายใต้คำสั่งเหล่านี้เป็นชนิด ข้อมูล Uint8 เช่นเดียวกับคำสั่ง imread และ imwrite สำหรับคำสั่งที่ใช้ในงานวิจัยนี้ เลือกใช้ คือ คำสั่ง imshow ซึ่งจะมีรายละเอียดที่แตกต่างกัน ตามชนิดของภาพอีกด้วย ดังนี้

Indexed Images	:	imshow (X,map)
Intensity Images	:	imshow (I)
Binary Images	:	imshow (BW)
RGB Images	:	imshow (RGB)

นอกจากนี้สามารถแสดงภาพได้โดยตรงจากแฟ้มกราฟฟิก ด้วยการใช้คำสั่ง imshow ตามด้วยชื่อแฟ้ม ดังนี้

```
imshow filename
```

ตัวอย่างรูปแบบ คำสั่ง : >>subplot, imshow

```
>>subplot('Position',[Left Bottom Width High];
>>imshow('animal25.tif');
%Rem subplot = Position of Image.
```

การแสดงผลแต่ละภาพ ไว้บนแต่ละฟีกเจอร์ (Figure) แยกออกจากกันทำได้ โดยระบุฟีกเจอร์เพิ่มขึ้นมา ซึ่งอาจจะระบุเป็นตัวควบคุม (Handle) ใดๆ ได้ด้วยคำสั่ง figure (handle) ดังตัวอย่างคำสั่ง

```
imshow (I);
figure;
imshow (I2);
```

สำหรับการแสดงแต่ละภาพไว้บนฟิกเจอร์เดียวกันสามารถทำได้โดยคำสั่ง `subplot ( m , n , p )` โดยจะแบ่งฟิกเจอร์เป็นส่วนๆ ตามขนาดเมตริกซ์ขนาด `mxn` และ `p` เป็นการระบุว่าจะให้แสดงภาพไว้ที่ตำแหน่งใดของเมตริกซ์ ดังคำสั่ง

```
subplot (1,2,1) , imshow (X,map)
```

```
subplot (1,2,2) , imshow (X2,map)
```

3) การเตรียมภาพ (Image Preprocessing)

ตามปกติโดยทั่วไป ภาพที่ได้มานั้นมักจะไม่มีควมคมชัดหรือมีจุดรบกวน (Noise) มาก อาจทำให้การวิเคราะห์ภาพที่จะกระทำต่อไปนั้นได้ผลลัพธ์ที่ออกมาผิดพลาดได้ จึงต้องนำข้อมูลภาพนั้นไปผ่านวิธีการที่จะทำให้เป็นภาพที่ชัดเจน (Enhanced Image) กว่าเดิม ซึ่งการเตรียมภาพนั้นสามารถแบ่งออกได้เป็น สอง วิธีการใหญ่ๆ คือ

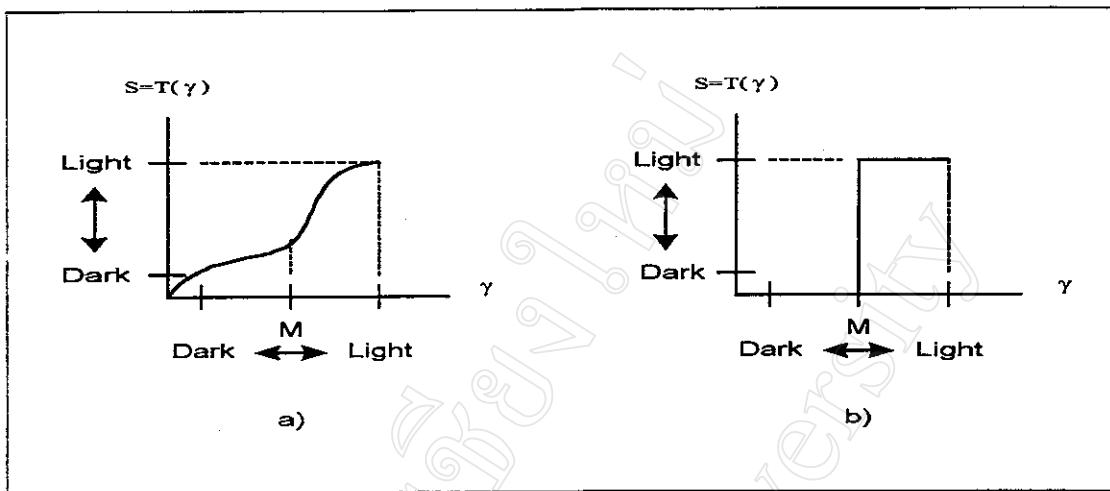
ก) การเตรียมภาพในมิติของระยะ (Spatial Domain Methods)

เป็นการทำการประมวลผลกับข้อมูลของภาพ ที่เก็บไว้ในมิติของระยะ ซึ่งผู้คิดค้นไว้มากมายหลายวิธี แต่ในที่นี้จะกล่าวถึงเฉพาะวิธีที่สำคัญๆ ได้แก่

- การปรับคอนทราสต์ (Contrast Stretching) โดยการกำหนดระดับความเข้มของจุดค่าหนึ่งขึ้นมาเป็นค่าตัดสิน (Threshold Level) สำหรับจุดในภาพที่มีระดับความเข้มต่ำกว่าค่าตัดสินนี้ก็จะเปลี่ยนให้เป็นจุดที่มีความเข้มเป็นศูนย์ และสำหรับจุดที่มีระดับความเข้มสูงกว่าค่าตัดสินนี้ก็จะเปลี่ยนให้เป็น จุดที่มีค่าความเข้มสูงสุด ดังแสดงการทำงานนี้ในกราฟในรูป 4.1 และจะได้ภาพผลลัพธ์ที่มีคอนทราสต์หรือความเข้มของจุดที่ตัดกันมากยิ่งขึ้น

● การทำภาพเนกาตีฟ (Image Negatives)

โดยการกลับค่าแสดงระดับความเข้มของจุดต่างๆในภาพ กล่าวคือ สำหรับจุดที่มีความเข้มมากก็จะกลับให้เป็นจุดที่มีความเข้มน้อย และจุดที่มีความเข้มน้อยก็จะกลับให้เป็นจุดที่มีความเข้มมาก



รูป 4.1 วิธีการทำงานของวิธีการปรับคอนทราสต์

ข) การเตรียมภาพในมิติของความถี่ (Frequency Domain Methods)

เป็นการทำการประมวลผลกับข้อมูลของภาพที่เก็บไว้ในมิติของความถี่ ซึ่งในบริเวณของภาพที่มีความเข้มของจุดไม่เปลี่ยนแปลงมากนัก ความถี่ที่ได้ก็จะต่ำ ส่วนในบริเวณที่เป็นขอบของวัตถุในภาพก็จะมี การเปลี่ยนแปลงของความเข้มมาก จึงได้ความถี่สูง เพื่อที่จะทำให้ได้ภาพที่ชัดเจนยิ่งขึ้นเราสามารถทำการกรอง (Filter) ความถี่ที่ไม่ต้องการออกไปจากข้อมูลภาพในมิติของความถี่นี้โดยถ้าทำการกรองความถี่ต่ำ (Low Pass Filter) ก็จะได้ภาพที่มีการเปลี่ยนแปลงของความเข้มที่นุ่มนวล (Smooth) ยิ่งขึ้น ถ้าทำการกรองความถี่สูงออกก็จะได้ผลลัพธ์ที่เป็นภาพที่ตรงบริเวณขอบภาพของวัตถุ ในภาพนั้นจะถูกขยายให้มีสีที่ตัดกันมากยิ่งขึ้น (Edge Enhancement) และถ้าต้องการปรับภาพ ณ บริเวณใดบริเวณหนึ่งก็ทำการกรองความถี่เป็นช่วง (Bandpass Filter) ได้ ทั้งนี้โดยจากสมการของข้อมูลภาพในมิติของความถี่ $f(x,y)$ ซึ่งตามปกติก็จะได้ไม่มีการเปลี่ยนแปลงแบบมีคาบคงที่ (Periodic Function) นั้นจะต้องทำการเปลี่ยนมิติให้เป็นสมการในมิติของฟูรีเยร์ (Fourier Domain) โดยการนำไปทำการเปลี่ยนฟูรีเยร์ (Fourier Transform) ให้เป็นสมการฟูรีเยร์ (Fourier Function, $F(u,v)$) ตามสมการ

$$F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \exp(-j2\pi(ux + vy)) dx dy \quad \dots(4.1)$$

ซึ่งมีคาบคงที่เสียก่อน จึงจะนำสมการฟูรีเยร์นี้ไปทำการประมวลผลต่อไป ได้อย่างสะดวกต่อไป

- การกรองความถี่ต่ำ (Lowpass Filtering)

สำหรับสมการแสดงข้อมูลภาพในมิติของฟูเรียร์ $F(u,v)$ ก็จะนำมาทำการปรับปรุงแบบคอนโวลูชัน (Convolution) ให้เป็นสมการ

$$G(u,v) = H(u,v)F(u,v) \quad \dots(4.2)$$

โดย $H(u,v)$ เป็นฟังก์ชันถ่ายโอน (Transfer Function) สำหรับตัวกรองความถี่ต่ำ (Lowpass Filter) โดย

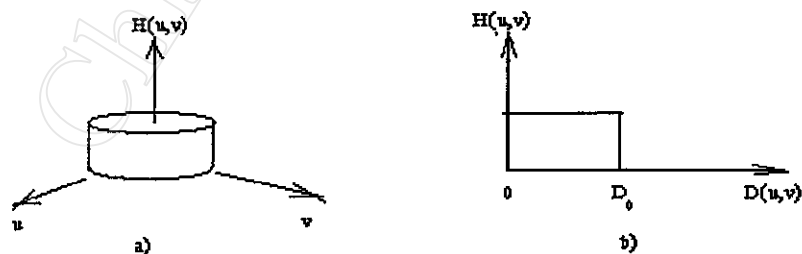
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_o \\ 0 & \text{if } D(u,v) > D_o \end{cases} \quad \dots(4.3)$$

โดย

$D(u,v)$ = ระยะจุด u,v ที่อยู่ห่างจากจุดเริ่มต้น (Origin) ในระนาบของความถี่

D_o = ค่าความถี่จุดตัด (Cut-off Frequency)

ดังแสดงคุณสมบัติการทำงานของตัวกรองความถี่ต่ำในรูปที่ 4.2



รูป 4.2 คุณสมบัติในการทำงานของตัวกรองความถี่ต่ำ

a) การทำงานในระบบ 2 มิติ

b) การทำงานในระบบมิติเดียว

และเมื่อได้ผลลัพธ์ซึ่งยังอยู่ในมิติของฟูเรียร์มาแล้ว ก็นำมาทำการเปลี่ยนกลับ (Fourier Inverse Transform) เพื่อให้ได้ผลการข้อมูลภาพที่ปรับปรุงแล้วในมิติของความถี่เช่นเดิมตามสมการ

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp(j2\pi(ux + vy)) du dv \quad \dots(4.4)$$

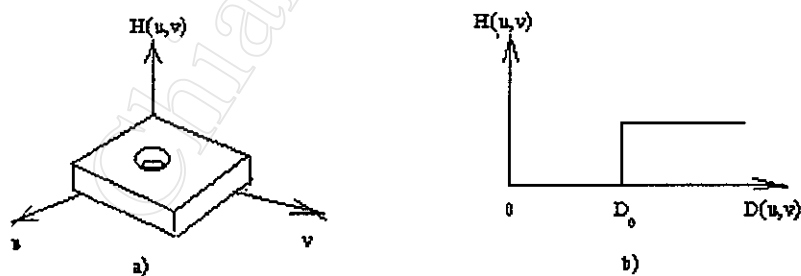
แล้วนำมาพล็อตกลับเป็นภาพ ก็จะได้ภาพที่คมชัดขึ้น

- การกรองความถี่สูง (Highpass Filtering)

เป็นการนำเอาสมการของข้อมูลภาพในมิติของฟูเรียร์ $F(u, v)$ มาทำการปรับปรุงแบบคอนโวลูชันด้วย $H(u, v)$ ตามสมการ (4.2) แต่คราวนี้ฟังก์ชันถ่ายโอนสำหรับ $H(u, v)$ เปลี่ยนไปเป็น

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases} \quad \dots(4.5)$$

ดังแสดงคุณสมบัติในการทำงานนี้ในรูป 4.3



รูป 4.3 คุณสมบัติในการทำงานของตัวกรองความถี่สูง

- การทำงานในระบบ 2 มิติ
- การทำงานในระบบมิติเดียว

ค) การปรับแต่งภาพ (Image Enhancement)

ภาพที่นำมาประมวลผลในระบบดิจิทัล จะประกอบด้วยโครงสร้างข้อมูลเมตริกซ์ ภาพที่ได้เป็นฟังก์ชันชนิดต่อเนื่อง $f(x,y)$ สอง พิกัด (2 Co-Ordinates) ซึ่งการแปลงภาพเป็นเชิงเลข (Image Digitizer) หมายถึงการแปลงฟังก์ชันต่อเนื่องเป็นเมตริกซ์ แทนด้วย M แถว (Row) N สดมภ์ (Column)

เทคนิคการปรับแต่งภาพ (Image Enhancement Techniques) เป็นการปรับแต่งทางด้านกายภาพ (Objectivity) เช่น การปรับแต่งอัตราส่วนภาพ ต่อสัญญาณรบกวนภาพ (Signal to Noise Ratio) และทางด้านความรู้สึก (Subjectivity) เช่นการปรับแต่ง สี (Color) และ อินเทนซิตี (Intensities)

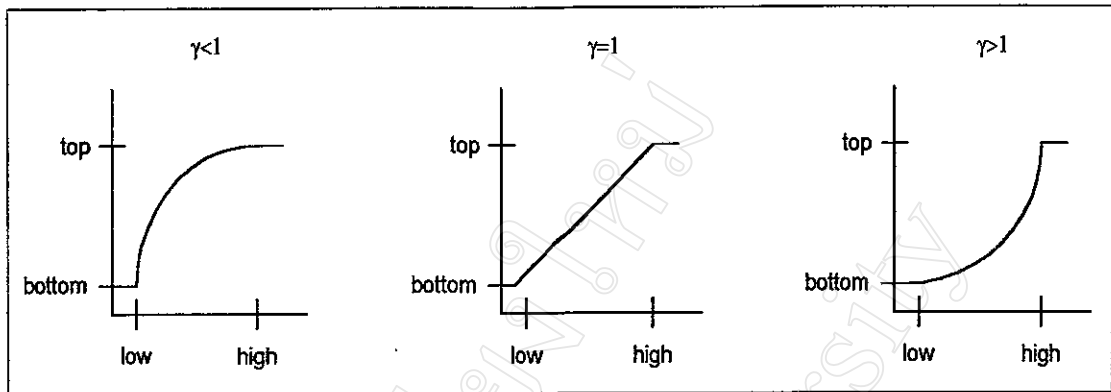
ในงานวิจัยนี้ ใช้การปรับแต่งภาพเพื่อนำไปเป็นภาพอินพุต โดยการปรับแต่ง ความสว่าง (Brightness) , ความแตกต่างระหว่างความสว่างกับความมืดภาพ (Contrast) และ ค่าแกมมา (Gamma) โดยมีรูปแบบคำสั่งดังนี้

$$\text{ImAdjust} = (I, [\text{Low High}] [\text{Bot Top}])$$

ค่าแกมมา เป็นองค์ประกอบในการปรับระดับความสัมพันธ์ระหว่างค่าอินพุต และ เอาต์พุต ที่อาจไม่เป็นเชิงเส้น เช่นค่ากึ่งกลางของ Low และ High อาจจะมากกว่า หรือน้อยกว่า ค่ากึ่งกลางของ Bot และ Top โดยที่ค่าแกมมาจะอยู่ในช่วง 0 ถึง อนันต์ มีค่าโดยปริยาย (Default) เท่ากับ 1 นั่นคือความสัมพันธ์ของภาพเป็นเชิงเส้น แต่ถ้าหากมันมีค่าน้อยกว่า 1 แนวโน้มของน้ำหนักภาพเอาต์พุตก่อนไปทาง ความสว่าง (Brighter) แต่หากมีค่ามากกว่า 1 แนวโน้มน้ำหนักภาพเอาต์พุตก่อนไปทางมืด (Darker)

คุณสมบัติของค่าแกมมาแสดงดังรูป 4.4 แสดงความสัมพันธ์ของ ค่า Low , High และ Bot Top เมื่อแกมมามีค่า น้อยกว่า , เท่ากับ และ มากกว่า 1 ในแต่ละกราฟแกน x แทนค่า อินเทนซิตี ของภาพอินพุต และ แกน y แทนค่า อินเทนซิตี ของภาพเอาต์พุตมีรูปแบบคำสั่งดังนี้ [13]

$$\text{ImAdjust} = (I, [\text{Low High}] [\text{Bot Top}], \text{Gamma})$$



รูป 4.4 คุณสมบัติของค่าแกมมา

4) การแบ่งส่วนภาพ (Image Segmentation)

หลังจากที่ได้นำข้อมูลภาพไปทำการปรับปรุงให้มีความชัดเจน คมชัดขึ้น แล้วจึงนำข้อมูลภาพไปทำการประมวลผลต่อไปเพื่อแยกวัตถุ (Objects) ออกมาจากในภาพซึ่งในที่นี้เรียกว่าเป็นการแบ่งส่วนภาพ เพื่อจะได้นำลักษณะ รูปร่าง และรายละเอียดของวัตถุในภาพแต่ละชิ้นไปวิเคราะห์ว่าเป็นอะไรต่อไป การแบ่งส่วนภาพนั้นได้มีผู้คิดค้นวิธีการมากมายหลายวิธี แต่ในที่นี้จะกล่าวถึงเฉพาะวิธีการหาเส้นขอบซึ่งจะเริ่มค้นด้วยการตรวจหาจุดของเส้นขอบ (Edge Point Detection) การเชื่อมต่อเส้นขอบ (Edge Linking) และการหารายละเอียดของวัตถุ (Object Detail Detection) ซึ่งในที่นี้จะไม่มีการหารายละเอียดของวัตถุเนื่องจากต้องการทราบเพียงรูปร่างของวัตถุเท่านั้น

ก) การตรวจหาจุดของเส้นขอบ (Edge Point Detection)

หลังจากที่ได้นำภาพไปทำการปรับปรุงสภาพให้ดีขึ้นตามวิธีการเตรียมภาพ แล้วก็นำมาตรวจหาจุดของเส้นขอบ (Edge Point) ของวัตถุ (Objects) ต่างๆ ที่ปรากฏในภาพ โดยมีวิธีการตรวจหาจุดของเส้นขอบ ได้แก่

- การตรวจหาเส้นขอบแบบเกรเดียนท์ (Gradient Edge Detector)

ทำการตรวจหาเส้นขอบด้วยการเลื่อนตรวจสอบการเปลี่ยนแปลงของค่าแสดงระดับสีเทาของจุดต่างๆในภาพ โดยจะกวาดตรวจ (Scan) ไปตามจุดในเส้นแนวนอน (Horizontal Line) จากซ้ายไปขวาจนสุดเส้นโดยเริ่มต้นจากเส้นบนสุดในภาพแล้วเลื่อนแนวเส้นลงมาเรื่อยๆจนถึงเส้นล่างสุด โดยจากความจริงที่ว่า ในบริเวณเบื้องหลัง (Background) ของภาพจะมีการเปลี่ยนแปลงของระดับสีเทาของจุดต่างๆน้อยมาก แต่ ณ. เส้นขอบของวัตถุนั้นจะมีการเปลี่ยน

แปลงของระดับสีเทาเป็นอย่างมาก ดังนั้นในวิธีการนี้จึงตรวจดูค่าแสดงระดับสีของจุดต่างๆด้วยการหาอนุพันธ์ (Differentiate) สมการข้อมูลภาพหนึ่งครั้ง โดยใช้เกรเดียนท์โอเปอเรเตอร์ ∇ (Gradient Operator)

โดย

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad \dots(4.6)$$

ในระบบ 2 มิติ ซึ่ง f คือ $f(x,y)$ โดย

$$|\nabla f| = \text{mag}(\nabla_f) = [G_x^2 + G_y^2]^{\frac{1}{2}} \quad \dots(4.7)$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad \dots(4.8)$$

ซึ่งจะเห็นว่าการทำงานตามฟังก์ชันเกรเดียนท์ กับสมการข้อมูลภาพนั้นจะต้อง ทำการหาอนุพันธ์ ข้อมูล ซึ่งการเขียนโปรแกรมคอมพิวเตอร์ให้ทำงานดังกล่าวจะต้องใช้เวลาในการทำงานนานมาก จึงมีการเปลี่ยนมาใช้ในการคอนไวลูชัน ข้อมูลภาพด้วยแมสก์ (Masks) แบบต่างๆซึ่งให้ผลลัพธ์เช่นเดียวกัน แต่การเขียนโปรแกรมสำหรับทำงานนี้จะสะดวกและรวดเร็วกว่ามากโดยมีขั้นตอนการทำงานดังนี้

- นำแมสก์ไปวางทับจุดที่จะหาค่าเกรเดียนท์
- นำค่าในแต่ละเซลล์ไปคูณกับค่าระดับสีเทาของแต่ละจุดที่เซลล์นั้นๆทับอยู่เพื่อนำมาคำนวณหาค่าเฉลี่ยเป็นค่าของ G_x และ G_y และค่าของ G ณ จุดนั้นได้
- หลังจากทีเลื่อนแมสก์ไปทับผ่านทุกๆจุดในภาพและทำการคำนวณค่า G สำหรับทุกจุดแล้วก็จะได้ผลลัพธ์เป็นเกรเดียนท์ของภาพนั้น

ทั้งนี้โดยแบบของแมสก์ที่เป็นที่นิยมได้แก่

ตาราง 4.1 แบบของแมสก์หาจุดเส้นขอบแบบเกรเดียนท์

- Edge Operator Convolution Mask

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

- Roberts Edge Mask

1	0
0	-1

G_x

0	1
-1	0

G_y

- Prewitt Edge Mask

-1	-1	-1
0	0	0
1	1	1

G_x

-1	0	1
-1	0	1
-1	0	1

G_y

- Sobel Edge Mask

-1	-2	-1
0	0	0
1	2	1

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

- การตรวจหาเส้นขอบแบบลาปลาเซียน (Laplacian Edge Detection)
จะทำการประมวลผลข้อมูลภาพเช่นเดียวกับในแบบเกรเดียนท์ แต่คราวนี้จะทำการหาอนุพันธ์ 2 ครั้ง ณ แต่ละจุดตามสมการ

$$\nabla^2 f(x, y) = \frac{d^2 f(x, y)}{dx^2} + \frac{d^2 f(x, y)}{dy^2} \quad \dots(4.9)$$

ซึ่งสามารถทำงานแทนด้วยการใช้ แมสก์แบบลาปลาเซียนมาทำ คอนโวลูชันกับข้อมูลภาพในมิติของระยะ โดยลักษณะของแมสก์เป็นดังนี้

ตาราง 4.2 แบบของแมสก์หาจุดเส้นขอบแบบลาปลาเซียน

- Laplacian Operator Convolution Mas

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

แต่อย่างไรก็ตาม การตรวจหาเส้นขอบแบบลาปลาเซียนนี้มีข้อเสียคือ มีผลจากจุดรบกวนในภาพมากขึ้นกว่าเดิม และแนวเส้นขอบที่หาได้ไม่มีการบอกทิศทางแต่อย่างใด

ข) การหาขอบภาพ (Edge Detection)

จะเป็นการหาขอบภาพชนิด Intensity Image และข้อมูลเป็นชนิด double หรือ Uint8 ก็ได้ โดยภายหลังการหาขอบภาพแล้วจะได้เป็นภาพชนิด Binary Image และ เป็นข้อมูลชนิด Uint8 เท่านั้น โดยที่บริเวณขอบของภาพแสดงด้วยสีขาว ส่วนอื่นๆ แทนด้วยสีดำ มีรูปแบบคำสั่งดังนี้

BW = edge (I,method)

BW = edge (I,method,thresh)

BW = edge (I,method,thresh,direction)

BW = edge (I,'log',thresh,sigma)

BW = edge (I,'zerocross',thresh,h)

[BW,thresh]=edge(...)

