บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ ได้แก่ ทฤษฎีของการประมวลผลภาพ มาตราฐานสี ระบบสี RGB ระบบสี HSV การแปลงระบบสี RGB เป็น HSV การทำเทรชโฮลด์ (Threshold) การขยายพิกเซลของภาพ (Dilation) และการลดพิกเซลของภาพ (Erosion) การหาขอบของรูปภาพ (Image edge detection) การลดขนาดความหนาของเส้น (Thinning) และทฤษฎีนิวรอลเน็ตเวิร์ก รวมทั้งตัวอย่างของงานวิจัยที่มีการนำนิวรอลเน็ตเวิร์กไปใช้ในการทำนายการหมุนของวัตถุ

2.1 การประมวลผลภาพ

การประมวลผลภาพเป็นเทคนิคในการจัดการภาพซึ่งอยู่ในรูปแบบของข้อมูลดิจิทัล โดยนำอัลกอริทึมต่างๆ มาใช้ในการจัดการภาพ เพื่อนำมาใช้ในการวิเคราะห์ และหาผลลัพธ์ของภาพตามที่ต้องการ เป็นต้น ซึ่งในที่นี้จะอธิบายเฉพาะพื้นฐานและเทคนิคที่ใช้ในการประมวลผลภาพที่เกี่ยวข้องกับงานวิจัยนี้ ได้แก่ มาตรฐานสี การทำเทรชโฮลด์ การขยายพิกเซลของภาพ และการลดพิกเซลของภาพ การหาขอบของรูปภาพ และการลดขนาดความหนาของเส้นขอบ

2.1.1 มาตรฐานสี

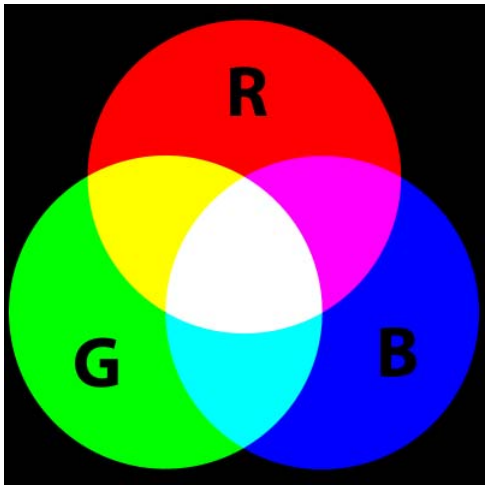
มาตรฐานของสีนั้นมีอยู่หลายระบบ เช่น RGB (Red, Green และ Blue) HSV (Hue, Saturation และ Value), CMYK (Cyan, Magenta, Yellow และ Black), CIE (International Commission on Illumination) และ HLS (Hue, Lightness และ Saturation) เป็นต้น โดยในงานวิจัยนี้จะขอกล่าวถึงเฉพาะระบบสีที่เกี่ยวข้องกับงานวิจัย ได้แก่ ระบบสี RGB และ HSV

2.1.1.1 ระบบสี RGB

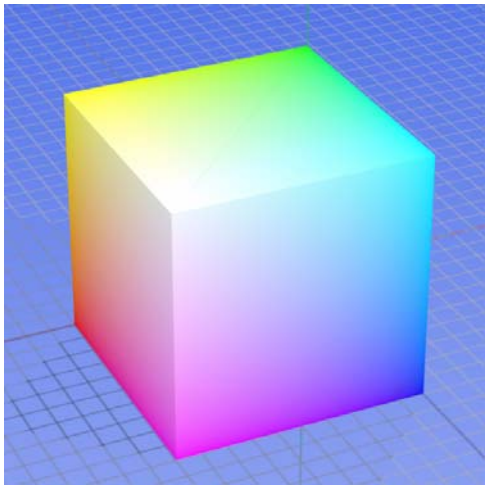
คือ ระบบแสงสีมีพื้นฐานจากหลักการของการมองเห็นแสงสี และการผสมของแสงสีในอัตราส่วนต่างๆ เป็นระบบสีที่นิยมใช้กันทั่วไป โดยอาศัยหลักการการรวมแม่สีของแสงเข้าด้วยกัน ได้แก่ แสงสีแดง แสงสีเขียว และแสงสีเหลือง มีค่าตั้งแต่ 0 ถึง 225 เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่ อีก 3 สี คือ สีมาเจนต้า (Magenta) สีฟ้าไซแอน (Cyan) และสีเหลือง เมื่อฉาย

แสงสีทั้งหมดรวมกันจะได้แสงสีขาว ดังภาพที่ 2.1 และโมเดลระบบสี RGB สามารถแสดงได้ดังภาพที่ 2.2 ซึ่งแกน X แทนแสงสีแดง แกน Y แทนแสงสีน้ำเงิน และแกน Z แทนแสงสีเขียว สำหรับภาพที่นำมาใช้ในงานวิจัยนี้เป็นภาพที่ถูกเก็บค่าสีในระบบ RGB ซึ่งค่าของสีแดง เขียว และเหลืองจะเปลี่ยนไปตามความเข้มแสง กล่าวคือ ถ้าแสงสว่างมาก ค่าของสีทั้งสามสีจะมีค่าเพิ่มขึ้น แต่หากมีแสงสว่างน้อย ค่าของสีทั้งสามจะมีค่าลดลง

ภาพที่ 2.1
การรวมของแสงสีในระบบ RGB



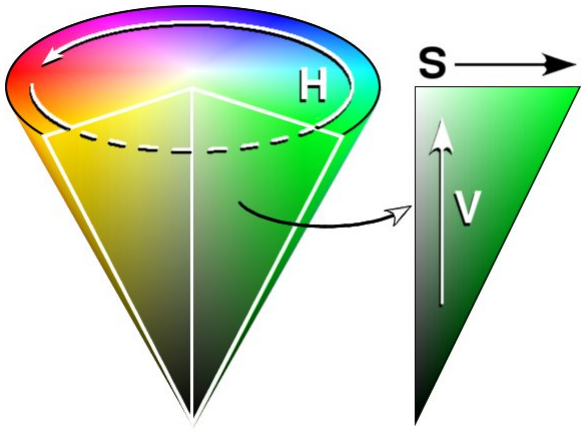
ภาพที่ 2.2
โมเดลระบบสี RGB



2.1.1.2 ระบบสี HSV

HSV เป็นระบบสีที่อาศัยหลักการใช้ ค่าของสี (Hue, H) ค่าความเข้มของเนื้อสี หรือค่าความบริสุทธิ์ของสี (Saturation, S) และ ค่าความสว่างของสี (Value, V) ซึ่งเสนอโดย A.R. Smith (1978) โดย Hue คือค่าสีหลัก เช่น สีแดง สีเขียว สีน้ำเงิน มีค่าอยู่ระหว่าง 0 ถึง 255 ถ้าเกิดค่าของสีมีค่าเท่ากับ 0 จะแทนให้เป็นสีแดง และเมื่อค่าของสีมีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนไปตามความถี่สเปกตรัมของสีจนถึง 256 แล้วจะกลับมาเป็นสีแดงเช่นเดิมอีกครั้ง และสามารถแทนให้อยู่ในรูปองศาได้ คือ สีแดง มีค่าเท่ากับ 0 องศา สีเขียว มีค่าเท่ากับ 120 องศา และสีน้ำเงิน มีค่าเท่ากับ 240 องศา สำหรับค่าความเข้มของเนื้อสีหรือค่าความเข้มของเนื้อสี เมื่อมีค่าเพิ่มขึ้น สีจะมีความเข้มมากขึ้นเรื่อยๆ และสุดท้ายค่าความสว่างของสี เมื่อมีค่าเพิ่มมากขึ้นภาพจะมีความสว่างเพิ่มขึ้น การบอกสีในระบบ HSV สามารถแสดงได้ดังภาพที่ 2.3

