ระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ

License plate recognition to build automatic car barrier gate model

ผู้ช่วยศาสตราจารย์ เกียรติศักดิ์ ลาภพาณิชยกุล

งานวิจัยนี้ได้รับทุนสนับสนุนจากงบประมาณกองทุนเพื่อการวิจัย

ประจำปีงบประมาณ พ.ศ. 2567

คณะบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ชื่อเรื่อง : ระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ
ผู้วิจัย : เกียรติศักดิ์ ลาภพาณิชยกุล
พ.ศ. : 2567

บทคัดย่อ

จากการวิจัยระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ ด้วย Deep Learning พบว่าในส่วนของการหาตำแหน่งของป้ายทะเบียน ระบบสามารถระบุตำแหน่งของป้ายทะเบียนได้ถูกต้องคิดเป็นร้อยละ 90 ในส่วนของป้ายทะเบียนรถที่อยู่ในลักษณะแนวตรงเท่านั้น ความผิดพลาดที่พบในการระบุตำแหน่งเนื่องจากเป็นการถ่ายรูปในลักษณะที่ไม่ได้อยู่ในแนวระดับสายตาทำให้ไม่สามารถจับป้ายทะเบียนและความคมชัด ส่งผลต่อการตรวจจับป้ายทะเบียนอย่างชัดเจน ส่วนของการแยกตัวเลขและตัวอักษร มีความถูกต้องคิดเป็นร้อยละ 98 และส่วนการรู้จำสามารถระบุตัวเลยและตัวอักษรได้ถูกต้องร้อยละ 95 โดยขบวนการวิจัยทั้งหมดใช้การประมวลผล

Title : License plate recognition to build automatic car barrier gate model
Researcher : Kreadtisak Lappanitchayakul
Year : 2024

Abstract

From the research on a vehicle license plate detection system to develop an automatic gate opening model using Deep Learning, it was found that in the aspect of locating the license plate, the system can correctly identify the position of the license plate at a rate of 90%, specifically for plates that are oriented straight. The errors encountered in locating the plates were due to images taken at angles that were not level with the eye, which made it difficult to capture the license plates. Additionally, image clarity significantly impacted the detection of the plates.

In terms of separating numbers and letters, the accuracy was found to be 98%. For recognition, the system could identify the characters correctly at a rate of 95%. The entire research process utilized data processing techniques.

กิตติกรรมประกาศ

งานวิจัยเรื่องนี้ได้รับการสนับสนุนหัวข้อการวิจัยจากคณะบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร และได้รับทุนอุดหนุนการวิจัยจากงบประมาณรายได้ ประจำปีงบประมาณ 2567 ผู้วิจัยขอขอบพระคุณไว้ ณ ที่นี้

ผู้ช่วยศาสตราจารย์เกียรติศักดิ์ ลาภพาณิชยกุล

สารบัญ

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปจจุบันจำนวนรถยนต์ในคณะบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร เพิ่มจำนวนมากขึ้น ทำให้ปัญหาการบริหารจัดการพื้นที่จอดรถในคณะบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร ต้องมีการดำเนินการด้วยเจ้าหน้าที่หน้าประตูมหาวิทยาลัยต่างๆ การที่มีระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ จะช่วยอำนายความสะดวกในการจอดรถยนต์ให้กับเจ้าหน้าที่ และอาจารย์ ที่ไม่ต้องรอให้เจ้าหน้าที่มาดันรั้วเหล็กเพื่อเปิดให้รถเข้ามาจอด ทำให้รถสามารถเข้ามายังภายในที่จอดได้อย่างอัตโนมัติ การจะอ่านทะเบียนรถให้ระบบสามารถอ่านภาพจากกล้องถ่ายภาพดิจิทัลหรือกล้องถ่ายวีดีโอแล้วส่งเข้าสู่การประมวลผลภาพ (Image Processing) ดวยคอมพิวเตอรและระบุหมายเลขของทะเบียนรถ

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 ระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ

1.3 ขอบเขตของการวิจัย

1.3.1 การวิจัยครั้งนี้ เป็นการวิจัยและพัฒนา โดยทำการพัฒนาระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ โดยผู้วิจัยใช้ระบบปัญญาประดิษฐ์กับอุปกรณ์ IoT ในการแก้ปัญหาดังกล่าว

1.3.2 ข้อมูลที่ใช้การทดลองประกอบด้วย ชุดข้อมูลตัวเลข 0-9 ตัวอักษร ก-ฮ

1.3.3 ข้อมูลชุดแผ่นป้ายทะเบียนรถ

1.4 วิธีดำเนินการวิจัย

1.4.1 ศึกษาอุปกรณ์สำหรับทำอุปกรณ์ตรวจจับทะเบียนรถยนต์

1.4.2 พัฒนาและออกแบบตัวอุปกรณ์ตรวจจับทะเบียนรถยนต์

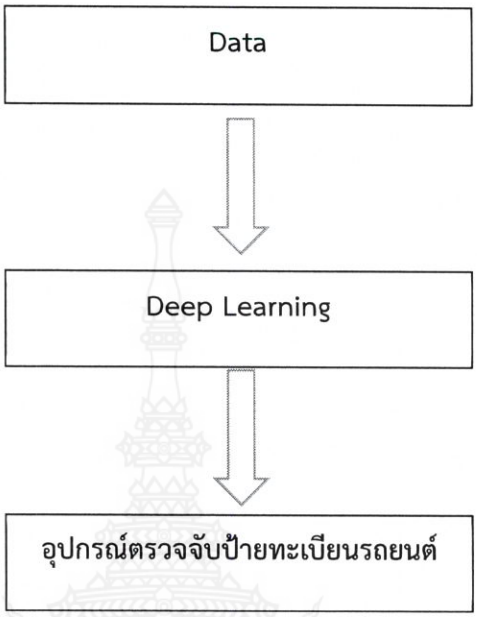
1.4.3 ศึกษาระบบการออกแบบไม้กั้นประตูรถยนต์อัตโนมัติ

1.4.4 ทำการพัฒนาระบบไม้กั้นประตูรถยนต์อัตโนมัติ

1.4.5 ทดสอบการทำงานของอุปกรณ์

1.5 กรอบแนวความคิดในการวิจัย

การวิจัยครั้งนี้ ได้ศึกษา เอกสาร ตำรา และงานวิจัยที่เกี่ยวข้อง กับการวิเคราะห์และพารัมนาระบบปัญญาประดิษฐ์ กับชุดข้อมูลตัวเลข เพื่อพัฒนาระบบตรวจจับตัวเลขบนป้ายทะเบียนรถยนต์ สำหรับการพัฒนาระบบตรวจจับป้ายทะเบียนรถยนต์ใช้หลักทฤษฎีของ Deep Learning และ Machine Learning และทำการประเมินประสิทธิภาพของอุปกรณ์ที่ได้พัฒนาขึ้น



รูปที่ 1-1 กรอบแนวความคิดวิจัย

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 อุปกรณ์ตรวจจับป้ายทะเบียนรถยนต์

1.6.2 การประยุกต์ใช้ Deep Learning กับนวัตกรรมสมัยใหม่

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎี

2.1.1 ทฤษฎีการประมวลผลภาพ (Image Processing)

ภาพ

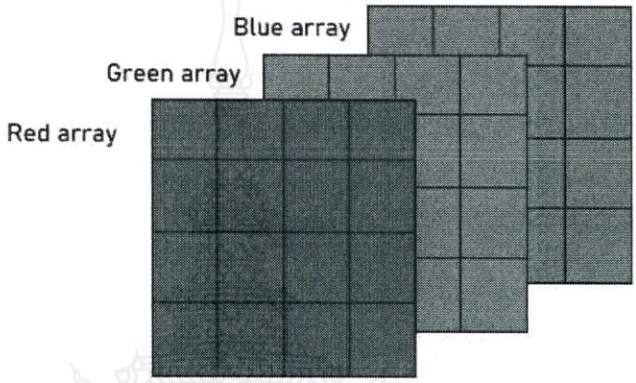
เป็นฟังก์ชันสองมิติ F (x, y)โดยที่ x และ y เป็นพิกัดเชิงพื้นที่และความกว้างของ F ที่พิ่กัดดูใดๆ (x, y) เรียกว่าความเข้มของภาพนั้น เมื่อ x, y และค่าความกว้างของ F มีจำกัด เราเรียกมันว่าภาพดิจิทัล กล่าวอีกนัยหนึ่งว่ารูปภาพสามารถถูกกำหนดโดยอาร์เรย์สองมิติที่จัดเรียงเป็นพิเศษในแถวและคอลัมน์ ภาพดิจิทัลประกอบด้วยจำนวนจำกัด ซึ่งภาพในระบบดิจิทัลจะมีข้อมูลขนาดเล็กๆ ที่เรียงกันในภาพ เรียกว่า พิกเซล ดังนั้น พิกเซลที่มีค่าส่องสว่างมาเรียงต่อๆ กัน เป็นจำนวนมาก จะทำให้เราสามารถมองเห็นภาพได้ในระบบคอมพิวเตอร์

ภาพดิจิทัล

เป็นตัวแทนของภาพเสมือนจริงเป็นชุดของตัวเลขที่สามารถจัดเก็บและจัดการโดยเป็นดิจิทัลคอมพิวเตอร์ ในการแปลภาพเป็นตัวเลขและจะถูกแบ่งออกเป็นพื้นที่เล็กๆ ที่เรียกว่า พิกเซล (องค์ประกอบภาพ) สำหรับแต่ละพิกเซลอุปกรณ์ถ่ายภาพจะบันทึกหมายเลขหรือชุดตัวเลขขขนาดเล็ก ซึ่งอธิบายคุณสมบัติบางอย่างของพิกเซลนี้เช่นความสว่าง (ความเข้มของแสง) หรือสีของภาพ ตัวเลขจะถูกจัดเรียงในอาร์เรย์ของแถวและคอลัมน์ที่สอดคล้องกับตำแหน่งแนวตั้งและแนวนอนของพิกเซลในภาพ

ภาพแบบ RGB

ภาพ RGB บางครั้งเรียกว่า ภาพสีจริง (TrueColor) ถูกเก็บไว้ใน MATLAB เป็นอาร์เรย์ข้อมูล M *N * 3 ที่กำหนดองค์ประกอบสีแดงสีเขียวและสีน้ำเงินสำหรับแต่ละพิกเซลแต่ละต่ละพิกเซล ภาพ RGB ไม่ได้ใช้จานสี สีของแต่ละพิกเซลถูกกำหนดโดยการรวมกันของความเข้มสีแดง, สีเขียวและสีน้ำเงินที่เก็บไว้ในระนาบสีแต่ละจุดที่ตำแหน่งของพิกเซล รูปแบบแบบไฟล์กราฟฟิกจัดเก็บภาพ RGB เป็นภาพ 24 บิตโดยที่องค์ประกอบสีแดงสีเขียวและสีน้ำเงินแต่ละชิ้นจะมี 8 บิต สิ่งนี้ทำให้ศักยภาพของ 16 ล้านสีมีความแม่นยำที่สามารถจำลองภาพในชีวิตจริงได้ สามารถดูได้สามภาพที่แตกต่างกัน (ภาพสเกลสีแดง ภาพสเกลสีเขียว และภาพสเกลสีฟ้า) ซ้อนกันอยู่ด้านบนของกันและกัน (GeeksforGeeks, 2562) [1]



รูปที่ 2-1 ระนาบสีแต่ละอันมีอาร์เรย์ M*N

ที่มา https://www.geeksforgeeks.org/matlab-rgb-image-representation/ [1]

ใน MATLAB ภาพ RGB นั้นโดยทั่วไปคืออาร์เรย์พิกเซล สี M * N * 3 ซึ่งแต่ละพิกเซลของสีมีความสัมพันธ์กับค่าสามค่าที่สอดคล้องกับองค์ประกอบ สีแดง น้ำเงิน และเขียวของภาพRGB

R = ระดับของแสงสีแดง

G = ระดับของแสงสีเขียว

B = ระดับของแสงสีน้ำเงิน

ดังนั้นสีของพิกเซลโดๆ จะถูกกำหนดโดยการรวมกันของความเข้ม สีแดง สีแดง และสีน้ำเงินที่เก็บไว้ในระนาบแต่ละสีที่ตำแหน่งของพิกเซล และระนาบสีแต่ละอันมีอาร์เรย์ M * N (Keim, 2561)

เป็นการแปลงภาพระบบสี RGB ที่แสดงถึงความเข้มสีในแต่ละองค์ประกอบสีให้เป็นภาพที่แสดงถึงค่าความสว่างของแสงเพียงอย่างเดียวโดยทั่วไปภาพระดับเทาสามารถแบ่งระดับของความสว่างได้แตกต่างกัน 256 ระดับ โดยพิกเซลที่มืดที่สุดมีค่าเท่า 0 และพิกเซลที่สว่างที่สุดมีค่าเท่ากับ 255

นอกจากนี้การแปลงภาพสี RGB เป็นภาพระดับเทาสามารถลดเวลาในการประมวลผลภาพดิจิทัลจากการลดจำนวนมิติของข้อมูลในภาพสี RGB ที่จัดเก็บด้วยเมทริกซ์สองมิติซ้อนทับกันสามชั้นให้ลดลงเหลืองเพียงเมทริกซ์สองมิติเพียงชั้นเดียว

2.1.2 การประมวลผลภาพ (Image Processing)

เป็นการนำภาพมาประมวลผลหรือคำนวณด้วยเครื่องคอมพิวเตอร์เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงปริมาณและในเชิงคุณภาพโดยมีขั้นตอนต่างๆที่สำคัญคือการทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในระยะเวลาอันสั้น จึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบ ตัวอย่างการนำการประมวลผลภาพไปใช้งาน เช่น ระบบจำลายนิ้วมือเพื่อตรวจสอบว่าภาพลายนิ้วมือที่มีอยู่นั้นเป็นของผู้ใดระบบตรวจกระดาษคำตอบ โดยมีการเปรียบเทียบภาพกระดาษคำตอบที่ถูกต้องกับกระดาษคำตอบที่จะตรวจว่ามีตำแหน่งตรงกันหรือไม่

2.1.3 การปรับปรุงคุณภาพของภาพ

ในการประมวลภาพนั้นการปรับปรุงคุณภาพของภาพได้กลายมาเป็นอีกปัจจัยหนึ่งที่มีความสำคัญและช่วยให้การประมวลผลภาพในลำดับต่อไปนั้นทำได้ง่ายขึ้น ภาพดิจิทัลโดยทั่วไปนั้นนั้นเกิดจากดิจิไตส์จากข้อมูลที่อยู่ในรูปแบบคลื่นให้กลายมาเป็นข้อมูลแบบจำกัด ดังนั้นบ่อยครั้งที่สภาพแวดล้อมต่างๆ ทำให้ภาพที่เราได้มานั้นไม่ได้เหมาะในการนำมาประมวลผลเพื่อให้ได้ผลลัพธ์ที่ดี ดังนั้นการปรับปรุงคุณภาพของภาพก่อนการประมวลผลจึงเป็นสิ่งที่จำเป็น โดยทั่วไปการปรับปรุงคุณภาพของภาพนั้นจะดำเนินการก่อนการประมวลผลภาพจริง หรือบางครั้งจะเรียกขั้นตอนนี้ว่า ขั้นตอนการประมวลผลภาพก่อนการประมวลผลจริง (image pre-processing)

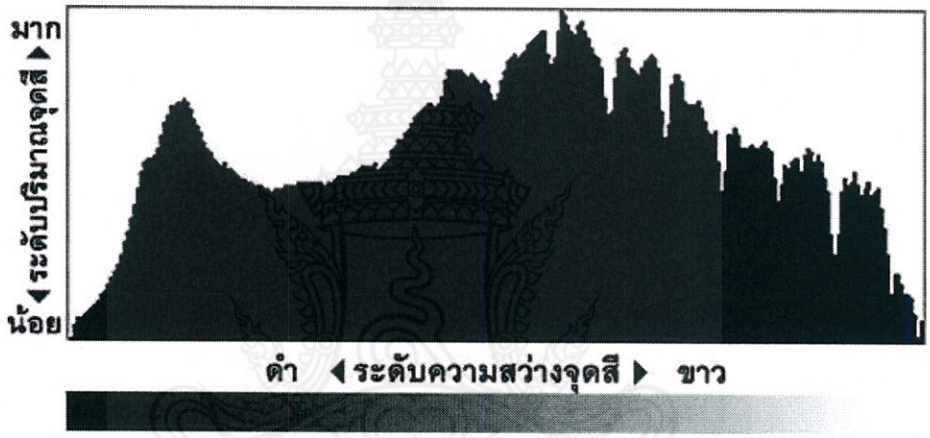
การปรับปรุงคุณภาพของภาพเป็นเทคนิคในการประมวลผลภาพในระดับล่างที่มักจะถูกนำไปใช้ในการประมวลผลภาพก่อนการประมวลผลจริง (pre-processing) โดยการปรับปรุงคุณภาพของ ภาพนั้นจะเป็นการเปลี่ยนแปลงคุณสมบัติของภาพไม่ว่าจะเป็นเรื่องของสี ความสว่าง หรือแม้แต่ ความคมชัดของภาพ รวมถึงการกำจัดสิ่งที่ไม่จำเป็นและอาจจะทำให้การประมวลผลในขั้นตอนถัดไปยากขึ้น

2.1.4 Image Histogram

เป็นกราฟที่แสดงจำนวน pixels ในแต่ละความสว่างต่างๆ หรือข้อมูลค่าสี R,G,B ของรูปภาพ digital ในภาพ gray scale ในแกนอนจะแสดงความสว่างดังกล่าว ซึ่งมีความสว่างตั้งแต่ 0-255 (แบ่งเป็น 256 ระดับความแตกต่างสี) โดยทางด้านซ้ายของกราฟจะมีค่าความสว่างน้อย ภาพจะมีสีเข้มเข้าใกล้สีดำ ส่วนทางด้านขวามือจะมีความสว่างสูง ภาพจะสว่างเข้าใกล้สีขาว ส่วนบริเวณตรงกลางกราฟแสดงส่วนน้ำหนักสีกลาง ส่วนในแนวแกนตั้งจะแสดงจำนวน pixels ในแต่ละความสว่างซึ่งในแกน

ตั้งนี้ ไม่มีขอบเขตจำกัด ถ้าหากภาพมีความมืดมาก กราฟจะไปกองรวมกันทางด้านซ้ายมือ โดยที่ไม่มีขอบเขตจำกัด

