บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ในการรู้จำใบเสร็จจากภาพ จะประกอบด้วยขั้นตอนการทำงานหลักๆ 2 ส่วนได้แก่ การตรวจจับข้อความบนแบบฟอร์มใบเสร็จ (Form Detection) จะเป็นขั้นตอนของการประมวลผลเบื้องต้น (Pre-processing) และการรู้จำข้อความบนใบเสร็จ (Text Recognition) ดังนั้นทฤษฎีที่เกี่ยวข้องที่จะกล่าวถึงในบทนี้ ได้แก่ เทคนิคการประมวลผลเบื้องต้นกับภาพ และการรู้จำข้อความบนภาพเอกสาร

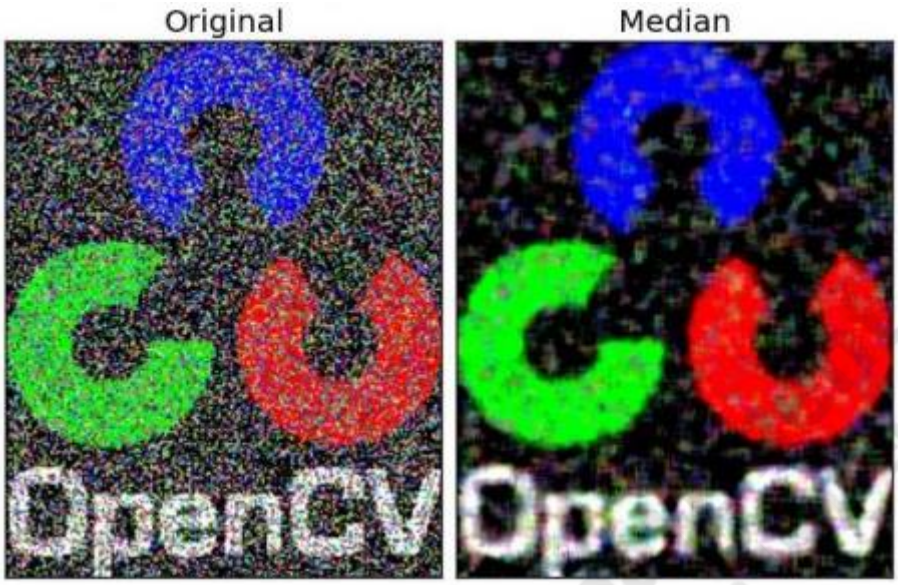
2.1.1 การประมวลผลเบื้องต้น

เป็นการนำภาพมาประมวลผลเพื่อให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพและปริมาณโดยมีขั้นตอนที่สำคัญ ได้แก่ การแปลงภาพระดับเทา การกำจัดสัญญาณรบกวนบนภาพ การแปลงภาพสองระดับ การจำแนกวัตถุ และการตัดบรรทัดข้อความ

สำหรับการตรวจจับข้อความบนใบเสร็จส่วนใหญ่จะมีสัญญารบกวนเกิดขึ้นที่ทำให้ภาพอาจไม่ชัด หรือมีริ้วรอยต่างๆ เกิดขึ้นได้ ดังนั้นเพื่อให้การรู้จำตัวข้อความและการสกัดข้อมูลทำงานได้ดียิ่งขึ้นนั้น จึงต้องมีการประมวลผลใบเสร็จเบื้องต้นก่อน โดยวิธีการประมวลผลเบื้องต้นที่ใช้ในการวิจัยนี้มีดังนี้

2.1.1.1 การปรับปรุงคุณภาพ (Image Enhancement)

เป็นการปรับปรุงคุณภาพของภาพให้เหมาะสมกับการประมวลผลต่างๆ ซึ่งภาพที่รับเข้ามาอาจมีสัญญาณรบกวนเกิดขึ้น โดยเทคนิคที่ใช้ในการปรับปรุงคุณภาพของภาพให้มีประสิทธิภาพที่ดี คือ Median Blur [3] ซึ่งวิธีนี้เป็นการกรองภาพด้วย Mask โดยจะนำเอาความเข้มแสงของจุดที่ตรงกันในภาพมาเรียงลำดับจากค่าน้อยไปหาค่ามาก แล้วหาค่ากลางนำไปแทนที่พิกเซลตรงกลางที่ตำแหน่ง Mask โดยจะทำแบบนี้และขยับ Mask ไปเรื่อยๆ จนครบทุกพิกเซล วิธีการนี้จะต้องใช้การเรียงลำดับซึ่งเป็นกระบวนการที่ใช้เวลาในการคำนวณสูง แต่ข้อดีคือไม่สูญเสียความคมชัด



ภาพประกอบที่ 2.1 ตัวอย่างภาพก่อนและหลังการปรับปรุงคุณภาพด้วย Median Blur (ที่มา: https://docs.opencv.org/)

2.1.1.2 การแปลงภาพสีเป็นภาพระดับเทา (Convert RGB to grayscale)

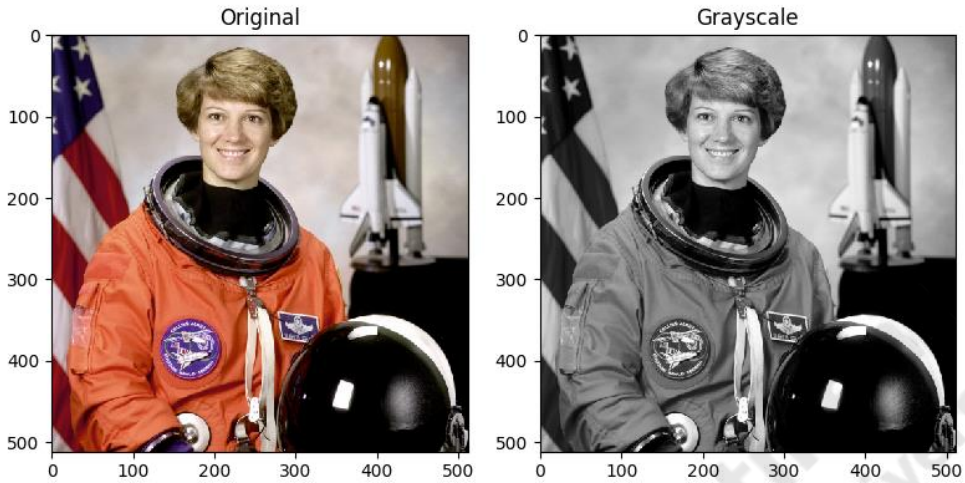
การแปลงภาพสี RGB เป็นภาพระดับสีเทา (Grayscale) [1] เป็นกระบวนการที่ทำเพื่อให้การประมวลผลภาพ มีความรวดเร็วและง่ายขึ้น ซึ่งหากนำภาพสีมาเข้ากระบวนการทำงานจะทำให้เกิดความล่าช้าเพราะว่าภาพสีแต่ละภาพจะประกอบไปด้วยช่องสี 3 ช่อง คือ ช่องสีแดง (Red) ช่องสีเขียว (Green) และช่องสีน้ำเงิน (Blue) ฉะนั้นการที่จะเข้าถึงภาพและประมวลผลก็จะต้องเข้าถึงข้อมูลทั้งสามช่องสี แต่ภาพระดับเทานั้นจะทำการดึงค่าของช่องสีในแต่ละจุดมาคำนวณให้ได้จุดสีเทาจุดเดียวดังสมการ

$$\text{Grayscale image} = 0.3R + 0.59G + 0.11B \quad (2.1)$$

โดยที่ R คือ ช่องสีแดง

G คือ ช่องสีเขียว

B คือ ช่องสีน้ำเงิน



ภาพประกอบที่ 2.2 ตัวอย่างการแปลงภาพสีเป็นภาพระดับเทา
(ที่มา: https://scikit-image.org/)