ภาพที่ 2.3
การบอกค่าสีในระบบ HSV



จะสังเกตว่าการบอกสีในระบบ HSV จะมีความคงทนต่อการเปลี่ยนแปลงของแสงในสภาพแวดล้อมมากกว่า การใช้สีในระบบ RGB เนื่องจากหากภาพที่ความสว่างมาก ค่าของสี และค่าความเข้มของเนื้อสีจะไม่มีการเปลี่ยนแปลง มีเพียงค่าความสว่างของสีเปลี่ยนแปลงเพียงค่าเดียวเท่านั้น

2.1.1.3 การแปลงระบบสี RGB เป็น HSV

เนื่องจากภาพที่นำมาใช้ในการประมวลผลนั้นเก็บค่าสีในระบบ RGB ซึ่งค่าสีที่อยู่ในระบบ RGB นั้นจะประกอบไปด้วยค่าสี ค่าแสง และค่าความสว่างซึ่งจะมีความซับซ้อนในการแยกแยะสี เนื่องจากมีค่าแสงและค่าความสว่างผสมอยู่ด้วยดังนั้น ในงานวิจัยนี้จึงได้ทำการแปลงระบบสีแบบ RGB ให้เป็นแบบ HSV เพื่อให้สามารถทำการแยกสีในกระบวนการทำเทรซโฮลด์ได้ดีกว่า โดยระบบสีแบบ RGB และ HSV สามารถแสดงความสัมพันธ์ได้ดังสมการ (2.1) (2.2) และ (2.3)

โดยที่

R G B แทนค่าของสีในระบบ RGB มีค่าระหว่าง 0.0 – 1.0

H S V แทนค่าของสีในระบบ HSV

max = ค่าสูงสุดใน (R, G, B)

min = ค่าต่ำสุดใน (R, G, B)

และ

$$H\begin{cases} 60\times\dfrac{G-B}{\max-\min} & \text{เมื่อ max = R} \\ 60\times\dfrac{B-R}{\max-\min}+120 & \text{เมื่อ max = G} \\ 60\times\dfrac{R-G}{\max-\min}+240 & \text{เมื่อ max = B} \end{cases} \qquad (2.1)$$

$$S=\frac{\max-\min}{\max}\times 100 \qquad (2.2)$$

$$V=\max\times 100 \qquad (2.3)$$

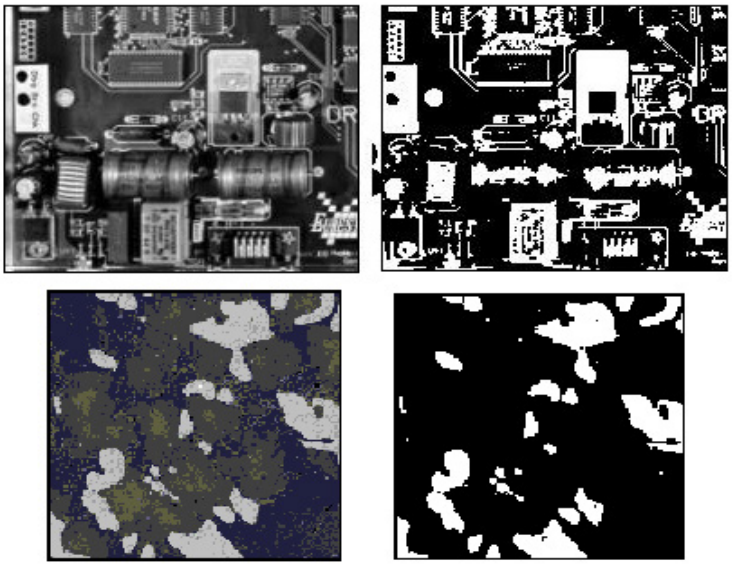
2.1.2 การทำเทรซโฮลด์

การทำเทรซโฮลด์เป็นการแยกบริเวณหรือวัตถุที่เราต้องการออกมาจากสิ่งแวดล้อมหรือฉากหลัง โดยหลักการทำเทรซโฮลด์ คือการเลือกค่าเทรซโฮลด์ที่เหมาะสม เพื่อแยกค่าสีที่ต้องการออกมาจากสิ่งแวดล้อม การทำเทรซโฮลด์แบ่งออกเป็น 2 รูปแบบใหญ่ๆ ได้แก่ การทำเทรซโฮลด์ค่าเดียวกันทั้งภาพ (Global Threshold) และการแบ่งภาพออกเป็นภาพย่อยๆ ซึ่งแต่ละภาพก็จะมีค่าเทรซโฮลด์เป็นของตัวเอง (Local Threshold) ส่วนการทำเทรซโฮลด์ ให้ได้ภาพดีและคมชัด ต้องเกิดจากการเลือกค่าเทรซโฮลด์ที่ถูกต้องและเหมาะสม ถ้าเลือกค่าเทรซโฮลด์ไม่เหมาะสม เช่น ค่าเทรซโฮลด์ที่มากหรือน้อยจนเกินไป ภาพที่ได้จะขาดความคมชัดหรืออาจทำให้รายละเอียดของภาพขาดหายไป หรือภาพที่ได้อาจจะมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน (Noise) เกิดขึ้น ทำให้ภาพผลลัพธ์ที่ได้ไม่ชัดเจน ตัวอย่างภาพที่ผ่านกระบวนการเทรซโฮลด์สามารถแสดงได้ดังภาพที่ 2.4

สำหรับงานวิจัยนี้จะทำการเทรซโฮลด์ภาพสี ดังนั้นในขั้นตอนแรกภาพในระบบ RGB จะถูกแปลงให้อยู่ในระบบ HSV และทำการเลือกค่าของสี, ค่าความเข้มของเนื้อสี และค่าความสว่างของสีที่เหมาะสมเพื่อแยกค่าสีที่ต้องการออกจากสิ่งแวดล้อม

ภาพที่ 2.4

ภาพการทำเทรซโฮลด์ (ก) ภาพต้นฉบับ (ข) ผลลัพธ์จากการทำเทรซโฮลด์



(ก) (ข)