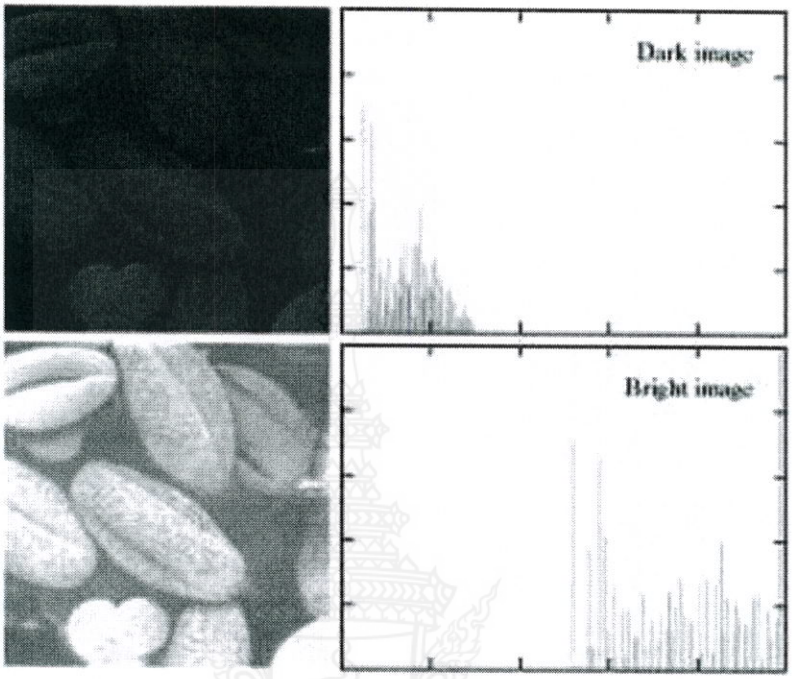
ฮิสโตรแกรมภาพ (image histogram) ภาพประกอบไปด้วยพิกเซลเล็กๆ ซึ่งแต่ละพิกเซลจะมีค่าส่องสว่างหรือว่าค่าสี ค่าสีของพิกเซลจะมีค่าที่แตกต่างกันออกไปขึ้นอยู่กับระบบสีที่ใช้ สำหรับภาพสีแบบ RGB ค่าของพิกเซล 1 พิกเซล จะประกอบไปด้วยค่าสี 3 ค่าด้วยกัน โดยค่าแต่ละค่าจะมีช่วงอยู่ระหว่าง 0-255 นอกจากเรา จะสามารถพิจารณาภาพในรูปแบบปกติแล้ว เรายังสามารถที่จะแสดงการกระจายของค่าสีของ พิกเซลที่อยู่ในภาพด้วย และสามารถแสดงออกมาในรูปแบบของตารางการแจกแจงหรือที่เรียกว่า ฮีสโตรแกรม



รูปที่ 2-2 กราฟที่แสดง จำนวน pixels ในแต่ละความสว่างต่างๆ

ที่มา https://imageprocessinggr3.wordpress.com/2014/11/22/image-histogram/ [2]

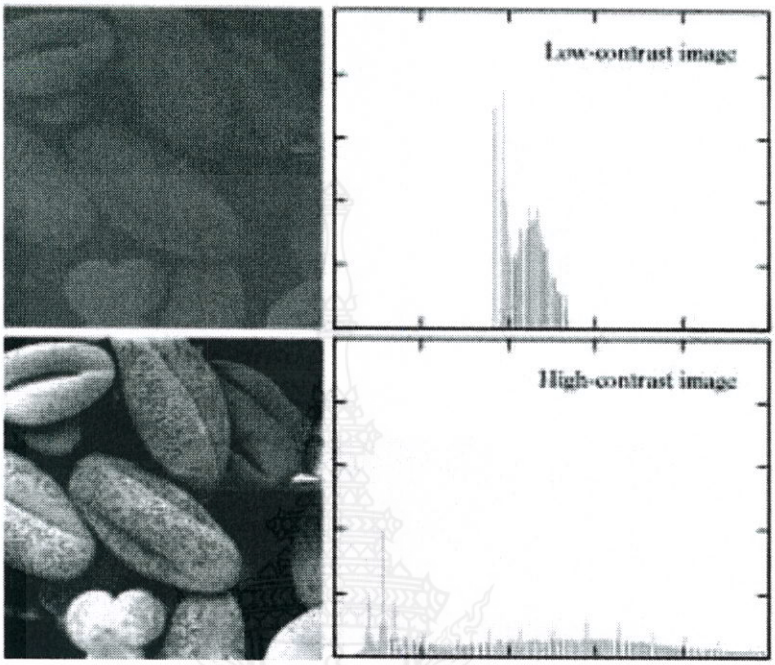
คุณสมบัติของ Histogram หากการกระจายส่วนใหญ่อยู่ทางด้านซ้ายของกราฟ แสดงว่าภาพนั้นมีความสว่างของภาพน้อย ในทางกลับกัน หากการกระจายส่วนใหญ่อยู่ทางด้านขวาของกราฟ แสดงว่าภาพนั้นมีความสว่างของภาพมาก



รูปที่ 2-3 High-Contrast

ที่มา http://www.ecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch03-part2.pdf [3]

หากการกระจายของกราฟเป็นกลุ่มแคบๆ แสดงว่าภาพนั้นเป็นภาพที่ Low-contrast และหากการกระจายของกราฟมีการกระจายอย่างสม่ำเสมอทั่วทั้งกราฟ แสดงว่าภาพนั้นเป็นภาพที่ High-contrast



รูปที่ 2-4 Low-Contrast

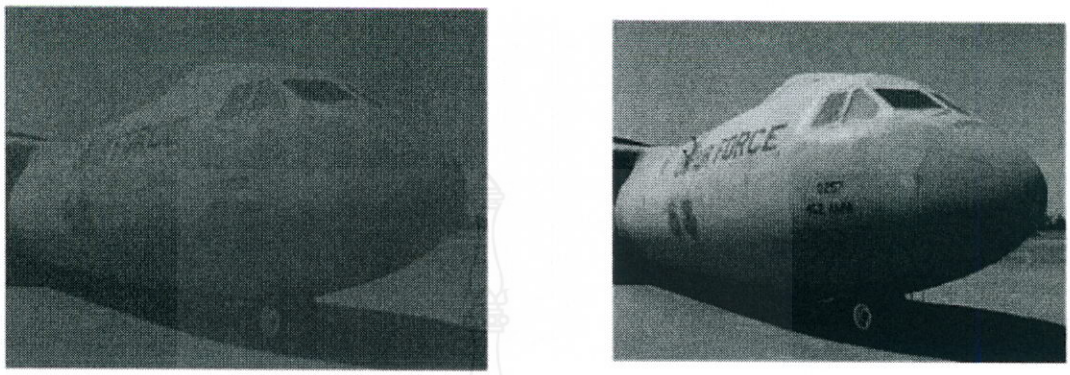
ที่มา http://www.ecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch03-part2.pdf [3]

ประโยชน์ของ Histogram

1. ภาพที่มองจากจอภาพอาจมีการคลาดเคลื่อนได้ แต่ข้อมูลจาก histogram จะบอกความสว่างความเข้มของรูปภาพได้อย่างแท้จริง
2. ข้อมูลนี้จะช่วยให้เราเลือกโหมดในการถ่ายภาพให้ดียิ่งขึ้น โดยช่วยในการเลือกการชดเชยแสงของภาพเมื่อต้องถ่ายภาพในที่ที่มีความสว่างของภาพสูงหรือต่ำมากได้
3. สามารถนำข้อมูลมาใช้ประกอบในการประมวลผลและปรับแต่งภาพได้

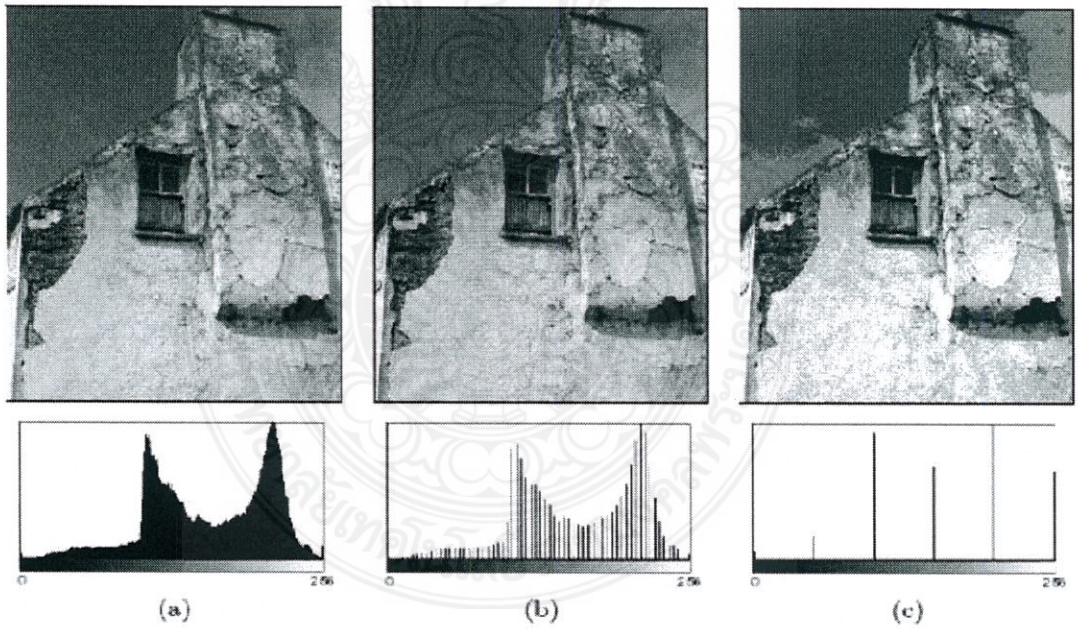
2.1.5 การเพิ่มคอนทราสต์ของภาพ (contrast enhancement)

เป็นอีกกระบวนการหนึ่งที่สามารถ ทำได้ด้วยการดำเนินการกับฮีสโตรแกรมของภาพ ซึ่งคอนทราสต์ของภาพนั้นจะทำให้เราสามารถ เห็นความแตกต่างระหว่างค่าสีที่มีความต่างกันได้ชัดเจนขึ้น โดยทั่วไปการเพิ่มคอนทราสต์ของภาพ นั้นจะดำเนินกับภาพที่มีค่าสีในภาพนั้นใกล้เคียงกัน และในบางครั้งการที่มีคอนทราสต์ที่ไม่เหมาะสมจะให้เราไม่สามารถเห็นรายละเอียดบางส่วนในภาพอย่างชัดเจน เช่น ตัวอย่างของการปรับปรุง คอนทราสต์ภาพแสดงดัง**รูปที่ 2-5**



รูปที่ 2-5 Contrast

Contrast หมายถึง ค่าความแตกต่างสี โดยในทาง histogram เมื่อกราฟที่ได้มีการกระจายความเข้มของสีมาก ก็จะได้ภาพที่คมชัดมากขึ้น หรือภาพที่มีความแตกต่างกันของระดับ สีในภาพน้อย จะได้ภาพที่มีสีซีด



รูปที่ 2-6 การปรับ Contrast

2.1.6 การทรานสฟอร์มภาพ (Image transformation)

การทรานฟอร์มภาพเป็นอีกกระบวนการทางด้านการประมวลผลภาพหนึ่งที่มีการนำมาปรับปรุงคุณภาพ ของภาพเพื่อให้การประมวลผลในขั้นตอนถัดไปนั้นง่ายขึ้น หรือลดความหลากหลาย ของภาพเพื่อให้การประมวลผลสามารถดำเนินการได้อย่างมีประสิทธิภาพ ในหัวข้อนี้จะอธิบาย หลักการและวิธีการการทรานสฟอร์มภาพด้วยการปรับภาพให้อยู่ในมาตรฐานหรือรูปแบบเดียวกัน เพื่อให้ง่ายต่อการประมวลผล ขั้นตอนในการทรานฟอร์ภาพจากคุณสมสมบัติของภาพ มีดังต่อไปนี้ คือ

1. ทำการเลือกภาพต้นแบบ
2. ทำการเปลี่ยนระบบสีภาพทั้งภาพต้นแบบ และ ภาพที่จะประมวลผล หรือ ภาพนำเข้า
3. คำนวณค่าสถิติของทั้ง 2 ภาพ
4. ทำการ ปรับการกระจายของค่าพิกเซลของภาพประมวลผลให้เป็นเหมือนภาพต้นแบบ โดยใช้ข้อมูลทางสถิติ
5. ทำการแปลงภาพประมวลผลกลับไปยังระบบสีเดิม เพื่อให้เห็นภาพถึงขั้นตอนในการประมวลผลการทรานสฟอร์มภาพ

2.1.7 Deblur

เป็นอีกกระบวนการทางด้านการประมวลผลภาพหนึ่งที่มีการนำมาปรับปรุง คุณภาพของภาพ เพื่อให้การประมวลผลในขั้นตอนถัดไปนั้นง่ายขึ้น หรือ ลดเบลอของภาพเพื่อให้การประมวลผลสามารถดำเนินการได้อย่างมีประสิทธิภาพ ในหัวข้อนี้จะอธิบาย หลักการและวิธีการการดีเบลอภาพด้วยการปรับให้มีความคมชัดมากขึ้น เพื่อให้ง่ายต่อการประมวลผล ขั้นตอนในดีเบลอภาพที่ใช้มีดังต่อไปนี้ คือ

1. เลือกภาพต้นแบบ
2. หาค่า threshold ของภาพ ด้วย Otsu อัลกอริทึม
3. หาเส้นของภาพด้วย canny age
4. ปรับค่าให้มีความคมชัดมากยิ่งขึ้นด้วยฟังก์ชั่น sharpening Kernel

การหาค่า threshold เป็นการเลือกจุดตัดที่เหมาะสมในการแบ่งส่วนของภาพ ซึ่งจะเป็นการคัดเลือกจุดตัดแบบภาพรวม (Global thresholding) และการคัดเลือกจุดตัดแบบกลุ่มย่อย(Local thresholding) ดังต่อไปนี้

Global Thresholding

เป็นการหาจุดตัดจากภาพรวมทั้งหมดของภาพ แบ่งเป็นวิธีหลักๆ 2 วิธี ได้แก่ Single thresholding และ Double thresholding

- Single thresholding เป็นการพิจารณาจากฮิสโตแกรมของระดับความเข้มของภาพเพื่อหาจุดตัดที่ เหมาะสม วิธีนี้ใช้ได้ดีกับการแยกภาพที่วัตถุกับพื้นหลังแยกกันอย่างชัดเจน

-Double Thresholding สำหรับกรณีที่ที่ต้องการหาค่า Threshold 2 ค่า

Local (Adaptive) Thresholding

สำหรับ Local Threshold หรือเรียกว่า Adaptive Threshold ซึ่งเป็นการหาค่า Threshold ภาพในพื้นที่ย่อยๆ ของแต่ละจุดภาพ ดังนั้นค่า Threshold จะปรับเปลี่ยนไปตามแต่ละจุดภาพ

ตารางที่ 2-1 ข้อดีและข้อเสียของวิธี Thresholding

วิธี	ข้อดี	ข้อเสีย
Global-Thresholding	- เป็นวิธีที่ง่าย - ทำงานได้เร็ว - เหมาะกับภาพที่ระดับสีมีการกระจายตัวสม่ำเสมอ (Uniform)	- กำจัดสัญญาณรบกวนได้ไม่ดี - ไม่เหมาะกับภาพที่ระดับสีมีการกระจายไม่สม่ำเสมอ - แบ่งระดับของสีได้ไม่มากนัก เหมาะกับภาพสีไม่ซับซ้อน
Local-Thresholding	- เป็นวิธีที่ง่าย - ทำงานได้ช้ากว่า Global - เหมาะกับภาพที่ระดับสีมีการกระจายตัวไม่สม่ำเสมอ	- แบ่งระดับของสีได้ไม่มากนัก เหมาะกับภาพสีไม่ซับซ้อน

การตรวจจับขอบวัตถุด้วยวิธีแคนนี (Canny Edge Detection)

วิธี Canny เป็นวิธีที่ประยุกต์จากอนพันธ์อันดับดับที่ 1 โดยการประยุกต์กับ Guassian เพื่อประมาณค่า Operator โดยมีขั้นตอน ดังนี้

1. กรองภาพด้วยอนุพันธ์ของ Guassian โดยการคำนวณค่า G_x และ G_y ดังสมการที่ (1) และ (2)

$$G_x = \frac{\partial}{\partial x}(f*g) = f*\frac{\partial}{\partial x}g = f*g_x \qquad (1)$$

$$G_y = \frac{\partial}{\partial y}(f*g) = f*\frac{\partial}{\partial y}g = f*g_y \qquad (2)$$

กำหนดให้ g(x,y) Guassian Function

g_x (x,y) และ g_y (x,y) คืออนุพันธ์ของ g(x,y) ตามแนวแกน x และ y ตามลำดับโดยที่

$$g_x(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{\partial^2}\ g(x,y), \qquad (3)$$

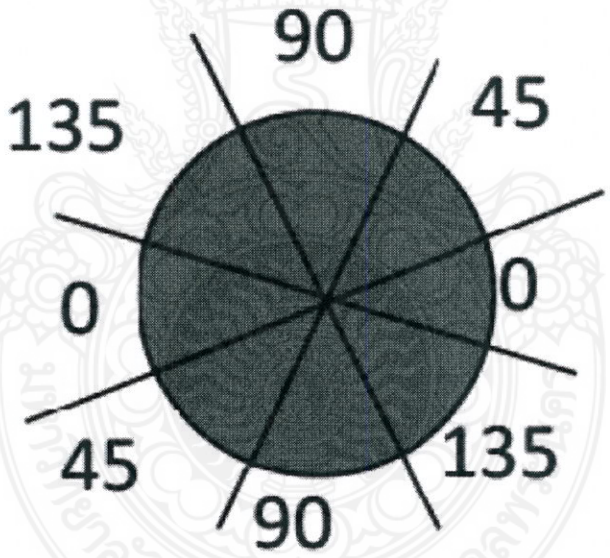
$$g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{\partial^2} \; g(x,y), \qquad (3)$$

2. หาค่าขนาด (Magnitude) และทิศทาง (Direction) ของแกรเดียนท์

$$|\nabla f| = \sqrt{G_x{}^2 + G_y{}^2}$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

โดยมี 4 ทิศทาง คือ 0,45,90 และ 135 องศา สำหรับการหาแกรเดียนท์ใช้วิธี Sobel



รูปที่ 2-7 มุม θ ของแกรเดียนท์

3. ประยุกต์การทำ Non-maximum suppression เพื่อให้ขอบวัตถุบางเหลือเพียงของวัตถุเพียงจุดเดียว

- ระบุทิศทางของขอบภาพจากค่ามุม 6 โดยมี 4 ทิศทาง คือ 0 , 45 , 90 และ 135 องศา สำหรับการหาแกรเดียนท์ใช้วิธี Sobel

- ตรวจสอบถ้าจุดภาพที่มีทิศทางเดียวกันโดยเช็คว่า ถ้าจุดภาพใดมีค่ามากที่สุดให้พิจารณาค่านั้นเป็นจุดภาพที่น่าจะป็นขอบภาพนำไปประมวลผลต่อ นอกนั้นก็ให้เป็น 0