2.1.1.3 การแปลงภาพระดับเทาเป็นภาพสองระดับ (Thresholding หรือ Binarization)

เทคนิคการรู้จำตัวอักษร (Optical Character Recognition, OCR) ส่วนใหญ่ทำงานได้ดีกับภาพขาวดำ ดังนั้นการแปลงภาพระดับเทาเป็นภาพสองระดับ ทำได้หลายวิธี ทั้งแบบ Global thresholding technique โดยการกำหนดค่า Threshold แบบคงที่ เช่นการใช้ค่าจุดกึ่งกลาง ได้แก่ Threshold = 128 หรือการหาค่า Threshold แบบอัตโนมัติ ซึ่งวิธีที่ได้รับความนิยม โดยใช้ Otsu's algorithm [3] โดยหลักการการเลือกจุดตัด (Threshold) ของ Otsu นั้นเป็นการหาค่าความแปรปรวนระหว่างกลุ่มข้อมูลสองกลุ่มที่มีค่ามากที่สุดแสดงดัง**ภาพประกอบที่ 2.4** และแสดงดังสมการ

$$g(x,y) = \begin{cases} 1 \;\; if \; g(x,y) < Th \\ 0 \;\; if \; g(x,y) \geq Th \end{cases} \tag{2.2}$$

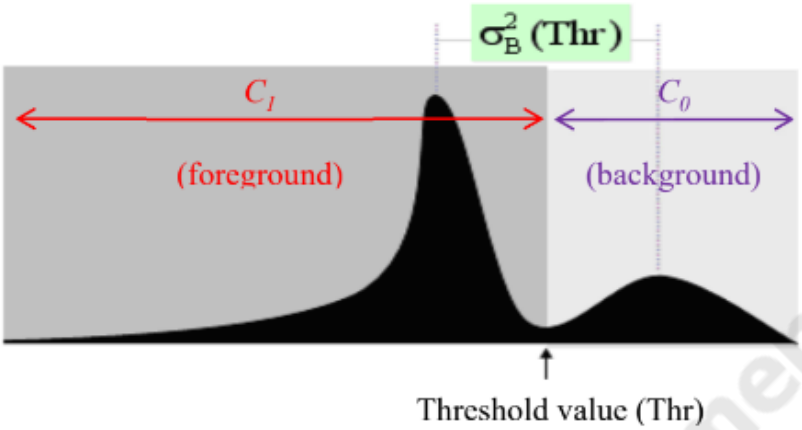
$$g(x,y) = \begin{cases} 1 \;\; if \; g(x,y) < Th \\ 0 \;\; if \; g(x,y) \geq Th \end{cases} \tag{2.2}$$

โดยที่ $g(x,y)$ คือ ค่าระดับความสว่างที่ตำแหน่ง (x,y)

Th คือค่า Threshold

1 คือค่า สีดำ ซึ่งเป็นส่วนของวัตถุ (Object)

0 คือค่า สีขาว ซึ่งเป็นส่วนของพื้นหลัง (Background)



ภาพประกอบที่ 2.3 แสดงจุดตัด (Threshold) จากฮิสโตแกรมของภาพด้วย Otsu Algorithm (ที่มา: เอกสารประกอบการสอนผู้ช่วยศาสตราจารย์ ดร.รพีพร ช่ำชอง เรื่อง การตรวจจับขอบวัตถุ https://drive.google.com/file/d/)

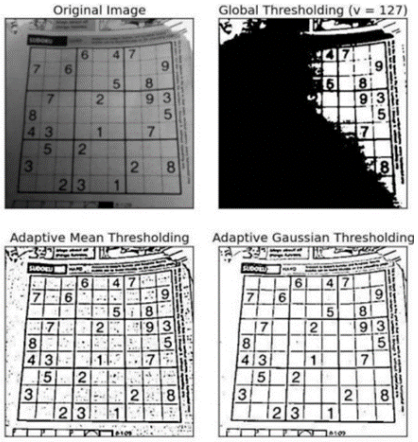
นอกจากการแปลงภาพระดับเทาด้วยวิธี Global thresholding แล้วยังมีวิธีแบบ Adaptive local thresholding ซึ่งเป็นการพิจารณาค่า Threshold ภายในพื้นที่ย่อยๆ ของจุดภาพ เพื่อปรับค่า Threshold ไปตามคุณลักษณะของพื้นที่ย่อยๆ ภายในภาพนั้นเช่น กรณีที่ภาพอาจมีแสงเงาเกิดขึ้น ไม่คงที่บนภาพ ดังนั้นการประยุกต์หาค่า Threshold แบบ Adaptive local thresholding จะช่วยแก้ปัญหาดังกล่าวได้ ซึ่งวิธีพื้นฐานที่ประยุกต์ใช้กันได้แก่ Average C-mean thresholding และ Gaussian C-mean thresholding แสดงดังสมการที่ 2.2 และผลการประมวลผลภาพดัง**ภาพประกอบที่ 2.4**

$$dst(x,y) = \begin{cases} 1 \ \ if \ f(x,y) > g(x,y) + C \\ 0 \ \ otherwise \end{cases} \tag{2.3}$$

โดยที่ C คือ ค่าคงที่

1 คือ ค่าสีขาว

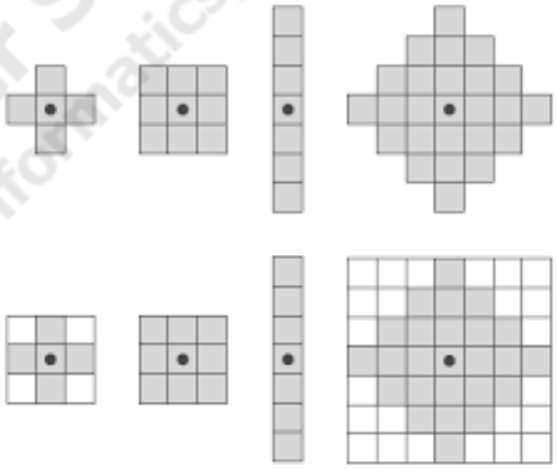
0 คือ ค่าสีดำ



ภาพประกอบที่ 2.4 ตัวอย่างการหาค่า Threshold แบบ Adaptive local thresholding
(ที่มา: http://56cjgj.blogspot.com/2017/02/thresholding.html)