● Method เป็นค่าสตริง (String) ที่ระบุถึงวิธีการหาขอบภาพ มีหลายวิธี ได้แก่

- 'sobel' (Default) การหาขอบภาพด้วยวิธี Sobel Approximation
- 'prewitt' การหาขอบภาพด้วยวิธี Prewitt Approximation
- 'roberts' การหาขอบภาพด้วยวิธี Robert Approximation
- 'log' การหาขอบภาพด้วยวิธี Zero Crossing ของ

การกรอง (Filter) ด้วยหลักของ Laplacian of Gaussin Filter การหาขอบภาพด้วยวิธีนี้ ต้องมีการกำหนดค่าซิกมา (Sigma) ที่เป็นค่าเบี่ยงเบนมาตรฐาน (Standard Deviation)

- 'zerocross' การหาขอบภาพด้วยวิธี Zero Crossing ของการกรองด้วยค่าที่เรากำหนดเอง (h) และ Thresh เป็นการกำหนดค่า Sensitivity Threshold ซึ่งทำการหาขอบภาพโดยจะไม่สนใจบริเวณขอบที่มีค่าน้อยกว่าค่า Threshold ถ้าไม่มีการกำหนดค่า Thresh ค่าตั้ง Edge จะเลือกค่าที่เหมาะสมด้วยตัวเอง

- Direction ค่าสตริงที่ระบุทิศทางของการหาขอบภาพ

แบ่งเป็น 'Horizontal' , 'Vertical' หรือ 'Both' (Default) สำหรับการหาขอบภาพโครงร่างภาพวัตถุ ของงานวิจัยนี้ได้กำหนดให้สามารถ เลือกวิธีการหาขอบภาพได้ 3 แบบคือ 'sobel' , 'prewitt' และ 'robert' ส่วนค่า Threshold ให้มีการกำหนดแบบอัตโนมัติ และทิศทางของการหาขอบภาพ (Direction) กำหนดให้หาขอบภาพทั้ง 2 ทิศทาง

ตัวอย่างรูปแบบ คำสั่ง : >>edge

```
>>BW = edge('animal25.tif','sobel');
```

```
%Rem BW = Image Edged Detection.
```

ค) การเชื่อมจุดของเส้นขอบ (Edge Point Linking)

เมื่อแยกจุดทุกจุดในภาพที่มีแนวโน้มนั้นจะประกอบกันขึ้นเป็นเส้นขอบออกมาได้แล้ว ในขั้นตอนนี้จะตรวจสอบดูว่าจุดใดบ้างที่จะสามารถเชื่อมต่อกันขึ้นมาเป็นเส้นขอบสั้นๆ ได้บ้าง เพื่อจะได้เส้นขอบต่างๆ ที่มีอยู่ในภาพขึ้นมา ซึ่งก็มีวิธีการตรวจที่เป็นที่นิยมได้แก่

- วิธีการเชื่อมขอบเฉพาะที่ (Local Edge Linking Method)

จากผลการหาจุดของเส้นขอบที่หาจากวิธีการเกรเดียนท์ซึ่งแต่ละจุดจะมีทั้งขนาดและทิศทาง ดังนั้นสำหรับจุดที่อยู่ติดกันและมีทิศทางที่ตรงกันหรือใกล้เคียงกันมาก ก็สามารถถือได้ว่าเป็นจุดที่วางอยู่บนเส้นขอบเดียวกัน และเมื่อพิจารณาทิศทางของจุดเส้นขอบไปจนครบทุกๆ จุดที่พบว่ามียู่ในภาพก็จะสามารถหาเส้นขอบที่ปรากฏทั้งหมดในภาพนั้นได้ แต่อย่างไรก็ตามวิธีการนี้ก็มีข้อเสียคือ ต้องเสียเวลาในการทำงานของโปรแกรมมาก และได้เส้นขอบที่ไม่ใช่เส้นจริงคิดมาด้วยเป็นจำนวนมาก

- วิธีการฮัฟทรานส์ฟอร์ม (Hough Transform Method)

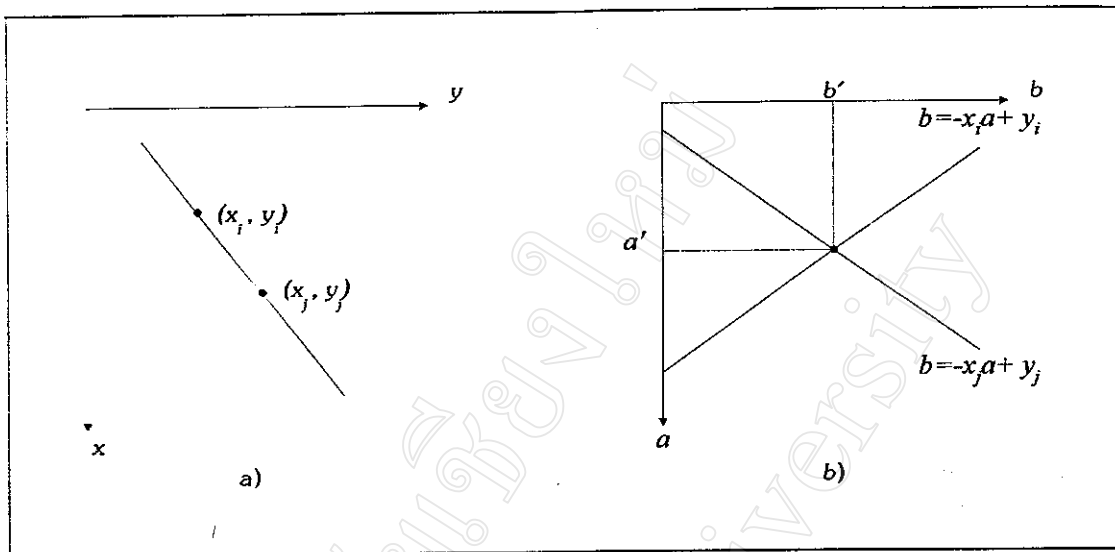
ทำการหาจุดที่วางอยู่บนเส้นขอบเดียวกันโดยการสำรวจหาจุดต่างๆ ที่มีค่าความชัน (Slope) หรือมุมโค้ง (Arc Angle) เท่ากัน

สำหรับกรณีเส้นตรง จากสมการของเส้นตรง

$$y = ax + b \quad \dots (4.10)$$

เป็นสมการของเส้นตรงเส้นหนึ่งที่ผ่านมาจุด x, y โดยมีค่าความชัน (a) และลากตัดผ่านแกน y ที่จุด b

พิจารณาจุด 2 จุดคือ x_1, y_1 และ x_2, y_2 ซึ่งในระนาบของแกน x, y จะสามารถพล็อตเส้นตรงต่างๆ ที่ลากผ่านจุดทั้งสองได้ดังแสดงในรูปที่ 4.5 a) และเส้นเหล่านี้เมื่อนำมาพล็อตในระนาบของแกน a, b ก็จะได้เป็นเส้นตรงอีก 2 เส้นดังในรูปที่ 4.5 b) และสำหรับเส้นตรงเฉพาะเส้นที่ลากผ่านจุดทั้งสองในระนาบของ x, y ดังในรูป a) ก็จะปรากฏเป็นจุดตัดของทั้งสองเส้นในระนาบของ a, b ดังในรูป b)



รูป 4.5 แสดงพล็อตของเส้นตรงต่างๆในระนาบของแกน x, y และ a, b

โดย ณ. จุดตัดนี้ในระนาบ a, b จะบอกทั้งค่าความชันและจุดตัดแกน y ของเส้นตรงดังกล่าวด้วย ดังนั้นจากหลักการนี้จากจุดขอบที่หาได้จากในภาพ จึงสามารถนำมาประมวลผลเพื่อหาเส้นตรงที่จุดเหล่านี้อาจจะประกอบกันขึ้นตามวิธีการดังนี้

- ทำการแบ่งช่วงในแกน a, b ออกเป็นแถวลำดับ 2 มิติ A
- ให้ทุกๆค่าในแถวลำดับ A เป็น 0 ไว้ก่อน
- สำหรับแต่ละจุดขอบที่หาได้ในภาพทำการคำนวณโดยแทนค่า a_i

สำหรับแต่ละช่วงของ a แล้วคำนวณหาค่า b_i ออกมาตามสมการ $y_i = ax + b_i$ เมื่อค่า a, b_i ที่ได้มานั้นตรงกับเซลล์ใดในเมตริกก็เพิ่มค่าในเซลล์นั้นอีก หนึ่ง กระทำเช่นนี้จนครบทุกๆ จุดขอบที่มีอยู่

- ค่า M ณ แต่ละเซลล์ก็จะบอกจำนวนจุดในระนาบของ x, y ที่วางอยู่ในเส้นตรง $y_i = ax + b_i$

แต่อย่างไรก็ตามวิธีการนี้จะใช้หาเส้นขอบได้เฉพาะที่เป็นเส้นตรงเท่านั้น ในกรณีที่ต้องการหาเส้นขอบที่เป็นเส้นโค้งก็จะใช้สมการของเส้นเป็น $x \cos \theta + y \sin \theta = \rho$ แทน

4) การจัดเก็บภาพ (Storage Images Files)

การจัดเก็บภาพ ในงานวิจัยนี้ได้ใช้กล้องถ่ายภาพชนิดดิจิทัล โดยการจับภาพ (Capture Image) มาเก็บไว้ในหน่วยจำของเครื่องไมโครคอมพิวเตอร์โดยมีรูปแบบของการ

เก็บบันทึกภาพ เป็นรูปแบบ 'TIF' (Tagged Image File Format) ขนาดของภาพ 100 x 100 Pixels ที่มีพาลีต (Palette) ของสีเพียงสองสี คือ สีดำ และ สีขาว ลงไว้ในแฟ้มข้อมูล (Files) เพื่อจะนำไปใช้วิเคราะห์ต่อไป การเก็บข้อมูลของภาพนั้นๆ มีอยู่ 2 ชนิด คือ

ก) Uint 8 Data

ปกติจะทำงานใน Uint 8 Data นี้ แต่จะเป็นเมตริกซ์ ที่ไม่สามารถนำไปกระทำการคำนวณใดๆ ได้ จะต้องทำการเปลี่ยนให้อยู่ในชนิด Double ก่อน

ข) Double

เป็นชนิดที่สามารถนำเมตริกซ์ไปคำนวณได้ สำหรับการแปลงชนิดของข้อมูล (Converting Between Data Types) ทั้งสองกลับไปมา สามารถกระทำด้วยคำสั่ง ดังแสดงในตารางที่ 3 และ ตารางที่ 4 ของภาคผนวก ก

นอกจากนี้ยังมีเมตริกซ์คัลเลอร์แมป (Matrix Color Map) ซึ่งเป็นเมตริกซ์ขนาด $m \times 3$ ของชนิด Double เพื่อใช้ในการกำหนดค่าสี แดง , เขียว และ น้ำเงิน (Red , Green and Blue : RGB) ดังนี้

$$\text{Color} = [R \ G \ B]$$

โดยค่า R , G และ B เป็นค่าจริง (Real Scalar) อยู่ในช่วง 0 (ค่า) ถึง 1 (ความเข้มสูงสุด)

4.1.2 การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่กลับ

การออกแบบ แบบจำลองโครงข่ายประสาทเทียมแบบแพร่กลับ แบ่งออกเป็นขั้นตอนดังนี้ [15]

- การสร้างโครงข่าย และ กำหนดส่วนประกอบของ โครงข่ายประสาทเทียม (Network)
- การป้อนข้อมูลอินพุต (Input Data)
- การระบุเป้าหมาย (Specify to Target)
- การเตรียมชุดการเรียนรู้ (PreTraining)
- การฝึกสอน (Training)

1) การสร้างโครงข่าย และ กำหนดส่วนประกอบของ โครงข่ายประสาทเทียม (Network)

ในการสร้างโครงข่ายประสาทเทียมนั้น ต้องมีการกำหนดค่า ตัวแปรที่จำเป็น ต้องใช้ กำหนดค่าในโครงข่าย ดังแสดงในรูปที่ 4.1 โดยตัวแปรทั้งหมด ต้องมีความสอดคล้องกัน โดย p กำหนดเป็น อินพุตเวกเตอร์ มีขนาดเท่ากับ R โดยแต่ละชั้นประกอบไปด้วยจำนวน นิวรอน (s) , ค่าถ่วงน้ำหนัก (lw สำหรับอินพุตเมตริกซ์ และ lw สำหรับชั้นเมตริกซ์ถัดมา) , ค่าไบแอส (b) , ฟังก์ชันการกระตุ้น (f) และ เอาต์พุต (a) [14]

คำสั่งที่ใช้ในการสร้างโครงข่าย คือ

$\gg [W_1, B_1, \dots, W_n, B_n] = \text{initff}(p, S_1, 'F_1', \dots, S_n, 'F_n')$

ตัวอย่างรูปแบบ คำสั่ง : $\gg \text{initff}$

```
 $\gg [W1, B1, W2, B2] = \text{initff}([mn], UB1, UT1, UB2, UT2);$ 
                                %Rem  $W_n, B_n = \text{Weight \& Bias.}$ 
                                %Rem  $UB_n = \text{Number of Neural per Layer.}$ 
                                %Rem  $UT_n = \text{Types of Activation Function.}$ 
```

โดยที่คำสัญลักษณ์ต่างๆ มีความหมายดังนี้

P : $R \times 2$ Matrix of Input Vectors.

ซึ่งเป็นขนาดมากที่สุด และน้อยที่สุด ที่จะเป็นไปได้ในข้อมูลที่เข้ามา
แต่ละตัว จำนวนหลักเท่ากับ R

S_i : Size of i th Layers

F_i : Activation Function (or Transfer Function) of the i th Layer
เป็นรูปแบบคำสั่งสตริงมี 3 รูปแบบ ที่ใช้งานวิจัยนี้

- 'purelin' คือ ไอเดนติตี้ฟังก์ชัน
- 'logsig' คือ ไบนารีซิกมอยด์
- 'tansig' คือ ไบโพลาร์ซิกมอยด์

W_i : Weight Matrix of the i th Layer

B_i : Bias (Column) Vector of the i th Layer

2) การป้อนข้อมูลอินพุต (Input Data)

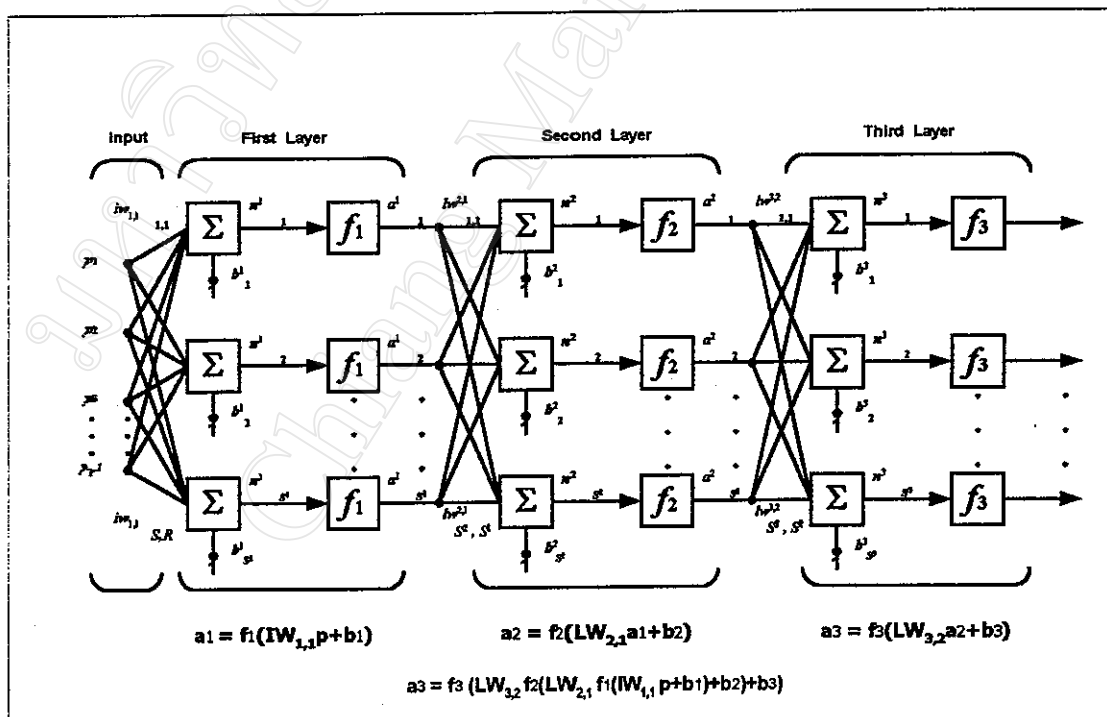
วิธีการป้อนข้อมูลมี สอง แบบคือ

ก) Concurrent

การป้อนข้อมูลที่ไม่มีเรื่องของเวลาเข้ามาเกี่ยวข้อง โปรแกรมจะมองว่าข้อมูลป้อนเข้าสู่ระบบ ที่เวลาเดียวกัน การป้อนข้อมูลเช่นนี้จะใช้เครื่องหมาย ‘[...]’ เช่น

$$p_1 = \begin{bmatrix} 1 & 2 & 2 & 3 \\ 2 & 1 & 3 & 1 \end{bmatrix}$$

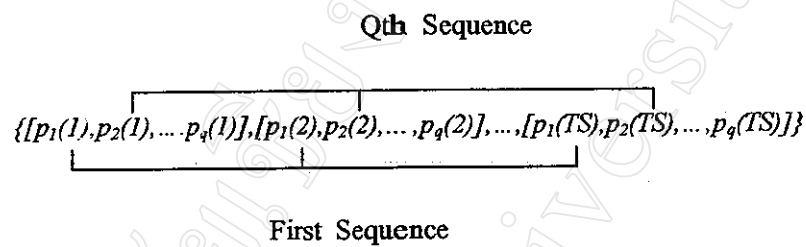
จะได้ $\gg p = [1 \ 2 \ 2 \ 3 ; 2 \ 1 \ 3 \ 1]$



รูป 4.6 โครงข่ายประสาทเทียมชนิดหลายชั้น [14]

๒) Sequential

การป้อนข้อมูลที่มีเรื่องเวลาเข้ามาเกี่ยวข้อง โปรแกรมจะมองว่าชุดข้อมูลใดเข้ามาก่อนหน้าหรือภายหลัง การป้อนข้อมูลเช่นนี้ใช้เครื่องหมายวงเล็บ '{...}' จะได้การจัดเรียงข้อมูล ดังนี้



จะได้

$$\begin{aligned} p_1(1) &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} & p_2(1) &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} & p_3(1) &= \begin{bmatrix} 2 \\ 3 \end{bmatrix} & p_4(1) &= \begin{bmatrix} 3 \\ 1 \end{bmatrix} \\ p_1(2) &= \begin{bmatrix} 2 \\ 3 \end{bmatrix} & p_2(2) &= \begin{bmatrix} 3 \\ 2 \end{bmatrix} & p_3(2) &= \begin{bmatrix} 3 \\ 4 \end{bmatrix} & p_4(2) &= \begin{bmatrix} 4 \\ 2 \end{bmatrix} \end{aligned}$$

จะได้ $\gg p = \{[1 \ 2 \ 2 \ 3; 2 \ 1 \ 3 \ 1] \ [2 \ 3 \ 3 \ 4; 3 \ 2 \ 4 \ 2]\}$

3) การระบุเป้าหมาย (Specify to Target)

เป้าหมายจะกำหนดเป็นเมตริกซ์ โดยจำนวนหลักระบุตามจำนวนอินพุตว่ามีเข้ามาที่ชุด และจำนวนแถวระบุตามจำนวนนิวรอนในชั้นเอาต์พุต เช่น

$$\gg p = [-1 \ -1 \ 2 \ 2; 0 \ 5 \ 0 \ 5]$$

$$\gg t = [-1 \ -1 \ 1 \ 1];$$

4) การเตรียมชุดการเรียนรู้ (PreTraining)

เป็นการเพิ่มประสิทธิภาพให้แก่โครงข่าย คือทำให้เกิดการเรียนรู้ได้รวดเร็วขึ้น โดยกำหนดช่วงของค่าถ่วงน้ำหนัก และ ค่าไบแอส

- `>>randS`

คำสั่งกำหนดช่วงการปรับค่าถ่วงน้ำหนักและค่าไบแอส (W&B) ให้อยู่ในช่วง $[-1, +1]$ โดยการปรับ W&B นั้นจะปรับอย่างสมดุลทั้งแถว และสดมภ์ของเมตริกซ์ การใช้คำสั่ง คือ

`>>[W,B]=randS(S,R)`

เมื่อ S มีค่าเท่ากับขนาดของนิวรอนในชั้นนั้นๆ

เมื่อ R มีค่าเท่ากับจำนวนอินพุตในแต่ละชุดที่เข้ามา

- `>>randnc`

คำสั่งที่มีการทำงานเช่นเดียวกับ randS แต่การปรับ W&B นั้นจะกำหนดค่าในสดมภ์ไว้ เปลี่ยนแปลงเฉพาะค่าในแถว ส่วนการใช้คำสั่งเหมือนกันเปลี่ยนเฉพาะคำสั่งเป็น randnc เท่านั้น

- `>>randnr`

คำสั่งที่มีการทำงานเช่นเดียวกัน แต่การปรับ W&B นั้นจะกำหนดค่าในแถวไว้เปลี่ยนแปลงเฉพาะค่าในสดมภ์ ส่วนการใช้คำสั่งเหมือนกันเปลี่ยนเฉพาะคำสั่งเป็น randnr เท่านั้น

- `>>nwtan`

คำสั่งในการสุ่มค่าเป็นการปรับค่า W&B แบบ Nguyen-Widrow สำหรับ Hyperbolic Tangent Sigmoid Transfer Function

- `>>nwlog`

คำสั่งในการสุ่มค่าเป็นการปรับค่า W&B แบบ Nguyen-Widrow สำหรับ Log Sigmoid Transfer Function