4. ประยุกต์การทำ Double thresholding โดยการนำค่า threshold 2 ค่า (Low และ high) ให้ค่า Low threshold (T_{lo}) ในการกำหดจุดเริ่มต้นของเส้นขอบและจุดปลายอยู่ที่ High threshold (T_{hi}) ซึ่งโดยปกติแล้ว $T_s \approx 2(T_{lo})$ มาเปรียบเทียบกับ Gradient magnitude เพื่อระบุค่าจุดภาพ 3 ชนิด (Weak. Strong และ Non-relevant) ดังต่อไปนี้

- จุดภาพที่มีระดับความเข้มสูง พิจารณาให้เป็น Strong pixel คือจุดที่คาดว่าจะเป็นของภาพ
- จุดที่มีระดับความเข้มไม่สูงละต่ำจัด จะพิจารณาเป็น Weak pixel
- นอกนั้นพิจารณาให้เป็น non-relevant

5. ประยุกต์การทำ Hysteresis ซึ่งเป็นเชื่อมขอบภาพ จากการทำ Double Thresholding เนื่องจากขั้นตอนการทำ Non-maximum suppression นั้นอาจยังมีสัญญาณรบกวนเกิดขึ้น โดยพิจารณาจุดภาพที่เป็น weak pixel และทาการเพิ่มขอบ (Contour) โดยพิจารณาจุดรอบๆ ข้างทั้ง 8 ทิศทางหากมีเพียง 1 จุดภาพที่เป็น Strong pixel ก็จะกำหนดให้จุดภาพปัจจุบันเป็นขอบภาพ (Strong Pixel) ดังตัวอย่าง

ข้อดี

1. เป็นวิธีที่ดีในการระบุขอบวัตถุ
2. เป็นการจำแมกคุณลักษณะ () โดยไม่เปลี่ยนแปลงคุณลักษณลักษณะของภาพ
3. ไวต่อสัญญาณรบกวนน้อยมาก

ข้อเสีย

1. อาจทำให้กรณี Zero Crossing เป็นเท็จได้
2. ใช้เวลาในการประมวลผลมากและชั้นตอนในการคำนวณซับซ้อน

คำสั่ง Canny ใน OpenCV

คำสั่งการหาขอบภาพด้วย Canny ดำเนินการได้ดังนี้

edges - c/2.Canny (image, threshold1, threshold2, [aperturesize, L2radient])

edges - cv2.Canny(dx, dy, threshold1, threshold2, [, L2radient])

Parameters :

Image :	ภาพนำเข้าขนาด 8 bit เป็น ndarray
dx:	ค่าอนุพันธ์ตามแนวแกน x (CV 16SC1 หรือ CV_165C3)
dy:	ค่าอนุพันธ์ตามแนวแกน y (CV 165C1 หรือ CV_16SC3)
threshold1 :	ค่า Low threshold ของ Hysteresis
threshold2 :	ค่า High threshold ของ Hysteresis

apertireSize : ขนาดของการค้านวณหาค่า Sobel ซึ่ง defailt-3

L2gationt : การหาค่า magnitude ด้วยรากของกำลังสอง ซึ่งมีค่าเป็น True หรือหา absolute ซึ่งมีค่าเป็น default-False

Result:

edges : ภาพผลลัพธ์ (edge map) ขนาด 8 bit หรือเท่ากับภาพนำเข้า

2.1.8 การเรียนรู้เชิงลึก (Deep Learning)

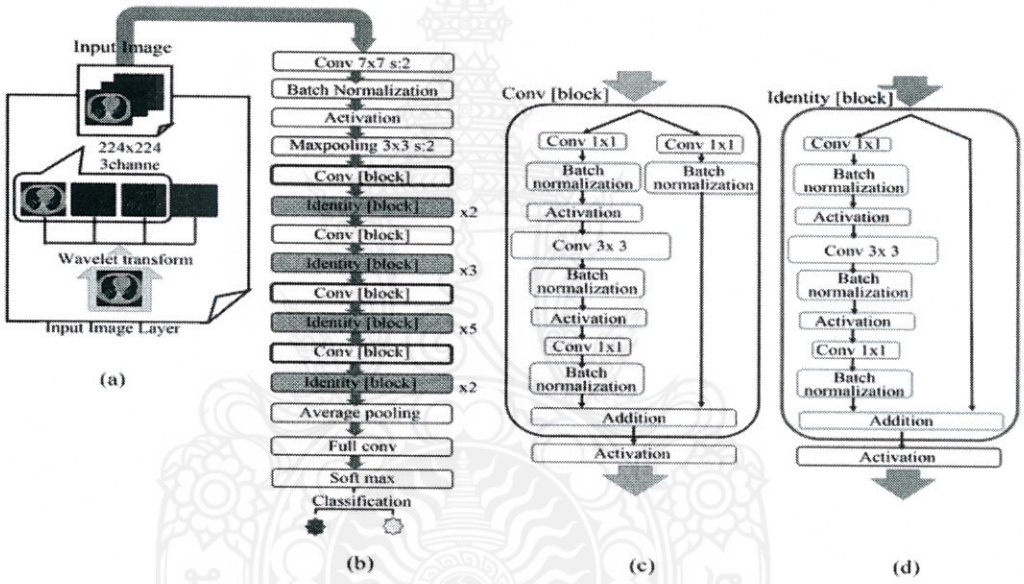
การเรียนรู้อย่างลึกซึ้งเป็นฟังก์ชันของ ปัญญาประดิษฐ์ที่เลียนแบบการทำงานของสมองมนุษย์ในการประมวลผลข้อมูลและสร้างรูปแบบเพื่อใช้ใช้ในการตัดสินใจ การเรียนรู้อย่างลึกซึ้งเป็นส่วนหนึ่งของการเรียนรู้ของเครื่องในปัญญาประดิษฐ์ (AI) เป็นเครือข่ายที่สามารถเรียนรู้ได้โดยที่ไม่ได้รับการสนับสนุนจากข้อมูลที่ไม่มีโครงสร้างหรือไม่มีป้ายกำกับ ยังเป็นที่รู้จักกันในนามการเรียนรู้ระบบประสาหลึกหรือเครือข่ายประสาทลึก

การเรียนรู้อย่างลึกซึ้งได้พัฒนาไปพร้อมๆ กับยุคดิจิทัลซึ่งทำให้เกิดการระเบิดของข้อมูลในทุกรูปแบบและจากทุกภูมิภาคของโลก ข้อมูลนี้เรียกได้ว่าเป็นข้อมูลขนาดใหญ่นั้นนำมาจากแหล่งต่างๆ เช่น โซเชียลมีเตียเครื่องมือค้นหาอินเทอร์เน็ตแพลตฟอร์มอีคอมเมิร์ซและโรงภาพยนตร์ออนไลน์เป็นต้นต้น ข้อมูลจำนวนมหาศาลนี้สามารถเข้าถึงได้อย่างง่ายดายและสามารถแบ่งปันผ่านแอพพลิเคชั่น fintech เช่น cloud computing การเรียนรู้เชิงลึกกับการเรียนรู้ของเครื่องในเทคนิค AI ที่ใช้กันมากที่สุดในการประมวลผลข้อมูลขนาดใหญ่คือการเรียนรู้ด้วยเครื่องซึ่งเป็นอัลกอริทึมการปรับตัวเองที่ได้รับการวิเคราะห์และรูปแบบที่ดีขึ้นด้วยประสบการณ์หรือด้วยข้อมูลที่เพิ่มเข้ามาใหม่

การเรียนรู้เชิงลึกหรือที่เรียกว่าการเรียนรู้แบบลำดับชั้นหรือการเรียนรู้เชิงโครงสร้างเป็นประเภทของการเรียนรู้ของเครื่องที่ใช้สถาปัตยกรรมอัลกอริทึมแบบแบ่งชั้นเพื่อวิเคราะห์ข้อมูลโนโมเดลการเรียนรู้เชิงลึกข้อมูลจะถูกกรองผ่านการเรียงซ้อนของหลายเลเยอร์โดยแต่ละเล เยอร์ที่ต่อเนื่องจะใช้เอาต์พุตจากเลเยอร์ก่อนหน้าเพื่อแจ้งผลลัพธ์ แบบจำลองการเรียนรู้แบบลึกสามารถมีความแม่นยำมากขึ้นเมื่อประมวลผลข้อมูลมากขึ้นโดยการเรียนรู้จากผลลัพธ์ก่อนหน้าเพื่อปรับแต่งความสามารถในการสร้างความสัมพันธ์และการเชื่อมต่อการเรียนรู้อย่างลึกซึ้ง นั้นขึ้นอยู่กับวิธีการที่เซลล์ประสาททางชีวภาพเชื่อมโยงถึงกันเพื่อประมวลผลข้อมูลในสมองของสัตว์ เช่นเดียวกับที่สัญญาณไฟฟ้าเคลื่อนที่ผ่านเซลล์ของสิ่งมีชีวิตแต่ละเลเยอร์ต่อมาจะถูกเปิดใช้งานเมื่อได้รับสิ่งเร้าจากเซลล์ประสาทข้างเคียงในเครือข่ายประสาทเทียม (ANNs) พื้นฐานสำหรับโมเดลการเรียนรู้เชิงลึกแต่ละชั้นอาจได้รับมอบหมายส่วนเฉพาะของถานกรมปลงและรัสมูลยาจคลื่อนที่เลธธรรหายครั้งเพื่อปรับแล่มห่นประชั้นประทัพรภาพยาที่พูดขั้นสูงสุด เลเยอร์"ซ่อนเร้น" เหล่านี้ทำหน้าที่ในการดำเนินการแปลทางคณิตศาสตร์ซึ่งเปลี่ยนข้อมูลดิบเป็นผลลัพธ์ที่มีความหมาย

CNN ResNet

ResNet50-v2 ResNet มาจาก Deep residual Network ถูกนำเสนอในงานวิจัย Deep residual learning for image recognition ซึ่งแก้ปัญหาเรื่อง vanishing gradient ซึ่งเกิดขึ้นกับโครงข่ายที่มี ความลึกค่อนข้างมากซึ่งมีจำนวนชั้นของ network ถึง 152 เลเยอร์ (8 เท่าของโมเดล VGG16) โดยใช้เทคนิคการออกแบบ module ที่มีลักษณะทางลัดลงใน network ตัวโครงข่ายนี้ประกอบด้วยกัน 4 block โดยจำนวนที่มีพารามิเตอร์สำหรับฝึกทั้งหมดคือชั้นที่เราใช้เรียกชื่อ เช่น ResNet50 จะหมายถึงจำนวน 50 เลเยอร์ซึ่งจะอธิบายขนาดว่า [3, 4, 6, 3] ซึ่งคือ (3 + + 6 + 3) x3 = 48 ชั้น + 2 ชั้น = 50 ซึ่ง ResNet ที่นิยมใช้จะเป็น ResNet18, ResNet34, ResNet50, ResNet101 และ ResNet152



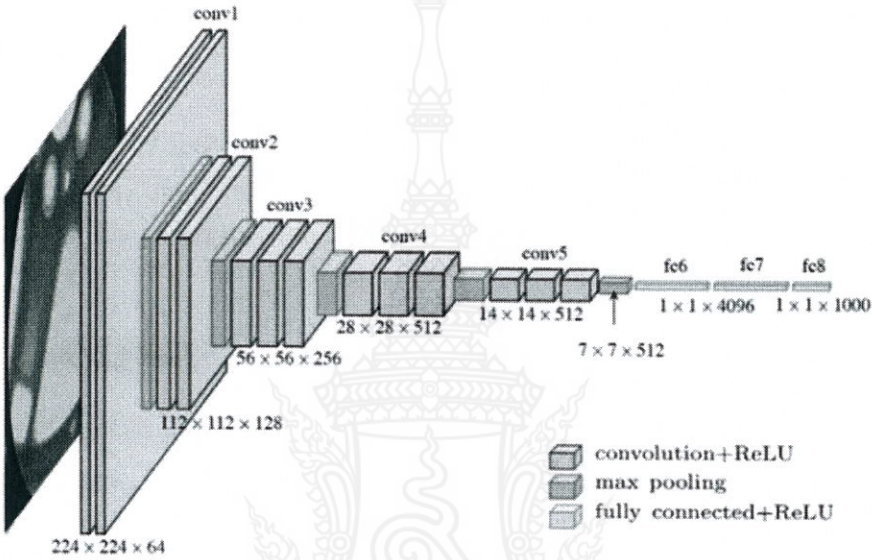
รูปที่ 2-8 ResNet 50 Architecture

Residual Network (ResNet) เป็นสถาปัตยกรรม Convolutional Neural Network (CNN) ซึ่งได้รับการออกแบบมาเพื่อเปิดใช้งานเลเยอร์ Convolutional หลายร้อยหรือหลายพันชั้น ในขณะที่สถาปัตยกรรม CNN รุ่นก่อนมีประสิทธิภาพของเลเยอร์เพิ่มเติมลดลง ResNet สามารถเพิ่มเลเยอร์จำนวนมากพร้อมประสิทธิภาพที่แข็งแกร่ง ResNet เป็นโซลูชันที่สร้างสรรค์สำหรับปัญหา "การไล่ระดับสีที่หายไป"

VGG16

โมเดล VGG ย่อมาจาก Visual Geometry Group ซึ่งเป็นกลุ่มนักวิจัยจาก Oxford ทำการพัฒนาสถาปัตยกรรมนี้ขึ้นมา และที่ได้รับความสนใจมากจากการแข่งขัน ILSVR ปี ค.ศ. 2014 และเป็นที่นิยมจนถึงปัจจุบัน โดยสิ่งที่เป็นจุดเด่นของ VGG16 คือการแทนที่ hyperparameter จำนวนมาก

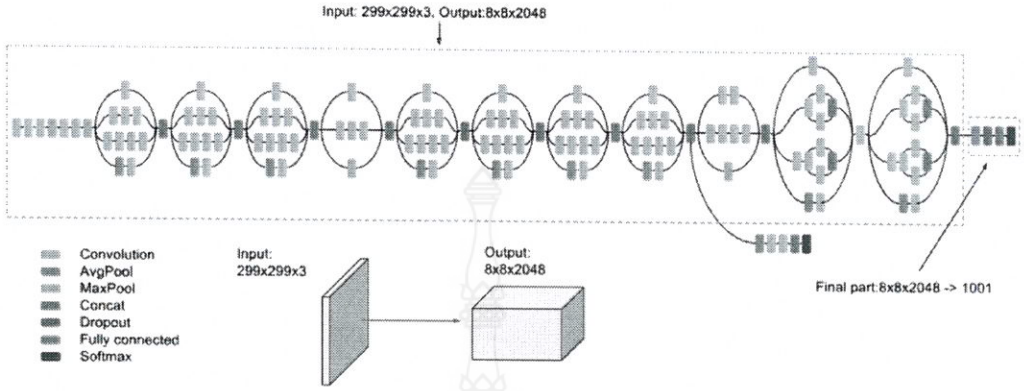
เน้นไปที่การออกแบบ เลเยอร์ conv2D 3x3 pixels, 1 stride และการใช้ same padding และ maxpooling 2x2 pixels, 2 stride แบบเดียวกันตอดทั้งโครงสร้าง โดยชื่อของ VGG16 หมายถึงมี16 ชั้นที่มีน้ำหนักเครือข่ายนี้เป็นเครือข่ายที่ใหญ่และมีพารามิเตอร์ประมาณ 138 ล้าน ดังรูปที่ 2-9



รูปที่ 2-9 VGG16 Architecture

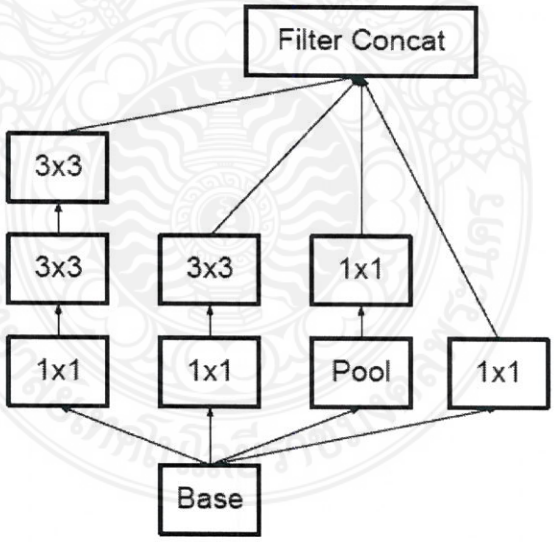
Inception V3

Inception-v3 เป็นโมเดลที่ได้รับการ พัฒนาโดย Google ซึ่งถูกต่อยอดจาก Inception 2,1 (ซึ่งมาจากการพัฒนามาจาก GoogLeNet 2012) โดยการลดโครงสร้างภายในออกเป็น 5 Step คือ Inception Module A จำนวน 5 Module (1), Grid Size of Reduction Step1 จำนวน 1 Module (2), Inception Module B จ้านวน 4 Module (3), Grid Size of Reduction Step2 จำนวน 1 Module (4), Inception Module C จำนวน 2 Module (5) และ Head (8x8x2048) สามารถแยก output ได้ 1,000 classes ดังรูปที่ 2-10



รูปที่ 2-10 Inception V3

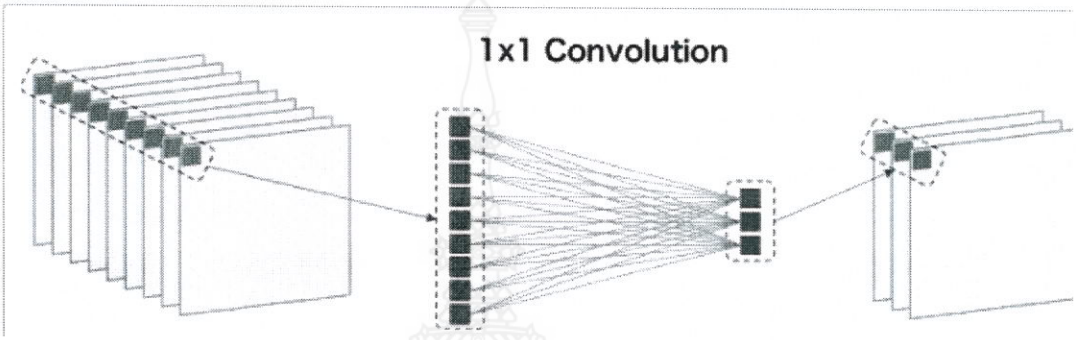
พัฒนาให้ Inception-v3 มี parameter ลดลงจากเดิม แต่ยังคงสิทธิภาพสูงการปรับปรุง convolutions เดิมที่ 5x5 ลงเหลือ 3x3 pixels และ maxpooling เดิมที่ 3x3 ลงเหลือ 2x2 pixels ซึ่งวัตถุประสงค์ของการออกแบบ 1x1 convolution ต้องการให้ shape outputs ขา ออกเป็น tensor เช่น กำหนด (N. F. H. W) ซึ่ง N คือ batch size, F คือ จำนวนของ convolution filters ส่วน H, W คือ spatial ของมิติ ดังรูปที่ 2-11