2.1.1.4 การเปลี่ยนแปลงลักษณะรูปร่างของภาพ (Morphological Operation)

เป็นการดำเนินการของเซต (Set Operation) [4] ซึ่งเป็นตัวดำเนินการที่ไม่เป็นเชิงเส้น ที่ประมวลผลกับภาพโดยอาศัยพื้นฐานของรูปร่างของภาพ เป็นการอาศัยองค์ประกอบโครงสร้าง (Structuring Element) ของภาพนำเข้าเพื่อสร้างภาพผลลัพธ์ โดยที่ Structuring Element นี้จะเป็นภาพสองระดับ (0 หรือ 1) ที่มีโครงสร้างเป็นเมทริกซ์ขนาดเล็ก และมีหลักการทำงานโดยการนำ Structuring Element มาทำการ Convolution กับภาพ



ภาพประกอบที่ 2.5 ตัวอย่าง Structuring Element
(ที่มา: https://inst.eecs.berkeley.edu/)

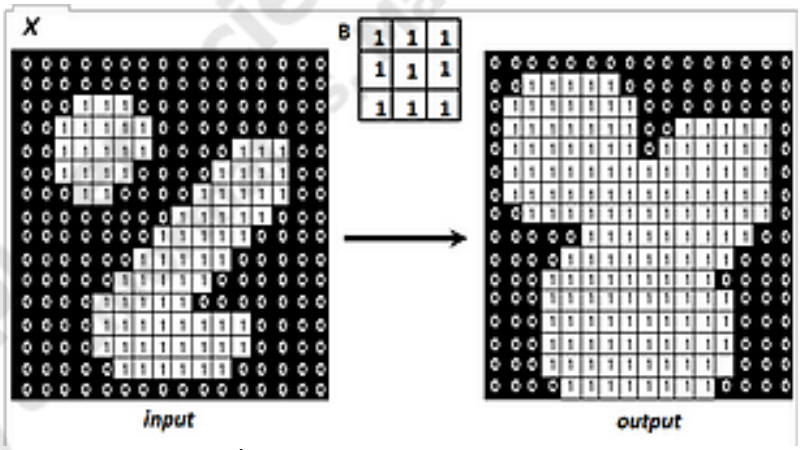
สำหรับตัวดำเนินการหลักมี 2 ประเภท ได้แก่ การทำไดเลชัน (Dilation) และการทำอิโรชัน (Erosion) ดังรายละเอียดต่อไปนี้

1) การทำไดเลชัน (Dilation)

Dilation [3] ใช้ในการเพิ่มขนาดของรูปร่างของภาพนำเข้า ดังนั้นผลลัพธ์เมื่อถูกดำเนินการจะทำให้วัตถุภายในภาพขยายขนาดใหญ่ขึ้น รวมทั้งเหมาะสมที่จะใช้ในการเชื่อมหรือขยายจุดภาพที่ขาดหายไป

ภาพประกอบที่ 2.6 ตัวอย่างก่อนทำไดเลชัน (Dilation)

ภาพประกอบที่ 2.7 ตัวอย่างหลังทำไดเลชัน (Dilation)



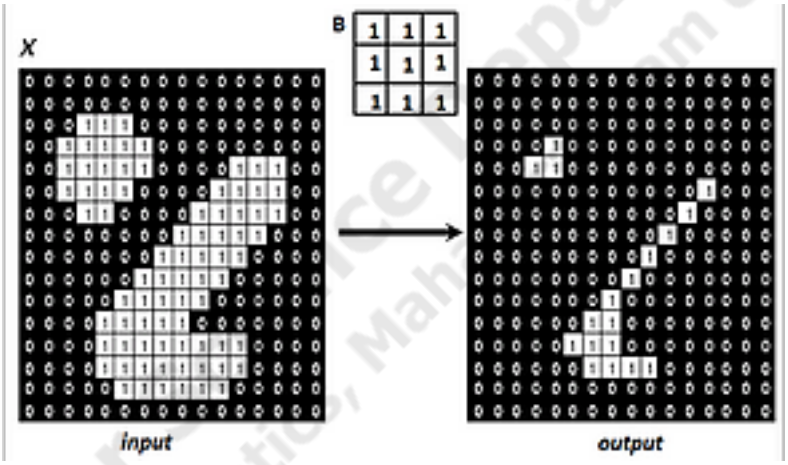
ภาพประกอบที่ 2.8 ภาพตัวอย่างการทำไดเลชัน (Dilation)
(ที่มา: https://humancominteracg1.wixsite.com/)

2) การทำอิโรชัน (Erosion)

Erosion [3] เป็นการกัดกร่อน หรือลดขนาดของวัตถุ ดังนั้นจึงสามารถประยุกต์ใช้ในการกำจัดขอบของวัตถุภายในภาพ และ กำจัดข้อมูลขนาดเล็กๆออกจากภาพได้

ภาพประกอบที่ 2.9 ตัวอย่างก่อนทำอิโรชัน (Erosion)

ภาพประกอบที่ 2.10 ตัวอย่างหลังทำอิโรชัน (Erosion)

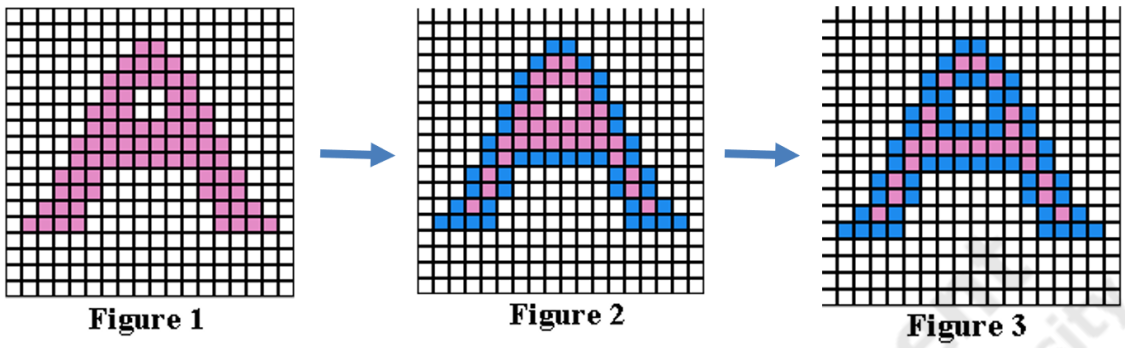


ภาพประกอบที่ 2.11 ภาพตัวอย่างการทำอิโรชัน (Erosion)
(ที่มา: https://humancominteracg1.wixsite.com/)