ตัวอย่างรูปแบบ คำสั่ง : `>>randS, nwtan, nwlog`

```
>>[Wn,Bn]=randS(UBn,[row x col]);
>>[Wn,Bn]=nwtan(UBn,[row x col]);
>>[Wn,Bn]=nwlog(UBn,[row x col]);
                                     %Rem Wn,Bn = Weight & Bias.
                                     %Rem UBn = Number of Neural per Layer.
```

5) การฝึกสอน (Training)

- Incremental Training (Adapt)

คำสั่งการฝึกสอนที่มีการปรับค่าถ่วงน้ำหนักและค่าไบแอส (W&B) ทุกๆ ครั้งที่อินพุตเข้ามา 1 ครั้ง และจะปรับค่าไปเรื่อยๆ จนหมดมีรูปแบบคำสั่งคือ

```
>>adaptwh
```

- Batch Training (Train)

รูปแบบการฝึกสอนที่จะมีการปรับค่า W&B เพียงครั้งเดียว ภายหลังจาก รับอินพุตทั้งหมดเข้ามาแล้ว ซึ่งการฝึกสอนแบบนี้มีรูปแบบคำสั่งดังนี้

```
>>trainbp
```

```
:train feed-forward network with backpropagation
```

```
>>trainbpx
```

```
:train feed-forward network with faster backpropagation
```

```
>>trainelm
```

```
:train Elman recurrent network
```

```
>>trainlm
```

```
:train feed-forward network with Levenberg-Marquardt
```

รูปแบบการเขียนโปรแกรม

```
>>[Wn1,Bn1,Wn2,Bn2,...,TE,TR]=TRAINBPX(W1,B1F1,W2,B2F2,...,P,T,TP)
```

```
>>TP[disp_freq,max_epoch,err_goal,Lr_inc,Lr_dec,momentum,err_ratio]
```

โดยค่าสัญลักษณ์ต่างๆ มีความหมายดังต่อไปนี้

W_i	--	Weight Matrix for the i th Layer
B_i	--	Bias Vector for the i th Layer
F_i	--	Transfer Function (String) for the i th Layer
P	--	$R \times Q$ Matrix of Input Vectors
T	--	$S \times Q$ matrix of Target Vectors
TP	--	Training Parameters (Optional)
W_{ni}	--	New Weights
B_{ni}	--	New Biases
TE	--	The Actual Number of Epochs Trained
TR	--	Training Record : [Row of Errors]

ในกระบวนการฝึกสอนนั้น จำเป็นที่จะต้องบอกขอบข่ายของความพอดี ในการฝึกสอน และความสามารถที่ตรงกันของการฝึกสอน ให้แก่โครงข่าย ดังนั้นต้องกำหนดพารามิเตอร์ (Parameter :TP) ที่เหมาะสมให้แก่โครงข่ายด้วย ซึ่งค่าพารามิเตอร์มีอยู่ 8 จำนวน คือ

ก) Disp_freq

จำนวนรอบที่จะแสดงข้อมูลการฝึกสอน

ข) Max_epoch

จำนวนการวนทำการปรับ W&B มากที่สุด

ค) Err_goal

ค่าของระดับความผิดพลาด ที่ยอมให้เกิดจากผลต่าง ระหว่างเอาต์

พุตและเป้าหมาย

ง) Lr

อัตราการเรียนรู้ โดยปกติประมาณ ≤ 0.01

จ) Lr_inc

ค่าตัวคูณเพิ่มของอัตราการเรียนรู้, โดยปกติประมาณ 1.05

ฉ) Lr_dec

ค่าตัวคูณลดของอัตราการเรียนรู้, โดยปกติประมาณ 0.7

ช) Momentum

ค่าโมเมนตัม, โดยปกติประมาณ 0.95

ข) Err_ratio

ค่าอัตราส่วนที่มากที่สุดของ ค่าความผิดพลาดใหม่ค่าความผิดพลาดเก่าที่ยอมให้ไม่มีการเปลี่ยนแปลงค่า W&B

4.1.3 การทดสอบโครงข่ายประสาทเทียมแบบแพร่กลับ (Test Backpropagation Network)

เมื่อทำการฝึกสอนให้โครงข่ายสามารถรู้จำโครงร่างภาพวัตถุแล้ว ก็จะได้ชุดข้อมูลค่าน้ำหนัก และค่าไบแอสในโครงข่าย

การทดสอบความสามารถของโครงข่าย โดยป้อนชุดข้อมูลค่าน้ำหนัก, ไบแอสและชุดข้อมูลอินพุต กลับเข้าไปในโครงข่าย เพื่อทำการคำนวณน้ำหนักในแต่ละชั้นซ่อน และ ชั้นเอาต์พุต มีคำสั่งดังนี้

```
>>Y = simuff(P, Wi, Bi, 'Fi' )
```

โดยที่

P -- Matrix of Input (Column) Vectors
W_i -- Weight Matrix of the *i*th Layer
B_i -- Bias (Column) Vector of the *i*th Layer
F_i -- Transfer Function of the *i*th Layer (String)
Y -- Output

ตัวอย่างรูปแบบ คำสั่ง : >>simuff

```
y1=simuff([p], W1,B2,UT1,W2,B2,UT2,W3,B3,UT3  

%rem Wn,Bn = Weight & Bias.  

%rem UBn = Number of Neural per Layer.  

%rem UTn = Types of Activation Function.
```

4.1.4 การเชื่อมโยงกับผู้ใช้ (User Interface)

การสร้าง Graphical User Interface (GUI) [4] , [12] เพื่อเชื่อมโยงกับผู้ใช้งาน มีอยู่หลายรูปแบบ เช่น uicontrol , uimenu งานวิจัยนี้เลือกใช้ uicontrol ทำหน้าที่เป็นตัวเชื่อมโยงการสร้าง uicontrol มีรูปแบบคำสั่งดังนี้

uicontrol (handle , 'PropertyName', PropertyValue,...)

ชนิดของ uicontrol มีด้วยกันหลายชนิดกำหนดการใช้งานโดยค่าของ พร็อพเพอร์ตี้ (Property) ประเภท style คือ

ก) Frame

สร้างเพื่อล้อมกรอบ พื้นที่ หรือ บริเวณ ที่ต้องการ เพื่อแยกตำแหน่ง ของปุ่มควบคุม ต่างๆ ที่อยู่ใน Figure เดียวกัน

ตัวอย่างรูปแบบ คำสั่ง : *Frame*

```
frame=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','frame',...
    'Position',[Left Bottom Width High]);
```

ข) Pushbutton

สร้างปุ่มที่เป็นลักษณะปุ่มคลิก การใช้งานให้ใช้ตัวชี้ของเมาส์เลือก และคลิกที่ปุ่ม ก็ให้ไปกระทำตามเงื่อนไขของโปรแกรม ที่กำหนดไว้

ตัวอย่างรูปแบบ คำสั่ง : *Pushbutton*

```
PushButton=uicontrol(gcf,...
    'Style','push',...
    'String','End.',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'close(all)']);
```

ค) Checkbox

เป็นปุ่มที่เลียนแบบการทำงาน คล้ายกับเป็นสวิตช์เลือก เปิด/ปิด คือ Checkbox ใดถูกเลือก จะมีลักษณะเป็นเปิด (On) ส่วน Checkbox ที่ไม่ถูกเลือกจะมีสถานะปิด (Off) โดยที่ Checkbox แต่ละอัน ไม่เกี่ยวข้องกัน

ง) Radiobutton

มีการทำงานที่คล้ายกันกับ Checkbox แต่สามารถเลือกได้เพียง 1 ทางเลือกเท่านั้น

นั้น

ตัวอย่างรูปแบบ คำสั่ง : *radio*

```
RadioButton=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Man',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(RadioButton,` ` Value'',1)']);
```

จ) Static text boxes

แสดงตัวอักษรหรือข้อความที่เป็นข้อมูลเพื่อใช้เป็นคำสั่งหรือคำแนะนำ โดยที่ผู้ใช้งานไม่สามารถจะเปลี่ยนแปลงแก้ไขได้

ตัวอย่างรูปแบบ คำสั่ง : *text*

```
Text=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','text',...
    'String','Good Morning',...
    'Position',[Left Bottom Width High]);
```

ฉ) Editable text boxes

เป็นอักขระ หรือข้อความ ที่ให้ผู้ใช้งานสามารถแก้ไขเปลี่ยนแปลงๆ ได้

ตัวอย่างรูปแบบ คำสั่ง : *edit*

```
EditText=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','edit',...
    'String','Filename',...
    'Position',[Left Bottom Width High]);
```

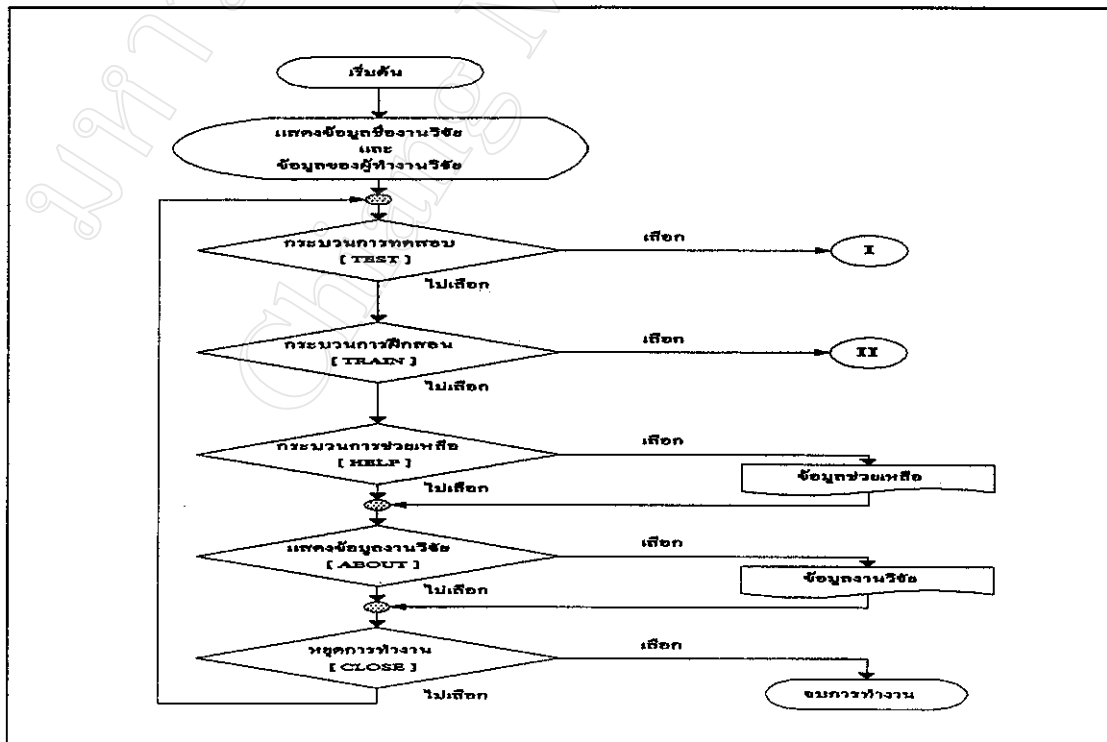
4.2 การออกแบบโปรแกรมการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์ค

การออกแบบสร้างโปรแกรมการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์ค สามารถแยกออกเป็นส่วนประกอบต่างๆ ดังนี้

- เมนูหลัก (Main Menu)
- กระบวนการฝึกสอน (Train)
- กระบวนการทดสอบ (Test)

4.2.1 เมนูหลัก (Main Menu)

ส่วนของ เมนูหลัก ในโปรแกรมการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์ค ทำหน้าที่เชื่อมโยง กระบวนการทำงานของส่วนย่อยที่สำคัญของโปรแกรม มีผังแสดงขั้นตอน การทำงานดังรูปที่ 4.7 โดยในส่วนของโปรแกรม ก็จะมีการแยกคำสั่งสร้างโปรแกรมขึ้นมาเป็นส่วนๆ ดังนี้



รูป 4.7 ผังการทำงานเมนูหลักของงานวิจัยการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์ค

รูปแบบ คำสั่งในงานวิจัย : ** เมนูหลัก **

```
%Rem Create Main Menu Interface.
%
%Rem Create Main Menu Windows.
Mainmenu=figure(...
    'Name','MAIN MENU',...
    'Color',[R G B]);
    'Position',[Left Bottom Width High],...

%Rem Create Frame userinterface.
Frame=uicontrol(gcf,...
    'Style','frame',...
    'BackgroundColor',[R G B]),...
    'Position',[Left Bottom Width High]);

%Rem Create Text.
Text=uicontrol(gcf,...
    'Style','text',...
    'String','MAIN MENU',...
    'BackgroundColor',[R G B]),...
    'Position',[Left Bottom Width High]);

%Rem Create Test Button.
TestButton=uicontrol(gcf,...
    'Style','push',...
    'String','TEST',...
    'Position',[Left Bottom Width High]),...
    'Callback',test_Program);

%Rem Create Train Button.
TrainButton=uicontrol(gcf,...
    'Style','push',...
    'String','TRAIN',...
    'Position',[Left Bottom Width High]),...
    'Callback',train_Program);

%Rem Create Help Button.
HelpButton=uicontrol(gcf,...
    'Style','push',...
    'String','HELP',...
    'Position',[Left Bottom Width High]),...
    'Callback',help_Program);

%Rem Create About Button.
AboutButton=uicontrol(gcf,...
    'Style','push',...
    'String','ABOUT',...
    'Position',[Left Bottom Width High]),...
    'Callback',about_Program);

%Rem Create Close Button.
CloseButton=uicontrol(gcf,...
    'Style','push',...
    'String','CLOSE',...
    'Position',[Left Bottom Width High]),...
    'Callback',close_Program);
```

คำอธิบาย คำสั่ง : เมนูหลัก

- *Create Main Menu Windows.*

สร้างหน้าต่าง เมนูหลัก ประกอบด้วย

- 'Name' : กำหนด ชื่อของหน้าต่าง
- 'Color' : กำหนด สีของหน้าต่าง
- 'Position' : กำหนด ตำแหน่งของหน้าต่าง

Left ระยะห่างจากขอบซ้าย , Bottom ระยะห่างจากขอบล่าง

Width ระยะความกว้าง , High ระยะความสูง

- *Create Frame UserInterfaces.*

สร้างกรอบล้อมพื้นที่ ประกอบด้วย

- 'Style' : กำหนด ชนิดของ UserInterfaces (Frame)
- 'Backgroundcolor' : กำหนด สีพื้น
- 'Position' : กำหนด ตำแหน่งของหน้าต่าง

- *Create Text.*

สร้างชุดข้อความ ประกอบด้วย

- 'Style' : กำหนด ชนิดของ UserInterfaces (Text)
- 'Backgroundcolor' : กำหนด สีพื้น
- 'String' : กำหนด ข้อความที่ต้องการแสดง
- 'Position' : กำหนด ตำแหน่งของหน้าต่าง

- *Create Push Button.*

สร้างปุ่มคลิก ประกอบด้วย

- 'Style' : กำหนด ชนิดของ UserInterfaces (Push)
- 'String' : กำหนด ชื่อปุ่มคลิก
- 'Position' : กำหนด ตำแหน่งของหน้าต่าง
- 'Callback' : กำหนด เงื่อนไขที่ต้องการเมื่อมีการคลิกปุ่ม
 - Test_Program โปรแกรมการทดสอบ
 - Train_Program โปรแกรมการฝึกสอน
 - Help_Program โปรแกรมช่วยเหลือ
 - About_Program โปรแกรมข้อมูลงานวิจัย
 - Close_Program โปรแกรมปิดการทำงาน

4.2.2 กระบวนการฝึกสอน (Train)

ผังขั้นตอนกระบวนการฝึกสอนของโครงข่ายประสาทเทียมแบบแพร่กลับ แสดงดังรูป 4.8 ซึ่งประกอบไปด้วยลำดับการทำงาน ดังนี้

ลำดับที่ 1 กำหนด, เก็บบันทึกค่าตัวแปรต่างๆ ของโครงข่ายประสาทเทียม

- กำหนด ค่าพารามิเตอร์
- กำหนด จำนวนชั้นซ่อน
- กำหนด จำนวนนิวรอนในแต่ละชั้น
- กำหนด ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น
- กำหนด ชนิดของการสุ่มค่าน้ำหนัก, ไบแอสเริ่มต้น
- กำหนด ชื่อแฟ้มข้อมูล ที่ใช้เก็บบันทึกค่าตัวแปร
- เก็บบันทึก ค่าตัวแปรของโครงข่ายฯ ลงแฟ้มข้อมูล
 - บันทึก ค่าพารามิเตอร์
 - บันทึก จำนวนชั้นซ่อน
 - บันทึก จำนวนนิวรอนในแต่ละชั้น
 - บันทึก ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น

ลำดับที่ 2 สร้างรูปแบบ และ วิธีการฝึกสอน

- รูปแบบการฝึกสอน
 - ฝึกสอนครั้งละ กลุ่ม
 - ฝึกสอนครั้งละ ภาพ
- วิธีการฝึกสอน
 - ฝึกสอน ครั้งแรก
 - ฝึกสอน ซ้ำเดิม

ลำดับที่ 3 สร้างค่าน้ำหนัก และ ไบแอส ให้กับโครงข่ายประสาทเทียม

- สุ่มค่าเริ่มต้นสำหรับ ค่าน้ำหนัก และ ไบแอส
- นำค่าน้ำหนัก และ ไบแอส จากแฟ้มข้อมูล

ลำดับที่ 4 ป้อนชุดข้อมูลที่ต้องการวิเคราะห์เข้าไปในชั้นอินพุต

- เลือกภาพครั้งละภาพ และครั้งละกลุ่ม เพื่อทำการฝึกสอนโครงข่าย
- ปรับแต่งภาพ
 - ปรับความสว่างภาพ

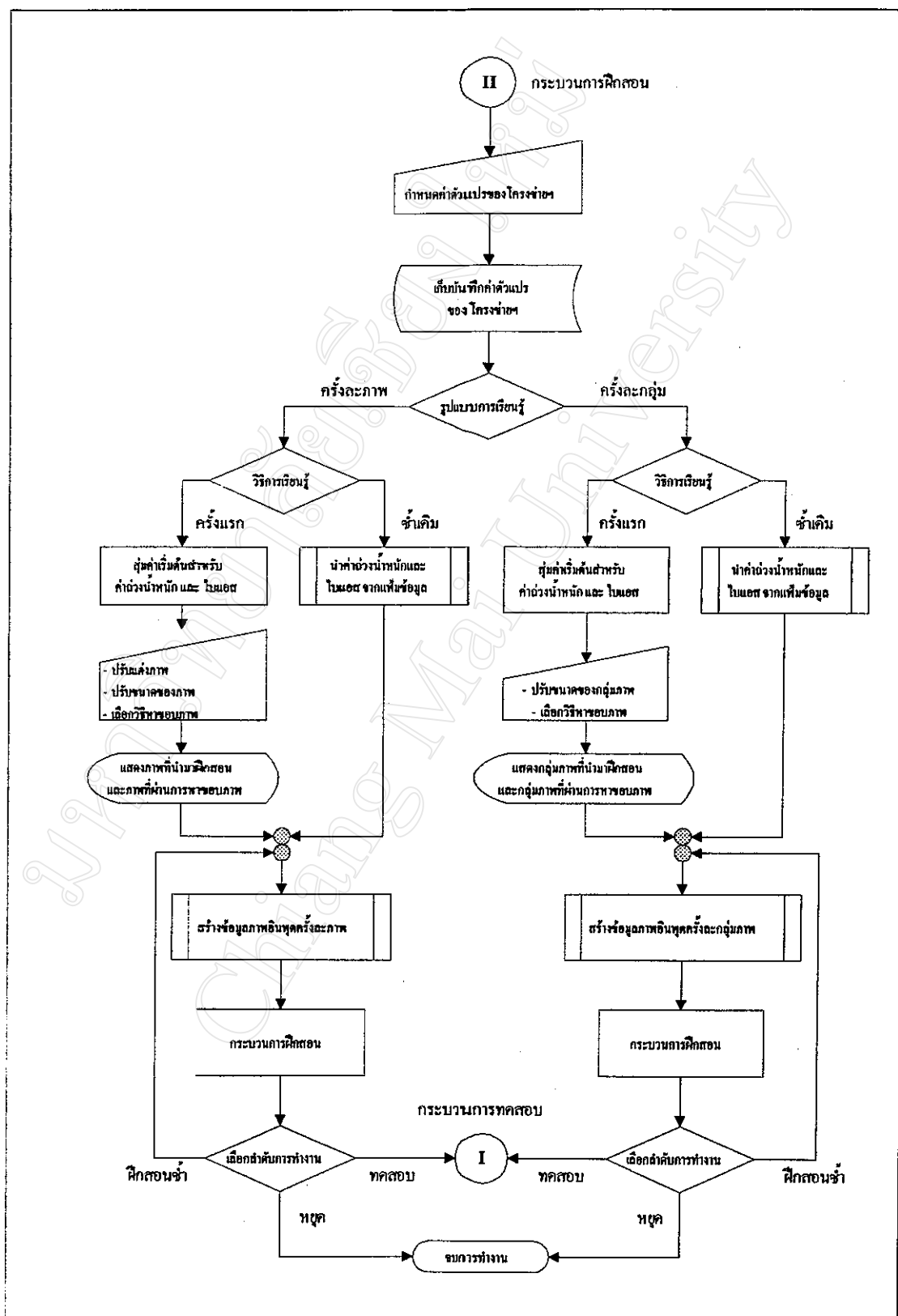
- ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ
- ปรับค่าองค์ประกอบ แกมมา ภาพ
- ปรับขนาดภาพ
 - ปรับขนาดภาพ
 - แบบอัตโนมัติ
 - แบบร้อยละ
 - แบบจัตุรัส
 - แบบขนาดใดๆ
 - กำหนดขนาดภาพ
 - กำหนดขนาดเมตริกซ์ภาพ
 - กำหนดขนาด แถว และ สดมภ์ เมตริกซ์ภาพ
 - คำนวณเมตริกซ์ภาพ
- เลือกวิธีการหาขอบภาพ
 - วิธีของ ‘ Sobel ’
 - วิธีของ ‘ Prewitt ’
 - วิธีของ ‘ Roberts ’
- แสดงภาพที่นำมาฝึกสอน และ ภาพที่ผ่านการหาขอบภาพ