รูปที่ 2-11 Inception Model

โดยภายในของ inception module กรณี input ข้อมูลเข้ามามี 3 สีคือ RGB ในที่นี้เราสามารถเปรียบเทียบได้กับการทำ feature map ดังรูปด้านซ้าย ซึ่งสมมุติว่าเราต้องการผลลัพธ์ feature

map เพียง 1 ค่านั้นถูกทำได้โดยใช้ convolution ขนาด 1x1 pixels ที่มีการขยับที่ 1 stride ดังรูปที่ 2-12 โดยค่า weight ที่คุณในแต่ละชั้นของ RGB นั้นจะมีค่าที่ไม่เท่ากันส่งผลให้ค่าที่ได้ใน แต่ละพิกเซล หลังการทำ feature map นี้จะมีที่เป็นลักษณะเฉพาะ



รูปที่ 2-12 การทำ Feature map 1x1 Convolution

2.1.9 โปรแกรมมายเอสคิวแอล (MySQL)

เกษศิรินทร์ โถวสกุล [4] อธิบายว่า MySQL เป็นโปรแกรมฐานข้อมูลที่ใช้จัดเก็บข้อมูลโปรแกรมหนึ่ง ทำงานในลักษณะ Client Server ทำงานบนระบบบ Telnet UM Linux Redhad หรือ Unix System และบน Win32 ทั่วไปบนระบบเครือข่ายอินเทอร์เน็ต และอินทราเน็ตและยังสามารถเรียกใช้บนเว็บบราวเซอร์ได้กรณีใช้ภาษาเป็นอินเตอร์เฟสในการเชื่อมภาษาที่ใช้เป็นอินเตอร์เฟส เช่น PHP, Perl, C,C++ เป็นต้น มายเอสคิวแอลเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) RDBMS คือ สามารถทำงานกับตารางข้อมูลหลายตารางพร้อมๆ กันโดยสามารถแสดงความสัมพันธ์ของตารางเหล่านั้นด้วย Field ที่ใช้ร่วมกัน

มายเอสคิวแอลเป็นโปรแกรมบริหารจัดการฐานข้อมูลหรือเรียกว่า DataBase Management System ซึ่งมักจะใช้คำย่อเป็น DBMS

มายเอสคิวแอลทำงานในลักษณะฐานข้อมูลเชิงสัมพันธ์ (Relational DataBase Management System: RDBMS) คำว่าว่าฐานข้อมูลเชิงสัมพันธ์ก็คือ ฐานข้อมูลที่แยกข้อมูลไปเก็บเอาไว้ในหน่วยย่อย ซึ่งเรียกว่า ตารางข้อมูล (Table) แทนที่จะเก็บข้อมูลทั้งหมดรวมกันเอาไว้แห่งเดียว แต่ละหน่วยย่อยที่ใช้เก็บข้อมูลต่างมีความสัมพันธ์เชื่อมโยงกันอยู่ ยกตัวอย่างเช่น ข้อมูลสินค้า ที่จัดเก็บแยกกันได้ แล้วอาศัยรหัสสินค้าในการเรียกค้นข้อมูลที่จัดเก็บแยกเอาไว้ การที่จะเข้าไปจัดการกับข้อมูล ต้องอาศัยภาษาคอมพิวเตอร์ที่เรียกกันว่า SQL ซึ่งย่อมาจาก Structured Query Language ชื่อ มายเอสคิวแอลก็สื่อให้ทราบว่าเกี่ยวกับภาษา SQL อยู่แล้ว ดังนั้นมายเอสคิวแอลจึงทำงานตามคำสั่งภาษาเอสคิวแอลได้อันเป็นมาตรฐานของโปรแกรมทางด้านฐานข้อมูลในยุคนี้ที่จะต้องมีความสามารถรองรับคำสั่งที่เป็นภาษาเอสคิวแอล

มายเอสคิวแอลมีจุดเด่นที่สามารถครองใจผู้ใช้คือ เร็ว ใช้งานง่ายและมีความเชื่อถือได้สูง ซึ่งมายเอสคิวแอลเองก็มีนิยามประจำตัวว่า MySQL is a very fast, multi-threaded, multi-user, robust SQL(Structured Query Language) database server and MySQL is free software. ซึ่งเมื่อเปรียบเทียบกับบรรดาโปรแกรมบริหารจัดการฐานข้อมูล ที่ทำงานเหมือนกันและมีอยู่ในท้องตลาดในปัจจุบัน เช่น MS SQL Server หรือ Oracle เป็นต้น จะพบว่าโดยรวมแล้ว การทำงางานของมายเอสคิวแอล ไม่ได้แย่กว่าหรือเหนือกว่าโปรแกรมเหล่านั้นเลย การทำงานของมายเอสคิวแอลในบางเรื่องหรือบางฟังก์ชันอาจจะแย่กว่าและในทำนองเดียวกันมายเอสคิวแอลก็ทำงานได้ดีกว่าในบางเรื่องบางฟังก์ชัน

คุณสมบัติของมายเอสคิวแอลที่น่าสนใจมีดังนี้

1. ทำงานแบบ multi-treaded หมายถึงการแบ่งการทำงานเป็นส่วนย่อยแยกออกไปต่างคนต่างทำงาน ทำให้สามารถทำงานได้เร็ว และการทำงางานมีความอิสระไม่ขึ้นต่อกันรวมทั้งสามารถนำไปใช้กับเครื่องที่มี CPU มากกว่า 1 ตัวได้

2. ใช้ได้กับภาษาโปรมกรมมิ่งหรือสคริปต์หลากหลายภาษา อาทิ C, C++, Eiffel, Java, Perl, PHP, VB, Delphi, VFP เป็นต้น

3. ทำงานกับฐานข้อมูลขนาดใหญ่ได้ เคยมีผู้ใช้กับตารางข้อมูลถึง 60,000 ตาราง มีจำนวนรายการข้อมูลถึง 5,000,000,000 รายการอย่างไม่มีปัญหา

4. รองรับชนิดข้อมูลที่หลากหลาย เช่น signed/unsigned, INTEGER ขนาด 1, 2, 3, 4 และ 8 ไบต์, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIME STAMP, YEAR, SET HOS ENUM

5. รองรับภาษา SQL มาตรฐาน

6. รองรับ ODBC (Open Database Connectivity)

7. ใช้ได้กับระบบปฏิบัติการหลากหลายระบบ เช่น Limas, Solaris, Mac OS X Server, OS/2 Warp, SunOS, Windows และระบบตระกูล Unix อีกมากมาย

2.2 งานวิจัยที่เกี่ยวข้อง

สมปอง, ธีรศักดิ์ และณัฏฐ์ (2567) [5] กล่าวถึง การทำงานในองค์กรจะมีการสร้างเอกสารเป็นจำนวนมากในแต่ละวัน ซึ่งการจัดเก็บเอกสารในรูปแบบของแฟ้มเอกสารทั่วไปอาจนำมาซึ่งปัญหา เช่นการเกิดเชื้อราจากความชื้น สีอักษรเกิดความซีดจาง กระดาษเสื่อมสภาพ ซึ่งเอกสารเหล่านี้มักถูกจัดเก็บในรูปแบบของไฟล์ภาพดิจิทัลด้วยการสแกนเอกสารด้วยเครื่องสแกนเนอร์ หรือกล้องถ่ายภาพดิจิทัล พบว่า ไฟล์ภาพที่ถูกจัดเก็บใช้พื้นที่จำนวนมาก มีความยากลำบากในการสร้างระบบค้นคืนเอกสาร ทำให้มีการพัฒนาระบบสืบค้นข้อมูลจากระบบจำตัวอักษรขึ้น ซึ่งสามารถแปลงตัวอักษรที่ประกอบกันเป็นข้อความภายในเอกสารที่อยู่ในรูปแบบของไฟล์ภาพดิจิทัลให้กลายเป็นข้อความตัวอักษรธรรมดา ซึ่งพบความแม่นยำสูง จากตัวอักษรที่เป็นการพิมพ์ แต่มักมีหลายองค์กรสร้างเอกสารข้อความเหล่านี้ด้วยการเขียนด้วยมือแทนการพิมพ์ ทำให้ข้อความที่ปรากฏไม่มีความคงที่ และแน่นอน ส่งผลให้การพัฒนาระบบ

สำหรับการรู้จำเป็นไปได้ยาก จากการศึกษางานวิจัยที่เกี่ยวข้องในการรู้จำตัวอักษรจากการเขียนด้วยลายมือ โดยเฉพาะภาษาไทยพบว่า มีการสกัดลักษณะเด่นด้วยวิธีต่างๆ มาใช้ร่วมกับวิธีรู้จำด้วยโครงข่ายประสาทเทียมเพื่อเพิ่มความยำ ยังมีการพัฒนาขั้นตอนวิธีในการรู้จำทั้งในลักษณะ Lexicon Driven Word Recognition [6] และการนำเทคนิคที่ผสมผสานระหว่าง Heuristic Rules กับโครงข่ายประสาทเทียม [7] เพื่อการรู้จำลายมือเขียนภาษาไทย การเรียนรู้เชิงลึกซึ่งสามารถแบ่งการทำงานเป็นสามส่วนหลักดังนี้

1) ส่วนของการเตรียมไฟล์ภาพตัวอักษร (Pre-Processing) ซึ่งจะเริ่มตั้งแต่การรับไฟล์ภาพเอกสารข้อความที่เขียนด้วยลายมือเข้ามาในระบบ จากนั้นทำการกำจัดสัญญาณรบกวนในภาพ (Noise Removing) แล้วแบ่งแบ่งภาพตามแนวเส้นบรรทัด (Line Segmentation) ตามด้วยการค้นหาข้อความในบรรทัดนั้นๆ

2) ส่วนของการรู้จำไฟล์ภาพของแต่ละตัวอักษร ซึ่งจะใช้โครงข่ายประสาทเทียมที่มีสถาปัตยกรรมแบบสังวัตนาการ (Convolutional Neural Network) [8]

3) ส่วนของการปรับปรุงข้อความที่ได้ให้มีความถูกต้องมากยิ่งขึ้น (Post-Processing) ด้วยโครงข่ายประสาทเทียมที่อยู่บนพื้นฐานโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network) [9] อย่างเช่น Long Short-Term Memory (LSTM) หรือ Gated Recurrent Units [10] เมื่อจบกระบวนการเหล่านี้จะทำให้สามารถแปลงภาพถ่ายเอกสารข้อความให้กลายเป็นข้อความรูปแบบของยูนิโคด (Unicode Plain Text) ได้

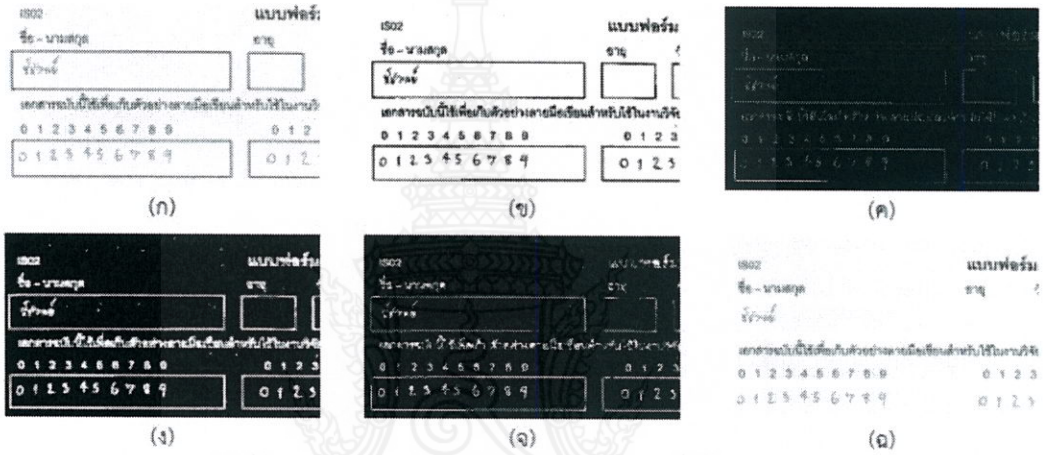
การเก็บรวบรวมข้อมูลสำหรับการทดลอง มีการชุดข้อมูลภาพ 68PersonsBMP [11] ซึ่งเป็นชุดข้อมูลที่รวบรวมสำหรับฝึกฝนในการแข่งขัน NSC2019 ประกอบด้วยภาพตัวอย่างการเขียนอักษรไทยจากผู้เขียน จำนวน 68 คน เนื่องด้วยมีบางตัวอย่างที่ให้ข้อมูลตัวอย่างอักษรมาไม่ครบ ผู้วิจัยจึงได้ทำการคัดเลือกมาจำนวน 50 คน คนละ 2 ตัวอย่าง ทำการคัดแยกออกเป็นคลาสต่างๆ ตัดอักษรที่ไม่ปรากฏในพจนานุกรมอย่างเช่น ข และ ค ทำให้ได้อักษรทั้งหมด 78 คลาด รวมทั้งสิ้น 7,800 ตัวอย่าง กระบวนการเตรียมภาพจะถูกดำเนินการ 3 ขั้นตอนดังนี้

1) การแปลงภาพนำเข้าให้มีระดับความเข้มเทา ภาพที่นำมาเป็น Input สำหรับการรู้จำอักษรจะได้จากการถ่ายภาพ หรือสแกนมาจากเอกสารกระดาษที่มักจะมีพื้นหลังเป็นสีขาวและมีสีตัวอักษรที่ตัดกันชัดเจน อีกทั้งในการรู้จำตัวอักษร จะใช้เพียงแนวทางเดินของเส้นที่สร้างเป็นตัวอักษรเพื่อแยกคลาสของแต่ละภาพโดยที่สีนั้นไม่มีความจำเป็นต่อการคัดแยก ด้วยเหตุนี้สีของอักษรจึงไม่มีนัยสำคัญต่อการรู้จำ ดังนั้น เพื่อให้การประมวลผลสามารถทำได้เร็วขึ้นภาพที่อยู่ในปริภูมิสี RGB จึงถูกแปลงให้อยู่ในปริภูมิระดับความเข้มเทา ซึ่งทำได้โดยใช้สมการที่ (1) [12]

$$y = (0.2989x\ x_R) + (0.5870\ x\ x_G)\ x\ (0.1140x\ x_B) \qquad (1)$$

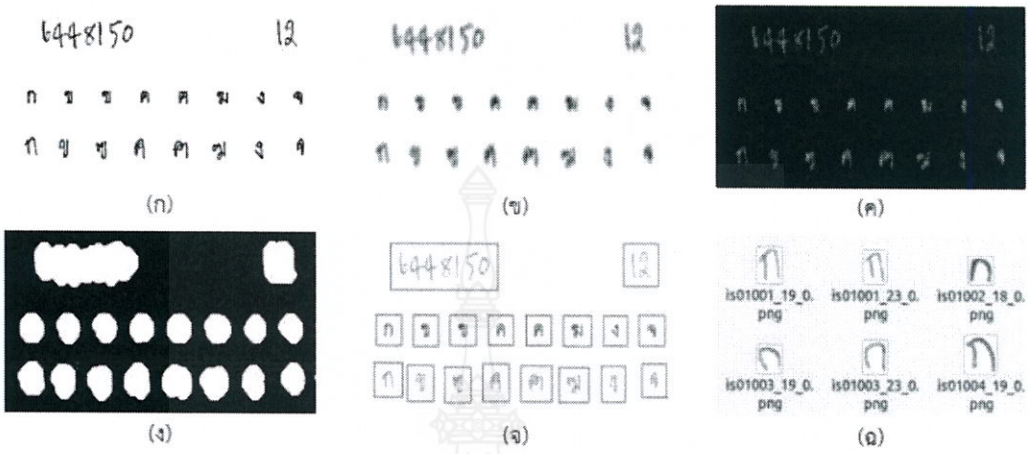
โดยที่ x คือแต่ละพิกเซลของภาพ Input ในปริภูมิสี RGB และ y คือแต่ละพิกเซลของภาพ Output ในปริภูมิระดับความเข้มเทา

2) การกำจัดขอบกล่องข้อความ เพื่อความแม่นยำในการค้นหาตำแหน่งของข้อความที่ปรากฏบนภาพ โดยภาพในปริภูมิสีระดับความเข้มเทาจากขั้นตอนก่อนหน้า (รูปที่ 2-13 (ก)) จะถูกกำจัดขอบกล่องข้อความออกโดยเริ่มจากการกำจัดสัญญาณรบกวนด้วยการแปลงภาพให้เป็นขาวดำโดยใช้ GaussianC [13] ทำให้ได้ผลลัพธ์ดังรูปที่ 2-13 (ข) ตามด้วย



รูปที่ 2-13 กระบวนของการกำจัด

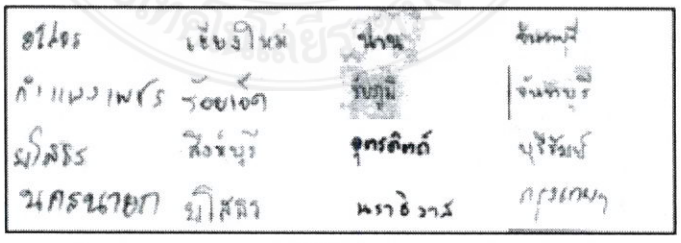
การใช้ CannyEdge [14] ในการหาขอบจะทำให้ได้ผลลัพธ์ดังรูปที่ 2-13 (ค) แล้วจึงทำ Morphological Gradient ได้ผลลัพธ์ดังรูปที่ 2-13 (ง) ตามด้วยการกร่อนภาพด้วย Erosion ดังรูปที่ 2-13 (จ) จากนั้นทำ Inverse ภาพที่ได้ต่อด้วยการทำ Threshold อีกครั้งแล้วจึงทำการค้นหา Contour เพื่อให้ได้ตำแหน่งของกล่องข้อความ ตำแหน่งข้อความที่ได้นี้จะถูกนำไปวาดเส้นสีขาวทับลงบนภาพต้นฉบับจะทำให้กล่องข้อความถูกเส้นสีขาวทับหายไปเหลือเพียงอักษรที่ต้องการบนเอกสาร ดังรูปที่ 2-13 (ฉ)



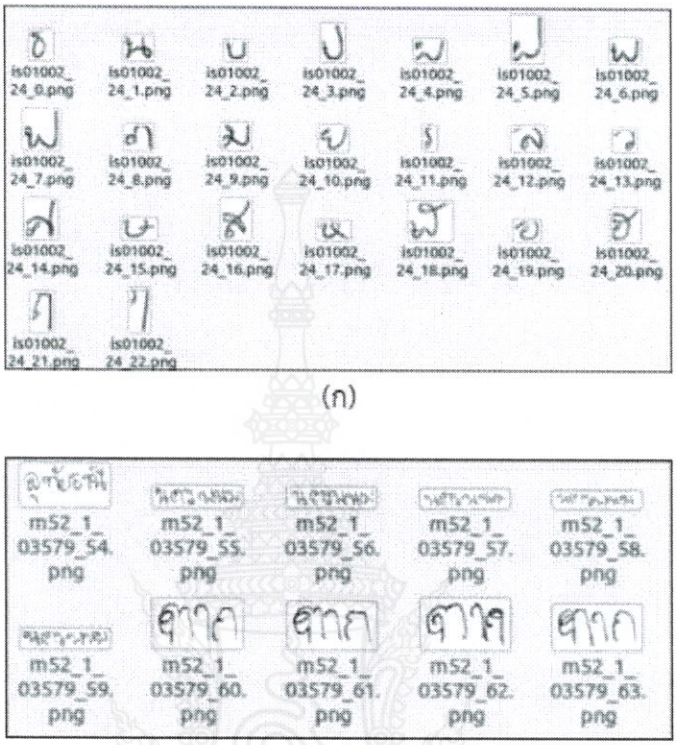
รูปที่ 2-14 ขั้นตอนการเตรียมภาพอักษรสำหรับทดสอบ