2.1.3 การขยายพิกเซลของภาพและการลดพิกเซลของภาพ

การขยายพิกเซลของภาพและการลดพิกเซลของภาพ ถือเป็นส่วนหนึ่งของ Morphological Image Processing ซึ่งเป็นการประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ ซึ่งมีโอเปอเรชันพื้นฐานโดยทั่วไป เช่น การขยายพิกเซลของภาพ คือ การขยายภาพโดยมีสัดส่วนเท่ากันทั่วทั้งภาพ (Uniform) การลดพิกเซลของภาพ คือการย่อภาพและการทำ Skeleton เป็นการหาโครงสร้างหลักของวัตถุ

2.1.3.1 การขยายพิกเซลของภาพ

การขยายพิกเซลของภาพ คือ การขยายพิกเซลของภาพโดยการสแกนค่าของ Structuring Element (SE) บนแต่ละพิกเซลของภาพ โดยทำการสแกนจากตำแหน่งบนซ้ายไปยังตำแหน่งล่างขวา ซึ่งกระบวนการจะทำการเปลี่ยนค่าของพิกเซลที่มีค่าเป็น 0 ให้มีค่าเป็น 1 เมื่อค่าของพิกเซลใดๆ พิกเซลหนึ่งบน SE มีค่าตรงกับค่าของพิกเซลของภาพ และจะมีค่าคงเดิม เมื่อทุกค่าของ SE มีค่าตรงกับทุกค่าของพิกเซลภาพ การทำงานของการขยายพิกเซลของภาพสามารถแสดงได้ดังภาพที่ 2.5 ตัวอย่างของภาพที่ผ่านการทำการขยายพิกเซลของภาพแสดงดังภาพที่ 2.6 ประโยชน์ของการทำการขยายพิกเซลของภาพ คือสามารถใช้แก้ไขภาพของตัวอักษรที่มีเส้นที่ไม่ต่อเนื่องกัน ให้เชื่อมต่อเนื่องกัน เป็นต้น สมการของการทำการขยายพิกเซลของภาพสามารถแสดงได้ดังสมการที่ (2.4)

$$A \oplus B = \bigcup_{x \in B} (A_x) \tag{2.4}$$

โดยที่

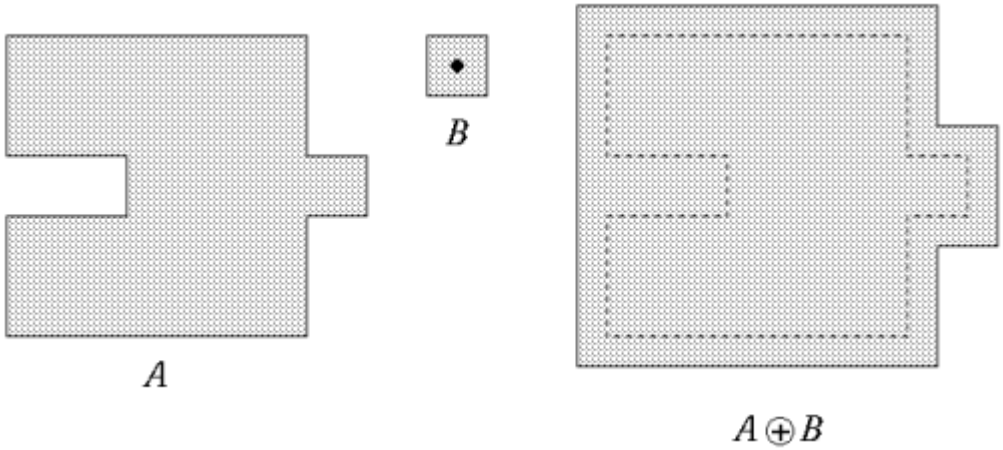
A คือ ภาพที่จะทำการขยายพิกเซลของภาพ

B คือ Structuring Element

ความหมายคือทุก ๆ พิกเซล $x \in B$ ทำการเคลื่อนย้ายไปยัง A โดยการยูเนี่ยนไปตามพิกัดของ x

ภาพที่ 2.5

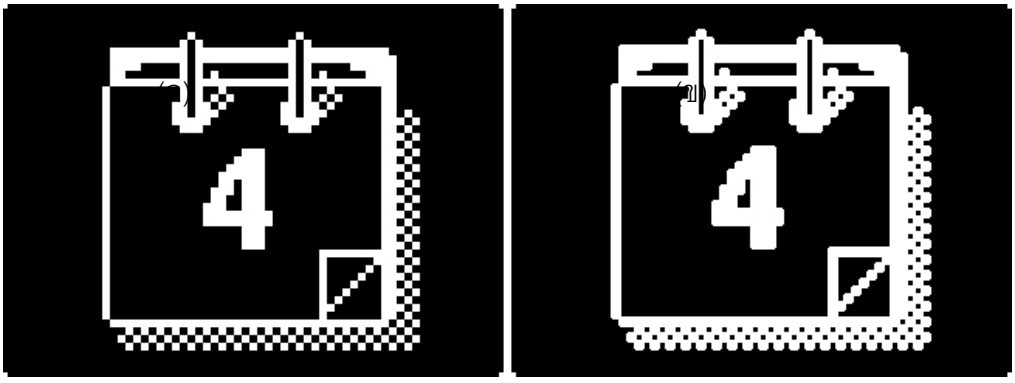
การทำงานของการขยายพิกเซลของภาพ



ภาพที่ 2.6

ภาพการทำการขยายพิกเซลของภาพ (ก) ภาพต้นฉบับ

(ข) ผลลัพธ์จากการทำการขยายพิกเซลของภาพ



(ก) (ข)

2.1.3.2 การลดพิกเซลของภาพ

การลดพิกเซลของภาพ เป็นวิธีการที่ตรงข้ามกับการขยายพิกเซลของภาพ คือ จะทำการลดขนาดของพิกเซล โดยการสแกนค่าของ SE บนแต่ละค่าของพิกเซลภาพ โดยทำการสแกนจากตำแหน่งบนซ้ายไปยังตำแหน่งล่างขวา จะเปลี่ยนค่าของพิกเซลที่มีค่าเป็น 1ให้มีค่าเป็น 0 เมื่อพิกเซลใดพิกเซลหนึ่งบน SE มีค่าตรงกับค่าของพิกเซลภาพ และจะมีค่าคงเดิม เมื่อทุก