ลำดับที่ 5 สร้างข้อมูลภาพอินพุต เพื่อป้อนเข้าไปยังโครงข่ายฯ

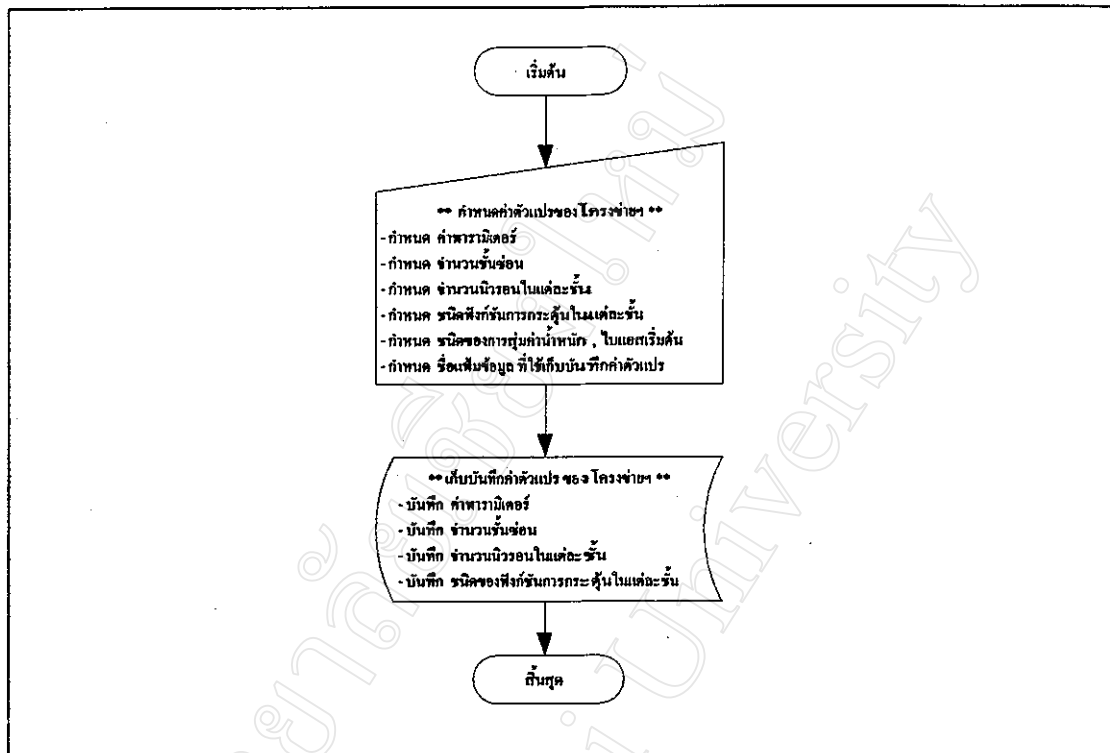
- สร้างข้อมูลภาพอินพุตครั้งละภาพ
- สร้างข้อมูลภาพอินพุตครั้งละกลุ่มภาพ

ลำดับที่ 6 กระบวนการฝึกสอนทำการคำนวณค่าน้ำหนักของชั้นซ่อนและชั้นเอาต์พุต เพื่อหาคำตอบเอาต์พุตของ โครงข่ายฯ

- กำหนดเป้าหมายเอาต์พุต ครั้งละภาพ , ครั้งละกลุ่ม
- กำหนดค่าน้ำหนัก และ ไบแอส เริ่มต้นให้แก่โครงข่าย
- ทำการฝึกสอนโครงข่าย
- แสดงกราฟระดับค่าความผิดพลาด ระหว่างเอาต์พุตและเป้าหมาย
- บันทึกค่าน้ำหนักและไบแอส ของโครงข่าย



รูป 4.8 ฟังก์ชันตอนกระบวนการฝึกสอนของโครงข่ายประสาทเทียมแบบแพร่กลับ



รูป 4.9 ฟังก์ชันตอนกระบวนการฝึกสอน ลำดับที่ 1

ลำดับที่ 1 เก็บบันทึก , กำหนดค่าตัวแปรต่างๆ ของโครงข่ายประสาทเทียม มีฟังก์ชันตอนดังรูป 4.9 ซึ่งค่าตัวแปรต่างๆ ประกอบไปด้วย

- กำหนด ค่าพารามิเตอร์ [TP]
 - Disp_freq Default = 25
 - Max_epoch Default = 3000
 - Err_goal Default = 0.01
 - Lr Default = 0.01
 - Lr_inc Default = 1.06
 - Lr_dec Default = 0.7
 - Momentum Default = 0.95
- กำหนด จำนวนชั้นซ่อน [nolay]
 - 1 Layer
 - 2 Layers

- กำหนดจำนวนนิวรอนในแต่ละชั้น [UBn]
 - UB1 จำนวนนิวรอนในชั้นซ่อนที่ หนึ่ง
 - UB2 จำนวนนิวรอนในชั้นซ่อนที่ สอง
 - UB3 จำนวนนิวรอนในชั้นเอาต์พุต
- กำหนด ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น [UTn]
ประกอบไปด้วยชนิด Tansig , Logsig , Purelin
 - UT1 ชนิดของฟังก์ชันการกระตุ้นชั้นซ่อนที่ หนึ่ง
 - UT2 ชนิดของฟังก์ชันการกระตุ้นชั้นซ่อนที่ สอง
 - UT3 ชนิดของฟังก์ชันการกระตุ้นชั้นเอาต์พุต
- กำหนด ชนิดของการสุ่มค่าน้ำหนัก และไบแอส เริ่มต้น [Wn , Bn]
ประกอบไปด้วยชนิด NwTansig , NwLogsig , Random
 - NwTansig ชนิดของการสุ่มน้ำหนัก และ ไบแอส เริ่มต้น
 - NwLogsig ชนิดของการสุ่มน้ำหนัก และ ไบแอส เริ่มต้น
 - Random ชนิดของการสุ่มน้ำหนัก และ ไบแอส เริ่มต้น
- กำหนด ชื่อแฟ้มข้อมูล ที่ใช้เก็บบันทึกค่าตัวแปร
 - Filename_n ชื่อแฟ้มข้อมูลที่ใช้จัดเก็บค่าตัวแปร
- เก็บบันทึกค่าตัวแปร ของโครงข่ายฯ ลงแฟ้มข้อมูล [Nfile]
 - TP ค่าพารามิเตอร์
 - nolay จำนวนชั้นซ่อน
 - UBn จำนวนนิวรอนในแต่ละชั้น
 - UTn ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น

รูปแบบ คำสั่งในงานวิจัย : ** กระบวนการฝึกสอน **

```
%Rem Create Training Processing.
%
%Rem Create Backpropagation Neural Network Windows.
Network=figure(...
    'Name', 'Create Backpropagation Neural Network',...
    'Color', [R G B],...
    'Position', [Left Bottom Width High]);
```

รูปแบบ คำสั่งในงานวิจัย : กำหนดค่าพารามิเตอร์

```
%Rem Set Configuration Networks Parameter.
Edit_Parameter=uicontrol (...
    'BackgroundColor', 'w',...
    'ForegroundColor', 'b',...
    'Style', 'edit',...
    'String', Parameter, ...
    'Position', [Left Bottom Width High]);
```

คำอธิบาย คำสั่ง : กำหนดค่า พารามิเตอร์

- 'Edit_Parameter'

เป็นส่วนที่ใช้กำหนดค่าพารามิเตอร์ทั้งหมด จัด จำนวน เพื่อกำหนดเงื่อนไข การฝึกสอนของโครงข่าย โดยการเติมค่า ที่ต้องการลงไป ซึ่งค่าตัวแปรเหล่านี้จะถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String' , 'Parameter' ,...

```
%Rem Parameter 1st = disp_freq.
%Rem Parameter 2nd = max_epoch.
%Rem Parameter 3rd = err_goal.
%Rem Parameter 4th = lerning rate.
%Rem Parameter 5th = lerning rate_increment.
%Rem Parameter 6th = lerning rate_decrement.
%Rem Parameter 7th = momentum.
```

รูปแบบ คำสั่งในงานวิจัย : กำหนด จำนวนชั้นซ่อน

```
%Rem Set Number of Hidden Layers.
Pop_layer=uicontrol(...
    'Style','popup',...
    'String','1 Layer|2 Layers',...
    'Position',[Left Bottom Width High],...
    'Callback',{...
        'nolay = popupstr(Pop_layer)'}]);
```

คำอธิบาย คำสั่ง : กำหนด จำนวนชั้นซ่อน

- 'Pop_layer'

เป็นเงื่อนไข ในการกำหนดจำนวนชั้นซ่อนของโครงข่าย ให้มีค่าเป็นจำนวน หนึ่ง ชั้นซ่อน หรือ จำนวน สอง ชั้นซ่อน คำตัวแปรจะถูกเก็บไว้เป็นค่า String ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', '1 Layer | 2 Layers',...

: Pop_layer = '1 Layer', เมื่อเลือกจำนวน หนึ่ง ชั้นซ่อน หรือ

: Pop_layer = '2 Layers', เมื่อเลือกจำนวน สอง ชั้นซ่อน

- 'Callback', 'nolay = popupstr(Pop_layer)');

: กำหนดให้ nolay เก็บค่า จำนวนชั้นซ่อนที่เลือกไว้เพื่อนำไปใช้กำหนดเงื่อนไขในการฝึกสอน ต่อไป

รูปแบบ คำสั่งในงานวิจัย : กำหนดจำนวน นิวรอน ในแต่ละชั้น

```
%Rem Set Number Neuron of Layers.
Edit_Neuron=uicontrol(...
    'BackgroundColor','w',...
    'ForegroundColor','b',...
    'Style','edit',...
    'String',UBn,...
    'Position',[Left Bottom Width High]);
```

คำอธิบาย คำสั่ง : กำหนดจำนวน นิวรอน ในแต่ละชั้น

- 'Edit_Neuron'

เป็นส่วนที่ใช้กำหนดค่าจำนวน นิวรอน ในแต่ละชั้นของโครงข่าย เพื่อ กำหนดเงื่อนไข การฝึกสอนของโครงข่าย โดยการเติมค่า ที่ต้องการลงไป ซึ่งค่าตัวแปรเหล่านี้จะ ถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'UBn', ...
 - : กำหนดให้ UB1 เก็บค่า จำนวน นิวรอน ในชั้นซ่อนที่ หนึ่ง
 - : กำหนดให้ UB2 เก็บค่า จำนวน นิวรอน ในชั้นซ่อนที่ สอง
 - : กำหนดให้ UB3 เก็บค่า จำนวน นิวรอน ในชั้นเอาต์พุต

รูปแบบ คำสั่งในงานวิจัย : กำหนด ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น

```
%Rem Set Activation Function of Layers.
%
Pop_Function_1=icontrol(...
    'Style','popup',...
    'String','tansig|logsig|Purelin',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'UTn = popupstr(Pop_Function)']);
```

คำอธิบาย คำสั่ง : กำหนด ชนิดฟังก์ชันการกระตุ้นในแต่ละชั้น

- 'Pop_Function'

เป็นเงื่อนไข ในการกำหนดชนิดฟังก์ชันการกระตุ้น ในชั้นของโครงข่าย ให้มีค่าเป็นชนิด Tansig หรือ Logsig หรือ Purelin ค่าตัวแปรจะถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'tansig | logsig | Purelin ', ...
 - : Pop_Function = 'tansig', เมื่อเลือก tansig หรือ
 - : Pop_Function = 'logsig', เมื่อเลือก logsig หรือ
 - : Pop_Function = 'purelin', เมื่อเลือก purelin
- 'Callback', 'UTn = popupstr(Pop_Function)');

: กำหนดให้ UT_n เก็บค่า ชนิดของฟังก์ชันการกระตุ้นในแต่ละชั้น
ของโครงข่ายไว้เพื่อนำไปใช้กำหนดเงื่อนไขในการฝึกสอนต่อไป

รูปแบบ คำสั่งในงานวิจัย : กำหนดชนิดของการสุ่มค่าน้ำหนัก, ไบแอสเริ่มต้น

```
%Rem Initial Random Weight & Bias.
Rb_Tan=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Nwtansig',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Tan,''Value'',1),'...
        'set(Rb_Log,''Value'',0),'...
        'set(Rb_Gen,''Value'',0)']]);
    %Rem Ramdom Weight&Bias by Nguyen-Widrow tan.

Rb_Log=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Nwlogsig',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Tan,''Value'',0),'...
        'set(Rb_Log,''Value'',1),'...
        'set(Rb_Gen,''Value'',0)']]);
    %Rem Ramdom Weight&Bias by Nguyen-Widrow log.

Rb_Gen=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Random',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Tan,''Value'',0),'...
        'set(Rb_Log,''Value'',0),'...
        'set(Rb_Gen,''Value'',1)']]);
```

คำอธิบาย คำสั่ง : กำหนดชนิดของการสุ่มค่าน้ำหนัก, ไบแอสเริ่มต้น

• 'Rb_Tan', 'Rb_Log', 'Rb_Gen'

เป็นเงื่อนไข ในการกำหนดชนิดของการสุ่มค่าน้ำหนัก และ ไบแอสเริ่มต้นสำหรับโครงข่าย ให้มีค่าเป็นชนิด Nwtansig หรือ NwLogsig หรือ Random ค่าตัวแปรจะถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'NWtansig',...
- 'String', 'NWlogsig',...
- 'String', 'Random',...
- 'Callback',
 - : Rb_Tan, 'Value' = '1', เมื่อเลือก NWtansig หรือ
 - : Rb_Log, 'Value' = '1', เมื่อเลือก NWlogsig หรือ
 - : Rb_Gen, 'Value' = '1', เมื่อเลือก Random

รูปแบบ คำสั่งในงานวิจัย : กำหนดชื่อแฟ้มข้อมูล ที่ใช้เก็บบันทึกค่าตัวแปร

```
%Rem Save Network Parameter to File.
Edit_Save_Filename=uicontrol(...
    'BackgroundColor', 'w',...
    'ForegroundColor', 'b',...
    'Style', 'edit',...
    'String', 'filename',...
    'Position', [Left Bottom Width High]);
    %Rem Fill Filename for Save .

Pb_Save=uicontrol(...
    'Style', 'push',...
    'String', 'Save',...
    'Position', [Left Bottom Width High],...
    'Callback', [...
        'file=filename',...
        'Nfile=strcat(Sfile, '_n'),'...'
        'save(Nfile,nolay,UTn,UBn,TP)']]);
```

คำอธิบาย คำสั่ง : กำหนดชื่อแฟ้มข้อมูล ที่ใช้เก็บบันทึกค่าตัวแปร

• 'Edit_Save_Filename'

เป็นส่วนที่ใช้กำหนดชื่อแฟ้มข้อมูล ที่ใช้เก็บบันทึกค่าตัวแปรของโครงข่ายฯ ลงไว้ในแฟ้มข้อมูล เพื่อที่สามารถเรียกกลับนำใช้ได้อีกในภายหลัง โดยการเติมชื่อแฟ้มข้อมูลที่ต้องการลงไป ซึ่งค่าตัวแปรเหล่านี้ถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'filename',...
 - : กำหนดให้ filename เป็นชื่อของแฟ้มข้อมูล

- 'Pb_Save'

เป็นปุ่มคลิกเพื่อทำตามเงื่อนไข ถูกกำหนดใช้ในการเก็บบันทึก คำตัวแปรของโครงข่ายฯ ลงไว้ใน แฟ้มข้อมูล

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'Save', ...

: ชื่อของปุ่มกำหนดเป็น Save

- 'Callback', [...

'Sfile = filename,'...

'Nfile = strcat(Sfile, '_n'),'...

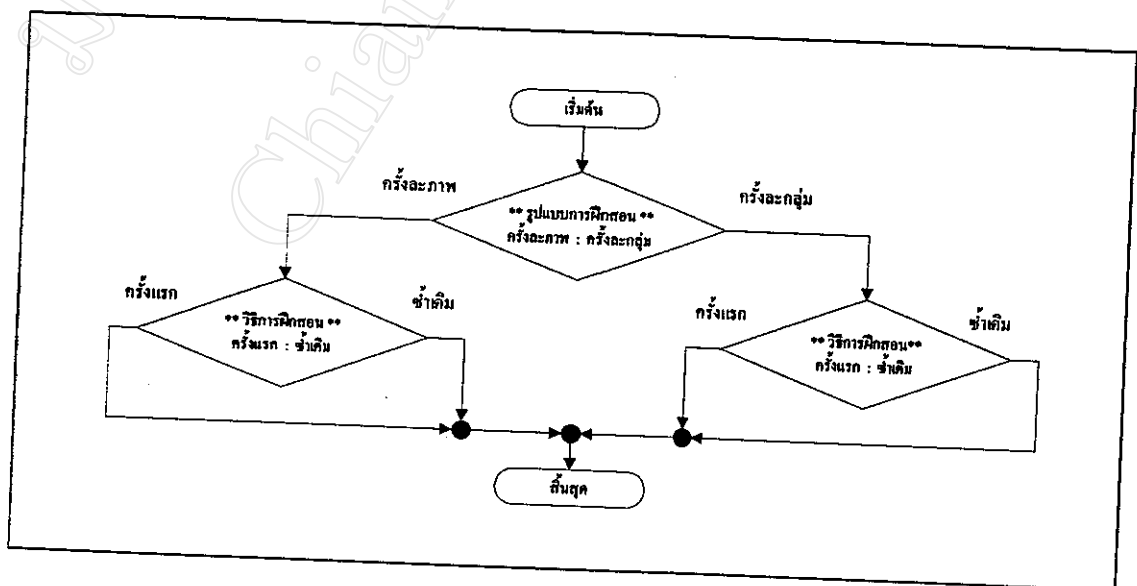
'save (Nfile, nolay, UTn, UBn, TP)');

: Sfile มีค่าเท่ากับชื่อของแฟ้มข้อมูล

: Nfile มีค่าเท่ากับชื่อของแฟ้มข้อมูลต่อท้ายด้วย _n โดยที่ คำสั่ง strcat ทำหน้าที่เอาค่า String ตัวแรก มาต่อเข้ากับ String ตัวที่สอง เพราะฉะนั้น

Nfile = filename_n เพื่อบอกให้ทราบว่า เป็น แฟ้มข้อมูลที่ใช้เก็บ คำตัวแปรของโครงข่ายฯ

: save ทำหน้าที่เก็บค่าของ จำนวนชั้นซ่อน , ฟังก์ชันการกระตุ้น , จำนวนนิวรอนในแต่ละชั้น และ ค่าพารามิเตอร์ ไว้ในแฟ้มข้อมูล ที่ถูกกำหนดชื่อโดย Nfile



รูป 4.10 ฟังก์ชันตอนกระบวนการฝึกสอน ลำดับที่ 2

ลำดับที่ 2 สร้างรูปแบบ และ วิธีการฝึกสอน เพื่อเลือกวิธีการป้อนข้อมูล อินพุต เข้าไปยังชั้นอินพุตของโครงข่ายฯ มีผังขั้นตอนการทำงาน ดังรูป 4.10 โดยมีวิธีการ ดังนี้

- รูปแบบการฝึกสอน
 - Group ฝึกสอนครั้งละกลุ่ม
 - One ฝึกสอนครั้งละภาพ
- วิธีการฝึกสอน
 - New train ฝึกสอนครั้งแรก
 - Train again ฝึกสอนซ้ำเดิม

รูปแบบ คำสั่งในงานวิจัย : รูปแบบการฝึกสอน ครั้งละกลุ่ม , ครั้งละภาพ

```
%Rem Networks Pattern Training.
%
%Rem Pattern Training by Group.
Rb_Group=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','group',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Group, 'Value',1),'...
        'set(Rb_One, 'Value',0)']);

%Rem Pattern Training by One.
Rb_One=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','one',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Group, 'Value',0),'...
        'set(Rb_One, 'Value',1)']);
```

คำอธิบาย คำสั่ง : รูปแบบการฝึกสอน ครั้งละกลุ่ม , ครั้งละภาพ

- 'Rb_One' , 'Rb_Group'

เป็นเงื่อนไข ในการกำหนดรูปแบบการฝึกสอนสำหรับโครงข่ายฯ ให้มีรูปแบบเป็น การฝึกสอนแบบครั้งละภาพ หรือ การฝึกสอนแบบครั้งละกลุ่ม หากเงื่อนไขใดถูกเลือก จะมีค่า Value = 1

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'Style', 'radio', ...
- 'String', 'One', ...
- 'String', 'Group', ...
 - : ชื่อของปุ่มกำหนดเป็น One
 - : ชื่อของปุ่มกำหนดเป็น Group
- 'Callback', [...
 - : Rb_One , 'Value' = '1' , เมื่อเลือกการฝึกสอนแบบครั้งละภาพ
 - : Rb_Group , 'Value' = '1' , เมื่อเลือกการฝึกสอนแบบครั้งละกลุ่ม

รูปแบบ คำสั่งในงานวิจัย : วิธีการฝึกสอน ครั้งแรก , ซ้ำเดิม

```
%Rem Networks Method Training.
%
%Rem Method Training by New Train.
%Rem Initial W&B by Random.
Rb_New=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','New Train',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_New, 'Value',1),...
        'set(Rb_Old, 'Value',0)']);

%Rem Method Training by Train Again.
Rb_Old=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Train Again',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_New, 'Value',0),...
        'set(Rb_Old, 'Value',1)']);
```

คำอธิบาย คำสั่ง : วิธีการฝึกสอน ครั้งแรก , ซ้ำเดิม

- 'Rb_New' , 'Rb_Old'

เป็นเงื่อนไข ในการกำหนดวิธีการฝึกสอนสำหรับโครงข่ายฯ ให้มีวิธีการ
เป็น การฝึกสอนครั้งแรก หรือ การฝึกสอนซ้ำเดิม หากเงื่อนไขใดถูกเลือก จะมีค่า Value = 1
ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

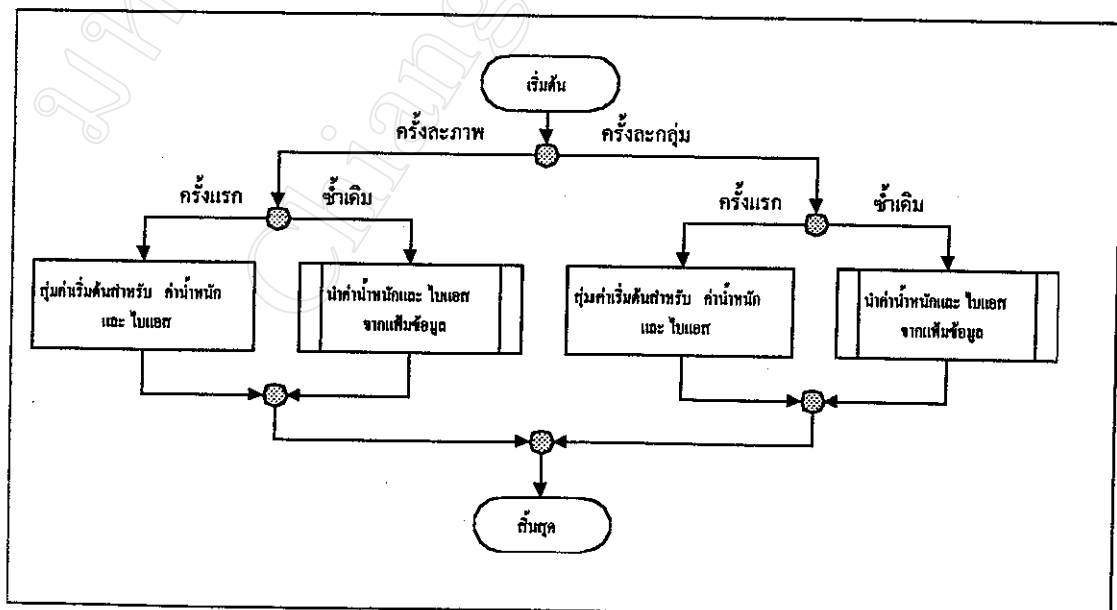
- 'Style' , 'radio' , ...
- 'String' , 'New Train' , ...
- 'String' , 'Train Again' , ...
 - : ชื่อของปุ่มกำหนดเป็น New Train
 - : ชื่อของปุ่มกำหนดเป็น Train Again
- 'Callback' , [...
 - : Rb_New , 'Value' = '1' , เมื่อเลือกวิธีการฝึกสอนครั้งแรก
 - : Rb_Old , 'Value' = '1' , เมื่อเลือกวิธีการฝึกสอนซ้ำเดิม