3) การตัดตัวอักษรเพื่อเตรียมข้อมูลสำหรับทดสอบ โดยจะเริ่มด้วยการสร้าง Mask สำหรับบอกตำแหน่งของตัวอักษรที่ปรากฏบนภาพ สามารถทำได้โดยการสร้าง Mask โดยภาพผลลัพธ์จากขั้นตอนก่อนหน้าจะถูกทำภาพให้เป็นขาวดำโดยใช้ GaussianC ซึ่งจะได้ผลลัพธ์ดังรูปที่ 2-14 (ก) จากนั้นจะถูกทำให้เบลอด้วย Guassian Blur ดังรูปที่ 2-14 (ข)เมื่อนำรูปที่ 2-14 (ข) มาลบกับรูปที่ 2-14 (ก) จะได้รูปที่ 2-14 (ค) ซึ่งจะถูกนำไปดำเนินการทางสัณฐานวิทยา (Morphological Operation) ได้ผลลัพธ์ดังรูปที่ 2-14 (ง) ที่แสดงตำแหน่งของตัวอักษรที่ปรากฏบนภาพ และเมื่อนำมาหาตำแหน่งของภาพจะได้ผลลัพธ์ดังรูปที่ 2-14 (จ) ซึ่งจะสามารถถูกตัดออกมาได้โดยง่าย ดังตัวอย่างในรูปที่ 2-14 (ฉ)

ข้อมูลสำหรับการดำเนินการระดับคำข้อมูลภาพตัวอย่างระดับคำสำหรับทดสอบได้จากการตัด (Crop) จากชุดข้อมูลภาพชื่อจังหวัด 77 จังหวัด (คลาส) จากชุดข้อมูลสำหรับฝึกฝนการแข่งขัน NSC2019 [15] (WD200-1, WD200-2, WD200-3 และ WD200-4) และรวบรวมเพิ่มเติมจากลายมือของนักเรียนระดับชั้นมัธยมศึกษาตอนปลายของโรงเรียนแห่งหนึ่งในจังหวัดศรีสะเกษ คลาสละ 70 คน รวมทั้งหมด 5,390 ตัวอย่าง รูปที่ 2-15 แสดงตัวอย่างของข้อมูลบางส่วนที่ถูกรวบรวมมา ตัวอย่างภาพหลังจากกระบวนการเก็บรวบรวมข้อมูลและตัดแบ่งแสดงดังรูปที่ 2-16 ซึ่งจะนำเข้าสู่กระบวนการ Pre-Processing ต่อไป

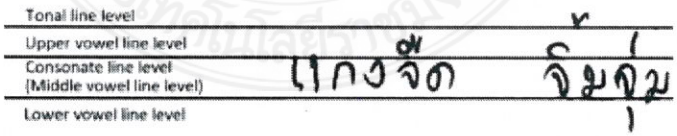


รูปที่ 2-15 ตัวอย่างภาพที่ใช้ในการทดสอบและฝึกฝนโมเดล

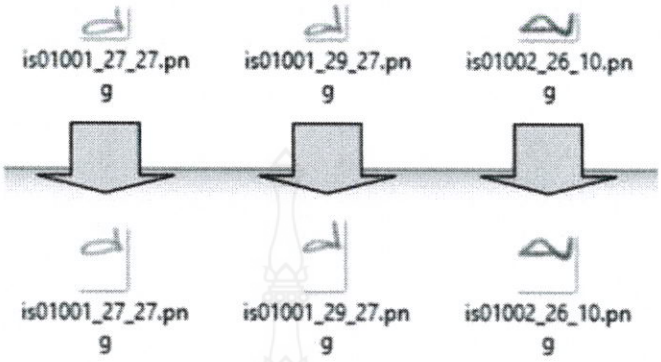


รูปที่ 2-16 ตัวอย่างภาพที่ได้จากการตัดแบ่ง (ก) ระดับตัวอักษร (ข) ระดับคำ

กระบวนการ Pre-Processing เนื่องจากลักษณะการเขียนประโยคในภาษาไทยมีการแบ่งระดับชั้นของอักขระออกเป็น 4 ระดับ ดังรูปที่ 2-17 และอักขระบางตัวเช่นในคำว่า "กล้า" กับคำว่า "น้ำ" จะเห็นว่าไม้โท *' อาจอยู่ได้ทั้งใน Tonal Line Level หรือ Upper Vowel Line Level ในได้แบ่งชั้นของอักขระออกเป็น 3 กลุ่ม ชั้นดังตารางที่ 2-1 และเนื่องด้วยอักขระ ข และ ค ไม่ปรากฏในพจนานุกรมราชบัณฑิตยสถาน จึงได้ตัดออกจากการวิจัย ภาพของแต่ละตัวอักษรที่ถูกตัดออกมาจะถูกดำเนินการเพิ่มพื้นที่ว่างเพื่อแยกกลุ่มอักษร ซึ่งจะได้ผลลัพธ์ดังรูปที่ 2-18



รูปที่ 2-17 ระดับอักขระที่ปรากฏในภาษาไทย



รูปที่ 2-18 การเติมพื้นที่ว่าง เพื่อแยกตัวอักษรกลุ่ม 0 (สระอี)

ตารางที่ 2-2 ตัวอักษรในแต่ละกลุ่มชั้น

กลุ่ม	ตัวอักษร
0	็, ่, ้, ๊, ๋, ิ, ี, ึ, ื, ุ, ู, ั
1	'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'ก', 'ข', 'ค', 'ฆ', 'ง', 'จ', 'ฉ', 'ช', 'ซ', 'ฌ', 'ญ', 'ฎ', 'ฏ', 'ฐ', 'ฑ', 'ฒ', 'ณ', 'ด', 'ต', 'ถ', 'ท', 'ธ', 'น', 'บ', 'ป', 'ผ', 'ฝ', 'พ', 'ฟ', 'ภ', 'ม', 'ย', 'ร', 'ล', 'ว', 'ศ', 'ษ', 'ส', 'ห', 'ฬ', 'อ', 'ฮ', 'ฯ', 'ะ', 'า', 'เ', 'โ', 'ใ', 'ไ', 'ๆ', 'ฤ', 'ฦ'
2	์, ็, ่, ้, ๊, ๋, ิ, ี, ึ, ื, ุ, ู, ั

การพัฒนาโมเดลที่ใช้ในการรู้จำโมเดลที่ใช้ในการรู้จำประกอบด้วยสองส่วน คือ ส่วนที่ใช้สำหรับสกัดลักษณะเด่นออกจากภาพข้อความ โดยเลือกใช้โครงข่ายประสาทเทียมแบบสังวัตนาการในการพัฒนาส่วนนี้ และส่วนที่สอง คือ ส่วนที่ใช้สำหรับรู้จ้าตัวอักษรจากลักษณะเด่นที่ได้จากส่วนแรกซึ่งได้เลือกใช้โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network; ANN) แบบ LSTM ผลลัพธ์ที่ได้จากโมเดลจะถูกนำไปทำ Post-Processing เพื่อเพิ่มความถูกต้องด้วย Word Beam Search Algorithm ดังรูปที่ 2-19 โมเดลทั้งสองถูกพัฒนาด้วยภาษาไพธอน (Python 3) ร่วมกับการใช้เฟรมเวิร์คเทนเซอร์โฟล์ (Tensorflow-gpu 2.0.0) และโอเพนซีวี (OpenCV4.1.2) เขียนโค้ดในรูปของเท็กซ์โหมดแอปพลิเคชัน (Text-Mode Application)

กระบวนการฝึกฝนและทดสอบดำเนินการบนระบบปฏิบัติการวินโดวส์ (Windows 10 Pro) CPU: Intel Core i5 (3.1GHz/gen4) Ram: DDR3-8GB VGA: NVIDIA RTX2060 SSD: 240GB

ผลการทดสอบการรู้จำกับภาพนำเข้าระดับอักษรการทดสอบการรู้จำกับภาพนำเข้าระดับอักษร ซึ่งทดลองกับภาพอักษรโดดๆ ขนาด 32 x 32 พิกเซล ที่มีความแตกต่างกัน 78 รูปแบบ

(คลาส) จำนวน 7,8000 ตัวอย่าง ทำการฝึกฝนโมเดลจำนวน 2,500 รอบ (Epochs) ซึ่งจะพบว่าโมเดล A และ B มีค่าความถูกต้องใกล้เคียงกัน แสดงดังตารางที่ 2-2 (คอลัมน์ที่ 2) โดยโมเดล B มีค่าความถูกต้องเฉลี่ยสูงกว่าโมเดล A เพียงแค่ร้อยละ 0.24 ส่วนโมเดล C ที่มีการบังคับทำ Regularization ด้วยการทำ Batch Normalization สามารถให้ค่าความถูกต้องเพิ่มขึ้นจากโมเดลได้อีกถึงร้อยละ 2.81 และเมื่อนำโมเดล C มาป้อนด้วยภาพที่ผ่านกระบวนการแบ่งกลุ่มระดับอักษรร่วมกับการคงอัตราส่วน [16] จะสามารถทำให้ผลลัพธ์เฉลี่ยเพิ่มได้อีกร้อยละ 4.13 ดังตารางที่ 2-2

ตารางที่ 2-3 ผลการทดสอบประสิทธิภาพโมเดลการรู้จำกับภาพนำเข้าระดับอักษร

โมเดล	ค่าเฉลี่ยร้อยละความถูกต้อง		ส่วนต่าง
	ไม่ผ่านการแบ่งกลุ่มระดับอักษรร่วมกับการคงอัตราส่วน	ผ่านการแบ่งกลุ่มระดับอักษรร่วมกับการคงอัตราส่วน	
A	85.32	89.42	+4.10
B	85.56	89.23	+3.67
C	88.37	92.50	+4.13

ผลการทดสอบการรู้จำกับภาพนำเข้าระดับคำการทดสอบการรู้จำกับภาพนำเข้าระดับคำ (ไม่ทำกาการแยกอักษร) ขนาดภาพ 400 x 80 พิกเซล ที่เขียนด้วยลายมือเขียนภาษาไทยซึ่งเป็นชื่อจังหวัดในประเทศไทยทั้ง 77 จังหวัด (คลาส) ที่ได้รวบรวมมา โดยได้ดำเนินการฝึกฝนโมเดลในการรู้จำข้อความจำนวน 1,0000 รอบด้วยการแบ่งข้อมูลสำหรับฝึกฝนและทดสอบออกเป็น 90 : 10 ทำให้ได้ข้อมูลสำหรับฝึกฝนจำนวน 4,851 ตัวอย่าง และข้อมูลสำหรับหรับทดสอบจำนวน 539 ตัวอย่าง ได้ผลการทดสอบประสิทธิภาพของโมเดลการรู้จำ ดังตารางที่ 2-3 และ 2-4 ซึ่งพบว่า โมเดลจะให้ค่าความถูกต้องเมื่อใช้ภาพข้อมูลนำเข้าในรูปแบบภาพความเข้มเทาที่สูงกว่าการใช้ภาพข้อมูลนำเข้ารูปแบบภาพขาวดำ

ตารางที่ 2-4 ค่าความถูกต้องของโมเดลและหลังทำ Post-Processing ในการรู้จำกับภาพนำเข้าระดับคำ

โมเดล	ภาพขาวดำ				ภาพความเข้มเทา			
	คงอัตราส่วน		ไม่คงอัตราส่วน		คงอัตราส่วน		ไม่คงอัตราส่วน	
	word	letter	word	letter	word	letter	word	letter
without wbs	89.05	92.39	73.46	79.38	94.99	95.92	93.69	97.00
with wbs	96.66	96.58	96.66	96.41	98.14	98.40	99.62	99.77
ผลต่าง	7.61	4.19	23.20	17.03	3.15	2.48	5.93	2.77

(wbs คือ word beam search)

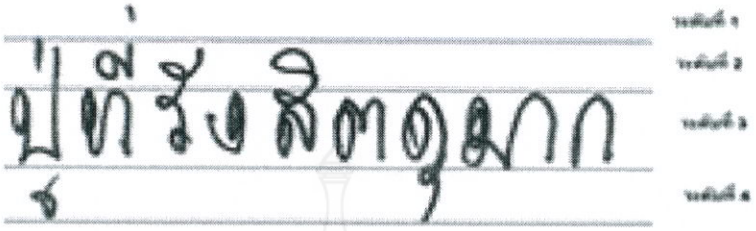
ตารางที่ 2-5 เวลาในการประมวลผลกับข้อมูลสำหรับทดสอบในการรู้จำกับภาพนำเข้าระดับคำ

โมเดล	ภาพขาวดำ		ภาพความเข้มเทา	
	คงอัตราส่วน	ไม่คงอัตราส่วน	คงอัตราส่วน	ไม่คงอัตราส่วน
without wbs	0.0190	0.0160	0.0158	0.0154
with wbs	0.8076	0.3949	0.2862	0.4685
ผลต่าง	0.7886	0.3789	0.2704	0.4531

(หน่วย คือ วินาที)

วิเชษฐ์รจน์ และรัชฎา (2556) [17] กล่าวถึง การรู้จำภาพตัวอักษรลายมือเขียน (handwriting recognition) เป็นงานวิจัยขั้นสูงต่อจากการรู้จำภาพตัวอักษรพิมพ์ (optical character recognition) โดยมีวัตถุประสงค์เพื่อช่วยให้ผู้ใช้สามารถจัดเก็บเอกสารที่มีอยู่เข้าสู่ระบบคอมพิวเตอร์ได้สะดวกขึ้น ซึ่งนอกจากจะเป็นเอกสารพิมพ์แล้ว ยังมีเอกลารจำนวนมากที่อยู่ในรูปของเอกสารลายมือเขียน เช่น ข้อมูลที่ผู้ใช้กรอกใบสมัครงาน รายงานการประชุม ข้อความจากการบรรยาย เป็นต้น งานวิจัยทางด้านการรู้จำภาพตัวอักษรลายมือเขียนในภาษาไทย (Phokharatkul and Kimpan, 2002) [18] เสนอวิธีการรู้จำตัวอักษรลายมือเขียนในภาษาไทยโดยใช้ตัวอธิบายรูปร่างแบบฟูริเยียร์ (Fourier descriptor) เป็นตัวอธิบายลักษณะเด่นของตัวอักษร จากนั้นฝึกฝนและรู้จำโดยใช้โครงข่ายประสาทเทียมแบบหลายชั้น ซึ่งค่าน้ำหนักของโหนดถูกพิจารณาโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (genetic algorithm) ผลการรู้จำเมื่อทดลองกับตัวอักษรลายมือเขียนไทยแบบบรรจงจำนวน 13,500 ตัวอักษร จากผู้ทดลอง 60 คน เท่ากับร้อยละ 99.12 แต่อย่างไรก็ตามตัวอักษร ลายมือเขียนที่ใช้เป็นแบบเขียนบรรจง ซึ่งไม่สอดคล้องกับการใช้งานจริง ต่อมา Methasate และ Sae-tang (2004) เสนอวิธีการจัดกลุ่มตัวอักษรลายมือเขียนด้วยการพิจารณาโครงสร้างของตัวอักษรที่คล้ายกัน ลักษณะที่นำมาพิจารณา ได้แก่ เส้นตัวอักษรในแนวตั้ง (vertical stroke) และการกระจายของพิกเซลภาพตัวอักษร (pixel distribution) จากนั้นนำมาฝึกฝนและรู้จำโดยใช้โครงข่ายประสาทเทียมแบบแพร่กระจายกลับ (back propagation neural network) ผลการวิจัยลามารถแบ่งกลุ่มตัวอักษรออกได้เป็น 21 กลุ่ม มีระดับความถูกต้องที่ร้อยละ 97.60 นอกจากนี้ Nopsuwanchai และคณะ (2006) [19] ได้ใช้ส่วนประกอบสำคัญ (principal component) ของตัวอักษรแต่ละตัว ร่วมด้วยรูปแบบต่างๆ ของตัวอักษร ได้แก่ ภาพกลับขั้วของตัวอักษร (polar transformed image) และภาพตัวอักษรหมุนขวา 90 องตา เป็นลักษณะเด่นของตัวอักษร จากนั้นนำไปฝึกฝนและรู้จำโดยใช้ แบบจำลองฮิดเด้นมาร์คอฟ โดยเปรียบเทียบระหว่างคลังข้อมูล 2 ชุดได้ผลการรู้จำถูกต้องร้อยละ 95.98 และ 95.13 ตามลำดับ

เนื่องจากระดับของการเขียนในภาษาไทยมีอยู่ 4 ระดับ ดังแสดงในรูปที่ 2-19 คือ ระดับวรรณยุกต์ ระดับสระบน ระดับพยัญชนะ/สระกลาง และระดับสระล่าง ดังนั้นโอกาลที่ตัวอักษรจะเขียนสัมผัสกันสามารถเป็นไปได้ทั้งแนวตั้ง เช่น ระหว่างระดับที่ 2 กับระดับที่ 3 ได้แก่ "รั และ "สิ" และในแนวนอน เช่น "มา" ดังนั้นการแยกตัวอักษรที่สัมผัสกัน เราควรทราบก่อนว่าแนวพยัญชนะกลางอยู่ตำแหน่งใด จากนั้นมาพิจารณาว่าส่วนของตัวอักษรตัวใดสัมผัสกันบ้าง โดยใช้หลักการต่อไปนี้



รูปที่ 2-19 ระดับการเขียนในภาษาไทย [13]

เมื่อเรานำภาพตัวอักษรมาหากรอบของวัดถุเรียบร้อยแล้ว เราจะใช้ตำแหน่งมุมบนซ้ายสุดของกรอบตัวอักษรเป็นตัวแทนจุดสูงสุดของแต่ละส่วน ดังแสดงด้วยรูป "*" สีน้ำเงินในรูปที่ 2-20ก จากนั้นลากเส้นตรงเชื่อมระหว่างจุดที่ได้ นำมาเข้าสมการการวิเคราะห์การถดถอยเชิงเส้น จากนั้นปรับความชันของเส้นที่ได้ให้เท่ากับ 0 จะได้เป็นเส้นชมพู ดังแสดงในรูปที่ 2-20 ข ซึ่งเส้นดังกล่าวเป็นเส้นบนของระดับกลาง ในทำนองเดียวกัน เราใช้ตำแหน่งมุมล่างขวาสุดของกรอบตัวอักษรเป็นตัวแทนจุดต่ำสุด