พิกเซลของ SE มีค่าตรงกับค่าของพิกเซลภาพ ประโยชน์ของวิธีการนี้คือสามารถใช้ในการกำจัดส่วนของภาพที่ไม่ต้องการออกไป สมการของการทำการลดพิกเซลของภาพสามารถแสดงได้ดังสมการที่ (2.5) การทำงานของการลดพิกเซลของภาพสามารถแสดงได้ดังภาพที่ 2.7 ตัวอย่างของภาพที่ผ่านการทำการลดพิกเซลของภาพ แสดงดังภาพที่ 2.8

$$A\Theta B = \{w : B_w \subset A\} \tag{2.5}$$

โดยที่

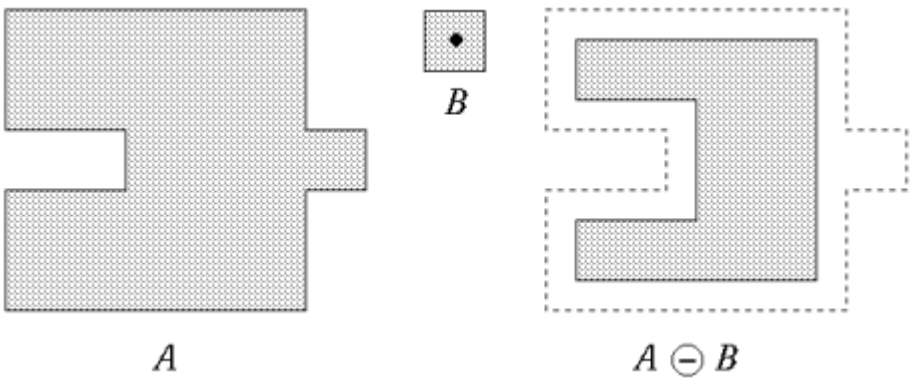
A คือ ภาพที่จะทำการลดพิกเซลของภาพ

B คือ Structuring Element

ความหมายคือ B_w เป็นสับเซตของ A โดยที่ค่าของ B จะต้องประกอบด้วยทุกๆ พิกเซลของ w มีพิกัดเป็น (x, y) ซึ่งค่า B_w จะต้องอยู่ใน A

ภาพที่ 2.7

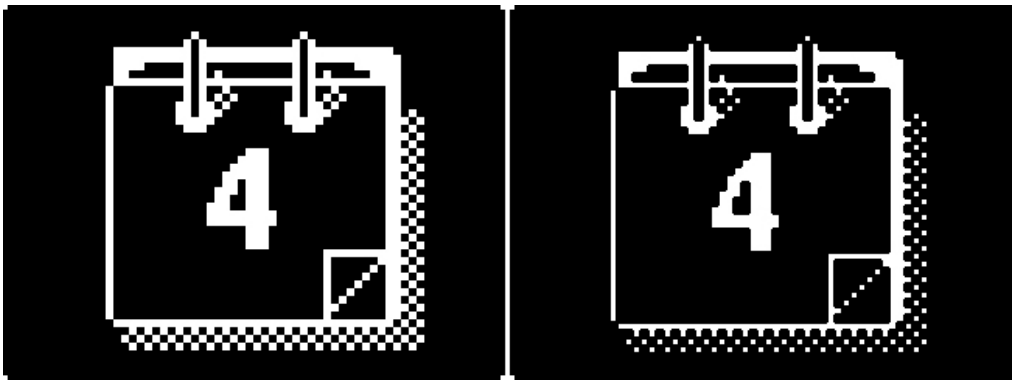
การทำงานของการลดพิกเซลของภาพ



ภาพที่ 2.8

ภาพการทำการลดพิกเซลของภาพ (ก) ภาพต้นฉบับ

(ข) ผลลัพธ์จากการทำการลดพิกเซลของภาพ



(ก) (ข)

2.1.4 **การหาขอบของรูปภาพ** (Image edge detection)

การหาขอบของรูปภาพคือการตรวจสอบหาเส้นรอบวัตถุที่อยู่ในภาพ โดยขอบของภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง หากความต่างนี้มีค่ามากขอบก็จะมีความชัดเจน แต่ถ้ามีค่าน้อยขอบก็จะไม่ชัดเจน การหาขอบจึงเป็นการวัดการเปลี่ยนแปลงของความเข้มแสงในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบนั้นมีด้วยกันหลายวิธี แต่อย่างไรก็ตามการหาขอบของรูปภาพสามารถแบ่งออกได้เป็น 2 กลุ่มหลัก คือ วิธีการแบบ Gradient และวิธีการแบบ Laplacian โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้

2.1.4.1 **วิธีการแบบ** Gradient

วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า เทรชโฮลด์ จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts edge detection, Prewitt edge detection, Sobel edge detection และ Canny edge detection เป็นต้น

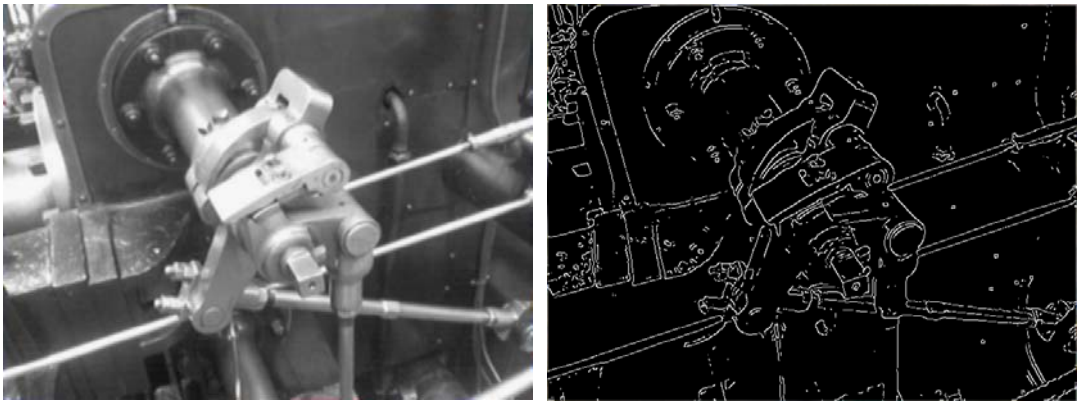
2.1.4.2 **วิธีการแบบ** Laplacian

วิธีนี้จะหาขอบโดยใช้อนุพันธ์อันดับสอง โดยใช้จุดที่ค่า y เป็น 0 (Zerocrossing) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

ตัวอย่างของภาพที่ผ่านกระบวนการหาขอบสามารถแสดงได้ดังภาพที่ 2.9

ภาพที่ 2.9

กระบวนการหาขอบ (ก) ภาพต้นฉบับ (ข) ผลลัพธ์จากการหาขอบ



(ก) (ข)

สำหรับการหาขอบของรูปภาพในงานวิจัยนี้ได้เลือกใช้วิธี Prewitt edge detection ซึ่งวิธีการนี้จะคำนวณหาขอบที่เป็นเกรเดียนต์เวกเตอร์ของทุกจุดบนภาพที่เป็นภาพต้นฉบับ โดยค่าระดับความเข้มสีเทาที่สูงกว่า (higher grey-level) จะแสดงถึงขอบระหว่างวัตถุสองชิ้น สำหรับฟิวเตอร์ (filter) ของ Prewitt edge detector ที่นำมาใช้ในการคำนวณค่าเกรเดียนต์จะเป็น SE ขนาด 3x3 พิกเซล ดังภาพที่ 2.10