ลำดับที่ 3 สร้างค่าน้ำหนัก และ ไบแอส ให้กับโครงข่ายฯ โดยมีการกำหนดวิธีการ จากวิธีการฝึกสอน ดังรูปผังขั้นตอนการทำงาน รูปที่ 4.11 และมีรายละเอียดดังนี้

- วิธีการฝึกสอน

กำหนดเงื่อนไข จากวิธีการฝึกสอน

- New train คู่ค่าเริ่มต้นสำหรับค่าน้ำหนักและไบแอส
- Train again นำค่าน้ำหนักและไบแอส จากแฟ้มข้อมูล



รูป 4.11 ผังขั้นตอนกระบวนการฝึกสอน ลำดับที่ 3

รูปแบบ คำสั่งในงานวิจัย : สุ่มค่าเริ่มต้นสำหรับ ค่าน้ำหนัก และไบแอส

```
%Rem Initial Random Weight & Bias.
Case 'Nwtansig', [Wn,Bn]=nwtan(UBn,(row * col);
Case 'Nwlogsig', [Wn,Bn]=nwlog(UBn,(row * col);
Case 'Random' , [Wn,Bn]=rands(UBn,(row * col);
```

คำอธิบาย คำสั่ง : สุ่มค่าเริ่มต้นสำหรับ ค่าน้ำหนัก และไบแอส

- Initial Random Weight & Bias

เป็นวิธีที่ถูกกำหนดเงื่อนไขจากวิธีการฝึกสอนแบบ ฝึกสอนครั้งแรก โดยการกำหนดค่าน้ำหนัก และ ไบแอส ของโครงข่ายฯ เริ่มต้น ให้เป็นแบบ Nwtan หรือ NwLog หรือ rands

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- Rb_Tan,Value, '1' , [Wn , Bn] = nwtan (UBn , (row x col) ',...
Rb_Log,Value, '1' , [Wn , Bn] = nwlog (UBn , (row x col) ',...
Rb_Gen,Value, '1' , [Wn , Bn] = rands (UBn , (row x col) ',...
: กำหนดชนิดของการสุ่มค่าน้ำหนักและไบแอสให้แก่โครงข่ายฯ
เพื่อใช้ เป็นข้อมูลในการฝึกสอน ต่อไป

รูปแบบ คำสั่งในงานวิจัย : นำค่าน้ำหนัก และไบแอส จากแฟ้มข้อมูล

```
%Rem Load Network Parameter and Weight & Bias From File.
%Rem Fill Filename of File.
Edit_Load_Filename=icontrol(...
    'BackgroundColor','w',...
    'ForegroundColor','b',...
    'Style','edit',...
    'String','filename',...
    'Position',[Left Bottom Width High]);

Pb_Load=icontrol(...
    'Style','push',...
    'String','Load',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'Lfile=filename','...
        'Nfile=strcat(Lfile, '_n'),'...
        'Wfile=strcat(Lfile, '_w'),'...
        'load(Nfile)',...
        'load(Wfile)']]);
```

คำอธิบาย คำสั่ง : นำค่าน้ำหนัก และ ไบแอส จากเพิ่มข้อมูล

- *'Edit_Load_Filename'*

เป็นส่วนที่ใช้กำหนดชื่อเพิ่มข้อมูล ที่ต้องการนำข้อมูลที่ถูเก็บบันทึก ค่าตัวแปรของโครงข่ายฯ จากเพิ่มข้อมูล เพื่อที่สามารถเรียกกลับนำใช้ได้อีกในภายหลัง โดยการเติมชื่อที่ต้องการลงไป ซึ่งค่าตัวแปรเหล่านี้ถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- *'String', 'filename', ...*

: กำหนดให้ filename เป็นชื่อของเพิ่มข้อมูล

- *'Pb_Load'*

เป็นปุ่มคลิกเพื่อทำตามเงื่อนไข ถูกกำหนดใช้ในการนำค่าตัวแปร ค่าน้ำหนัก และไบแอส ของโครงข่ายที่ถูกเก็บบันทึก จากเพิ่มข้อมูล

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- *'String', 'Load', ...*

: ชื่อของปุ่มกำหนดเป็น Load

- *'Callback', [...*

'Lfile = filename,' ...

'Nfile = strcat(Lfile, '_n'),' ...

'Wfile = strcat(Lfile, '_w'),' ...

'Load (Nfile),' ...

'Load (Wfile)')];

: Lfile มีค่าเท่ากับชื่อของเพิ่มข้อมูล

: Nfile มีค่าเท่ากับชื่อของเพิ่มข้อมูลต่อท้ายด้วย *_n* โดยที่ คำสั่ง strcat ทำหน้าที่เอาค่า String ตัวแรก มาต่อเข้ากับ String ตัวที่สอง เพราะฉะนั้น

Nfile = filename_n เพื่อบอกให้ทราบว่า เป็นเพิ่มข้อมูลที่ใช้เก็บ ค่าตัวแปรของโครงข่ายฯ

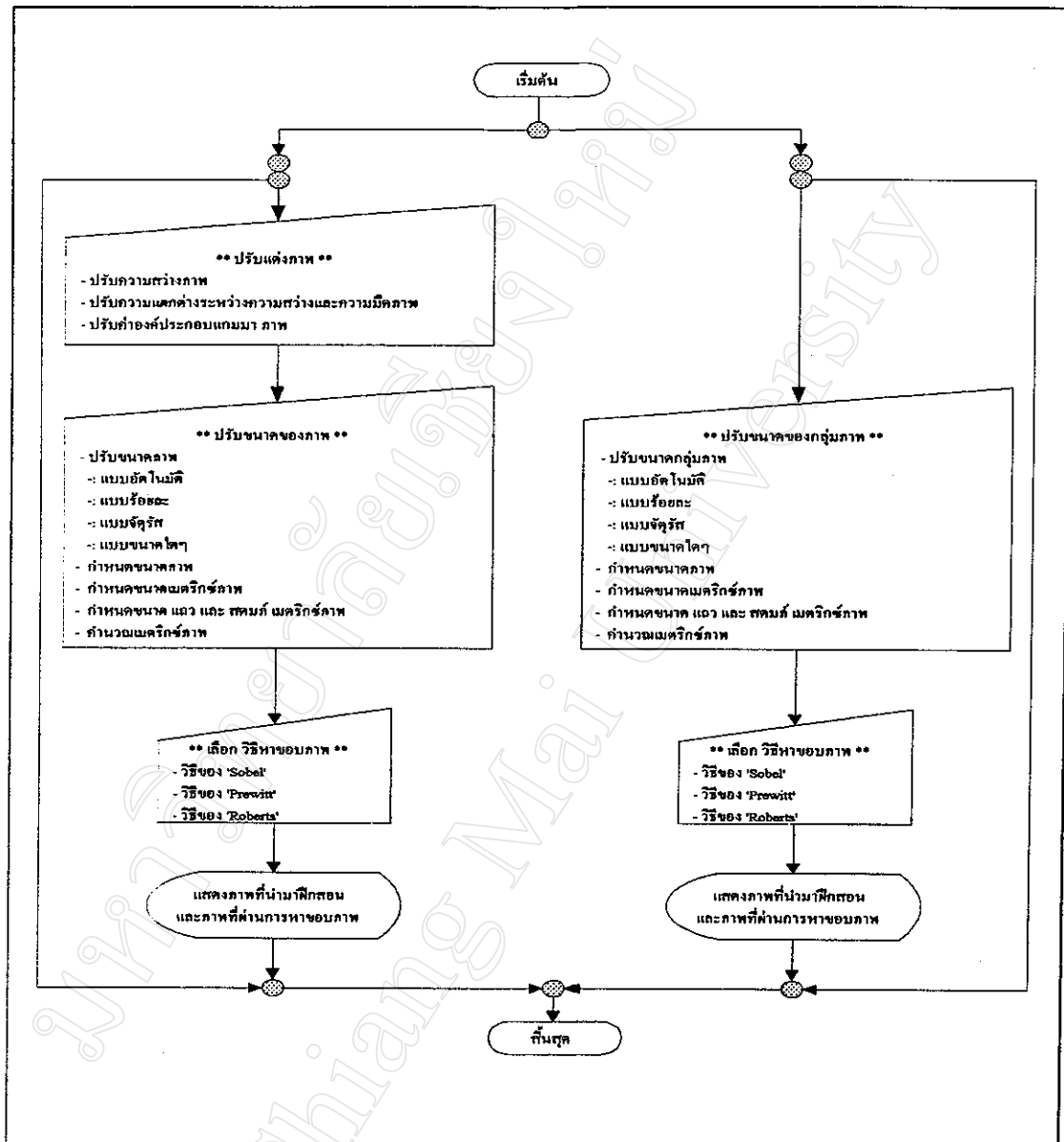
: Wfile มีค่าเท่ากับชื่อของเพิ่มข้อมูลต่อท้ายด้วย *_w* โดยที่ คำสั่ง strcat ทำหน้าที่เอาค่า String ตัวแรก มาต่อเข้ากับ String ตัวที่สอง เพราะฉะนั้น

Wfile = filename_w เพื่อบอกให้ทราบว่า เป็นเพิ่มข้อมูลที่ใช้เก็บ ค่าน้ำหนัก และ ไบแอสของโครงข่ายฯ

: Load ทำหน้าที่นำค่าของ จำนวนชั้นซ่อน , ฟังก์ชันการกระตุ้น , จำนวนนิวรอนในแต่ละชั้น และ ค่าพารามิเตอร์ จากเพิ่มข้อมูล ที่ถูกกำหนดชื่อโดย Nfile ส่วนค่าของ น้ำหนัก และไบแอส นำมาจากเพิ่มข้อมูล ที่ถูกกำหนดชื่อโดย Wfile

ลำดับที่ 4 ป้อนชุดข้อมูลที่ต้องการวิเคราะห์เข้าไปในชั้นอินพุตเป็นขั้นตอนกระบวนการจัดการภาพที่ต้องการนำมาป้อนเข้าไปในโครงข่ายเพื่อทำการฝึกสอนมีผังขั้นตอนการทำงาน รูปที่ 4.12 และมีรายละเอียดดังนี้

- เลือกภาพครึ่งละภาพ และครึ่งละกลุ่ม เพื่อทำการฝึกสอนโครงข่าย
- ปรับแต่งภาพ
 - ปรับความสว่างภาพ
 - ปรับความแตกต่างระหว่างความสว่างกับความมืด ภาพ
 - ปรับค่าองค์ประกอบ แกมมา ภาพ
- ปรับขนาดภาพ
 - ปรับขนาดภาพ
 - แบบอัตโนมัติ
 - แบบร้อยละ
 - แบบจัดรูป
 - แบบขนาดใดๆ
 - กำหนดขนาดภาพ
 - กำหนดขนาดเมตริกซ์ภาพ
 - กำหนดขนาด แถว และ สดมภ์ เมตริกซ์ภาพ
 - กำหนดเมตริกซ์ภาพ
- เลือกวิธีการหาขอบภาพ
 - วิธีของ ' Sobel '
 - วิธีของ ' Prewitt '
 - วิธีของ ' Roberts '
- แสดงภาพที่นำมาฝึกสอน และ ภาพที่ผ่านการหาขอบภาพ



รูป 4.12 ฟังก์ชันคอนเวอร์ชันการฝึกสอน ลำดับที่ 4

รูปแบบ คำสั่งในงานวิจัย : เลือกภาพที่นำมาฝึกสอน

```

%Rem Select Images to Trained One by One.
Pop_filename=uicontrol(...
    'Style', 'popup',...
    'String', 'man|animal|article',...
    'Position', [Left Bottom Width High],...
    'Callback', [...
        'filename = popupstr(Pop_filename)']);
  
```

```

Pop_n_ber=uicontrol(...
    'Style','popup',...
    'String','1-30',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'n_ber = popupstr(Pop_n_ber)']);
im=strcat(filename,n_ber, '.tif'),

```

คำอธิบาย คำสั่ง : เลือกภาพที่นำมาฝึกสอน

- 'Pop_filename'

เป็นเงื่อนไข ในการกำหนดชื่อของภาพ ที่ต้องการนำมาป้อนให้โครงข่าย โดยแบ่งออกเป็น 3 กลุ่มตัวอย่าง คือ man หรือ animal หรือ article ค่าตัวแปรจะถูกเก็บไว้เป็นค่า String

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', 'man | animal | article',...

: Pop_filename = 'man' ,เมื่อเลือกชื่อกลุ่มภาพ man หรือ

: Pop_filename = 'animal' ,เมื่อเลือกชื่อกลุ่มภาพ animal หรือ

: Pop_filename = 'article' ,เมื่อเลือกชื่อกลุ่มภาพ article

- 'Callback', 'filename = popupstr(Pop_filename)'];

: คำสั่ง popupstr นำค่า String ที่เก็บไว้ใน Pop_filename มาเก็บไว้ในตัวแปรชื่อ filename

- 'Pop_n_ber'

เป็นเงื่อนไข ในการกำหนดลำดับที่ของภาพ ที่ต้องการนำมาป้อนให้โครงข่าย โดยมีค่าตั้งแต่รูปที่ 1 ถึงรูปที่ 30 ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'String', '1-30',...

: Pop_n_ber = '1-30' ,เมื่อเลือกลำดับที่ของภาพ

- 'Callback', 'n_ber = popupstr(Pop_n_ber)'];

: คำสั่ง popupstr นำค่า String ที่เก็บไว้ใน Pop_n_ber มาเก็บไว้ในตัวแปรชื่อ n_ber

- `im = strcat (filename , n_ber , '.tif') ;`

เป็นการใช้คำสั่ง `strcat` กำหนด ค่า String ตัวแรก , ค่า String ตัวที่สอง และค่า String ตัวสุดท้ายมาเรียงต่อกัน เป็น `filamen_ber.tif` เพื่อใช้อ้างอิงชื่อแฟ้มข้อมูลภาพ มีประโยชน์เมื่อใช้ร่วมกับคำสั่ง `Load (filename)` และ `Save (filename)` แฟ้มข้อมูล

รูปแบบ คำสั่งในงานวิจัย : เลือกกลุ่มภาพ ที่นำมาฝึกสอน

```
%Rem Select Images to Trained by Group.
Pop_group=uicontrol(...
    'Style','popup',...
    'String','1|2|3|4|5|6|7|8|9|10',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'n_group = popupstr(Pop_group)']);
```

คำอธิบาย คำสั่ง : เลือกกลุ่ม ภาพที่นำมาฝึกสอน

- `'Pop_Group'`

เป็นเงื่อนไข ในการกำหนดเลือกกลุ่มภาพ ที่ต้องการนำมาป้อนให้โครงข่าย โดยแบ่งออกเป็น 10 กลุ่มตัวอย่าง คือแต่ละกลุ่มจะประกอบด้วยภาพอินพุต 7 ภาพ ซึ่งเงื่อนไขถูกกำหนดไว้ดังนี้

- `'String','1|2|3|4|5|6|7|8|9|10',...`
- `'Callback','n_group = popupstr(Pop_Group)'] ;`

: คำสั่ง `popupstr` นำค่า String ที่เก็บไว้ใน `Pop_Group` มาเก็บไว้ในตัวแปรชื่อ `n_group`

รูปแบบ คำสั่งในงานวิจัย : ปรับแต่งภาพ

```
%Rem Image Enhancement.
%
%Rem Select None Adjustment.
Rb_None=uicontrol(...
    'Backgroundcolor',[R G B],...
    'Style','radio',...
    'String','None Adjustment',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_None, 'Value',1),...
    ]);
```

```

        'set(Rb_Adjust, 'Value', 0)');

%Rem Select Intensity Adjustment.
Rb_Adust=icontrol(...
    'BackgroundColor', [R G B],...
    'Style', 'radio',...
    'String', 'Intensity Adjustment',...
    'Position', [Left Bottom Width High],...
    'Callback', [...
        'set(Rb_Adjust, 'Value', 1), '...
        'set(Rb_None, 'Value', 0), '...
        'I=imread('Filename.tif'),'...
        'Img=double(I)/16, '...
        'high=max(Img(:)), '...
        'low=min(Img(:)), '...
        'bot=0, '...
        'top=1, '...
        'gamma=1']]);

```

คำอธิบาย คำสั่ง : ปรับแต่ง ภาพ

นำภาพที่ต้องการฝึกสอนจากแฟ้มข้อมูล มาเปลี่ยนชนิดจาก uint8 เป็นชนิด double และสร้างค่า ความสว่าง , ความแตกต่างระหว่างความสว่าง กับ ความมืด และ ค่าองค์ประกอบ แกมมา จากข้อมูลภาพ

- 'Rb_None' , 'Rb_Adjust'

เป็นเงื่อนไข ในการกำหนดเลือกการปรับแต่งภาพ ที่นำฝึกสอนสำหรับ โครงข่าย ให้มีภาพที่มี อินเทนซิตี ต่างกัน ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

'Rb_None' = 1 เมื่อเลือก ไม่ต้องการปรับแต่งภาพ ที่นำมาฝึกสอน

'Rb_Adjust' = 1 เมื่อเลือก ต้องการปรับแต่งภาพ ที่นำมาฝึกสอน

รูปแบบ คำสั่งในงานวิจัย : ปรับความสว่าง ภาพ

```

%Rem Increase Brightness.
Pb_IncBright=icontrol(...
    'Style', 'push',...
    'String', '+ Brightness',...
    'Position', [Left Bottom Width High],...
    'Callback', [...
        'change = 0.1;', '...
        'if top >= bot, '...
            'if (bot+change > 1), '...
                'return, '...           % Check Maximum Brighter
            'end, '...
        'else, '...
            'if (top+change > 1), '...
                'return, '...           % Check Maximum Brighter
    ]]);

```

```

        'end,'...
    'end.,'...
    'top = top + change;','...
    'bot = bot + change;]);

```

```

%Rem Decrease Brightness.
Pb_DecBright=uicontrol(...
    'Style','push',...
    'String','- Brightness',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1;','...
        'if top >= bot,'...
            'if (top-change < 0),'...
                'return,'... % Check Maximum darker
            'end,'...
        'else,'...
            'if (bot-change < 0),'...
                'return,'... % Check Maximum darker
            'end,'...
        'end.,'...
    'top = top - change;','...
    'bot = bot - change;]);

```

รูปแบบ คำสั่งในงานวิจัย : ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ

```

%Rem Increase Contrast.
Pb_IncContrast=uicontrol(...
    'Style','push',...
    'String','+ Contrast',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1(high - low);','...
        'low = low + change;','...
        'high = high - change;]);

%Rem Decrease Contrast.
Pb_DecContrast=uicontrol(...
    'Style','push',...
    'String','- Contrast',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1(high - low);','...
        'low = max(low - change,0);','...
        'high = min(high + change,1);]);

```

รูปแบบ คำสั่งในงานวิจัย : ปรับค่าองค์ประกอบ แกมมา ภาพ

```
%Rem Increase Gamma.
Pb_IncGamma=uicontrol(...
    'Style','push',...
    'String','+ Gamma',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'gamma = gamma * 1.2;]);

%Rem Decrease Gamma.
Pb_DecGamma=uicontrol(...
    'Style','push',...
    'String','− Gamma',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'gamma = gamma * 0.8;]);
```

คำอธิบาย คำสั่ง : ปรับ ความสว่าง ภาพ

: ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ

: ปรับค่าองค์ประกอบ แกมมา ภาพ

- 'Pb_IncBrighte' , 'Rb_DecBrightt' ,
- 'Pb_IncContrast' , 'Pb_DecContrast' ,
- 'Pb_IncGamma' , 'Pb_DecGamma'

เป็นการคลิกปุ่ม เพื่อกระทำตามเงื่อนไข ในการกำหนดเลือกการปรับ
แต่งภาพ ที่นำมาฝึกสอนสำหรับ โครงข่าย ถูกกำหนดไว้ดังนี้

'Pb_IncBrighte' , 'Rb_DecBrightt'

: กำหนดการเลือกปรับแต่งความสว่างของภาพ

'Pb_IncContrast' , 'Pb_DecContrast' ,

: กำหนดการเลือกปรับแต่งความแตกต่างระหว่างความสว่างกับ
ความมืดภาพ

'Pb_IncGamma' , 'Pb_DecGamma'

: กำหนดการเลือกองค์ประกอบ แกมมา

รูปแบบ คำสั่งในงานวิจัย : ปรับขนาดภาพ

```
%Rem Adjust Size of Images to Train.
%
%Rem Adjust Size by Automatic 100 x 100 Pixel.
Rb_Auto=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','automatic',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Auto,''Value'',1),'...
        'set(Rb_Percent,''Value'',0),'...
        'set(Rb_Square,''Value'',0),'...
        'set(Rb_Anysize,''Value'',0)']]);
%Rem Adjust Size by Percent of Original Images.
Rb_Percent=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','percent (%)',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Auto,''Value'',0),'...
        'set(Rb_Percent,''Value'',1),'...
        'set(Rb_Square,''Value'',0),'...
        'set(Rb_Anysize,''Value'',0)']]);
```

```
%Rem Adjust Size by Square Matrix Images.
Rb_Square=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','square',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Auto,''Value'',0),'...
        'set(Rb_Percent,''Value'',0),'...
        'set(Rb_Square,''Value'',1),'...
        'set(Rb_Anysize,''Value'',0)']]);
%Rem Adjust Size by Anysize Images.
Rb_Anysize=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','size (mxn)',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Auto,''Value'',0),'...
        'set(Rb_Percent,''Value'',0),'...
        'set(Rb_Square,''Value'',0),'...
        'set(Rb_Anysize,''Value'',1)']]);
```

คำอธิบาย คำสั่ง : ปรับขนาดภาพ

- 'Rb_Auto' , 'Rb_Percent' , 'Rb_Square' , 'Rb_Anysize'

เป็นเงื่อนไข ในการกำหนดวิธีการปรับขนาดของภาพที่นำฝึกสอน
สำหรับโครงข่าย ให้มีขนาดของภาพที่ต่างกัน ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'Style' , 'radio' , ...
- 'String' , 'automatic' , ...
- 'String' , 'percent (%)' , ...
- 'String' , 'square' , ...
- 'String' , 'size (mxn)' , ...
- 'Callback' , [...