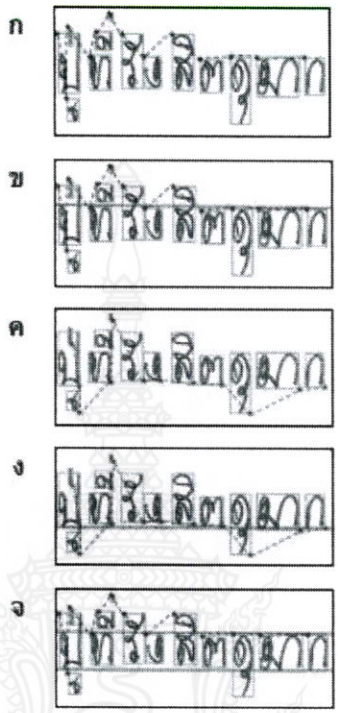
สมการการวิเคราะห์การถดถอยเชิงเส้น

$$y = bx + a \qquad \text{(สมการที่ 1)}$$

หาความสัมพันธ์ระหว่างตัวแปรทั้งสอง
ซึ่งมีสูตรในการหา a และ b ดังนี้

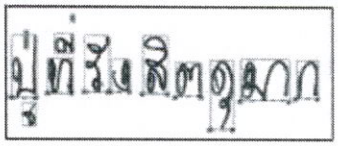
$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \qquad \text{(สมการที่ 2)}$$

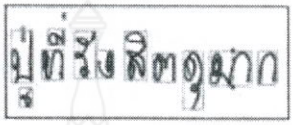
$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \qquad \text{(สมการที่ 3)}$$



รูปที่ 2-20 การวิเคราะห์ตัวอักษรที่สัมผัสกันในแนวตั้ง

การวิเคราะห์ตัวอักษรที่สัมผัสกันในแนวนอน (โดยใช้ความกว้างมัธยฐานของตัวอักษร) นำภาพตัวอักษรที่มีกรอบของวัตถุ เราจะใช้ตำแหน่งมุมล่างซ้ายสุดและขวาสุดของกรอบตัวอักษรเป็นตัวแทนความกว้างของแต่ละวัตถุ ดังแสดงด้วยรูป “*” สีน้ำเงินในรูป 2-21 ก ได้ความกว้างของแต่ละวัตถุจากภาพข้อความ เก็บค่าของแต่ละวัตถุไว้ในอาร์เรย์ แล้วนำค่าที่ได้จากตัวอักษรแต่ละตัวมาเรียงกันแล้วนำเข้าสมการหาค่ามัธยฐาน (median) ของข้อมูลวัตถุบนภาพข้อความ จากนั้นนำค่ามัธยฐานที่ได้มาเทียบตัวอักษรแต่ละตัว ถ้าพบว่ามีความกว้างมากกว่าเท่าครึ่งหรือ 11% ซึ่งเกินค่ามัธยฐานแสดงว่าวัตถุนั้นเป็นตัวอักษรที่ติดกันในแนวนอน จากนั้นทำการลากเส้นเชื่อมจุดตัดที่มีค่าเท่ากับค่ามัธยฐานของตัวอักษรดังแสดงด้วยรูป "*" สีแดงในรูป 2-21 ข

ก 

ข 

รูปที่ 2-21 การวิเคราะห์ตัวอักษรที่สัมผัสกันในแนวนอน

การหาค่ามัธยฐานของข้อมูลที่มีจำนวนคี่
สมการหาตำแหน่งค่ามัธยฐาน

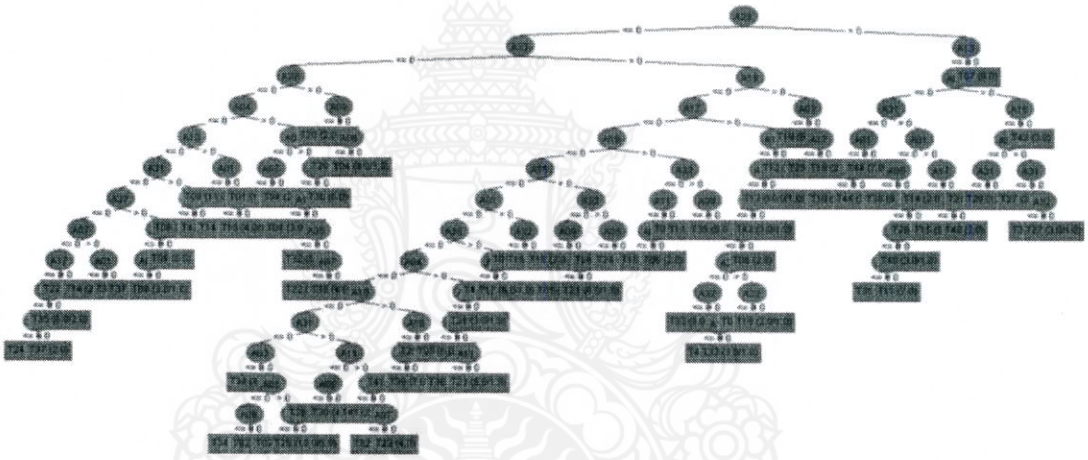
$= (n + 1) * 2$ (สมการที่ 4)

การหาค่ามัธยฐานของข้อมูลที่มีจำนวนคู่
สมการหาค่าเฉลี่ยของข้อมูลที่อยู่ในตำแหน่งที่
$= n \div 2$ และ $(n + 1) \div 2$ (สมการที่ 5)

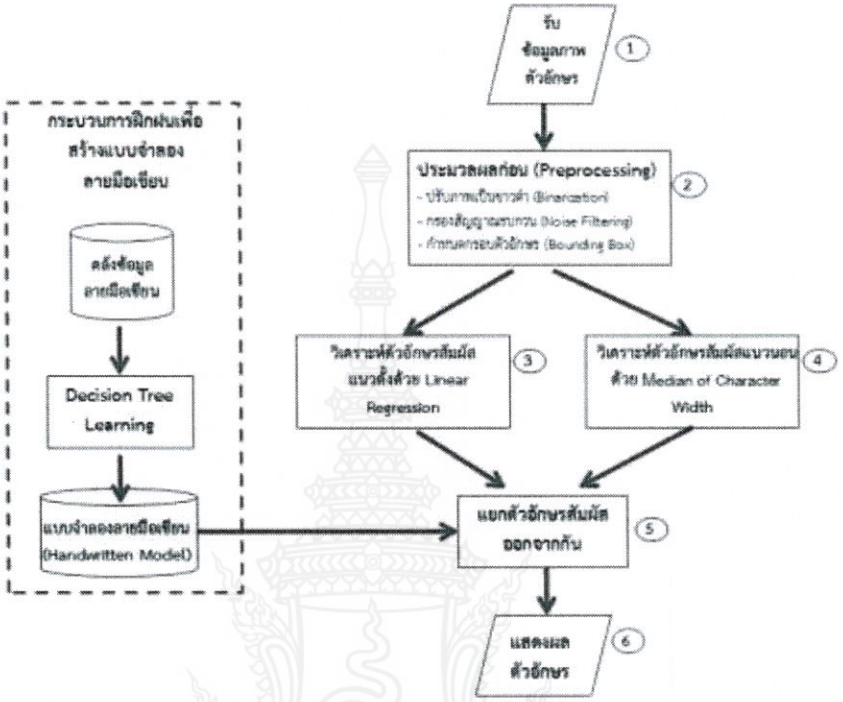
ต้นไม้ตัดสินใจ (decision tree classifiers, J48) เป็นโครงสร้างต้นไม้ที่ประกอบไปด้วยโหนดตัดสินใจ (decision nodes) แล้วเชื่อมต่อกันด้วยสาขา (branches) โดยขยายจากโหนดราก (root node) มายังโหนดใบ (leaf nodes) ค่าคุณสมบัติ (attribute) ที่ใช้สำหรับเปรียบเทียบเงื่อนไขจะอยู่ในโหนดตัดสินใจ ซึ่งผลลัพธ์การเปรียบเทียบจะอยู่ในสาขา (Branches) และในแต่ละสาขาจะนำไปสู่โหนดตัดสินใจอื่น หรือโหนดใบ ซึ่งโหนดใบจะเก็บผลลัพธ์ของการจำแนกไว้ (Quinlan,1986;Quinlan, 1990) โดยกำหนดให้ภาพตัวอักษร ก-ฮ เป็นภาพจำลองที่ได้จากลายมือเขียนของกลุ่มข้อมูลตัวอย่าง ตัวอย่างละ 10 ตัวอักษร ตรวจเช็คคุณลักษณะเด่นตัวอักษรภาษาไทยแต่ละตัว นำคุณลักษณะตัวอักษรแทนข้อมูลเวกเตอร์ (eigenvector) ทำการแปลงโครงสร้างเมตริกซ์ ข้อมูลไปเป็นเวกเตอร์แถว คำนวณหาไอเกนเวกเตอร์ที่สอดคล้องกันกับค่าไอเกน โดยนำผลรวมของคุณลักษณะต่างๆ ของตัวอักษรที่ซ้ำกันนำมาเรียงลำดับไอเกนเวกเตอร์ที่สอดคล้องกับค่าไอเกนจากมากไปน้อย แล้วดัดเอาเฉพาะค่าที่มีคุณลักษณะเด่น เข้าสมการหาค่า eigenvalue ดังตารางที่ 2-5 จากนั้นแล้วนำเอาลักษณะเด่นของภาพ (feature extraction) ทุกตัวอักษรเพื่อเข้าโปรแกรม Weka สร้างต้นไม้ตัดสินใจของตัวอักษรลายมือเขียน ก-ฮ ดังรูปที่ 2-22 เพื่อทำการรู้จำของตัวอักษรต่อไป

ตารางที่ 2-6 การหาคุณลักษณะของตัวอักษร “ฌ”

คุณลักษณะตัวอักษร ฌ	N Loop ซ้ายบน	N Loop ขวาบน	N Loop ขวาล่าง	ลูป 2	Loop ช่อง 7	Loop ช่อง 8	เปิดบน 2	เปิดล่าง 2	เปิดบนขวา	N Loop ซ้ายล่าง	เปิดล่างซ้าย	เปิดล่าง	Loop ช่อง 4	หัวหยักบน
code	A05	A06	A08	A10	A17	A18	A33	A31	A26	A07	A27	A04	A14	A30
รวม	10	10	10	10	10	10	10	9	8	7	6	2	1	1
ค่า Eigen vector	0.2688	0.2688	0.2688	0.2688	0.2688	0.2688	0.2688	0.24	0.215	0.188	0.16	0.05	0.027	0.027



รูปที่ 2-22 ผลลัพธ์ต้นไม้ตัดสินใจของตัวอักษรลายมือเขียน ก-ฮ



รูปที่ 2-23 ขั้นตอนการทำงานของระบบ

ภาพรวมของระบบและขั้นตอนการทำงานมีวิธีการดำเนินงานดังนี้

1 รับข้อมูลภาพตัวอักษรเข้าสู่ระบบ

2 ประมวลผลก่อนโดยปรับภาพสีของตัวอักษรบนรูปภาพให้เป็นตัวอักษรสีขาวพื้นหลังสีดำ (binanzation) นำมากรองสัญญาณรบกวนออกจากภาพขาวดำ (noise filtering) ทำการตีกรอบรอบแต่ละวัตถุบนรูปภาพ (bounding box)

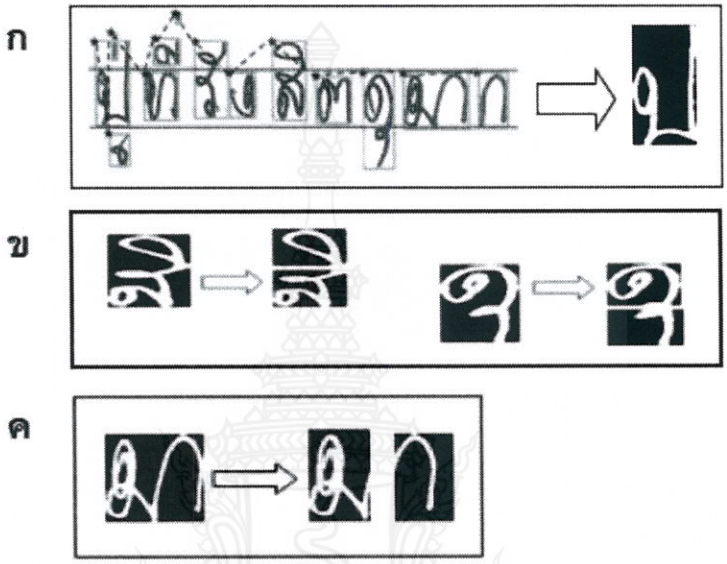
3 วิเคราะห์ตัวอักษรที่สัมผัสกันในแนวตั้งด้วยเส้นจากการวิเคราะห์ถดถอยเชิงเส้น

4 การวิเคราะห์ตัวอักษรที่สัมผัสกันในแนวนอนด้วยความกว้างมัธยฐานของตัวอักษร (median of character width)

5 นำแบบจำลองลายมือเขียน (hand written model) มาแยกตัวอักษรที่สัมผัสออกจากกัน ด้วยการใช้กฎที่ได้จากต้นไม้ตัดสินใจ

6 แสดงผลตัวอักษร นำวัตถุบนภาพได้จากการ bounding box ไปทำการตรวจสอบระดับพยัญชนะด้วยเส้นจากการวิเคราะห์ถดถอยเชิงเส้นบนพยัญชนะและเส้นจากการวิเคราะห์ถดถอยเชิงเส้นล่างพยัญชนะ ถ้าวัตถุนั้นมีความสูงมากกว่าความสูงของเส้นจากการวิเคราะห์ถดถอยเชิงเส้นทั้งสองเส้นจะได้วัตถุตัวติดหรืออักษรเดี่ยวหางยาวดังรูปที่ 2-24 ก ไม่ต้องนำไปตัด ส่วนอักษรที่สัมผัสกันในแนวตั้ง

ทำการตัดด้วยเส้นจากการวิเคราะห์ถดถอยเชิงเส้นดังรูปที่ 2-24 ข และตัวอักษรที่สัมผัสกันในแนวนอนตัดด้วยความกว้างมัธยฐานของตัวอักษรจะได้อักษรแต่ละตัวออกมาดังรูปที่ 2-24 ค



รูปที่ 2-24 แสดงผลตัวอักษร

ผลการวิจัยและวิเคราะห์ในการนำตัวอักษรลายมือเขียนมาเทียบด้วยต้นไม้ตัดสินใจ ซึ่งมีความรู้จำน้อยกว่าตัวอักษรพิมพ์หรือตัวอักษรที่คัดตัวบรรจงทั้งหมด จากการวิเคราะห์พบว่าตัวอักษรลายมือเขียนบางตัวมีคุณลักษณะไม่ครบเหมือนตัวพิมพ์ เช่น การเขียน ร, ล, ย, พ, บ ไม่มีหัว หรือมีหางยาวเกินกำหนด จึงทำให้การรู้จำคลาดเคลื่อนลดลงได้การนำตัวอักษรลายมือเขียนที่ตัดได้ตรวจสอบคุณลักษณะก่อนว่าตัวอักษรแต่ละตัวมีคุณลักษณะอะไรบ้าง เช่น "ม" เป็นลักษณะตัวอักษรลายมือเขียน มีคุณลักษณะเปิดบนขวา ตัวอักษรสั้น ไม่มีหัวแตก มีลูปช่องที่ 1, 4 และช่องที่ 7 เมื่อนำมาเทียบด้วยต้นไม้ตัดสินใจของตัวอักษรลายมือเขียน ก-ฮ ที่เตรียมไว้ ผลลัพธ์ที่ได้คือ T33 เท่ากับอักษร "ม" ส่วนตัวอักษรที่ติดด้านหลังตัวอักษร "ม" นำตัวอักษรนั้นมาตรวจสอบคุณลักษณะด้วยต้นไม้ตัดสินใจตัวติดด้านหลังที่เตรียมไว้เพื่อให้ทราบว่ามีคุณลักษณะเหมือนตัวอักษรใด พบว่าเป็นลักษณะตัวอักษรลายมือเขียน มีคุณลักษณะเปิดล่าง ตัวอักษรสั้น ไม่มีหัวแตก ไม่มีลูป ผลลัพธ์ที่ได้คือ T46 เท่ากับอักษรสระอา (-า) ผลลัพธ์การทดลองตัดตัวอักษรลายมือเขียนที่ผ่านการสแกนมีจำนวนทั้งหมด 1,234 ตัว ใช้ Matlab 2010 ทำการแยกตัวอักษรได้ 1,116 ตัว คิดเป็นร้อยละ 90.44 ตามตารางที่ 2-6 ดังนี้

ตารางที่ 2-7 ผลลัพธ์การทดลองตัดตัวอักษรลายมือเขียนจำแนกตามพยัญชนะ สระและวรรณยุกต์

ตัวอักษร	จำนวนทั้งหมด	จำนวนที่แยกได้	คิดเป็นร้อยละ
พยัญชนะ	772	701	90.80
สระล่าง	59	54	91.53
สระข้าง	124	115	92.74
สระบน	225	197	87.56
วรรณยุกต์	54	49	90.74
รวม	1,234	1,116	90.44

บทที่ 3
วิธีการดำเนินการวิจัย

วิธีการดำเนินงานวิจัยระบบตรวจจับป้ายทะเบียนรถยนต์จะแบ่งหัวข้อวิธีการดำเนินงานวิจัยออกเป็น 5 กระบวนการดังนี้

1 การออกแบบระบบฮาร์ดแวร์

2 การออกแบบระบบซอฟต์แวร์

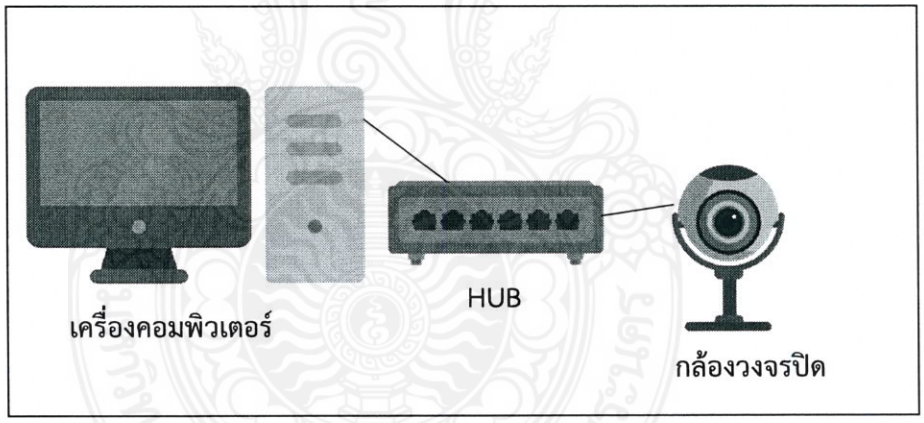
3 การเก็บภาพข้อมูลตัวอย่าง

4 การสร้างชุดข้อมูลเรียนรู้

5 การตรวจจับป้ายทะเบียนและเรียนรู้จากตัวอักษร

3.1 การออกแบบระบบฮาร์ดแวร์

ผู้วิจัยได้ทำการสร้างแบบจำลองระบบฮาร์ดแวร์ด้วยการใช้เครื่องคอมพิวเตอร์สำหรับจัดเก็บฐานข้อมูลชุดข้อมูลรูปภาพตัวหนังสือ ก-ฮ สระ และตัวเลข 0-9 พร้อมกับจำลองฐานข้อมูลทะเบียนรถที่อนุญาตให้ระบบเปิดประตูไม้กระดก มีอุปกรณ์กล้องสำหรับการตรวจจับทะเบียนรถยนต์ Hub