2.1.2 การตรวจจับข้อความด้วยการ Contour

การ Contour [4] เป็นเหมือนการหาวัตถุสีดำจากพื้นหลังสีขาว โดยขั้นตอนการ Contour นั้น จะหาพิกเซลสีดำและประกาศเป็น "จุดเริ่มต้น" ของพิกเซล ตำแหน่งของพิกเซล "เริ่มต้น" สามารถทำได้หลายวิธี ซึ่งหนึ่งในนั้นจะทำโดยเริ่มต้นที่มุมซ้ายด้านล่างของตาราง การสแกนพิกเซลจะสแกนจากด้านล่างขึ้นไป จากคอลัมน์ซ้ายสุดไปคอลัมน์ขวาสุดและจะดำเนินการต่อไปเรื่อยๆ จนกว่าจะพบพิกเซลสีดำ เมื่อพบพิกเซลสีดำก็จะกำหนดจุดที่พบปัจจุบันเป็นจุด "เริ่มต้น" ทั้งนี้ทั้งนั้นเราสามารถเลือกพิกเซลเริ่มต้นได้ตามความพึงพอใจ โดยจะมีข้อจำกัดในการเลือกพิกเซลเริ่มต้น ดังต่อไปนี้

ข้อจำกัดที่สำคัญเกี่ยวกับทิศทางการเลือกพิกเซล "เริ่มต้น" สามารถเลือกจุดเริ่มต้นที่เป็นพิกเซลสีดำที่ตำแหน่งใดก็ได้ แต่มีข้อจำกัดอยู่ว่าในกรณีที่เลือกพิกเซลเพื่อกำหนดให้เป็นจุดเริ่มต้น พิกเซลที่ใกล้เคียงอยู่ทางด้านซ้ายของจุดเริ่มต้นต้องไม่ใช่พิกเซลสีดำ แต่ต้องเป็นพิกเซลสีขาว



ภาพประกอบที่ 2.12 การ Contour รูปภาพ

ภาพประกอบที่ 2.13 ตัวอย่างก่อนและหลังการทำ Contour

2.1.3 การรู้จำตัวอักษร (Optical Character Recognition หรือ OCR)

OCR ย่อมาจาก Optical Character Recognition (OCR) [5] ซึ่งเป็นกระบวณการของการแปลงสื่อสิ่งพิมพ์ เช่น กระดาษ นิตยสาร สัญญา หรือข้อมูลอะไรก็ตามที่อยู่ในรูปของเอกสารกระดาษให้กลายเป็นข้อความให้มีความฉลาดมากขึ้นกว่าการเป็นข้อความธรรมดา หรือสามารถบันทึกไปเป็นไฟล์ประมวลผลคำที่สามารถแก้ไขได้ง่ายและบันทึกเก็บไว้ได้ เทคโนโลยี OCR ช่วยเพิ่มประสิทธิภาพอย่างมากในการจัดเก็บข้อมูล แบ่งปันข้อมูลและแก้ไขข้อมูล โครงสร้างของระบบ OCR ประกอบไปด้วยขั้นตอนการทำงานหลัก 2 ขั้นตอน ได้แก่

1.การประมวลผลขั้นต้น (Pre-process) เช่น การปรับแต่งข้อมูล (Normalization) การกรองข้อมูลแทรกซ้อน (Noise Filtering) การตรวจจับวัตถุ (Object Detection) เป็นต้น ซึ่งวิธีการประมวลผลขั้นต้นได้กล่าวมาแล้วในหัวข้อ 2.1.1 และ 2.1.2

2. การรู้จำตัวอักษร (Character Recognition) เช่น วิธีทางโครงข่ายประสาทเทียม และการเรียนรู้เชิงลึก เป็นต้น

ปัจจุบันได้มีเครื่องมือที่ได้รับการพัฒนาทางด้าน OCR มาใช้กันอย่างแพร่หลาย รองรับหลากหลายภาษา เช่น Tesseract ดังจะได้กล่าวต่อไปนี้

2.1.3.1 Tessreact

Tesseract [5] เป็นซอฟต์แวร์และไลบรารีที่ใช้ในการแปลงภาพข้อความที่มนุษย์เข้าใจให้ไปเป็นข้อความที่คอมพิวเตอร์เข้าใจ หรือเรียกอีกอย่างหนึ่งว่า OCR (Optical Character

Recognition) ซึ่งเป็น Engine ใช้สำหรับรู้จำอักขระที่ถูกพัฒนาโดยบริษัท HP ระหว่างปี 1984-1985 เริ่มต้นจากโปรเจควิจัยในระดับปริญญาเอกในห้องปฏิบัติการณ์ของ HP โดยพัฒนาเพื่อนำไปทำเป็นเครื่องแสกนเนอร์ ซึ่งต่อมาในปี 2005 HP ได้เปิด Open Source โดยมี Google เป็นผู้สนับสนุน Tesseract ถือเป็นเครื่องมือ OCR ที่มีความแม่นยำสูงอีกชนิดหนึ่ง

สาเหตุที่ Tesseract ได้รับความนิยม เพราะเป็นซอฟต์แวร์เสรี (Free software) และมีประสิทธิภาพดี โดยการเรียกใช้งานผ่านทาง Command line ก็ได้ หรือนำไปเชื่อมต่อกับ API ของงานที่ทำได้ ซึ่งภาพที่ใช้เป็น Input ให้กับ Tesseract ต้องเป็นภาพที่มีการปรับแต่งมาให้เหมาะกับการอ่านข้อความคือหมุนมาค่อยข้างตรง มีการปรับแสงและสีให้อ่านได้ง่าย พื้นหลังสีขาวหรือสีอ่อน ตัวอักษรสีดำ

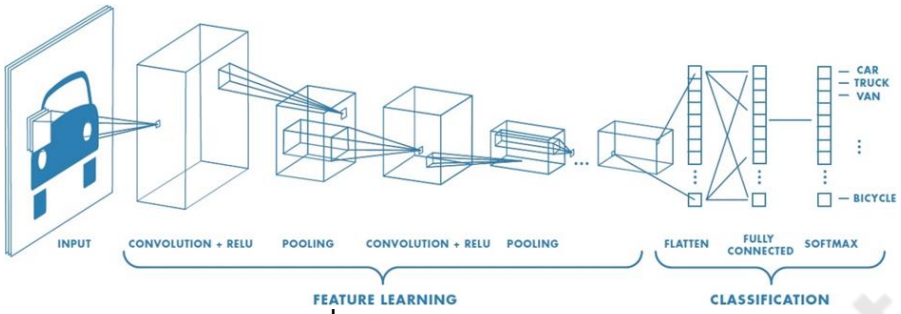
Tesseract รุ่นที่รองรับภาษาไทยตั้งแต่รุ่นที่ 3 ขึ้นไป โดยในรุ่นที่ 4 สามารถใช้โมเดล Deep Learning แบบ LSTM ได้