: Rb_Auto , 'Value' = '1' , เมื่อเลือก ขนาด 100 x 100 Pixel

: Rb_Percent , 'Value' = '1' , เมื่อเลือก ขนาดร้อยละ

: Rb_Square , 'Value' = '1' , เมื่อเลือก ขนาดจัตุรัส

: Rb_Anysize , 'Value' = '1' , เมื่อเลือก ขนาดใดๆ (แถว x สดมภ์)

รูปแบบ คำสั่งในงานวิจัย : กำหนดขนาดภาพ

```
%Rem Size Matrix Processing.
%
[x,y]=size(BW),
Case 'automatic' , row=100,col=100;
Case 'percent' , row=(x*percent)/100,col=(y*percent)/100;
Case 'square' , row=col;
Case 'anysize' , row=m , col=n;
```

คำอธิบาย คำสั่ง : กำหนดขนาดภาพ

- Case 'automatic'

: กำหนดขนาดของภาพ เป็นขนาดมาตรฐาน

แถว = 100 , สดมภ์ = 100

- Case 'percent'

: กำหนดขนาดของภาพ เป็นขนาดร้อยละของภาพต้นฉบับ

- *Case 'square'*

: กำหนดขนาดของภาพ เป็นขนาดจัตุรัส

- *Case 'anysize'*

: กำหนดขนาดของภาพ เป็นขนาดใดๆ

รูปแบบ คำสั่งในงานวิจัย : กำหนดขนาดเมตริกซ์ ภาพ

```
%Rem Setup Row Matrix.
%
Row_x=row,
Col_y=col,
Row_11=ceil(x/Row_x),
Row_22=floor(x/Row_x),
While Row_22 ~= Row_11,
    x=x+1,
    Row_11=ceil(x/Row_x),
    Row_22=floor(x/Row_x),
end,

%Rem Setup Colume Matrix.
%
Col_11=ceil(y/Col_y),
Col_22=floor(y/Col_y),
While Col_22 ~= Col_11,
    y=y+1,
    Col_11=ceil(y/Col_11),
    Col_22=floor(y/Col_22),
end;
```

คำอธิบาย คำสั่ง: กำหนดขนาดเมตริกซ์ ภาพ

- *Row Matrix & Col Matrix*

: คำนวณหาสัดส่วน แถว และ สดมภ์ ของภาพต้นฉบับ กับ ขนาดของภาพ ที่ต้องการปรับขนาด

รูปแบบ คำสั่งในงานวิจัย : กำหนดขนาด แถว และ สดมภ์ เมตริกซ์ภาพ

```
%Rem Marks End Block Matrix.
%
BW(x,y)=0,
%Rem Make Mirror Row Matrix.
%
M_row=zeros(Row_x,x),
j=0,
for i=1:Row_x ,
    for k=1:Row_11,
        j=j+1,
        M_row(i,j)=1,
    end,
end,

%Rem Make Mirror Colume Matrix.
%
N_col=zeros(Col_y,y),
i=0,
for j=1:Col_y ,
    for k=1:Col_11,
        i=i+1,
        N_col(i,j)=1,
    end,
end,
```

คำอธิบาย คำสั่ง : กำหนดขนาด แถว และ สดมภ์ เมตริกซ์ภาพ

- Make Mirror Row Matrix

: สร้างเมตริกซ์ ให้ทุก พิกเซล มีค่าเท่ากับ ศูนย์ โดยมีขนาด

แถว เท่ากับ ขนาดของภาพ ที่ต้องการปรับขนาด ,

สดมภ์ เท่ากับ ขนาดของภาพ ดั้งเดิม

: แทนค่าในพิกเซลให้มีค่าเท่ากับ 1 ในตำแหน่งที่เป็นสัดส่วนของภาพ

- Make Mirror Col Matrix

: สร้างเมตริกซ์ ให้ทุก พิกเซล มีค่าเท่ากับ ศูนย์ โดยมีขนาด

แถว เท่ากับ ขนาดของภาพ ดั้งเดิม ,

สดมภ์ เท่ากับ ขนาดของภาพ ที่ต้องการปรับขนาด

: แทนค่าในพิกเซลให้มีค่าเท่ากับ 1 ในตำแหน่งที่เป็นสัดส่วนของภาพ

รูปแบบ คำสั่งในงานวิจัย : คำนวณเมตริกซ์ภาพ

```
%Rem Set Images for Cut Matrix.
%
BW1=double(BW),
BW2=(M_row*BW1*N_col),
BW=sign(BW2),
```

คำอธิบาย คำสั่ง : คำนวณเมตริกซ์ภาพ

- Set Images for Cut Matrix

: double(BW) เพื่อแปลงข้อมูลของภาพจาก uint 8 เป็น double ซึ่งสามารถนำไปคำนวณได้

: คำนวณสร้างเมตริกซ์ ให้ได้ภาพตามขนาดที่ต้องการ เพื่อนำไปใช้ฝึกสอนโดยภาพมี ขนาด แถว และ สดมภ์ ตามขนาดที่ต้องการ

รูปแบบ คำสั่งในงานวิจัย : เลือกวิธีการหาขอบภาพ

```
%Rem Edge Dectection Method.
Pop_Method=uicontrol(...
    'Style','popup',...
    'String','Sobel|Prewitt|Roberts',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'meth=popupstr(Pop_Method),'...
        'pict=strcat(filename,n_ber,``.tif''),'...
        'im=strcat('`G`'',im);','...
        'I=imread(im);','...
        'BW=edge(I,meth)']]);
```

คำอธิบาย คำสั่ง : เลือกวิธีการหาขอบภาพ

- 'Pop_Method'

เป็นเงื่อนไข ในการกำหนดวิธีการหาขอบภาพที่น่าฝึกสอนสำหรับโครงข่าย ให้มีวิธีหาขอบภาพ 3 วิธีคือ Sobel, Prewitt, Roberts ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- 'Style', 'popup', ...
- 'String', 'Sobel | Prewitt | Roberts', ...
- 'Callback', [...

- : หาก Pop_method = 'Sobel' ค่าตัวแปร meth มีค่าเท่ากับ Sobel
- Pop_method = 'Prewitt' ค่าตัวแปร meth มีค่าเท่ากับ Prewitt
- Pop_method = 'Roberts' ค่าตัวแปร meth มีค่าเท่ากับ Roberts
- : สร้างชื่อเพิ่มภาพ ที่ต้องการนำไปหาขอบ
- im = filename_ber.tif
- : อ่านภาพ จากเพิ่มข้อมูล เพื่อนำไปหาขอบภาพ
- : หาขอบภาพตามวิธีที่กำหนดเลือก

รูปแบบ คำสั่งในงานวิจัย : แสดงภาพที่นำมาฝึกสอน และผ่านการหาขอบแล้ว

```
ImgAdj = imadjust(Img,[low;high][bot;top],gamma);
%
imshow(ImgAdj);

BW=edge(ImgAdj, meth),
imshow(BW),
```

คำอธิบาย คำสั่ง: แสดงภาพที่นำมาฝึกสอน และภาพที่ผ่านการหาขอบภาพ

- คำสั่ง imadjust ปรับค่า ความสว่างของภาพ , ปรับความแตกต่างระหว่างความสว่าง กับ ความมืด และค่าองค์ประกอบ แกมมา
- หาขอบภาพที่ผ่านการปรับแต่งแล้วด้วยคำสั่ง edge(ImgAdj, meth)
- แสดงข้อมูลภาพ ด้วยคำสั่ง imshow

ลำดับที่ 5 สร้างข้อมูลภาพอินพุต ครั้งละภาพ และ ครั้งละกลุ่ม โดยสร้างเมตริกซ์ของภาพ มีฟังก์ชันการทำงาน ดังรูป 4.13

รูปแบบ คำสั่งในงานวิจัย : สร้างข้อมูลภาพอินพุต ครั้งละภาพ

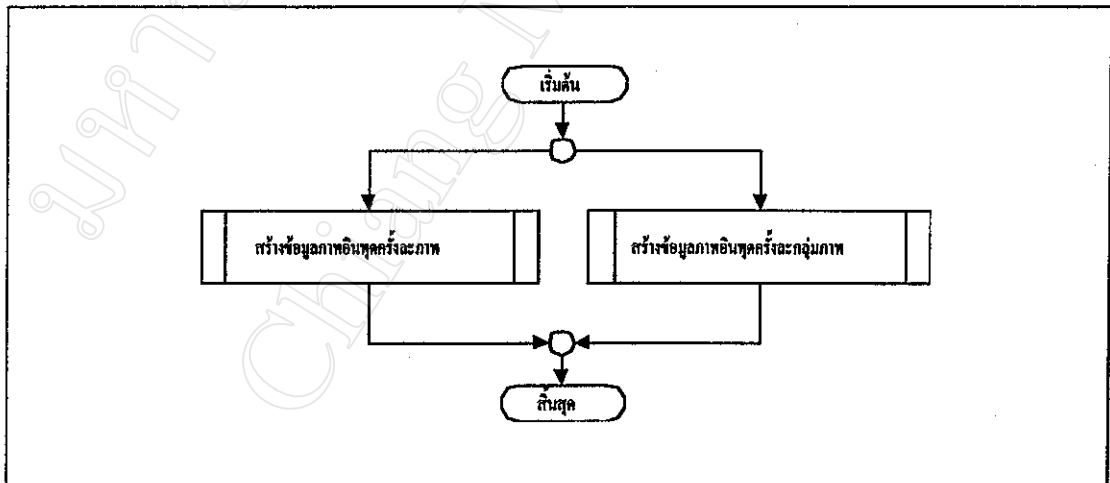
```
%Rem Initail Input Pattern by One by One Image.
[x,y]=size(BW),
row=x,
col=y,
z=row*col,
p=zeros(z,1),
i=0,
for a=1:row,
    for b=1:col,
        i=i+1,
        p=(i,1)=BW(a,b);end,end.
```

คำอธิบาย คำสั่ง : สร้างข้อมูลภาพอินพุต ครั้งละภาพ

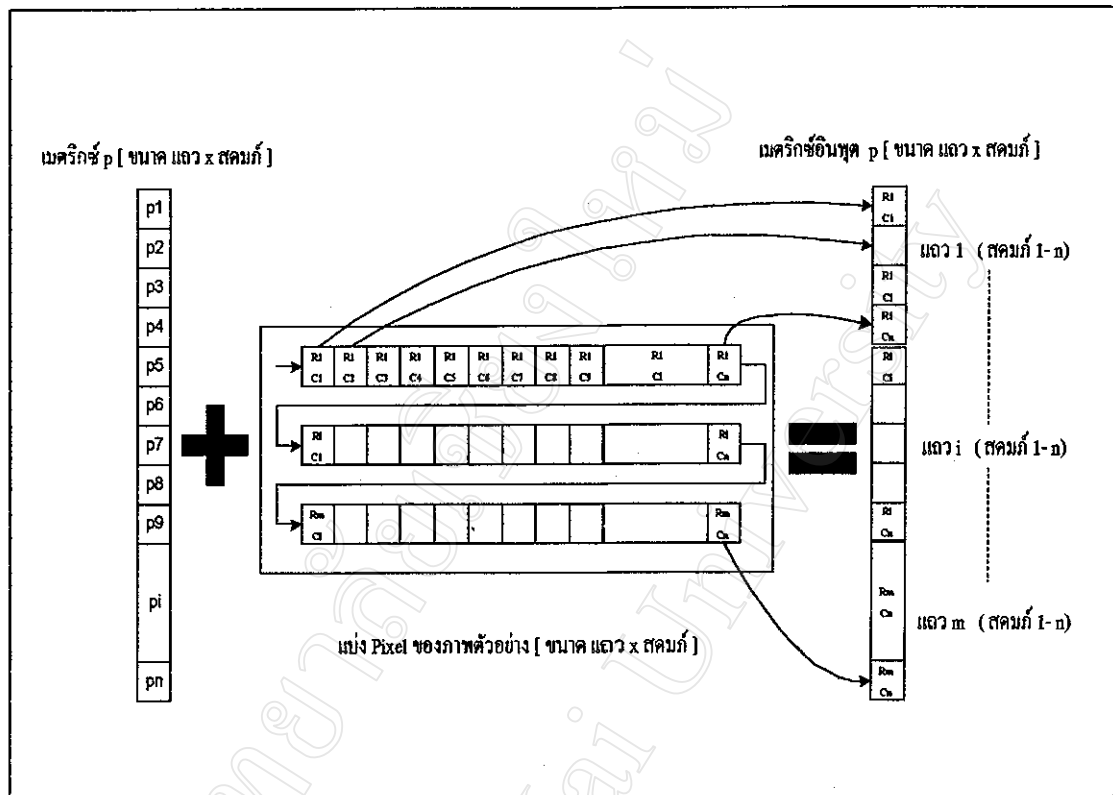
● *Initial Input Pattern by One by One Image.*

เป็นวิธีการสร้างข้อมูลอินพุตจากภาพที่ผ่านการหาขอบภาพโดยมีขนาดของข้อมูลอินพุตเท่ากับ ขนาดของภาพ (แถว x สดมภ์) วิธีการ ทำได้ ดังนี้

- สร้าง ขนาดข้อมูลอินพุต ($z = \text{row} \times \text{col}$)
- สร้าง เมตริกซ์อินพุต (p) ที่มีขนาดเท่ากับข้อมูลอินพุต (z)
- นำข้อมูลอินพุตที่มาจากการหาขอบภาพ มาเรียงไว้ในเมตริกซ์ อินพุต (p) โดยเรียงจาก แถวที่ 1 สดมภ์ที่ 1 ไปจนถึง แถว สุดท้าย สดมภ์สุดท้าย ดังรูปที่ 4.14



รูป 4.13 ฟังก์ชันตอนกระบวนการฝึกสอน ลำดับที่ 5



รูป 4.14 วิธีการสร้างข้อมูลอินพุตจากภาพที่ผ่านการหาขอบภาพ

รูปแบบ คำสั่งในงานวิจัย : สร้างข้อมูลภาพอินพุต ครั้งละกลุ่มภาพ

```
%Rem Make Filenames for Group.
%Rem Initial Input Pattern by Group Image.
%
s=strvcat('G_man*.tif','G_animal*.tif','G_article*.tif'),
s=cellstr(s),
%
%Rem Create Men Group.
s1=s(1,1),
s2=strrep(s1,'*',group)'
c=char(s2),
I=imread(c),
imshow(I),
BW=edge(I,meth),
imshow(BW),
[x,y]=size(BW),
row=x;
col=y;
z=row*col;
p1=zeros(z,1);
i=0;
```

```

for a=1:row,
    for b =1:col,
        i =i+1,
        p1=(i,1)=BW(a,b);
    end,
end.
%Rem Create Animals Group.
%
s1=s(2,1),
s2=strrep(s1, '*', group)'
c=char(s2),
I=imread(c),
imshow(I),
BW=edge(I, meth),
imshow(BW),
[x,y]=size(BW),
row=x;
col=y;
z=row*col;
p2=zeros(z,1);
i=0;
for a=1:row,
    for b =1:col,
        i =i+1,
        p2=(i,1)=BW(a,b);
    end,
end.
%Rem p2 = 2Legs Animals Group.
%Rem p3 = 4Legs Animals Group.
%Rem p4 = Bugs Animals Group.

```

```

%Rem Create Article Group.
%
s1=s(5,1),
s2=strrep(s1, '*', group)'
c=char(s2),
I=imread(c),
imshow(I),
BW=edge(I, meth),
imshow(BW),
row=x;
col=y;
z=row*col;
p5=zeros(z,1);
i=0;
for a=1:row,
    for b =1:col,
        i =i+1,
        p5=(i,1)=BW(a,b);
    end,
end.
%Rem p5 = Chairs Article Group.
%Rem p6 = Aircrafts Article Group.
%Rem p7 = Vehicles Article Group.

%Rem Group Matrix.
p=[p1 p2 p3 p4 p5 p6 p7],

```

คำอธิบาย คำสั่ง : สร้างข้อมูลภาพอินพุต ครั้งละกลุ่มภาพ

การสร้างชื่อของแฟ้มข้อมูลภาพ จะเรียงชื่อเป็นกลุ่มข้อมูล โดยใช้คำสั่ง `strvcat` จัดเรียงกลุ่มชื่อแฟ้มข้อมูล

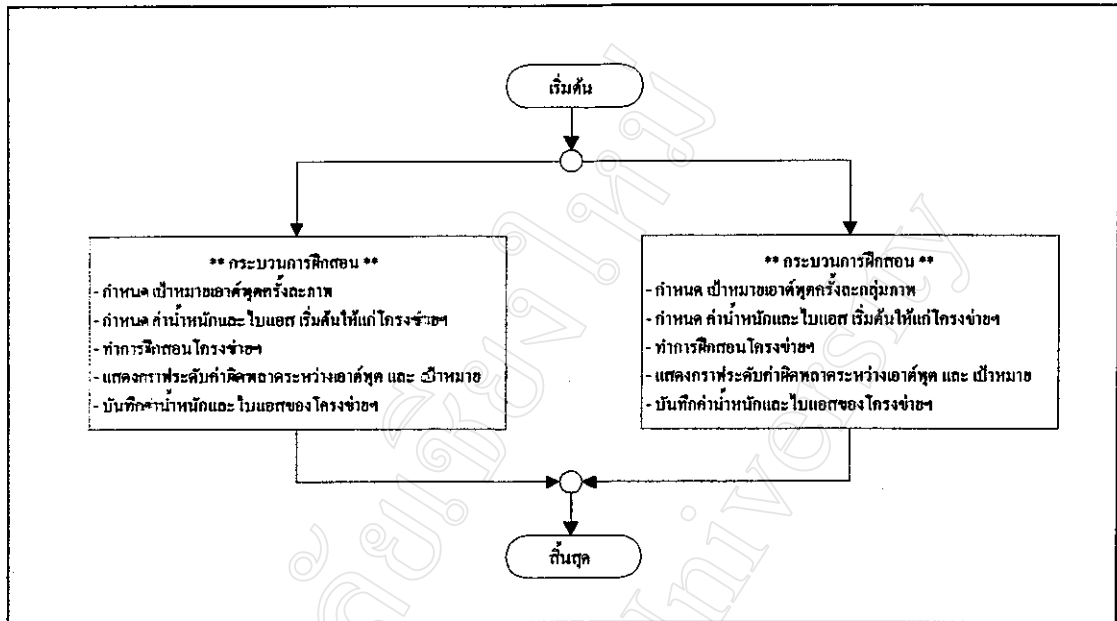
- *Initial Input Pattern by Group Image.*

เป็นวิธีการสร้างข้อมูลอินพุตครั้งละกลุ่มภาพ จากภาพที่ผ่านการหาขอบภาพ โดยได้ขนาดของข้อมูลอินพุตเท่ากับ ขนาดของภาพ (แถว x สดมภ์) วิธีการ ทำได้ ดังนี้

- สร้าง ขนาดข้อมูลอินพุต ($z = \text{row} \times \text{col}$)
- สร้าง เมตริกซ์อินพุต (p) ที่มีขนาดเท่ากับข้อมูลอินพุต (z)
- นำข้อมูลอินพุตที่มาจากการหาขอบภาพ มาเรียงไว้ในเมตริกซ์อินพุต (p) โดยเรียงจาก แถวที่ 1 สดมภ์ที่ 1 ไปจนถึง แถว สุดท้ายสดมภ์สุดท้าย

ลำดับที่ 6 กระบวนการฝึกสอน ทำการคำนวณค่าน้ำหนักของชั้นซ่อน และชั้นเอาต์พุต เพื่อหาคำตอบของโครงข่ายฯ แสดงดังรูป 4.15 มีรายละเอียดดังนี้

- กำหนดเป้าหมายเอาต์พุต ครั้งละภาพ , ครั้งละกลุ่ม
- กำหนดค่าน้ำหนัก และไบแอส เริ่มต้นให้แก่โครงข่าย
- ทำการฝึกสอนโครงข่าย
- แสดงกราฟระดับค่าความผิดพลาด ระหว่างเอาต์พุตและเป้าหมาย
- บันทึกค่าน้ำหนักและไบแอส ของโครงข่าย



รูป 4.15 ฟังก์ชันตอนกระบวนการฝึกสอน ลำดับที่ 6

รูปแบบ คำสั่งในงานวิจัย : กำหนดเป้าหมายเอาต์พุต ครั้งละภาพ

```

%Rem Determine Target Matrix of One by One Image.
Switch filename
Case 'man' ,t = [1;1;1],

%Rem Case 'animal'
Case '2Legs' ,t = [0;0;1],
Case '4Legs' ,t = [0;1;0],
Case 'Bug' ,t = [0;1;1],

%Rem Case 'article' ,
Case 'Chair' ,t = [1;0;0],
Case 'Aircraft',t= [1;0;1],
Case 'Vehicle' ,t= [1;1;0],
end.
  
```

รูปแบบ คำสั่งในงานวิจัย : กำหนดเป้าหมายเอาต์พุต ครั้งละกลุ่ม

```

%Rem Determine Target Matrix of Group Images.
t=[1 0 0 0 1 1 1;1 0 1 1 0 0 1;1 1 0 1 0 1 0],
%
%Rem t=[1      0      0      0      1      1      1
%      1      0      1      1      0      0      1
%      1      1      0      1      0      1      0 ]
%      [man 2Legs 4Legs bug Chair aircraft vehicle]
  
```

รูปแบบ คำสั่งในงานวิจัย : กำหนดค่าน้ำหนักและไบแอสเริ่มต้นให้แก่โครงข่าย

```
%Rem Make Rx2 Matrix of Input Vector.
z=row*col,
[Head,Tail]=size(z),
mn=zeros(Head,2),
for i = 1:Head,
    mn(i,2)=1,
end,
%Rem Random Weight and Bias.
Case 'tansig'
    [W1,B1]=nwtan(UB1,z),
    [W2,B2]=nwtan(UB2,UB1),
    [W3,B3]=nwtan(UB3,UB2),
Case 'logsig'
    [W1,B1]=nwlog(UB1,z),
    [W2,B2]=nwlog(UB2,UB1),
    [W3,B3]=nwlog(UB3,UB2),
Case 'Random'
    [W1,B1]=rands(UB1,z),
    [W2,B2]=rands(UB2,UB1),
    [W3,B3]=rands(UB3,UB2),

%Rem Initialize Feed Forward Network.
[W1,B1,W2,B2,W3,B3]=initff([mn],UB1,UT1,UB2,UT2,UB3,UT3),
```

คำอธิบาย คำสั่ง : กำหนดค่าน้ำหนักและไบแอสเริ่มต้นให้แก่โครงข่าย

- *Make Rx2 Matrix of Input Vector.*

เป็นวิธีการกำหนดขอบเขตความเป็นไปได้ค่าต่ำสุด และค่าสูงสุด ของข้อมูลอินพุต ที่ต้องการป้อนเข้าโครงข่ายเพื่อทำการฝึกสอน

: กำหนดขอบเขตความเป็นไปได้ของข้อมูลอินพุต คือ 0 กับ 1 โดยสร้างเมตริกซ์ [mn] เป็นขนาด แถว x 2

- *Initialize Feed Forward Network*

เป็นวิธีการนำค่าน้ำหนักและไบแอส ที่สุ่มมาได้ กำหนดลงบนโครงข่าย เพื่อเป็นค่าเริ่มต้น ในการฝึกสอน มีวิธีการดังนี้

- คำสั่ง initff (Initialize Feed Forward Network) ประกอบด้วยตัวแปร ดังนี้

: W_n = ค่าน้ำหนักของชั้นโครงข่าย

- : B_n = ค่าไบแอสของชั้นโครงข่าย
- : $[mn]$ = ขอบเขตความเป็นไปได้ของค่าอินพุต
- : UB_n = จำนวนนิวรอนของชั้นโครงข่าย
- : UT_n = ฟังก์ชันการกระตุ้นชั้นโครงข่าย

รูปแบบ คำสั่งในงานวิจัย : ทำการฝึกสอนโครงข่าย

```
%Rem Train Feed Forward Network.
[W1,B1,W2,B2,W3,B3,TE,TR]=trainbpx(W1,B1,UT1,W2,B2,UT2,...
                                     W3,B3,UT3,[p],t,TP),
```

คำอธิบาย คำสั่ง : ทำการฝึกสอนโครงข่าย

- *Train Feed Forward Network*

เป็นวิธีการฝึกสอนของโครงข่ายแบบแพร่กลับ ในการฝึกสอนประกอบ

ด้วยตัวแปรดังนี้

- คำสั่ง trainbpx (Train Feed Forward Network Backpropagation)

ประกอบด้วยตัวแปร ดังนี้

- : W_n = ค่าน้ำหนักของชั้นโครงข่าย
- : B_n = ค่าไบแอสของชั้นโครงข่าย
- : UT_n = ฟังก์ชันการกระตุ้นชั้นโครงข่าย
- : $[p]$ = เมตริกซ์อินพุต ภาพ
- : t = เป้าหมายเอาต์พุต
- : TP = ค่าพารามิเตอร์

และมีการส่งคืนค่าตัวแปรดังนี้

- : W_n = ค่าน้ำหนักของชั้นโครงข่าย
- : B_n = ค่าไบแอสของชั้นโครงข่าย
- : TE = จำนวนรอบการฝึกสอนของโครงข่าย
- : TR = ค่าบันทึกข้อมูลการฝึกสอน

รูปแบบ คำสั่งในงานวิจัย : แสดงกราฟระดับค่าความผิดพลาด ระหว่าง เอคต์พุต และ เป้าหมาย

```
%Rem Plot Graph Train Record.
Ploterr(TR,0.01,Hf_graph),
```

คำอธิบาย คำสั่ง : แสดงกราฟระดับค่าความผิดพลาด ระหว่างเอคต์พุต และเป้าหมาย

- Plot Graph Train Record.