รูปที่ 3-1 สถาปัตยกรรมของระบบ

3.2 การสร้าง Thai Recognition

3.2.1 การสร้าง Model

ข้อมูลสำหรับการทำ Model นี้จะมีข้อมูล 2 ส่วน ตามนี้

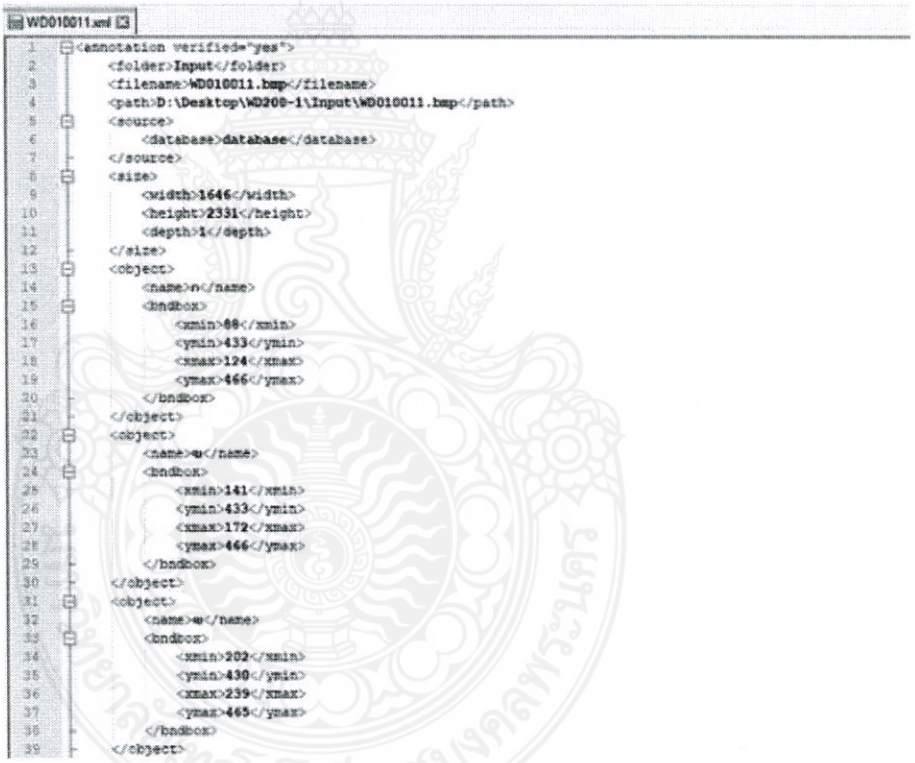
1. Public Data
2. Generate Data

โดยข้อมูลแต่ละแบบก็จะทำการ Label ข้อมูลก่อนว่าในรูปภาพนั้น ตัวอักษรไหนเป็น Class อะไร จากนั้นเมื่อ Label เสร็จแล้วก็จะได้ XML File ของแต่ละรูปภาพที่บอกว่าแต่ละตำแหน่ง (xmin, ymin) , (xmax, ymax) ของแต่ละรูปเป็น Class อะไร

จากนั้นก็ทำการ Crop รูปภาพตามแต่ละ Class ด้วยการอ่าน XML File ที่ระบุตำแหน่งไว้ออกมาแยกกันไว้เป็น Folder ละ Class

จากนั้นก็ทำ Data Augmentation โดยการ Rotation , Width Shifting , Height Shifting และการ Zoom เพื่อให้ได้ข้อมูลมากและหลากหลายเพียงพอสำหรับการสร้าง Model

การทำ Label Image ต่างๆ ที่ได้มาจาก Dataset จะได้ดังรูปที่ 3-2



รูปที่ 3-2 ตัวอย่าง XML ที่ได้จากการ Label

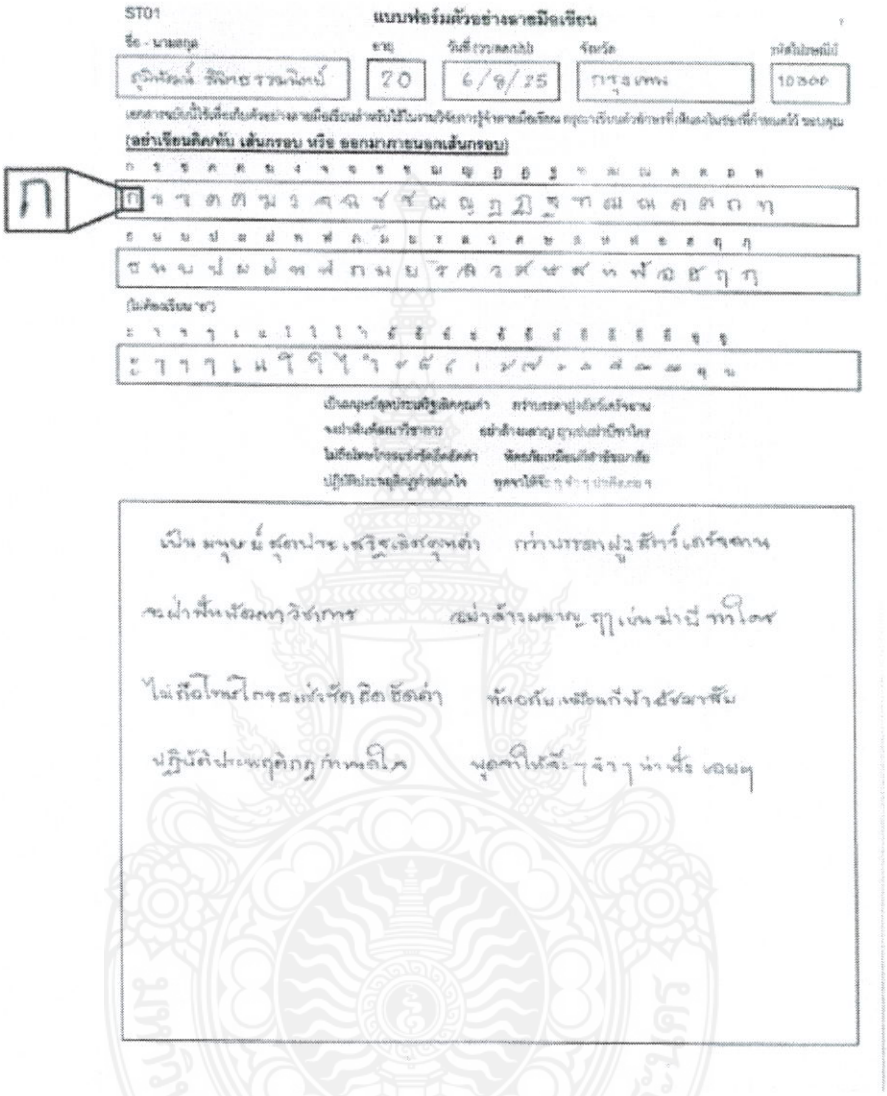
โดยในแต่ละ <object> จะประกอบไปด้วย

1. <name> : จะเป็น tag ที่บอกว่า object นี้เป็น Label Class ไหน
2. <bndbox> : จะเป็น tag ที่เก็บ (xmin,ymin) , (xmax,ymax) ของ Class นี้ไว้

3.2.1.1 Public Data

ผู้วิจัยได้นำข้อมูล Public Data มาจากเว็บไซต์ https://thailang.nectec.or.th /best/best2019-handwrittenrecognition-trainingset/?source=post_page-----4d1dbefedf55----------

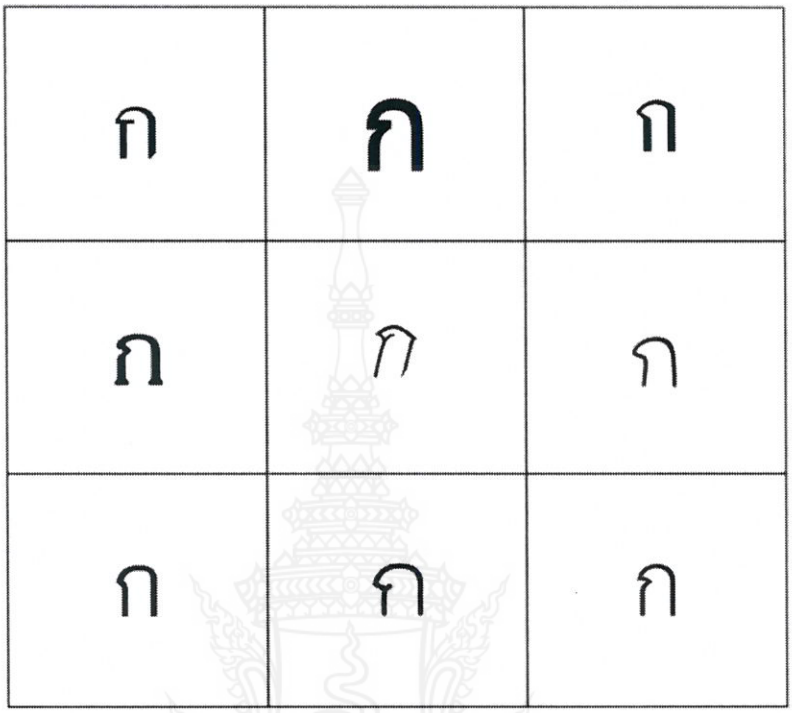
ซึ่งเมื่อลองเข้าไปดู Dataset ตัวนี้ก็จะเป็น Image Dataset ที่มีนามสกุลไฟล์ .bmp ออกมาหลาย Folder และนี่ก็คือตัวอย่างข้อมูลจาก Dataset ดังรูปที่ 3-3



รูปที่ 3-3 ข้อมูลตัวอย่างจาก NECTEC Dataset

3.2.1.2 Generate Data

สำหรับ Dataset ตัวนี้ Generate เพิ่มเติมมาจาก Microsoft Word โดยแบ่งแต่ละ Class เป็น 9 Grid โดยในแต่ละ Grid ใช้ Font ที่ต่างกัน เพื่อให้ได้ความหลากหลายของข้อมูลมากขึ้น ดังรูปที่ 3-4



รูปที่ 3-4 ข้อมูลจากการ Generate

3.2.2 Crop Image

การ Crop Image ต่างๆ ที่ได้มาจาก Dataset ทั้ง 2 ส่วน แต่จาก Step ก่อนหน้า โดย Class ต่างๆ ที่ทำการ Label ไว้นั้นอยู่ใน Format XML ซึ่งจะทำการแปลงเป็น Txt file ที่เก็บตำแหน่งต่างๆ ของแต่ละ Class ไว้ก่อน

3.2.2.1 สร้าง Function สำหรับการแปลง XML เป็น TXT

```
import xml.etree.ElementTree as ET
import glob
import numpy as np

def xml_to_txt(xml_folder, folder_out):
    xml_paths = glob.glob(xml_folder+'/*.xml')
    for i in  range(len(xml_paths)):
        print(xml_paths[i])
        tree = ET.parse(xml_paths[i])
        root = tree.getroot()

        size = root.find('size')
        print(size.text)
        img_width = int(size.find('width').text)
        img_height = int(size.find('height').text)

        yolos = []
        for obj in root.findall('object'):
            name = obj.find('name').text
            if name == 'helmet':
                name = 1
            xmin = int(obj.find('bndbox').find('xmin').text)
            ymin = int(obj.find('bndbox').find('ymin').text)
            xmax = int(obj.find('bndbox').find('xmax').text)
            ymax = int(obj.find('bndbox').find('ymax').text)
            yolo = f"{name} {xmin} {xmax} {ymin} {ymax}"
            yolos.append(yolo)

        outpath = folder_out+"/"+xml_paths[i].split("\\")[1].split(".")[0]+".txt"
        print('out_path', outpath)
        with open(outpath, mode='wt') as f:
            f.write('\n'.join(yolos))

if __name__ == "__main__":
    xml_to_txt('ST200-1 xml', 'ST200-1 label')
```

รูปที่ 3-5 Function สำหรับการแปลง XML เป็น TXT

3.2.2.2 สร้าง Function สำหรับการ Crop Image

```
import glob
import cv2

foldername = "ST200-1"
label_paths = glob.glob(foldername + " label/*.txt")

def crop_img(img, label_data):
    cls_name = label_data[0]
    xmin = int(label_data[1])
    xmax = int(label_data[2])
    ymin = int(label_data[3])
    ymax = int(label_data[4])
    img_crop = img[ymin:ymax, xmin:xmax]
    img_crop = cv2.resize(img_crop, (95, 95))
    return img_crop, cls_name

for label_path in label_paths:
    img_path = foldername+' img/'+ label_path.split('\\')[1].split('.')[0] + '.bmp'
    filename = label_path.split('\\')[1].split('.')[0]
    labels_data = []
    with open(label_path) as f:
        for row in f.readlines():
            labels_data.append(row.replace('\n', '').split(" "))
    img = cv2.imread(img_path)
    for idx, labels_data in enumerate(labels_data):
        img_crop, cls_name = crop_img(img, labels_data)
        out_folder = foldername+' img_out/'
        out_fname = cls_name+'_'+filename+'-'+str(idx)+'.jpg'
        savename = out_folder + out_fname
        print("crop img:", out_fname)
        cv2.imwrite(savename, img_crop)
```

รูปที่ 3-6 Function สำหรับการ Crop Image

เมื่อทำการ Crop Image ตาม Position ที่ได้จาก XML ตามรูปที่ 3-7

รูปที่ 3-7 ข้อมูลที่ทำการ Crop มาจาก NECTEC Dataset

3.2.3 Data Augmentation

จากกระบวนการทั้งหมดจะได้ Raw Data ที่เป็น Image ของ Thai Character ที่เป็นแบบลายมือ และแบบ Font ซึ่งถ้าจะทำเป็น Model ให้สามารถพร้อมรับกับการใช้งานจริง ก็จะพบว่า ตัวอักษร ที่แต่ละคนเขียนมานั้นมีขนาดไม่เท่ากัน มีความเอียงของตัวอักษรไม่เท่ากัน และอื่นๆ อีกมากมาย ดังนั้นจากข้อมูลที่มีสามารถนำมาทำ Augmentation ทั้งหมด 4 แบบ ดังนี้

1. Rotation
2. Width Shifting
3. Height Shifting
4. Zoom

3.2.3.1 สร้าง Function สำหรับการอ่าน Image

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from matplotlib.pyplot import imread, imshow, subplots, show , imsave
import glob

def read_img(img):
    full_path = img.split("/")
    path = full_path[0] + "/" + full_path[1]
    full_filename = full_path[2].split(".")
    filename = full_filename[0]
    file_extension = full_filename[1]
    image = imread(img)
    images = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    return images,filename,file_extension
```

รูปที่ 3-8 สร้าง Function สำหรับการอ่าน Image

3.2.3.2 สร้าง Function สำหรับการ Rotation

```
# 1. Rotation
def rotation(images):
  print("rotation process")
  data_generator = ImageDataGenerator(rotation_range=20)
  plot(images,data_generator,"rotation")
```

รูปที่ 3-9 สร้าง Function สำหรับการ Rotation

3.2.3.3 สร้าง Function สำหรับการ Width Shifting

```
# 2. Width Shifting
def width_shifting(images):
  print("width_shifting process")
  data_generator = ImageDataGenerator(width_shift_range=0.3)
  plot(images,data_generator,"width_shifting")
```

รูปที่ 3-10 สร้าง Function สำหรับการ Width Shifting

3.2.3.4 สร้าง Function สำหรับการ Height Shifting

```
# 3. Height Shifting
def height_shifting(images):
  print("height_shifting process")
  data_generator = ImageDataGenerator(height_shift_range=0.3)
  plot(images,data_generator,"height_shifting")
```

รูปที่ 3-11 สร้าง Function สำหรับการ Height Shifting

3.2.3.5 สร้าง Function สำหรับการ Zoom

```
# 4. Zoom
def zoom(images):
  print("zoom process")
  data_generator = ImageDataGenerator(zoom_range=[0.5, 1.5])
  plot(images,data_generator,"zoom")
```

รูปที่ 3-12 สร้าง Function สำหรับการ Zoom

3.2.3.6 สร้าง Function สำหรับการ Zip file

```
import zipfile
import os
import sys

def zipfolder(foldername, target_dir):
    zipobj = zipfile.ZipFile(foldername + '.zip', 'w', zipfile.ZIP_DEFLATED)
    rootlen = len(target_dir) + 1
    for base, dirs, files in os.walk(target_dir):
        for file in files:
            fn = os.path.join(base, file)
            zipobj.write(fn, fn[rootlen:])
```

รูปที่ 3-13 สร้าง Function สำหรับการ Zip file

3.2.3.7 สร้าง Function สำหรับการทำ Augmentation