2.1.4 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก [6] เป็นวิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาทเทียม (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจำแนกข้อมูล (Classify the Data) เพื่อที่จะทำให้ Neural Network นั้นสามารถคิดและประมวลผลซับซ้อนได้เหมือนสมองมนุษย์ ชั้นที่เป็น Hidden Layer จึงต้องมีหลายๆ ชั้น ส่งข้อมูลประมวลผลต่อๆ กันไป จึงทำให้คอมพิวเตอร์สามารถทำนายหรือตรวจจับรูปภาพได้เสมือนสมองของมนุษย์ ซึ่งการเรียนรู้เชิงลึกที่จะกล่าวถึงในที่นี้ได้แก่ Convolutional Neural Network และ Recurrent Neural Network มีรายละเอียด ดังนี้

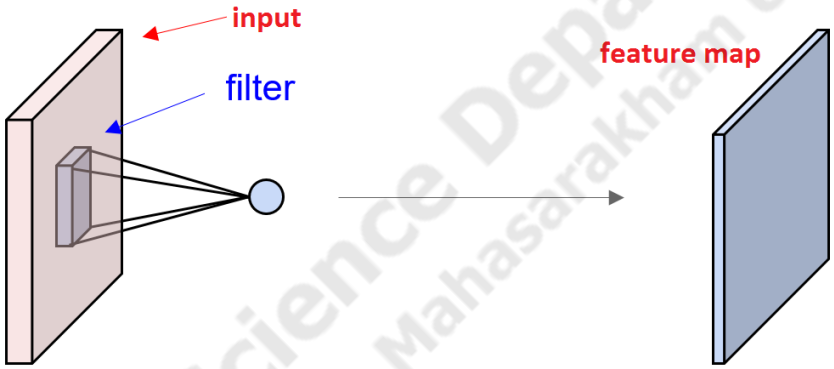
2.1.4.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network, CNN)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน [3] เป็นระบบที่มีลักษณะคล้ายกับระบบโครงข่ายประสาทเทียมพื้นฐานแต่โครงข่ายประสาทเทียมแบบคอนโวลูชัน สามารถประมวลผลกับรูปภาพได้มีประสิทธิผลมากกว่าเนื่องจากความสามารถในการสกัดเอา Feature หรือลักษณะเด่นต่างๆ ออกมาจากรูปภาพเพื่อใช้ในการเป็น Input ให้กับขั้นตอนการจำแนก (Classification) ต่อไป โดยภายในระบบ CNN จะประกอบไปด้วย 3 Layer ได้แก่ Convolutional layer, Pooling layer และ Fully connected layer ดังนี้



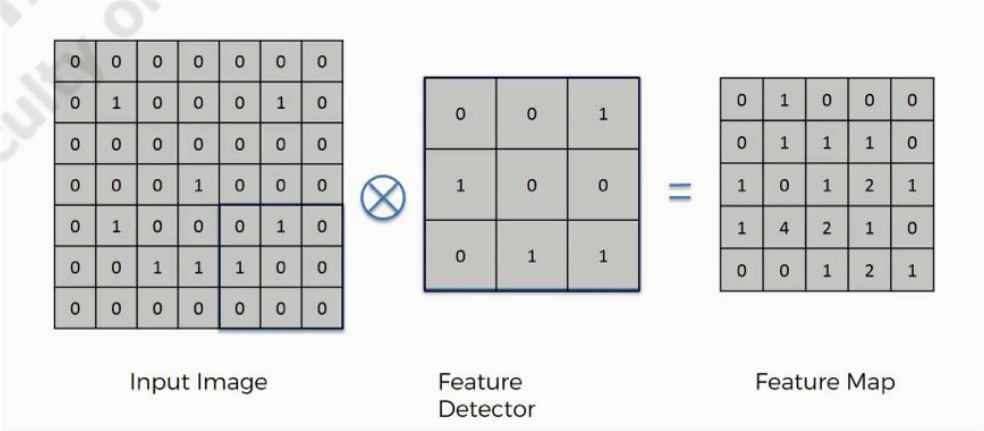
ภาพประกอบที่ 2.14 กระบวนการทำงานของ CNN
(ที่มา: https://www.thaiprogrammer.org/)

ชั้นคอนโวลูชัน (Convolution Layer) เป็น Layer ที่ทำหน้าที่ในการสกัด Feature ออกมาจากรูปภาพที่ใช้ เป็น Input เพื่อนำมาสร้างเป็นผังคุณลักษณะ (Feature map)



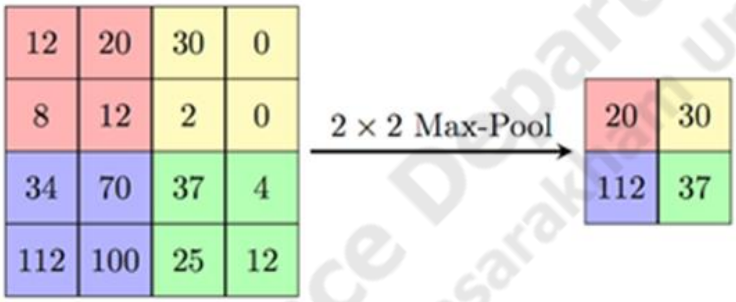
ภาพประกอบที่ 2.15 ผังคุณลักษณะ (Feature map)
(ที่มา: https://www.quora.com/)

โดยในการสกัด Feature นั้นทำโดยการแบ่งรูปภาพ ออกเป็นส่วนๆ แต่ละส่วนจะถูกเรียกว่า Cell จากนั้นนำแต่ละ Cell มาผ่านกระบวนการ Convolution กับ Filter เพื่อให้ได้ Feature ออกมา เช่น ขอบของรูปภาพ เป็นต้น

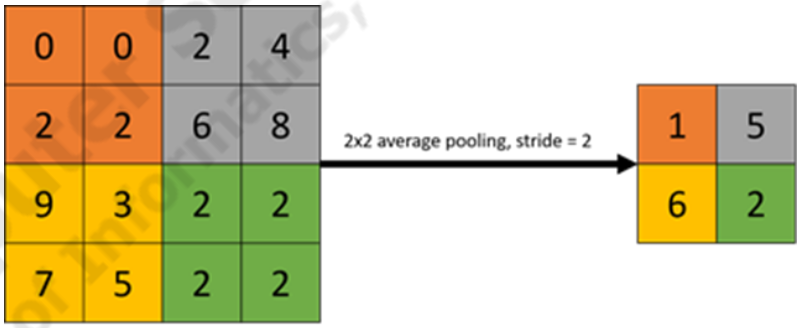


ภาพประกอบที่ 2.16 แสดง filter ที่ใช้ในการสร้าง feature maps ด้วย CNN
(ที่มา: https://www.andreaperlato.com/)