เป็นวิธีการแสดงความสัมพันธ์ของค่าการฝึกสอน , ค่าผิดพลาดของโครงข่าย โดยนำข้อมูลการฝึกสอนจากตัวแปร TR

รูปแบบ คำสั่งในงานวิจัย : บันทึกค่าน้ำหนักและไบแอสของโครงข่าย

```
%Rem Save Configuration of Trained Network.
Save(Wfile,'W1','B1','W2','B2','W3','B3',...
      'row','col','meth');
```

คำอธิบาย คำสั่ง : บันทึกค่าน้ำหนักและไบแอสของโครงข่าย

- Save Configuration of Training Network.

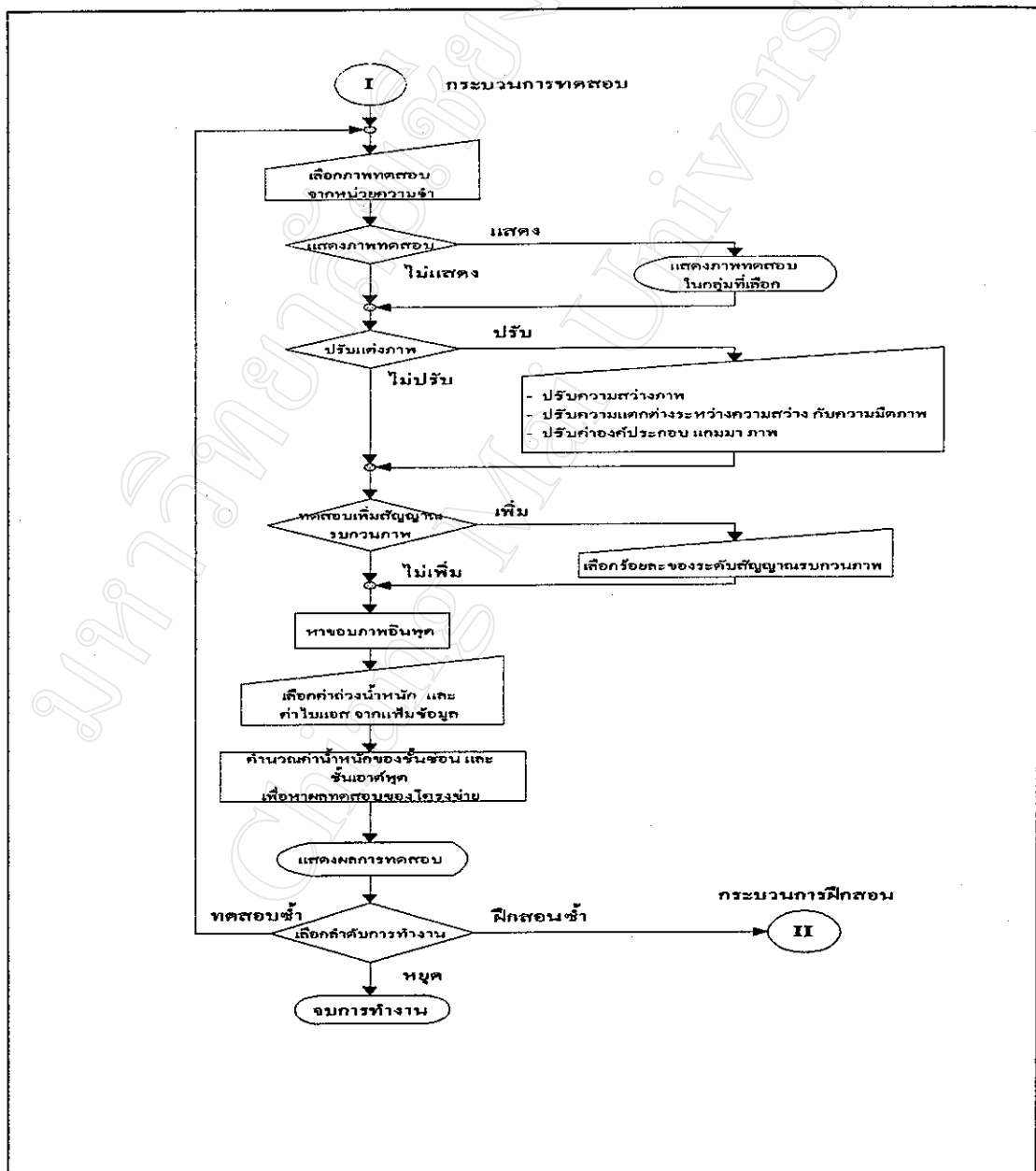
ใช้ในการบันทึก ค่าตัวแปรของโครงข่าย ลงไว้ใน แฟ้มข้อมูล ซึ่งเงื่อนไขถูกกำหนดไว้ดังนี้

: Wfile มีค่าเท่ากับชื่อของแฟ้มข้อมูลต่อท้ายด้วย _w เพื่อบอกให้ทราบว่าเป็น แฟ้มข้อมูลที่เก็บ ค่าน้ำหนัก และไบแอสของโครงข่าย

: Save ทำหน้าที่เก็บค่าน้ำหนัก, ค่าไบแอส, ขนาดแถว, ขนาดสดมภ์, วิธีการหาขอบภาพ ไว้ในแฟ้มข้อมูล ที่ถูกกำหนดชื่อโดย Wfile

4.2.3 กระบวนการทดสอบ (Test)

ผังขั้นตอนกระบวนการทดสอบการฝึกสอนของโครงข่ายประสาทเทียม แสดงดังรูป 4.16 ซึ่งประกอบไปด้วยลำดับการทำงาน ดังนี้



รูป 4.16 ผังขั้นตอนกระบวนการทดสอบโครงข่ายประสาทเทียมแบบแพร่กลับ

ลำดับที่ 1 เลือกและแสดงภาพทดสอบจากแฟ้มข้อมูล

- แสดงภาพทดสอบในกลุ่มที่เลือก

ลำดับที่ 2 เลือกปรับแต่งภาพ

- ปรับความสว่างภาพ
- ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ
- ปรับค่าองค์ประกอบ แกมมาภาพ

ลำดับที่ 3 เลือกเพิ่มสัญญาณรบกวนภาพเข้าไปยังภาพทดสอบ

ลำดับที่ 4 ป้อนชุดข้อมูลทดสอบเข้าไปยังชั้นอินพุต

- หาขอบภาพอินพุต
- เลือกค่าน้ำหนัก และ ไบแอสจากแฟ้มข้อมูล

ลำดับที่ 5 คำนวณค่าน้ำหนักของชั้นซ่อน และชั้นเอาต์พุตเพื่อหาผลการทดสอบของโครงข่าย

ลำดับที่ 6 แสดงผลการทดสอบ

- แสดงค่าเอาต์พุต เมตริกซ์
- แสดงค่าเอาต์พุต เป้าหมาย
- แสดงค่าความสามารถการรู้จำโครงร่างภาพวัตถุ
- แสดงค่า สถิติการทดสอบของโครงข่ายฯ

ลำดับที่ 1 เลือกและแสดงภาพที่ต้องการทดสอบจากแฟ้มข้อมูล เป็นขั้นตอนการจัดการ ป้อนภาพที่ต้องการเข้าไปในโครงข่าย มีผังขั้นตอนการทำงาน รูป 4.17 มีรายละเอียดดังนี้

- เลือกภาพที่ต้องการทดสอบ จากแฟ้มข้อมูล
- แสดงภาพทดสอบในกลุ่มที่เลือก

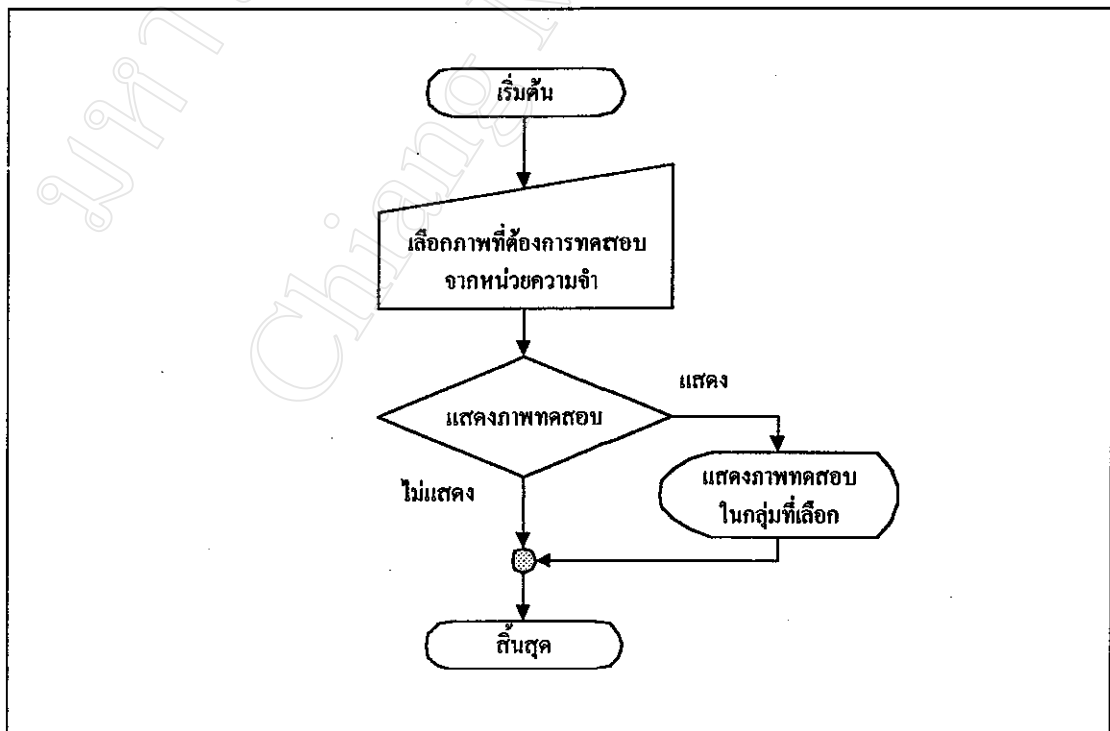
รูปแบบ คำสั่งในงานวิจัย : ** กระบวนการทดสอบ **

```
test_1=figure(...
    'Name', 'Select Objects to Test',...
    'Color', [R G B],...
    'Position', [Left Bottom Width High]);
```

รูปแบบ คำสั่งในงานวิจัย : เลือกภาพทดสอบ จากแฟ้มข้อมูล

```
%Rem Select Images Filenames to Show & Test
Rb_man=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','MAN',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_man, 'Value'',1),'...
        'set(Rb_animal, 'Value'',0),'...
        'set(Rb_article, 'Value'',0)']]);

Rb_animal=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','ANIMAL',...
    'Position',[Left Bottom Width High],...
    'Value',0,...
    'Callback',[...
        'set(Rb_man, 'Value'',0),'...
        'set(Rb_animal, 'Value'',1),'...
        'set(Rb_article, 'Value'',0)']]);
```



รูป 4.17 ฟังก์ชันตอนกระบวนการทดสอบ ลำดับที่ 1

```

Rb_animal=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','ANIMAL',...
    'Position',[Left Bottom Width High],...
    'Value',0,...
    'Callback',[...
        'set(Rb_man, 'Value',0),'...
        'set(Rb_animal, 'Value',1),'...
        'set(Rb_article, 'Value',0)']);

Rb_article=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','ARTICLE',...
    'Position',[Left Bottom Width High],...
    'Value',0,...
    'Callback',[...
        'set(Rb_man, 'Value',0),'...
        'set(Rb_animal, 'Value',0),'...
        'set(Rb_article, 'Value',1)']);

Rb_one=uicontrol(gcf,...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','1',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_one, 'Value',1)']);

```

คำอธิบาย คำสั่ง: เลือกภาพทดสอบ จากแฟ้มข้อมูล

- 'Rb_man', 'Rb_animal', 'Rb_article'

เป็นเงื่อนไข ในการกำหนดชื่อของภาพ ที่ต้องการนำมาป้อนให้อินพุตของโครงข่าย โดยแบ่งออกเป็น 3 กลุ่มตัวอย่าง คือ man หรือ animal หรือ article ซึ่งเงื่อนไขถูกกำหนดไว้ดังนี้

- 'String', 'man | animal | article ',...

- 'Callback',

: Rb_man(Value) = 1 ,เมื่อเลือกชื่อกลุ่มภาพ man หรือ

: Rb_animal(Value) = 1 ,เมื่อเลือกชื่อกลุ่มภาพ animal หรือ

: Rb_article(Value) = 1 ,เมื่อเลือกชื่อกลุ่มภาพ article

- *'Rb_one'*

เป็นเงื่อนไข ในการกำหนดลำดับที่ของภาพ ที่ต้องการนำมาป้อนให้
โครงข่าย โดยมีค่าตั้งแต่รูปที่ 1 ถึงรูปที่ 30 ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- *'String'*, *'1-30'*,...
- *'Callback'*,
: *n_ber = '1-30'* ,เมื่อเลือกลำดับที่ของภาพ

รูปแบบ คำสั่งในงานวิจัย : แสดงภาพทดสอบในกลุ่มที่เลือก

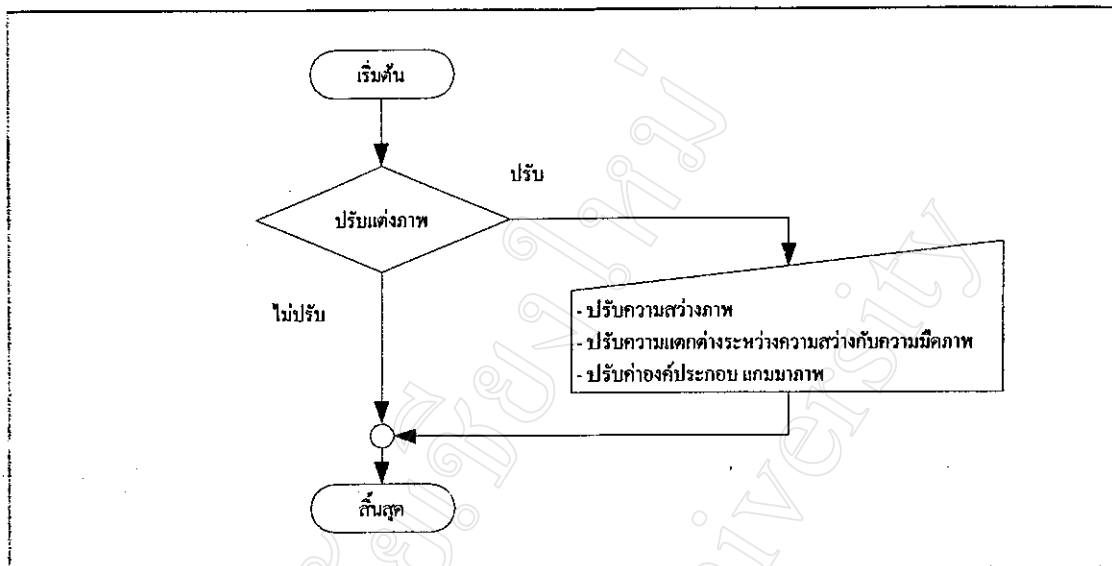
```
%Rem Show Images
Pb_Show=uicontrol(gcf,...
    'Style','push',...
    'String','Show Images',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'im=strcat(filename,n_ber,`.tif`)',...
        'I=imread(im)',...
        'imshow(I)']);
```

คำอธิบาย คำสั่ง : แสดงภาพทดสอบในกลุ่มที่เลือก

- *'Pb_Show'*

เป็นการแสดงภาพทดสอบ ในกลุ่มที่เลือก

- : ใช้คำสั่ง *strcat* กำหนด ค่า String ตัวแรก , ค่า String ตัวที่สอง และค่า String ตัวสุดท้ายมาเรียงต่อกัน เป็น *filenamen_ber.tif* เพื่อใช้อ้างอิงชื่อแฟ้มข้อมูลภาพ
- : อ่านข้อมูลภาพจากแฟ้มข้อมูล ภาพ ด้วยคำสั่ง *imread*
- : แสดงข้อมูลภาพ ด้วยคำสั่ง *imshow*



รูป 4.18 ฟังก์ชันตอนกระบวนการทดสอบ ลำดับที่ 2

ลำดับที่ 2 เลือกปรับแต่งภาพ โดยปรับความสว่างภาพ , ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ และ ปรับค่าองค์ประกอบ แกมมาภาพ มีฟังก์ชันการทำงานรูป 4.18

รูปแบบ คำสั่งในงานวิจัย : ปรับแต่งภาพ

```

%Rem Image Enhancement.
%
%Rem Select None Adjustment.
Rb_None=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','None Adjustment',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_None,'Value',1),'...
        'set(Rb_Adjust,'Value',0)']);

%Rem Select Intensity Adjustment.
Rb_Adust=uicontrol(...
    'BackgroundColor',[R G B],...
    'Style','radio',...
    'String','Intensity Adjustment',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'set(Rb_Adjust,'Value',1),'...
        'set(Rb_None,'Value',0),'...
  
```

```

'I=imread('Filename.tif'),'...
'Img=double(I)/16,'...
'high=max(Img(:)),'...
'low=min(Img(:)),'...
'bot=0,'...
'top=1,'...
'gamma=1']]);

```

คำอธิบาย คำสั่ง : ปรับแต่ง ภาพ

นำภาพที่ต้องการฝึกสอนจากเพิ่มข้อมูล มาเปลี่ยนชนิดจาก uint8 เป็นชนิด double และสร้างค่า ความสว่าง , ความแตกต่างระหว่างความสว่าง กับ ความมืด และ ค่าองค์ประกอบ แกมมา จากข้อมูลภาพ

- 'Rb_None' , 'Rb_Adjustt'

เป็นเงื่อนไข ในการกำหนดเลือกการปรับแต่งภาพ ที่นำฝึกสอนสำหรับ โครงข่าย ให้มีภาพที่มี อินเทนซิตี ต่างกัน ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

'Rb_None' = 1 เมื่อเลือก ไม่ต้องการปรับแต่งภาพ ที่นำมาฝึกสอน

'Rb_Adjustt' = 1 เมื่อเลือก ต้องการปรับแต่งภาพ ที่นำมาฝึกสอน

รูปแบบ คำสั่งในงานวิจัย : ปรับแต่ง ความสว่าง ภาพ

```

%Rem Increase Brightness.
Pb_IncBright=uicontrol(...
    'Style','push',...
    'String','+ Brightness',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1;','...
        'if top >= bot,'...
            'if (bot+change > 1),'...
                'return','...           % Check Maximum Brighter
            'end','...
        'else','...
            'if (top+change > 1),'...
                'return','...           % Check Maximum Brighter
            'end','...
        'end.,','...
    'top = top + change;','...
    'bot = bot + change;]);

```

```
%Rem Decrease Brightness.
Pb_DecBright=uicontrol(...
    'Style','push',...
    'String','- Brightness',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1;','...
        'if top >= bot,'...
            'if (top-change < 0),'...
                'return','...      % Check Maximum darker
            'end','...
        'else','...
            'if (bot-change < 0),'...
                'return','...      % Check Maximum darker
            'end','...
        'end.,','...
        'top = top - change;','...
        'bot = bot - change;]);
```

รูปแบบ คำสั่งในงานวิจัย : ปรับความแตกต่างระหว่างความสว่างกับความมืด
ภาพ

```
%Rem Increase Contrast.
Pb_IncContrast=uicontrol(...
    'Style','push',...
    'String','+ Contrast',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1(high - low);','...
        'low = low + change;','...
        'high = high - change;]);
```

```
%Rem Decrease Contrast.
Pb_DecContrast=uicontrol(...
    'Style','push',...
    'String','- Contrast',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'change = 0.1(high - low);','...
        'low = max(low - change,0);','...
        'high = min(high + change,1);]);
```

รูปแบบ คำสั่งในงานวิจัย : ปรับค่าองค์ประกอบ แกมมา ภาพ

```
%Rem Increase Gamma.
Pb_IncGamma=uicontrol(...
    'Style','push',...
    'String','+ Gamma',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'gamma = gamma * 1.2;]);

%Rem Decrease Gamma.
Pb_DecGamma=uicontrol(...
    'Style','push',...
    'String','- Gamma',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'gamma = gamma * 0.8;]);
```

คำอธิบาย คำสั่ง : ปรับ ความสว่าง ภาพ

: ปรับความแตกต่างระหว่างความสว่างกับความมืดภาพ

: ปรับค่าองค์ประกอบ แกมมา ภาพ

- 'Pb_IncBrighte', 'Rb_DecBrightt',
- 'Pb_IncContrast', 'Pb_DecContrast',
- 'Pb_IncGamma', 'Pb_DecGamma'

เป็นการคลิกปุ่ม เพื่อกระทำตามเงื่อนไข ในการกำหนดเลือกการปรับ
แต่งภาพ ที่นำมาฝึกสอนสำหรับโครงข่าย ถูกกำหนดไว้ดังนี้

'Pb_IncBrighte', 'Rb_DecBrightt'