ภาพที่ 2.10

SE ขนาด 3x3 พิกเซล ของ Prewitt edge detection

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

X

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

Y

พิจารณาตัวอย่างรูปภาพขนาด 3x3 ดังภาพที่ 2.11

ภาพที่ 2.11
รูปภาพขนาด 3x3 พิกเซล

a1	a2	a3
a4	a5	a6
a7	a8	a9

โดยที่

a1 … a9 คือระดับความเข้มสีเทา (grey level) ของแต่ละพิกเซลของรูปภาพ ค่า X และ Y สามารถคำนวณที่ (2.6) และ (2.7) ตามลำดับ

$$X = - (1*a1) + (1*a3) – (1*a4) + (1*a6) – (1*a7) + (1*a9) \quad (2.6)$$

$$Y = (1*a1) + (1*a2) + (1*a3) - (1*a7) – (1*a8) - (1*a9) \quad (2.7)$$

สามารถคำนวณหาค่า Prewitt gradient ได้ดังสมการที่ (2.8)

$$\text{Prewitt gradient} = \text{SQRT}((X*X) + (Y*Y)) \quad (2.8)$$

หลังจากนั้นพิกเซลทุกๆ พิกเซลบนรูปภาพจะถูกคำนวณ โดยพิกเซลที่อยู่บริเวณขอบของรูปภาพจะถูกจำลองค่าเพื่อให้มีข้อมูลเพียงพอสำหรับการคำนวณ

2.1.5 การลดขนาดความหนาของเส้น

หลังจากผ่านกระบวนการหาขอบของรูปภาพ ในบางครั้งขอบของรูปภาพมีขนาดหนามากกว่า 1 พิกเซล ในขั้นตอนกระบวนการลดขนาดความหนาของเส้นขอบนั้น จะทำการลดความหนาของเส้นขอบให้เหลือเพียง 1 พิกเซล เพื่อนำไปใช้ในการวางตำแหน่งโทเคนต่อไป

โดยการทำการลดขนาดความหนาสามารถทำได้โดยอาศัยกฎ 2 ข้อคือ P1 และ P2 ตามสมการที่ 2.9 และ 2.10 ตามลำดับ ขั้นแรกจะใช้กฎ P1 โดยการนำ structure element ขนาด 3x3 สแกนไปตามข้อมูลภาพ และทำการพิจารณาพิกเซลบริเวณขอบภาพว่าสามารถลบได้หรือไม่ ถ้าลบได้ให้หมายเหตุไว้แต่ยังไม่ทำการลบทันที หลังจากทำการสแกนทั่วทั้งภาพจึงจะทำการลบข้อมูลภาพที่ได้หมายเหตุไว้ ขั้นตอนที่สองใช้กฎ P2 และดำเนินการเหมือนการใช้กฎข้อที่ P1 เมื่อทำการลบข้อมูลภาพที่มีไว้ในหมายเหตุแล้ว ก็ให้ทำซ้ำต่อไปเรื่อยๆ จนไม่สามารถลบข้อมูลภาพออกได้อีก

ลักษณะของข้อมูลที่นำมาพิจารณาจะมีขนาดเท่ากับ 3x3 ดังภาพ 2.12 โดยที่พิกเซลปัจจุบันคือพิกเซลตรงกลางดังนี้คือ

ภาพที่ 2.12
ภาพที่นำมาพิจารณาในการทำการลดขนาดความหนาของเส้น

p_8	p_1	p_2
p_7	p_0	p_3
p_6	p_5	p_4

กำหนดให้ $N(p_0) = \sum_{i=1}^{8} p_i$ เป็นจำนวนของพิกเซลรอบ p_0 เมื่อ p_0 = 0, 1 และ i =0,1,2…,8

$T(p_0)$ แสดงถึงการเปลี่ยนแปลงของข้อมูลจาก 0 ไปเป็น 1 เมื่อพิจารณาข้อมูลใน $p_1, p_2, p_3, \ldots p_8, p_1$ ตามลำดับ

สำหรับกฎ P1 และ P2 จะมีลอจิกแสดงได้ดังสมการที่ (2.9) และ (2.10)

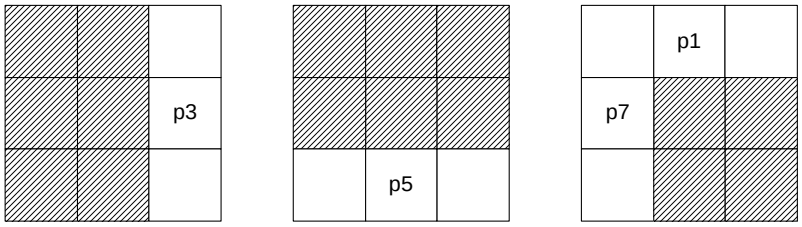
P1: $(2 <= N(p_0) <= 6) \&\& (T(p_0) = 1) \&\& (p_1.p_3.p_5 = 0) \&\& (p_3.p_5.p_7 = 0)$ (2.9)

P2: $(2 <= N(p_0) <= 6) \&\& (T(p_0) = 1) \&\& (p_1.p_3.p_7 = 0) \&\& (p_1.p_5.p_7 = 0)$ (2.10)

สำหรับเงื่อนไขแรกของ P1 และ P2 คือพิกเซลตรงกลาง (p_0) สามารถลบได้ก็ต่อเมื่อค่าของ p_1 ถึง p_8 มีค่าเท่ากับ 1 (เป็นสีดำ) มากกว่า 1 ตำแหน่งและต้องไม่มากกว่า 6 ตำแหน่ง สำหรับเงื่อนไขที่ 2 กำหนดให้ $T(p_0) = 1$ หมายถึงต้องมีการเปลี่ยนแปลงของข้อมูลจาก 0 ไปเป็น 1 ของข้อมูล $p_1, p_2, p_3, ... p_8, p_1$ เพียงครั้งเดียวเท่านั้น และในเงื่อนไขที่ 3 และ 4 คือ $(p_1.p_3.p_5 = 0)$ และ $(p_3.p_5.p_7 = 0)$ จะใช้ได้สำหรับเงื่อนไขที่ p_3 =0 หรือ p_5 =0 หรือ (p_1 =0 และ p_7 =0) ตามตัวอย่างดังกล่าวนี้แสดงไว้ดังรูปที่ 2.13 ซึ่งจะเห็นว่าเมื่อ p_3 =0 จะเป็นลักษณะของขอบด้านขวา เมื่อ p_5 =0 จะเป็นขอบด้านล่าง และเมื่อ p_1 =0 และ p_7 =0 จะเป็นมุมบนซ้าย สำหรับใน P2 นั้นเราสามารถลบพิกเซล p_0 ได้ก็ต่อเมื่อ p_1 =0 หรือ p_7 =0 หรือ (p_3 และ p_5 =0) ดังแสดงในรูปที่ 2.14 ซึ่งจะเห็นว่าเมื่อ p_1 =0 จะเป็นขอบบน p_7 =0 จะเป็นขอบด้านซ้าย และเมื่อ (p_3 =0 และ p_5 =0) จะเป็นมุมล่างขวา

ภาพที่ 2.13

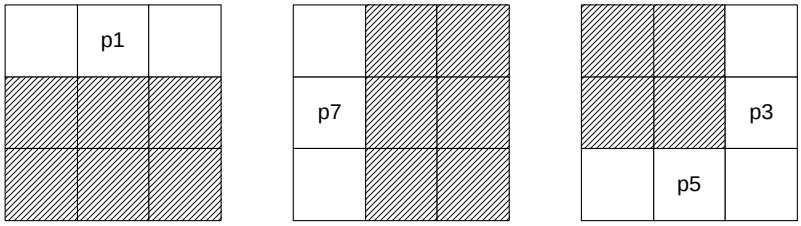
ตำแหน่งของพิกเซลที่ต้องเท่ากับ 0 จึงจะลบพิเซล p_0 ของกฎ P1