ชั้นพูล (Pooling layer) เป็นชั้นที่ทำหน้าที่ในการปรับขนาดและปริมาณของข้อมูลตัวอย่าง (Sample) ให้ลดลงก่อนนำส่งเข้าสู่ Layer ถัดไปเพื่อให้สามารถวิเคราะห์และเก็บรายละเอียดของภาพได้อย่างครบถ้วนโดยที่ไม่สูญเสียข้อมูล ยิ่งไปกว่านั้นกระบวนการนี้ยังช่วยลดโอกาสเกิดเหตุการณ์ Overfitting ได้อีกด้วย ในการ Pooling นั้นจะมีกระบวนการที่คล้ายกับกระบวนการสร้าง Feature maps คือการแบ่ง Feature map ออกเป็น Cell จากนั้นนำ Cell ไปผ่านกระบวนการ Pooling โดยการทำ Convolution กับ Filter อีกครั้งเหมือนกับ Convolutional layer การ Pooling ที่นิยมกระทำในปัจจุบันมีอยู่สองรูปแบบคือ Max pooling, Average pooling โดยกระบวนการทั้งสองสามารถอธิบายได้ดังภาพประกอบที่ 2.17 และ ภาพประกอบที่ 2.18

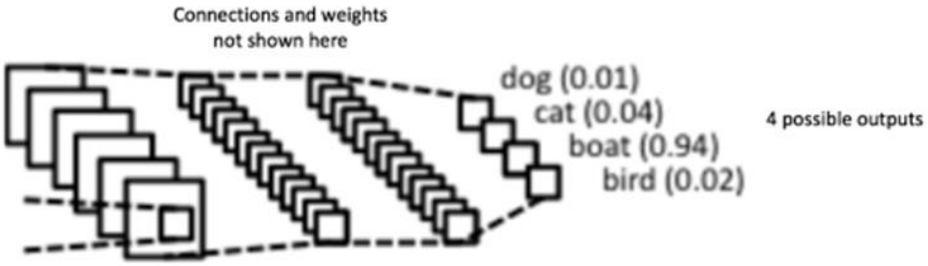


ภาพประกอบที่ 2.17 แสดงการทำงานของ Max Pooling
(ที่มา: https://paperswithcode.com/method/max-pooling)



ภาพประกอบที่ 2.18 แสดงการทำงานของ Average pooling
(ที่มา: https://www.kaggle.com/)

ชั้นฟูลลีคอนเนก (Fully connected layer) เป็น Layer ที่ประกอบด้วยระบบ Multilayer perceptron (MLP) ในการประมวลผลข้อมูลที่ได้มาจาก 2 layer ก่อนหน้านี้เพื่อสังเคราะห์และทำการแยกแยะรูปภาพออกเป็นหมวดหมู่



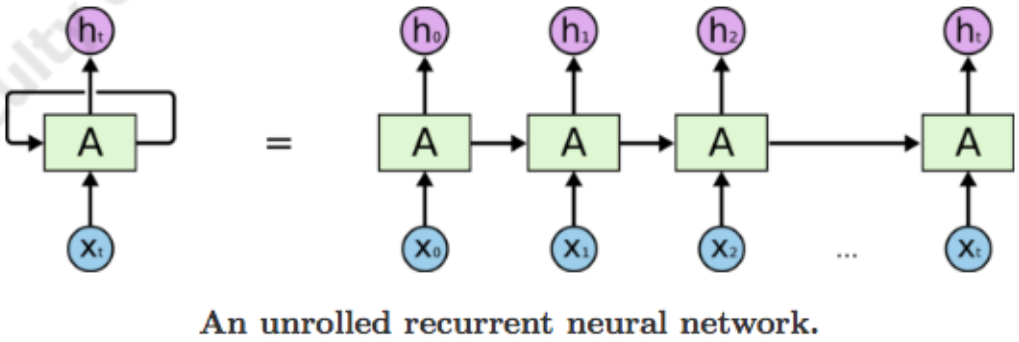
ภาพประกอบที่ 2.19 แสดงการทำงานของ Fully connected layer
(ที่มา: https://ujjwalkarn.me/)

สำหรับ CNN สามารถใช้ได้ทั้งในการสกัดคุณลักษณะของวัตถุ (Feature Extraction) และรู้จำหรือจำแนกประเภท (Recognition or Classification) หากต้องการเพียงการสกัดคุณลักษณะของวัตถุ ไม่ต้องทำขั้นตอน Fully connected layer ก็จะสามารถนำคุณลักษณะของวัตถุดังกล่าวไปประมวลผลต่อด้วยวิธีการรู้จำหรือจำแนกประเภทด้วยวิธีอื่นๆ ต่อไปได้

2.1.4.2 Recurrent Neural Network (RNN)

RNN [8] เป็นวิธีการที่ถูกนำมาใช้ในการวิจัยเกี่ยวกับการรู้จำเสียง (Speech Recognition) การประมวลผลภาษาธรรมชาติ (Natural Language Processing) การรู้จำลายมือเขียน (Handwritten Recognition) และการประมวลผลกับข้อมูลที่มีลำดับ (Sequence) การทำงานของ RNN คือการเอาผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลขาเข้าอีกครั้ง ซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลเสียง ข้อความ หรือแม้แต่รูปภาพเองก็ตาม

RNN ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับงานที่ข้อมูลที่มีลำดับ โดยใช้หลักการนำสถานะภายใน ของโมเดล กลับมาเป็นข้อมูลเข้าใหม่คู่กับข้อมูลเข้าแบบปกติ เรียกว่า สถานะซ่อน (Hidden State) หรือสถานะภายใน (Internal State) ช่วยให้โมเดลรู้จำรูปแบบ (Pattern) ของลำดับข้อมูลเข้า (Input Sequence)



An unrolled recurrent neural network.

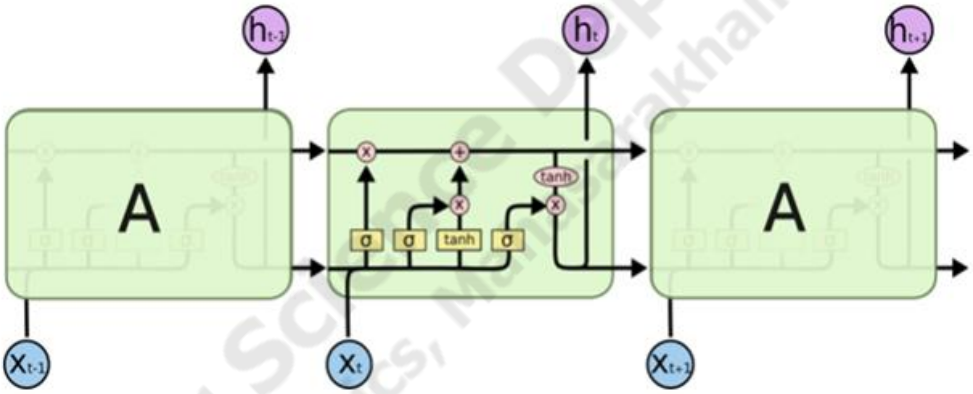
ภาพประกอบที่ 2.20 กระบวนการทำงานของ RNN
(ที่มา: https://ichi.pro/th/)