: กำหนดการเลือกปรับแต่งความสว่างของภาพ

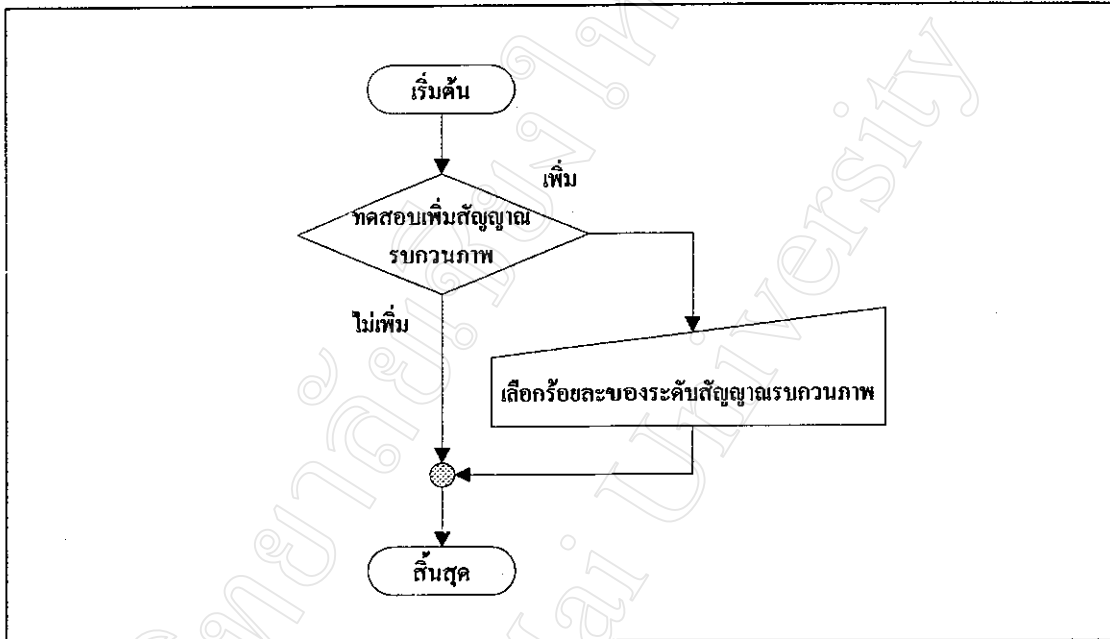
'Pb_IncContrast', 'Pb_DecContrast',

: กำหนดการเลือกปรับแต่งความแตกต่างระหว่างความสว่างกับ
ความมืดภาพ

'Pb_IncGamma', 'Pb_DecGamma'

: กำหนดการเลือกองค์ประกอบ แกมมา

ลำดับที่ 3. เลือกเพิ่มสัญญาณรบกวนภาพเข้าไปยังภาพทดสอบ มีผังขั้นตอนการทำงาน ดังรูป 4.19



รูป 4.19 ผังขั้นตอนกระบวนการทดสอบ ลำดับที่ 3

รูปแบบ คำสั่งในงานวิจัย : เลือกเพิ่มสัญญาณรบกวนภาพเข้าไปยังภาพทดสอบ

```

%Rem Fill number of Noise
E_Density=uicontrol(gcf,...
    'BackgroundColor','w',...
    'ForegroundColor','b',...
    'Style','edit',...
    'String','Density',...
    'Position',[Left Bottom Width High]);

%Rem Add Noise.
Pop_Noise=uicontrol(gcf,...
    'Style','popup',...
    'String','No Add Noise|Salt & Pepper',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'Case 'No Add Noise',
        'Case 'Salt&Pepper']]);
  
```

```
Pb_Noisy=icontrol(gcf,...
    'Style','push',...
    'String','Add Noise',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'I=imread(im);','...
        'Case 'Salt & Pepper',
        'Psp=Density','...
        'Psp=(Psp/100)','...
        'noisy=imnoise(I,'salt&pepper',Psp);...
        'imshow(noisy)','...
    'end','...
    'Case 'No Add Noise',
    'imshow(im)','...
    'end']);
```

คำอธิบาย คำสั่ง : เลือกเพิ่มสัญญาณรบกวนภาพเข้าไปยังภาพทดสอบ

- *Add Noise*

เป็นเงื่อนไข ในการเลือกเพิ่มสัญญาณรบกวนเข้าไปยังภาพทดสอบโดยใช้สัญญาณรบกวนชนิด salt & pepper เป็นลักษณะการทำงาน on and off pixel มีการกำหนดค่าความหนาแน่นของสัญญาณรบกวน เป็นร้อยละของขนาดภาพ ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

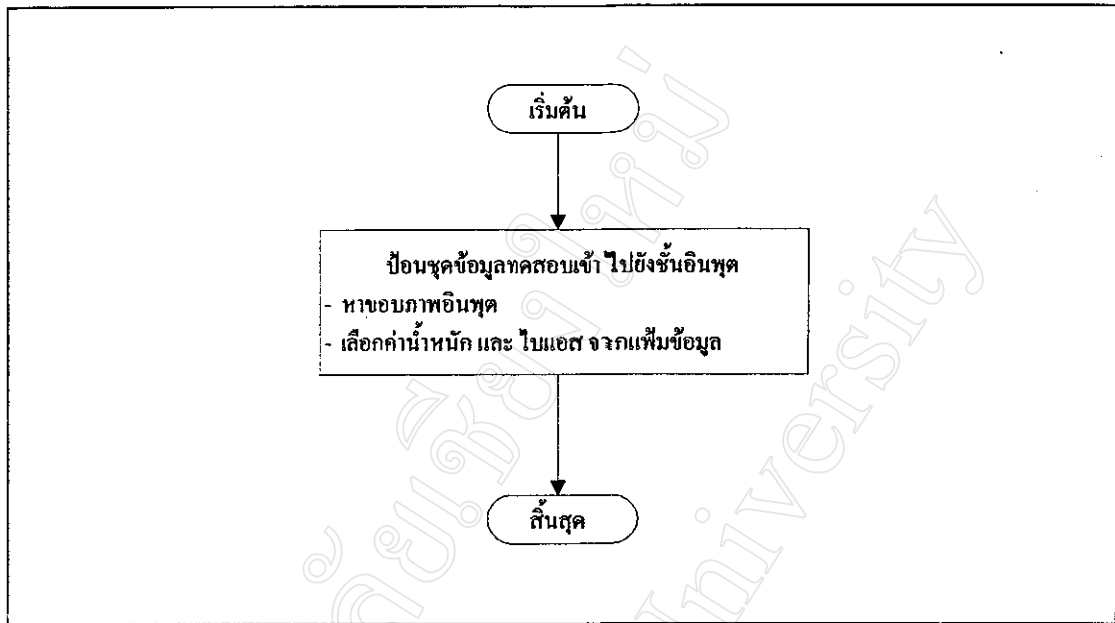
- 'imnoise(filename.tif, 'salt & pepper', 'percent', '...'
 - : filename.tif = ชื่อของแฟ้มข้อมูลภาพ
 - : salt & pepper = ชนิดของสัญญาณรบกวน
 - : percent = ความหนาแน่นของสัญญาณรบกวนคิดเป็นร้อยละของขนาดภาพ

ลำดับที่ 4. ป้อนชุดข้อมูลทดสอบเข้าไปยังชั้นอินพุต

หาขอบภาพอินพุต เลือกค่าน้ำหนัก และ ไบแอส จากแฟ้มข้อมูล ที่ผ่านการฝึกสอน แล้วแสดงผังการทำงานรูปที่ 4.20

รูปแบบ คำสั่งในงานวิจัย : การหาขอบแล้ว

```
I=noisy;
BW=edge(I, meth),
```



รูป 4.20 ผังขั้นตอนกระบวนการทดสอบ ลำดับที่ 4

รูปแบบ คำสั่งในงานวิจัย : เลือกค่าน้ำหนัก และ ไบแอส จากแฟ้มข้อมูล

```

%Rem Select Weight & Bias From Files.
Edit_WB=uicontrol(gcf,...
    'BackgroundColor','w',...
    'ForegroundColor','b',...
    'Style','edit',...
    'String','filename',...
    'Position',[Left Bottom Width High]);

%Rem Load Weight & Bias From Files.
Pb_Load=uicontrol(gcf,...
    'Style','push',...
    'String','Load',...
    'Position',[Left Bottom Width High],...
    'Callback',[...
        'Lfile=filename','...
        'Wfile=strcat(Lfile, '_w');','...
        'Nfile=strcat(Lfile, '_n');','...
        'Load(Wfile),'...
        'load(Nfile)']);
  
```

คำอธิบาย คำสั่ง : เลือกค่าน้ำหนัก และ ไบแอส จากแฟ้มข้อมูล

- *'Edit_WB'*

เป็นส่วนที่ใช้กำหนดชื่อแฟ้มข้อมูล ที่ต้องการนำค่าน้ำหนัก และไบแอส ที่ถูกเก็บบันทึกไว้ในแฟ้มข้อมูล โดยการเติมชื่อที่ต้องการลงไป

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- *'String', 'filename', ...*

: กำหนดให้ filename เป็นชื่อของแฟ้มข้อมูล

- *'Pb_Load'*

เป็นปุ่มคลิกเพื่อทำตามเงื่อนไข ถูกกำหนดใช้ในการนำค่าน้ำหนัก และไบแอส ของโครงข่ายที่ถูกเก็บบันทึก จากแฟ้มข้อมูล

ซึ่งเงื่อนไข ถูกกำหนดไว้ดังนี้

- *'String', 'Load', ...*

: ชื่อของปุ่มกำหนดเป็น Load

- *'Callback', [...*

'Lfile = filename, ' ...

'Wfile = strcat(Lfile, '_w'), ' ...

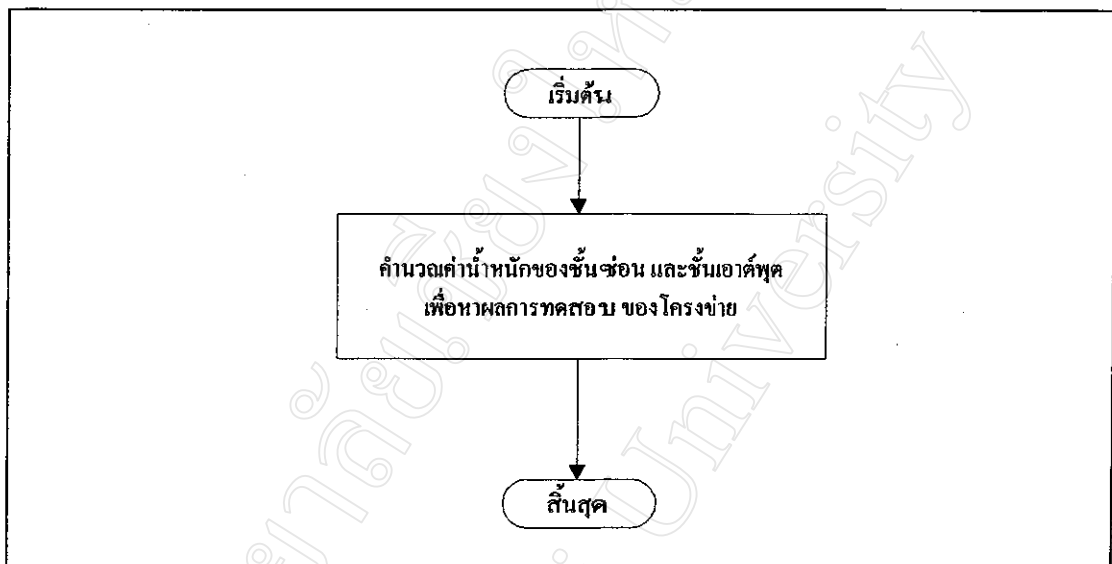
'Load (Wfile) ']);

: Lfile มีค่าเท่ากับชื่อของแฟ้มข้อมูล

: Wfile มีค่าเท่ากับชื่อของแฟ้มข้อมูลต่อท้ายด้วย _w โดยที่ คำสั่ง strcat ทำหน้าที่เอาค่า String ตัวแรก มาต่อเข้ากับ String ตัวที่สอง เพราะฉะนั้น Wfile = filename_w เพื่อบอกให้ทราบว่า เป็นแฟ้มข้อมูลที่ใช้เก็บ ค่าน้ำหนัก และ ไบแอสของโครงข่าย

: Load ทำหน้าที่นำค่าน้ำหนัก และไบแอส มาจากแฟ้มข้อมูล ที่ถูกกำหนดชื่อโดย Wfile

ลำดับที่ 5 คำนวณค่าน้ำหนักของชั้นซ่อน และชั้นเอาต์พุต เพื่อหาผลการทดสอบของโครงข่าย มีผังขั้นตอนการทำงานดังรูป 4.21



รูป 4.21 ผังขั้นตอนกระบวนการทดสอบ ลำดับที่ 5

รูปแบบ คำสั่งในงานวิจัย : คำนวณค่าน้ำหนักของชั้นซ่อนและชั้นเอาต์พุต เพื่อหาผลการทดสอบของโครงข่าย

```

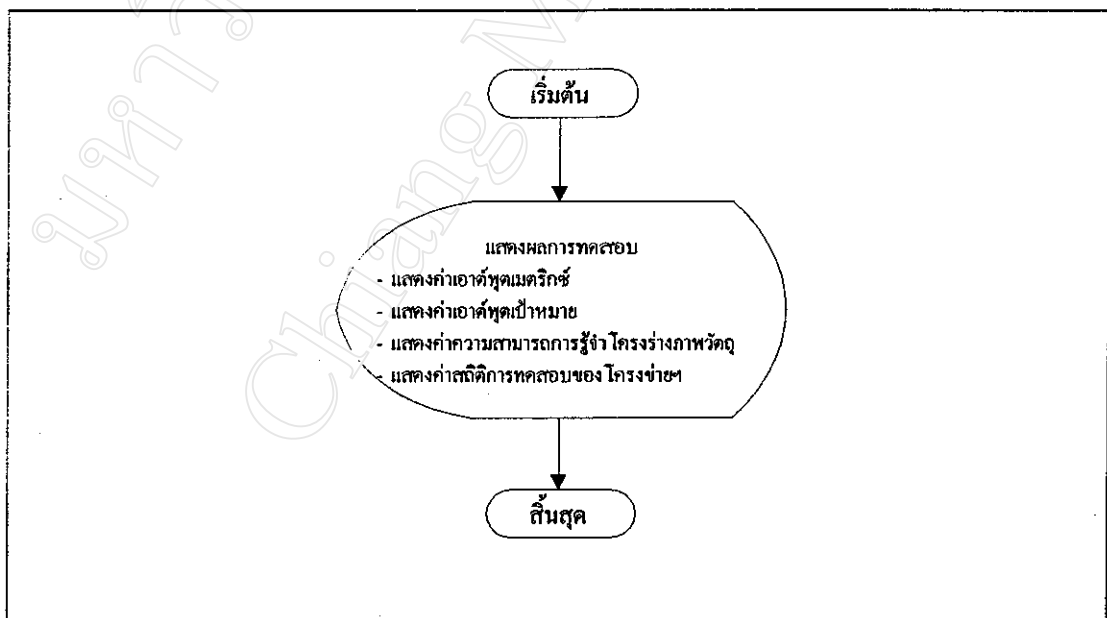
%Test Images by Use Weight & Bias From Trained.
Pb_ans=uicontrol(gcf,...
    'Style','push',...
    'String','Answer',...
    'Position','[Left Bottom Width High]',...
    'Callback',[...
        'I=noisy;',...
        'BW=edge(I, meth);',...
        '[x,y]=size(BW)']);
  
```

คำอธิบาย คำสั่ง : คำนวณค่าน้ำหนักของชั้นซ่อน และชั้นเอาต์พุต เพื่อหาผลการทดสอบของโครงข่าย

● *Test Images by Use Weight & Bias*

เป็นการป้อนภาพทดสอบที่ผ่านวิธีการหาขอบภาพแล้ว เข้าไปยังชั้นอินพุตของโครงข่าย แล้วนำค่าน้ำหนัก และไบแอส ออกจากแฟ้มข้อมูลที่ได้เก็บบันทึกขั้นตอนกระบวนการฝึกสอน จากนั้นจึงนำไปคำนวณหาค่าน้ำหนัก,ไบแอส ของชั้นซ่อน และชั้นเอาต์พุต ด้วยคำสั่ง *simuff* (Simulate Feed Forward Network)

ลำดับที่ 6 แสดงผลการทดสอบ โดยเปรียบเทียบ เอาต์พุตเมตริกซ์ , เอาต์พุตเป้าหมาย กับ ค่าเอาต์พุตของโครงข่าย , แสดงค่าความสามารถในการรู้จำโครงร่างภาพวัตถุ และ แสดงค่าสถิติการทดสอบของโครงข่ายฯ มีผังขั้นตอนการทำงาน ดังรูป 4.22



รูป 4.22 ผังขั้นตอนกระบวนการทดสอบ ลำดับที่ 6

รูปแบบ คำสั่งในงานวิจัย : แสดงค่าเอาต์พุตเมตริกซ์

```
%Comparator find Target Value
ref=0.5;
z=row*col;
p=zeros(z,1);

%Rem Load Weight & Bias From Files.
load(wfile);

%Rem Simulate Feed-Forward Network.
y1=simuff([p],W1,B1,UT1,W2,B2,UT2,W3,B3,UT3);

%Rem Output Matrix Checker
if [y1] >= [1.25 1.25 1.25] ,
    y1=[0;0;0];
elseif [y1] >=[-0.25 -0.25 -0.25] ,
    y1=[0;0;0];
else
    for i = 1:3;
        if ref-y1(i,1)>=0 ,
            y1(i,1)=0;
        else
            y1(i,1)=1;
        end,
    end,
end.
```

คำอธิบาย คำสั่ง : แสดงค่าเอาต์พุตเมตริกซ์

ค่าเอาต์พุตที่ได้จากการคำนวณอาจจะมีค่าผิดพลาดจากเป้าหมายเอาต์พุตที่ได้กำหนดไว้ในขั้นตอนการฝึกสอน จึงมีการจัดค่าอ้างอิงเพื่อใช้ ตรวจสอบ ค่าเอาต์พุตของโครงข่าย

ในขั้นตอนการเขียนคำสั่งได้กำหนดให้ $ref=0.5$ เป็นค่าตรวจสอบ เนื่องจากเป้าหมายเอาต์พุตได้กำหนดในขั้นตอนการฝึกสอนมีจำนวน สอง ค่าคือ 0 กับ 1

ดังนั้นหากค่าเอาต์พุตที่ได้จากการคำนวณมีค่า น้อยกว่า ref หรือ 0.5 นั้นหมายความว่า สมมติฐานการตัดสินใจของเอาต์พุตโครงข่ายคือ 0

ในทำนองเดียวกัน หากค่าเอาต์พุตที่ได้จากการคำนวณมีค่า มากกว่า ref หรือ 0.5 นั้นคือ สมมติฐานการตัดสินใจของเอาต์พุตโครงข่ายคือ 1

รูปแบบ คำสั่งในงานวิจัย : แสดงค่าเอาต์พุตเป้าหมาย

```
%Rem Determine Target Matrix
t1=[0;0;1];      %Rem animal (2Legs)
t2=[0;1;0];      %Rem animal (4Legs)
t3=[0;1;1];      %Rem animal (Bugs)
t4=[1;0;0];      %Rem article (Chairs)
t5=[1;0;1];      %Rem article (Aircraft)
t6=[1;1;0];      %Rem article (Vehicle)
t7=[1;1;1];      %Rem man
%
%Rem say Answer
if      [y1]==t1 , say='It is a 2legs';
elseif [y1]==t2 , say='It is a 4legs';
elseif [y1]==t3 , say='It is a Bugs';
elseif [y1]==t4 , say='It is a Chairs';
elseif [y1]==t5 , say='It is an Aircraft';
elseif [y1]==t6 , say='It is a Vehicle';
elseif [y1]==t7 , say='This is a Man';
else      , say='I 'm not sure';
end.
```

คำอธิบาย คำสั่ง : แสดงค่าเอาต์พุตเป้าหมาย

ผลจากการเปรียบเทียบ ค่าเอาต์พุตของโครงข่ายฯ [y1] กับ เอาต์พุตเป้าหมาย tn ได้ค่าคอมเปรียบเทียบเป้าหมายที่ถูกกำหนดไว้ในกระบวนการฝึกสอน

รูปแบบ คำสั่งในงานวิจัย : แสดงค่าความสามารถการรู้จำโครงร่างภาพวัตถุ

```
%Rem Check say Answer Ability.
if      [y1]==t1 ,
    say='It is a 2legs'; Recognition = Recognition + 1;
elseif [y1]==t2 ,
    say='It is a 4legs'; Recognition = Recognition + 1;
elseif [y1]==t3 ,
    say='It is a Bugs'; Recognition = Recognition + 1;
elseif [y1]==t4 ,
    say='It is a Chairs'; Recognition = Recognition + 1;
elseif [y1]==t5 ,
    say='It is an Aircraft'; Recognition = Recognition + 1;
elseif [y1]==t6 ,
    say='It is a Vehicle'; Recognition = Recognition + 1;
elseif [y1]==t7 ,
    say='This is a Man'; Recognition = Recognition + 1;
else , say='I 'm not sure';
end.
```

คำอธิบาย คำสั่ง : แสดงค่าความสามารถการรู้จำโครงร่างภาพวัตถุ

ผลจากการเปรียบเทียบ ค่าเอาต์พุตของโครงข่ายฯ [y1] กับ เอาต์พุตเป้าหมาย tn ได้คำตอบเปรียบเทียบกับเป้าหมายที่ถูกกำหนดไว้ในกระบวนการฝึกสอน และเพิ่มจำนวนตัวนับความสามารถในการรู้ $Recognition = Recognition + 1$

รูปแบบ คำสั่งในงานวิจัย : แสดงค่าสถิติการทดสอบของโครงข่ายฯ

```
Number_pic = Number_pic + 1;
Number_pic_Percentage = (Number_pic/Number_pic)*100;
Recognition = Recognition +1;
Recognition_Percentage= (Recognition/Number_pic)*100;
UnRecognition = UnRecognition +1;
UnRecognition_Percentage= (UnRecognition/Number_pic)*100;
UnKnow = UnKnow + 1;
UnKnow_Percentage= (UnKnow/Number_pic)*100;
```

คำอธิบาย คำสั่ง : แสดงค่าสถิติการทดสอบของโครงข่ายฯ

เป็นการเก็บบันทึก และ แสดงค่าสถิติของการทดสอบโครงข่ายฯโดยใช้คำตอบเป็นจำนวนของภาพรวม (Number_pic), จำนวนของภาพรวมคิดเป็นร้อยละ (Number_pic_Percentage), จำนวนภาพที่สามารถรู้จำ (Recognition) , จำนวนของภาพที่สามารถรู้จำคิดเป็นร้อยละ (Recognition_Percentage), จำนวนของภาพที่จำไม่ได้ (UnRecognition) , จำนวนของภาพที่จำไม่ได้คิดเป็นร้อยละ (UnRecognition_Percentage), จำนวนของภาพที่ไม่รู้จัก (UnKnow) และจำนวนของภาพที่ไม่รู้จักคิดเป็นร้อยละ (UnKnow_Percentage) เพื่อให้ทราบถึง ความสามารถในการวิเคราะห์โครงร่างภาพวัตถุ ว่ามีประสิทธิภาพดีเพียงใด วิธีการใช้งานโปรแกรมการรู้จำโครงร่างภาพวัตถุด้วยนิวรอลเน็ตเวิร์คแสดงในภาคผนวก ค.