(ก) (ข) (ค)

ภาพที่ 2.14

ตำแหน่งของพิกเซลที่ต้องเท่ากับ 0 จึงจะลบพิเซล p_0 ของกฎ P2

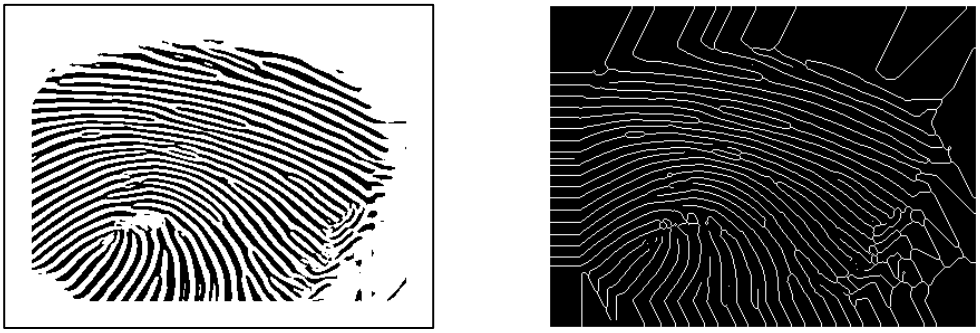


(ก) (ข) (ค)

ตัวอย่างภาพที่ผ่านกระบวนการลดขนาดความหนา สามารถแสดงได้ดังภาพที่ 2.15

ภาพที่ 2.15

กระบวนการลดขนาดความหนา (ก) ภาพต้นฉบับ (ข) ผลลัพธ์จากการลดขนาดความหนา



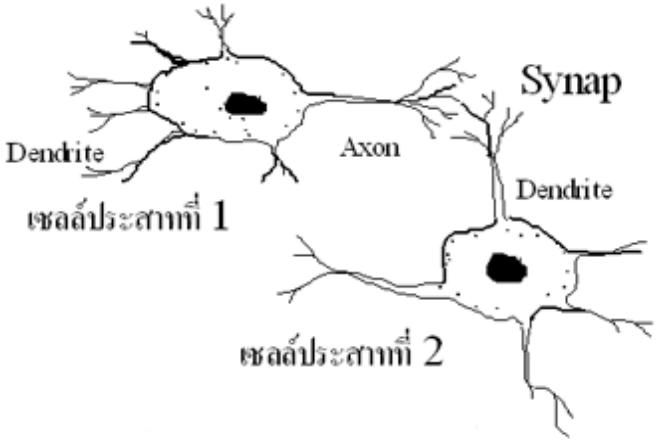
(ก) (ข)

2.2 นิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กหรือโครงข่ายประสาทเทียมเป็นศาสตร์แขนงหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence, AI) ซึ่งเลียนแบบการทำงานของเซลล์สมองหรือระบบประสาทของมนุษย์ สำหรับระบบประสาทมีลักษณะดังภาพที่ 2.16 โดยมีส่วนปลายประสาทรับรู้สัญญาณไฟฟ้า (Dendrite) ตัวรวบรวมข้อมูลที่ได้จากปลายประสาทรับรู้สัญญาณไฟฟ้า (Soma หรือ cell body) และท่อส่งข้อมูลเพื่อไปยังปลายประสาทรับรู้สัญญาณไฟฟ้าตัวอื่นๆ (Axon) โดยส่วนปลายประสาทรับรู้สัญญาณไฟฟ้าเปรียบเสมือนอินพุต (input) ที่ใช้ในการรับข้อมูลเข้ามา ตัวรวบรวมข้อมูลที่ได้จากปลายประสาทรับรู้สัญญาณไฟฟ้าเปรียบเสมือนตัวรวบรวมน้ำหนักของ

สัญญาณไฟฟ้าที่ได้มาจากปลายประสาทรับรู้สัญญาณไฟฟ้าหลายๆ ตัวเข้ามา และท่อส่งข้อมูลเพื่อไปยังปลายประสาทรับรู้สัญญาณไฟฟ้าตัวอื่นๆ เปรียบเสมือนเอาต์พุต (output) ที่จะส่งผลลัพธ์ที่ได้ไปยังนิวรอลตัวอื่นๆ ในโครงข่ายต่อไป

ภาพที่ 2.16
ระบบประสาท



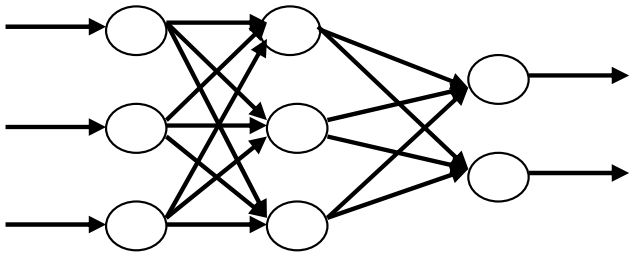
ถึงแม้นิวรอลเน็ตเวิร์กจะมีความแตกต่างจากระบบประสาทในสมองทั้งในแง่ของโครงสร้าง ความซับซ้อน และปริมาณของหน่วยย่อย แต่อย่างไรก็ดีก็มีความเหมือนกันในแง่ที่นิวรอลเน็ตเวิร์ก คือการรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อยๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญา รวมทั้งลักษณะการทำงานของนิวรอลเน็ตเวิร์กก็มีความใกล้เคียงกับเซลล์สมองหรือระบบประสาทโดยเฉพาะของมนุษย์ สำหรับแหล่งข้อมูลที่นำมาใช้เพื่อให้ระบบนิวรอลเน็ตเวิร์กเกิดการรู้จำอาจนำมาจากหลายแหล่ง ทั้งจากข้อมูลปัจจุบัน ข้อมูลในอดีต หรือข้อมูลที่ได้จากแบบจำลอง

2.2.1 **นิวรอลเน็ตเวิร์กแบบหลายชั้น** (Multilayers Neural Networks)

นิวรอลเน็ตเวิร์กสามารถนำมาต่อกันเป็นโครงข่ายหลายชั้นโดยแต่ละชั้นมีหน่วยประสาทเทียมหรือโหนดมากกว่าหนึ่งตัวได้ สัญลักษณ์ของนิวรอลเน็ตเวิร์กแบบหลายชั้นสามารถแสดงได้ดังภาพที่ 2.17 โดยชั้นที่รับข้อมูลเข้ามาเรียกว่าชั้นอินพุต (input layer) ส่วนชั้นสุดท้ายที่ส่งข้อมูลออกไปเป็นผลลัพธ์เรียกว่าชั้นเอาต์พุต (output layer) และชั้นที่อยู่ระหว่างชั้นอินพุตและชั้นเอาต์พุตเรียกว่าชั้นแฝง (hidden layer) จำนวนโหนดของแต่ละชั้นไม่จำเป็นต้องเท่ากัน ฟังก์ชัน