ในแต่ละโหดของ RNN จะมีข้อมูลเข้าสองอย่าง ได้แก่ ข้อมูลเข้า และผลลัพธ์ที่ได้จากการคำนวณในโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทาง คือ ผลลัพธ์ที่ออก ณ โหนดนั้นๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าในโหนดถัดไป

สำหรับเทคนิค RNN นี้ได้มีการพัฒนาต่อเป็นโครงข่าย Long Short-Term Memory และ Gated Recurrent Units ดังจะได้กล่าวในหัวข้อต่อไป

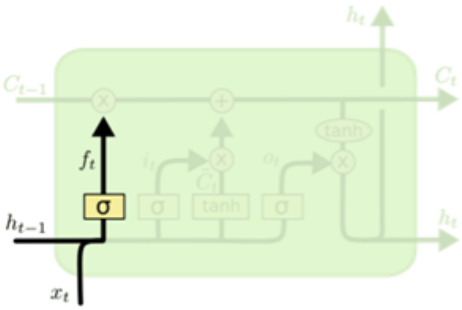
2.1.4.3 Long Short-Term Memory (LSTM)

LSTM [10] เป็นโครงข่ายประสาทเทียมที่เกิดซ้ำซึ่งแก้ปัญหาการไล่ระดับสีที่หายไปของ RNN หลักการ LSTM คือ การเก็บค่าสถานะของแต่ละโหดเอาไว้ เผื่อว่าตอนย้อนกลับมาจะได้รู้ว่าค่านี้แท้จริงแล้วเป็นค่าอะไรมาก่อน ลักษณะที่โดดเด่นของ LSTM คือ การที่สามารถเลือกได้ว่าข้อมูลไหนควรที่จะจดจำ ข้อมูลไหนควรกำจัดทิ้ง ใน LSTM จะประกอบด้วยฟังก์ชัน 3 ฟังก์ชัน ซึ่งมีรายละเอียดดังนี้



ภาพประกอบที่ 2.21 กระบวนการทำงานของ LSTM
(ที่มา: https://ichi.pro/th/)

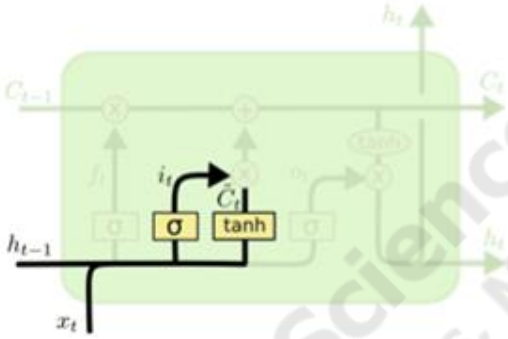
Forget Gate เป็นฟังก์ชันที่จะตัดสินใจว่าข้อมูลที่รับเข้ามาจากโหนดนั้นๆ จะเก็บไว้หรือไม่ ดังภาพประกอบที่ 2.22 ผลลัพธ์ของฟังก์ชันนี้จะอยู่ระหว่าง [0,1] ซึ่งค่า 0 หมายถึง ไม่มีข้อมูลใดที่จะสามารถไหลผ่าน Gate ไปได้เลย ในขณะที่ 1 หมายถึง ปล่อยให้ข้อมูลที่เข้ามาไหลผ่าน Gate ไปได้ทั้งหมด



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

ภาพประกอบที่ 2.22 Forget Gate LSTM
(ที่มา: https://ichi.pro/th/)

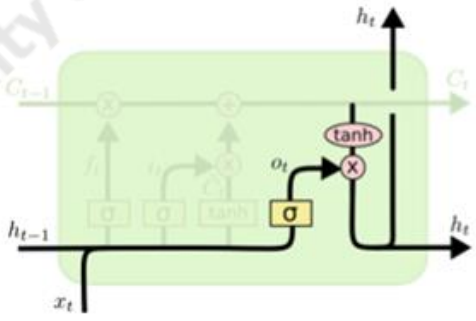
Input Gate เป็นฟังก์ชันที่ใช้รับข้อมูลที่เข้ามาใหม่เพื่อให้บันทึกลงไปในแต่ละโหนด เพื่อแก้ไขหน่วยความจำ ฟังก์ชัน Sigmoid จะตัดสินใจว่าจะให้ค่าใดผ่าน 0,1 และฟังก์ชัน tanh จะให้น้ำหนักกับค่าที่ส่งผ่านมาเพื่อกำหนดระดับความสำคัญตั้งแต่ -1 ถึง 1



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

ภาพประกอบที่ 2.23 Input Gate LSTM
(ที่มา: https://ichi.pro/th/)

Output Gate เป็นเช่นเดียวกับ Gate อื่นๆ คือการนำข้อมูลจากโหนดที่แล้วที่เข้ามารวมกับข้อมูลขาเข้าในโหนดนั้นๆ ผ่านฟังก์ชัน ใช้ในการตัดสินใจเพื่อหาค่า Output โดยหลังการคำนวณค่าฟังก์ชัน Sigmoid จะนำไปประมวลผลร่วมกับฟังก์ชัน tanh ก่อนจนได้ผลลัพธ์ออกมา



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

ภาพประกอบที่ 2.24 Output Gate LSTM
(ที่มา: https://ichi.pro/th/)

2.1.4.4 Gated Recurrent Units (GRU)

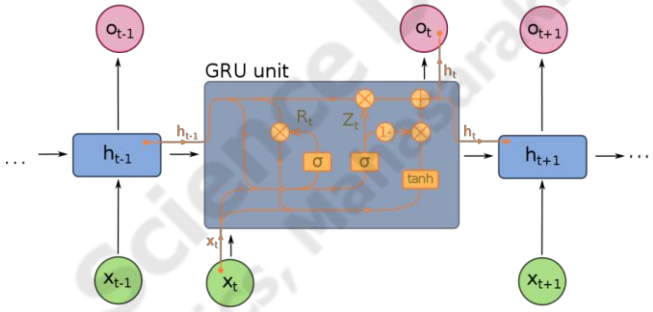
Gated Recurrent Units [6] สามารถเรียนรู้การพึ่งพาระยะยาว โดยมีกลไกภายในที่เรียกว่า Gate เพื่อควบคุมการไหลของข้อมูลเช่นการจดจำบริบทในช่วงเวลาต่างๆ ติดตามว่าข้อมูลใดในอดีตสามารถเก็บสิ่งที่ลืมได้ ซึ่งพัฒนาต่อมาจาก LSTM เพื่อลดขั้นตอนการทำงานภายใน Gate และช่วยให้การประมวลผลเร็วขึ้น โดย GRU จะแบ่งออกเป็น 2 Gate คือ Update gate และ Reset gate ซึ่งมีรายละเอียดดังนี้