การแปลงถ่ายทอดของแต่ละชั้นก็ไม่จำเป็นต้องเหมือนกัน นิวรอลเน็ตเวิร์กดังภาพที่ 2.17 เป็นแบบส่งผ่านข้อมูลไปข้างหน้า (feed-forward neural networks) ส่วนนิวรอลเน็ตเวิร์กที่มีการนำข้อมูลเอาต์พุตย้อนกลับมาเป็นข้อมูลอินพุตเรียกว่านิวรอลเน็ตเวิร์กแบบรีเคอร์เรนต์ (recurrent neural networks)

ภาพที่ 2.17
โครงสร้างของนิวรอลเน็ตเวิร์ก



2.2.2 **อัลกอริทึมแบ็กพรอพาเกชัน** (Backpropagation Algorithm)

อัลกอริทึมแบ็กพรอพาเกชันเป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของนิวรอลเน็ตเวิร์กวิธีหนึ่งที่นิยมใช้ในนิวรอลเน็ตเวิร์กแบบหลายชั้น ข้อมูลจากชั้นอินพุตจะถูกคำนวณและส่งผ่านฟังก์ชันจากชั้นแฝงไปยังชั้นเอาต์พุต ซึ่งหลักการสำคัญของการเรียนรู้คือ การเปลี่ยนแปลงค่าน้ำหนักของแต่ละเส้นเชื่อมระหว่างโหนด โดยในการปรับแก้ค่าน้ำหนักจะขึ้นกับความแตกต่างระหว่างค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ สำหรับขั้นตอนในการปรับแก้ค่าน้ำหนักมีขั้นตอนดังต่อไปนี้

1. กำหนดค่าอัตราการเรียนรู้ และค่าโมเมนตัม
2. ปรับแก้ค่าน้ำหนักของแต่ละเส้นเชื่อมระหว่างโหนดตามสมการที่ (2.11)

$$\Delta w_{ji}(n+1) = \eta\delta(n)\cdot y_j(n) + \alpha\Delta w_{ji}(n) \qquad (2.11)$$

โดยที่

x_i คือ ค่าข้อมูลด้านเข้าที่โหนด i

w_i คือ ค่าถ่วงน้ำหนักที่โหนด i

Δw_{ji} คือ ค่าปรับแก้ค่าถ่วงน้ำหนักระหว่างโหนด i และ j

η คือ ค่าอัตราการเรียนรู้

α คือ ค่าโมเมนตัม

δ_j คือ ค่าผลต่างระหว่างค่าจริงกับค่าที่ได้จากการคำนวณในรูปของอนุพันธ์ ของฟังก์ชันถ่ายโอน (Transfer function) ของโหนด j

y_j คือ ค่าผลลัพธ์ของแบบจำลองที่โหนด j

$n,\ n+1$ คือ ค่าที่แสดงถึงรอบของการปรับแก้ที่ n หรือ $n+1$

2.2.3 ข้อดีและข้อจำกัดของนิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กเป็นจักรกลเรียนรู้ (Machine Learning) ชนิดหนึ่งที่นิยมนำมาใช้กันมากในศาสตร์ทางด้านปัญญาประดิษฐ์ อย่างไรก็ตามนิวรอลเน็ตเวิร์กมีทั้งข้อดีและข้อจำกัดซึ่งสรุปได้ดังต่อไปนี้

2.2.3.1 ข้อดีของนิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กเป็นแบบจำลองโครงข่ายประสาทเทียม ซึ่งทำงานเลียนแบบระบบความจำของมนุษย์ ข้อดีของนิวรอลเน็ตเวิร์กมีดังต่อไปนี้

1. ใช้ในการสร้างระบบคอมพิวเตอร์อัจฉริยะอย่างได้ผล (Intelligent Computer System)
2. มีประสิทธิภาพในการประยุกต์ที่เกี่ยวข้องกับการคำนวณและจดจำ
3. สามารถจำลองความสัมพันธ์ของฟังก์ชั่นที่ซับซ้อนได้ถูกต้องในระดับใด ๆ ที่ต้องการ โดยไม่จำเป็นต้องทราบรูปแบบความสัมพันธ์ล่วงหน้า หรือสามารถหาความสัมพันธ์ระหว่างรูปแบบของข้อมูลอินพุตและเอาต์พุตได้ โดยเรียนรู้จากข้อมูลที่มีอยู่
4. สามารถประยุกต์ใช้กับการทำนายระดับน้ำ การจำลองกระบวนการเกิดฝนและน้ำท่า การทำนายความเค็มบริเวณปากแม่น้ำ ทำนายปริมาณการใช้ไฟฟ้าของเมือง ทำนายสภาพอากาศและความเร็วลม เป็นต้น
5. มีลักษณะเป็นโมเดลกล่องดำ (Black Box Model) ซึ่งไม่ใช่ลักษณะโมเดลแบบธรรมดาทั่วไป (Conventional model) ซึ่งช่วยให้ประหยัดเวลาและค่าลงทุนมากกว่า

2.2.3.2 ข้อจำกัดของนิวรอลเน็ตเวิร์ก

ถึงแม้นิวรอลเน็ตเวิร์กจะมีความสามารถที่น่าสนใจหลายอย่างแต่อย่างไรก็ดี นิวรอลเน็ตเวิร์กก็มีข้อจำกัดคือ

1. หลักการกำหนดค่าเริ่มต้นของค่าถ่วงน้ำหนักและกระบวนการเรียนรู้เพื่อปรับค่าถ่วงน้ำหนักที่เชื่อมโยงทุกๆ หน่วยยังต้องได้รับการพัฒนา

2. ในโครงข่ายแบ็กพรอพาเกชันไม่สามารถประกันได้ว่าจะให้ค่าคำตอบที่ดีที่สุดของปัญหา (Global optima) และมีความเป็นไปได้สูงที่คำตอบจะตกอยู่ในผลลัพธ์ที่ดีที่สุดในระดับท้องถิ่น (Local optima)

3. ในโครงข่ายแบ็กพรอพาเกชันสำหรับการประยุกต์ใช้กับปัญหาหนึ่งๆ ควรจะกำหนดว่าจะมีจำนวนชั้นแฝง และจำนวนโหนดในชั้นนี้เท่าไร การแปลงข้อมูลหรือสัญญาณอินพุตควรใช้ฟังก์ชันชนิดใด กระบวนการเรียนรู้วิธีใดมีประสิทธิภาพที่สุด รูปแบบของข้อมูลใดที่ใช้ในกระบวนการเรียนรู้ได้อย่างมีประสิทธิภาพ และจะมีวิธีการเลือกฟังก์ชันการกระตุ้น (Activation function) อย่างไร

4. พื้นฐานทางทฤษฎียังมีข้อจำกัดในแง่ของความช้าในการลู่เข้าหาคำตอบสุดท้าย ซึ่งยังท้าทายต่อการวิจัยเพื่อพัฒนา

5. การเรียนรู้เกินความจำเป็น (Overtraining) อาจทำให้นิวรอลเน็ตเวิร์กมีความเฉพาะเจาะจงกับข้อมูลที่ใช้ในการเรียนรู้มากเกินไป ทำให้ไม่สามารถที่จะจำแนกรูปแบบของข้อมูลเป้าหมายในช่วงการทดสอบข้อมูลได้

2.3 งานวิจัยที่เกี่ยวข้อง

Rae and Ritter (1998) งานวิจัยนี้ได้เสนอการรู้จำการหมุนของศีรษะมนุษย์โดยใช้นิวรอลเน็ตเวิร์กชนิด Local Linear Map (LLM) ซึ่งสามารถทำการคำนวณเพื่อบ่งชี้การหมุนของศีรษะมนุษย์ โดยการเรียนรู้จากข้อมูลตัวอย่าง โดยนิวรอลเน็ตเวิร์กที่นำมาใช้ประกอบด้วยนิวรอลเน็ตเวิร์ก 3 ตัว ตัวแรกใช้ในการทำการแยกสี (color segmentation) ตัวที่สองสำหรับจับภาพบริเวณหน้าของมนุษย์ ตัวที่สามใช้ในการรู้จำการหมุนของศีรษะมนุษย์ ขั้นตอนในการรู้จำการหมุนของศีรษะมนุษย์เริ่มจากภาพต้นฉบับขนาด 200 x 200 พิกเซล ซึ่งจะถูกนำมาผ่านกระบวนการแยกสี เพื่อแยกสีผิวมนุษย์ออกมาจากภาพ หลังจากนั้นนิวรอลเน็ตเวิร์กจะทำการจับภาพขนาด 120 x120 พิกเซล ในบริเวณที่เราสนใจ (Region of Interest, ROI) ซึ่งเป็นการประมาณแบบหยาบๆ เพื่อให้การประมาณมีความถูกต้องแม่นยำมากขึ้น นิวรอลเน็ตเวิร์กจะถูก

ฝึกสอนเพื่อให้สามารถสร้างวงรีล้อมรอบในบริเวณส่วนที่เป็นใบหน้าของมนุษย์ได้ ในการทดลองนำภาพจำนวน 100 ภาพ สำหรับใช้ในการฝึกสอนนิวรอลเน็ตเวิร์กให้เกิดการรู้จำ และอีกจำนวน 100 ภาพ สำหรับเป็นชุดทดสอบ โดยรูปภาพที่นำมาใช้ในการฝึกสอนและและใช้เป็นชุดทดสอบจะใช้รูปจากคนคนเดียวกันแต่แตกต่างกันที่องศาการหมุน ระยะห่างระหว่างกล้องและคน จากผลการทดลองพบว่าในกรณีการหมุนศีรษะในแนวราบ มุมเอียงเท่ากับ 0 องศา ให้ค่าความถูกต้องเฉลี่ยเท่ากับ ±9 องศา ส่วนในกรณีการหมุนศีรษะในแนวราบ ศีรษะเงยเท่ากับ 30 องศา ให้ค่าความถูกต้องเฉลี่ยเท่ากับ ±7 องศา สำหรับปัญหาที่พบในการทดลองคือระบบไม่สามารถรู้จำใบหน้าได้ถ้าศีรษะไม่อยู่ในตำแหน่งที่ถูกต้องในบริเวณที่กำหนด

Liang Zhao, Gopal Pingali และ Ingrid Carlbom (2002) งานวิจัยนี้ได้เสนอระบบการคาดคะเนองศาการหมุนของศีรษะมนุษย์ โดยทำการฝึกสอนนิวรอลเน็ตเวิร์กจำนวนสองตัวเพื่อให้สามารถหาฟังก์ชันในการจับคู่ภาพการหมุนศีรษะที่ใกล้เคียงได้ งานวิจัยนี้ได้ทำการทดลองหาผลของการหมุนศีรษะทีละ 10 องศา ที่ระยะห่าง 3 ถึง 10 ฟุตจากกล้อง นิวรอลเน็ตเวิร์กทั้งสองตัวที่นำมาใช้ในการทดลองมีสถาปัตยกรรมแบบ Multi-layered perceptron (MLP) มีจำนวนชั้นแฝงเท่ากับ1 ชั้น จำนวนโหนดในชั้นอินพุตมีจำนวน 48 x 48 โหนด ซึ่งทำการรับภาพส่วนศีรษะมนุษย์ซึ่งเป็นอินพุตของนิวรอลเน็ตเวิร์ก จากการทดลองปรับเปลี่ยนจำนวนโหนดในชั้นแฝงพบว่าในกรณีการหมุนศีรษะในแนวราบ จำนวนโหนดในชั้นแฝงเท่ากับ 5 โหนดให้ผลความถูกต้องในการบอกองศาดีที่สุด ส่วนในกรณีการหมุนศีรษะในแนวราบ แต่มีการเอียงของศีรษะพบว่าจำนวนโหนดในชั้นแฝงเท่ากับ 6 โหนดให้ผลความถูกต้องในการบอกองศาดีที่สุด สำหรับจำนวนโหนดในชั้นเอาท์พุตมีจำนวนโหนดเท่ากับ 19 โหนด ซึ่งเป็นค่าองศาในการหมุนทีละ 10 องศา ตั้งแต่ 0 องศาจนถึง 180 องศา ขั้นตอนการประมวลผลภาพอินพุตเริ่มจากภาพสีจะถูกแปลงให้เป็นภาพ grayscale และใช้กระบวนการ histogram normalization ในการลดผลกระทบเรื่องแสงที่ไม่เท่ากันของภาพ จากนั้นทำการฝึกสอนนิวรอลเตเวิร์กด้วยอัลกอริทึมแบ็กพรอพาเกชัน ข้อมูลที่นำมาใช้ในการทดลองทำการเก็บภาพคนแต่ละคนที่มีลักษณะท่าทางการหมุนศีรษะที่ต่างกันไป ทั้งการหมุนศีรษะในแนวราบ และการหมุนศีรษะในแนวราบแต่มีการเอียงของศีรษะ โดยแบ่งเป็นผู้หญิง 4 คน ผู้ชาย 9 คน ภาพที่ได้จะเก็บภาพภายใต้สภาพแวดล้อมที่ใช้แสงฟลูออเรสเซนต์ (fluorescent) และภายใต้แสงอินแคนเดสเซนต์ (incandescent) นิวรอลเน็ตเวิร์กจะถูกฝึกสอนด้วยภาพที่ใช้แสงฟลูออเรสเซนต์ และทำการทดสอบกับภาพที่ใช้แสงฟลูออเรสเซนต์และภาพที่ใช้แสงอินแคนเดสเซนต์ ผลการทดลองพบว่าภาพที่ถ่ายภายใต้แสงฟลูออเรสเซนต์ให้ผลความถูกต้องในการทำนายแม่นยำกว่าโดยในกรณีการหมุนศีรษะในแนวราบและไม่มีการ

เอียงของศีรษะให้ค่าความผิดพลาดเฉลี่ย (average error) เท่ากับ 9.29 และการหมุนศีรษะในแนวราบแต่มีการเอียงของศีรษะให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 9.02