1) Update gate

ทำหน้าที่พิจารณาว่าควรจะเก็บข้อมูล State ก่อนหน้าไว้มากน้อยเพียงใด และจะเพิ่มข้อมูลอะไรใหม่การสรุปความที่มีการนำมาเรียบเรียงใหม่ (Abstraction-Based Summarization

2) Reset gate

ทำหน้าที่คำนวณว่าจะนำข้อมูลจาก State ก่อนหน้ามาพิจารณาร่วมกับข้อมูล Input ปัจจุบันมากน้อยเพียงใด



ภาพประกอบที่ 2.25 กระบวนการทำงานของ GRU
(ที่มา: https://www.bualabs.com/)

2.2 งานวิจัยที่เกี่ยวข้อง/ระบบงานที่เกี่ยวข้อง (Related Word)

ผลงานวิจัยของ Sudharshan Chandra Babu [4] เรื่อง Automating Receipt Digitization with OCR and Deep Learning ได้นำเสนอวิธีการใช้ CUTIE (Convolutional Universal Text Information Extractor) เป็นเทคนิคการเรียนรู้ที่จะทำความเข้าใจเอกสารด้วยการใช้ Convolutional Neural Networks (CNN) ในการสกัดข้อมูลข้อความจากเอกสาร CUTIE Model จะเสนอเป็น CUTIE-A ซึ่งจะนำเสนอเป็น CNN ที่มีความจุสูง ใช้กับภาพที่มีความละเอียดสูง และ CUTIE-B จะเป็น CNN ที่มีหลายมุมมอง โดยอาศัยโมดูล Atrous Spatial Pyramid Pooling (ASPP) เพื่อสร้างภาพหลายขนาด โดยทั้งสองวิธีเป็นการสร้างการเข้ารหัสพร้อมกับการทำ Word embedding ในขั้นตอนแรกของเครือข่าย จากผลการทดลองโดยใช้ใบเสร็จจาก 3 ร้าน ได้แก่ แท็กซี่ ME และโรงแรม สรุปได้ว่า CUTIE-B ให้ผลค่าความแม่นยำ (AP) ได้ดีกว่า CUTIE-A โดยในใบเสร็จแท็กซี่จะอยู่ที่ 94.0

ใบเสร็จของ ME จะอยู่ที่ 81.5 และ ใบเสร็จโรงแรมจะอยู่ที่ 74.6 ส่วนค่าความแม่นยำที่วัดโดย softAP CUTIE-A จะให้ผลค่าแม่นยำได้ดีกว่า CUTIE-B ในใบเสร็จของโรงแรมและใบเสร็จของ ME

งานวิจัยเรื่อง GCNs for VRDs (Graph Convolution for Multimodal Information Extraction from Visually Rich Documents) [4] เป็น Convolution-based model เพื่อรวมข้อมูลที่เป็นข้อความและข้อมูลที่อยู่บนภาพในเอกสาร โดยกราฟจะถูกนำเข้าเพื่อสอนและสรุปเนื้อหาของข้อความที่จำแนกได้ในเอกสารโดยรวมกับการนำเข้าของผลเฉลย สำหรับกราฟ Convolution จะสรุปเนื้อหาของข้อความที่ถูกจำแนกมาจากเอกสาร โดยรวมข้อมูลนำเข้ากับข้อความผลเฉลย โดยใช้โมเดล BiLSTM-CRF ซึ่งโมเดลนี้จะรวมข้อมูลที่เป็นในรูปภาพกับข้อความเข้าด้วยกัน ผลจากการทดลองโดยใช้ตัวอย่างใบเสร็จ 2 ชนิดคือ ใบกำกับภาษีมูลค่าเพิ่ม และใบเสร็จการซื้อขายระหว่างประเทศ และใช้ค่า F1 score ในการวัดประสิทธิภาพ ซึ่งผลคะแนน F1 score ของโมเดล BiLSTM ในใบกำกับภาษีมูลค่าเพิ่มมีค่าเท่ากับ 0.873 และในใบเสร็จรับเงินการค้าระหว่างประเทศมีค่าเท่ากับ 0.836

จากงานวิจัยที่มีการประยุกต์ใช้ Faster-RCNN ร่วมกับ AED (Attention-based Encoder-Decoder) [4] เป็นวิธีการเรียนรู้เชิงลึกสำหรับการรู้จำใบเสร็จรับเงินอีกประเภทหนึ่งที่สแกน ระบบรู้จำมีสองโมดูลหลัก การตรวจจับข้อความโดยใช้ Connectionist Text Proposal Network และการรู้จำข้อความโดยใช้ Attention-based Encoder-Decoder โดยชุดข้อมูลนำมาจาก SROIE 2019 โดยแบ่งข้อมูลเป็นชุดการฝึกสอน ชุดvalidation และชุด testing ซึ่ง 80% (500 ใบเสร็จ) ใช้สำหรับการฝึกสอน 10%(63 ใบเสร็จ) เป็นชุดvalidation ที่เหลือเป็นชุดทดสอบ (63 ใบเสร็จ) จากผลการทดสอบการตรวจจับข้อความจาก 3 โมเดล พบว่าในการประเมินประสิทธิภาพด้วย Recall โมเดล pre – processing + CTPN ให้ค่า Recall ที่ดีที่สุดในทั้งสามโมเดล ส่วนการประเมินประสิทธิภาพด้วย Precision โมเดล pre – processing + CTPN + OCR verification ดีที่สุด และลำดับสุดท้ายประเมินด้วย F1 Score โมเดล pre – processing + CTPN ให้ค่า F1 Score สูงสุด

โครงงานปริญญานิพนธ์ของ นางสาวกมลทิพย์ เทศทอง เรื่อง การประมวลผลลายมือเขียนเป็นตัวพิมพ์อัตโนมัติ เวอร์ชั่น 2 (Automatic Handwritten Recognition; Auto HWR v.2) ได้ทำการแปลงรูปภาพลายมือเขียนภาษาไทยออกมาเป็นตัวพิมพ์อัตโนมัติ โดยออกแบบสถาปัตยกรรมในการรู้จำตัวอักษรด้วยการเรียนรู้เชิงลึก (Deep Learning) โดยมีสถาปัตยกรรมการทำงานของ CNN ร่วมกับ GRU และ CTC จะมี CNN จำนวน 5 ชั้น และ GRU แบบ Bidirectional GRU 3 ชั้น จากนั้นได้ทำการวัดประสิทธิภาพโดยใช้ Levenshtein edit distance และใช้ชุดข้อมูลเฉพาะลายมือเขียนทั้งสิ้น 10,792 รูปภาพ ข้อมูล Generate Font จำนวนทั้งหมด 6,432 ข้อความ จากผลการทดสอบพบกว่าในการใช้ตัวถอดรหัสแบบ Beam Search Decoding ร่วมกับ LM มีค่า Character Error Rate อยู่ที่ 2.53% และการใช้ Best Path Decoding มีค่า Character Error Rate อยู่ที่ 2.5