```
read_path = "extract/*"
read_path_img = "/*.jpg"
zipname = 'thw_written'
output_path = "extract_out/"

folder_count = 0
for folder in glob.glob(read_path):
    folder_count += 1

# Read image all directory
for idx , folder in enumerate(glob.glob(read_path)):
    folder_name = folder.split("/")[1]

    print("Process folder {} / {} ".format(idx+1, folder_count))
    print("Process folder : " + folder_name)

    sub_path =  folder + read_path_img
    output_path = "extract_out/"
    output_path = output_path + folder_name

    file_count = 0
    for file in glob.glob(sub_path):
        file_count += 1

    for idxx , img in enumerate(glob.glob(sub_path)):
      print("Process img {} / {} ".format(idxx+1, file_count))
      images = read_img(img)
      rotation(images)
      width_shifting(images)
      height_shifting(images)
      zoom(images)

zipfolder(zipname, 'extract_out')
```

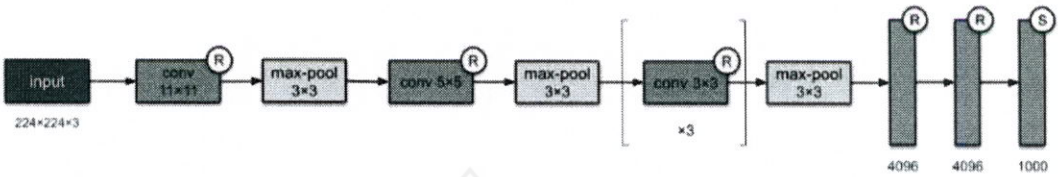
รูปที่ 3-14 สร้าง Function สำหรับการทำ Augmentation

จากนั้นเมื่อทำการ Execute แล้วจะได้ Zip file อยู่ที่ Path “extract_out/” พอทำการ Extract zip file แล้วก็จะได้ Image ที่ผ่านการทำ Augmentation ทั้ง 4 แบบ

3.2.3 Train Deep Learning Model

เริ่มสร้าง Model จาก Data ที่เตรียมมาทั้งหมดจะต้องทำการแยก Data แต่ละ Class ออกเป็น Folder ของใครของมันก่อน เพราะว่า Keras API มี Require 1 Class ต่อ 1 Folder นั่นเอง

AlexNet Architecture



รูปที่ 3-15 AlexNet Architecture

สำหรับการสร้าง Model จะใช้ AlexNet Architecture เป็นต้นแบบในการสร้าง Model นี้ โดยเจ้า AlexNet นี้จะประกอบไปด้วย

3.2.3.1 การทำ Feature Extraction

- Input Image Size 224 x 224 x 3(Chanel RGB)
- Convolution Layer (Filter Size 11x11 , Activation Function : ReLu)
- Max Pooling (Filter Size 3x3 ไปหยิบค่า Max ของ Conv. ก่อนหน้า)
- Convolution Layer (Filter Size 5x5 , Activation Function : ReLu)
- Max Pooling (Filter Size 3x3 ไปหยิบค่า Max ของ Conv. ก่อนหน้า)
- Convolution Layer (Filter Size 3x3 , Activation Function : ReLu , ทำ 3 รอบ)
- Max Pooling (Filter Size 3x3 ไปหยิบค่า Max ของ Convert ก่อนหน้า)

3.2.3.2 การ Classify ของ Model

- Fully Connected (Dense Layer 4096 Node, Activation Function: ReLu ทั้งหมด 2 Layer)
- Fully Connected (Dense Layer 1000 Node, Activation Function: SoftMax)

3.2.3.3 การทำ Feature Extraction

- Input Image Size 256 x 256 x 1(Grey Scale)

3.2.3.4 การ Classify Model

- Fully Connected (Dense Layer 128 Node, Activation Function: ReLu)
- Fully Connected (Dense Layer 55 Node, Activation Function: SoftMax)

จะเห็นได้ว่าการเปลี่ยนในส่วนของ Feature Extraction ในการรับ Input Image เป็น Grey Scale แทน และเปลี่ยนในส่วนของ Classify Model

ด้วยการตัด Fully Connected ออกไป 1 Layer และ Layer ที่เหลือลดจำนวน Node เหลือ 128 Node และ 55 Node ตาม Output Class

3.2.3.5 Model Training

จากข้างบนมีการอธิบายถึง Model ต้นแบบ และการปรับปรุง Model ต้นแบบสำหรับงาน

(1) Import lib ที่จะใช้งาน

```
%tensorflow_version 1.x

from google.colab import drive
drive.mount('/content/drive')

#ครั้งแรกสำหรับ unzip file on google drive
#!unzip -O utf-8 "/content/drive/DL/augment_multi.zip"

import keras
from keras.preprocessing.image import ImageDataGenerator, img_to_array
from keras.models import Sequential
from keras.layers import Dense, Dropout, Conv2D, MaxPooling2D, Flatten
from keras import backend as K

import numpy as np
from numpy.random import seed
import pandas as pd
import functools

import matplotlib.pyplot as plt
%matplotlib inline

import tensorflow as tf
tf.compat.v1.set_random_seed(2475)
print(tf.__version__)
```

(2) เตรียมข้อมูลสำหรับการ Train

```
img_size = 256
batch_size = 512

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   validation_split=0.2)

train_set=train_datagen.flow_from_directory('./aument_multi',
                                            target_size=(img_size,img_size),
                                            color_mode='grayscale',
                                            batch_size=batch_size,
                                            shuffle=True,
                                            subset='training')

test_set = train_datagen.flow_from_directory('./aument_multi',
                                            target_size = (img_size,img_size),
                                            color_mode='grayscale',
                                            batch_size = batch_size,
                                            shuffle=True,
                                            subset='validation')
```

(3) สร้าง Model Architecture

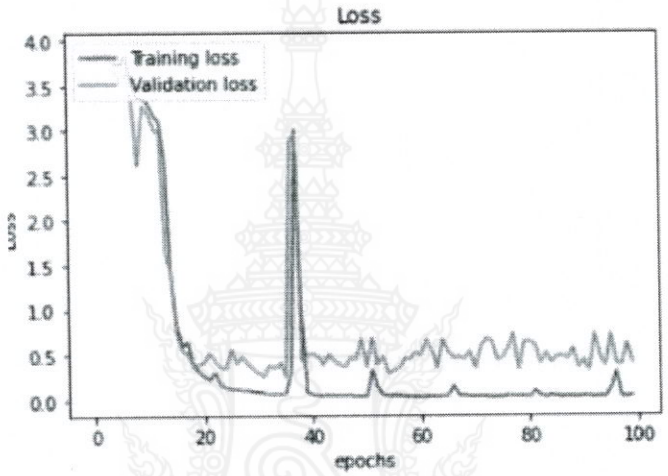
```
#AlexNet
model = Sequential()

# 1st Convolutional Layer
model.add(Conv2D(filters=96, input_shape=(img_size,img_size, 1), kernel_size=(11,11), strides=(4,4), padding='valid', activation='relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))

# 2nd Convolutional Layer
model.add(Conv2D(filters=256, kernel_size=(11,11), strides=(1,1), padding='valid', activation='relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))

# 3rd Convolutional Layer
model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid', activation='relu'))

# 4th Convolutional Layer
model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid', activation='relu'))

# 5th Convolutional Layer
model.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), padding='valid', activation='relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))

# Passing it to a Fully Connected layer
model.add(Flatten())
# 1st Fully Connected Layer
model.add(Dense(128, input_shape=(img_size*img_size*1,), activation='relu'))
# Add Dropout to prevent overfitting
model.add(Dropout(0.4))

# 2nd Fully Connected Layer
#model.add(Dense(128, activation='relu'))
# Add Dropout
#model.add(Dropout(0.4))

# 3rd Fully Connected Layer
model.add(Dense(55, activation='softmax'))
# Add Dropout
#model.add(Dropout(0.4))

# Output Layer
#model.add(Dense(55, activation='softmax'))
print(model.summary())
```

(4) Train Model

```
top5_acc = functools.partial(keras.metrics.top_k_categorical_accuracy, k=5)
top5_acc.__name__ = 'top5_acc'

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy', top5_acc])

step_size_train=train_set.n//train_set.batch_size
step_size_test=test_set.n//test_set.batch_size
history = model.fit_generator(train_set,
                              steps_per_epoch=step_size_train,
                              epochs = 100,
                              validation_data = test_set,
                              validation_steps=step_size_test)
```

(5) Model Evaluation (Loss & Accuracy)

```
plt.plot(history.history['loss'], label='Training loss')
plt.plot(history.history['val_loss'], label='Validation loss')
plt.title('Loss')
plt.ylabel('Loss')
plt.xlabel('epochs')
plt.legend(loc="upper left")
plt.show()
```

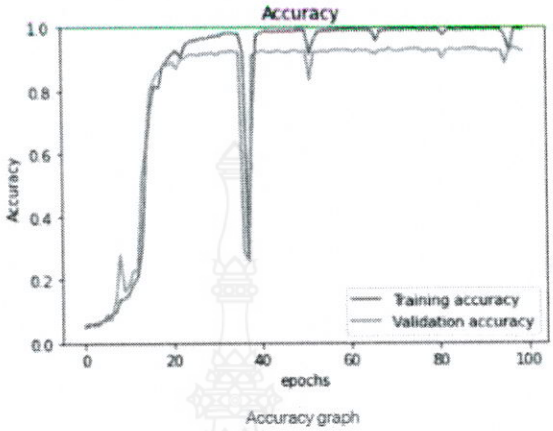


Loss graph

รูปที่ 3-16 Loss graph

จะเห็นว่าในแต่ละ Epochs(แกน X) และ Loss(แกน Y) จะเห็นว่าช่วง 0–20 epochs แรก นั้น Loss ลดลงเรื่อยๆ แต่พอช่วงเกือบๆ จะ epochs 30–40 กลับพุ่งขึ้นไปแล้วก็ลดลงมาจนนิ่ง

```
plt.plot(history.history['accuracy'], label='Training accuracy')
plt.plot(history.history['val_accuracy'], label='Validation accuracy')
plt.ylim((0, 1))
plt.title('Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('epochs')
plt.legend(loc="lower right")
plt.show()
```

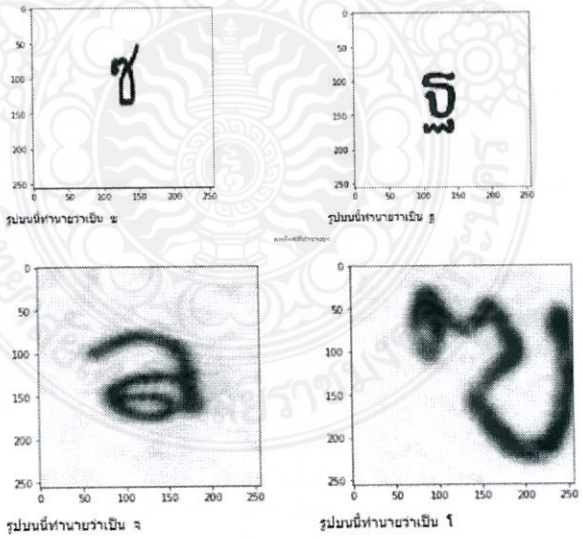


รูปที่ 3-17 Accuracy graph

จะเห็นว่าในแต่ละ Epochs(แกน X) และ Accuracy(แกน Y) จะเห็นว่าช่วง 0–20 epochs แรกนั้น Accuracy เพิ่มขึ้นเรื่อยๆ แต่พอช่วงเกือบๆจะ epochs 30–40 กลับพุ่งตกลงไปแล้วก็เพิ่มขึ้นมาจนนิ่งจนถึงประมาณ 92%

(6) Test Result

จากขั้นตอนก่อนหน้าพบว่า Model มีความแม่นยำอยู่ที่ 92% เราลองไปดูผลลัพธ์ที่ถูกและผิด



รูปที่ 3-18 ผลลัพธ์ของการทำนาย

(7) Save Model

จากนั้นก็ทำการ Save model เป็น json file และ Weights เป็น h5 เพื่อนำไปใช้ต่อ

```
model.save_weights("model.h5")
model_json = model.to_json()
with open("model_json.json", "w") as json_file:
    json_file.write(model_json)
```

3.2.2 การทำ Model Deployment

3.2.2.1 สร้าง load.py สำหรับอ่าน Model

(1) Import lib ทั้งหมดที่เราจะใช้ หลักๆ ก็จัดการเกี่ยวกับรูปภาพและ Model

```
from imageio import imsave, imread
from PIL import Image
import numpy as np
import tensorflow as tf
import keras
import tensorflow.keras.models
from keras.models import model_from_json
from tensorflow.python.keras.backend import set_session
from tensorflow.python.framework import ops
```

(2) สร้าง Tensorflow session สำหรับการ load model ที่ทำการ train มาใช้

```
session = tf.Session()

def init():
    # init and clear session tf keras
    init = tf.global_variables_initializer()
    session = keras.backend.get_session()
    session.run(init)

    # set default graph
    graph = tf.get_default_graph()

    # load model
    model_thw = load_model_thw()

    return model_thw,graph,session
```

(3) สร้าง function สำหรับการ load model

```
def load_model_thw():
    json_file = open('./model.json','r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)

    #load weights into new model
    loaded_model.load_weights("model.h5")

    #compile and evaluate loaded model
    loaded_model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

    return loaded_model
```

3.2.2.2 สร้าง app.py สำหรับการ Predict

(1) Import lib ทั้งหมดที่เราจะใช้ หลักๆ ก็จะมี request , cross_origin นอกนั้นก็จะเป็นการจัดการเกี่ยวกับรูปภาพและ Model

```
from flask import Flask, render_template, request , jsonify
from flask_cors import CORS, cross_origin
from imageio import imsave, imread
from PIL import Image
import numpy as np
import keras.models
import re
import sys
import os
import base64

sys.path.append(os.path.abspath("./model"))
from load import *

app = Flask(__name__)
app.config['CORS_HEADERS'] = "Content-Type"
cors = CORS(app,resources={r"/foo": {"origins": "*"}})

global graph, model , sess
model, graph , sess = init() #เรียกจาก load.py
```

(2) สร้าง Default route

```
@app.route('/',methods=['GET'])
@cross_origin(origin='*',headers=['Content-Type','Authorization'])
def test():
    return "Hello World."


if __name__ == '__main__':
  app.run(debug=True, port=8000)
```

(3) สร้าง Function สำหรับการ Predict

```
@app.route('/predict/',methods=['GET','POST'])
@cross_origin(origin='*',headers=['Content-Type','Authorization'])
def predict():
    imgData = request.get_json().encode()
    convertImage(imgData)
    imgFromOutPut = imread('output.png' , pilmode = "L")

    imgInvert = np.invert(imgFromOutPut)
    imgResize = np.array(Image.fromarray(imgInvert).resize(size=(95,95)))
    imgReshape = imgResize.reshape(1,95,95,1)

    with graph.as_default():
        set_session(sess)
        out = model.predict(imgReshape)
        response = get_thai_char_by_idex(out)
        return jsonify({'response': response})
```

ภายใน predict() จะมีการอ่าน request image data เป็น data_url base64 แล้วมาทำการ encode() จากนั้นทำการ save รูปนี้ไว้ที่ root path ด้วยชื่อ output.png

จากนั้นทำการอ่านรูปนั้นขึ้นมาเป็น byte array แล้วทำการ resize to 95,95 และ reshape ให้สามารถใช้งานกับ Model ได้ ณ ที่นี้ resize to 1,95,95,1

จากนั้นทำการ Predict รูปที่ทำการ reshape มาด้วย model.predict () พอได้ response มาแล้วนำไป map กับ ผลลัพธ์ตัวอักษรที่กำหนดไว้

(4) สร้าง Function สำหรับการ Save image

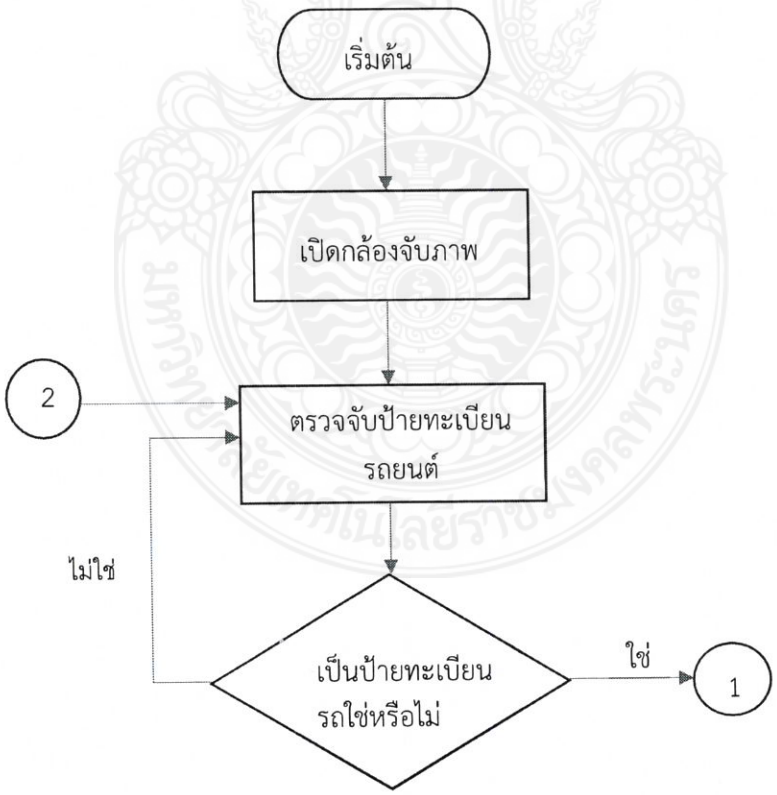
```
def convertImage(imgData1):
    print("convertImage")
    imgreg = re.search(b'base64,(.*)',imgData1)
    imgstr = imgreg.group(1)
    imgstr_64 = base64.b64decode(imgstr)
    with open('output.png','wb') as output:
            output.write(imgstr_64)
```

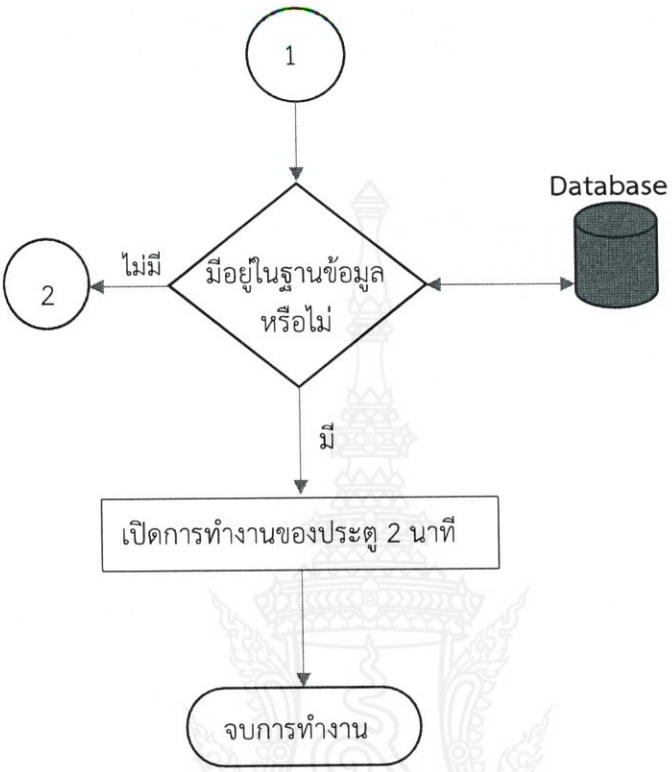
(5) สร้าง Function สำหรับการ Map response

```
def get_thai_char_by_idex(arr_idx):
    thai = ["ก","ข","ฃ","ค","ฅ","ฆ","ง","จ","ฉ","ช","ซ","ฌ","ญ","ฎ","ฏ","ฐ","ฑ","ฒ","ณ","ด","ต",
    "ถ","ท","ธ","น","บ","ป","ผ","ฝ","พ","ฟ","ภ","ม","ย","ร","ล","ว","ศ","ษ","ส","ห","ฬ","อ","ฮ"]

    result = []
    idx = arr_idx[0]

    for i in range(len(thai)):
        result.append((thai[i],idx[i]))

    sorted_by_second = sort_tuple(result)
    sorted_by_invert = sorted_by_second[::-1][:3]

    t = ""
    for i in range(len(sorted_by_invert)):
        t = t + sorted_by_invert[i][0]  + ":" + str(sorted_by_invert[i][1]) + "|"

    return t


def sort_tuple(tup):
    tup.sort(key = lambda x: x[1])
    return tup
```

เท่านี้ก็จะเสร็จเรียบร้อย ก็จะได้ API ที่อ่าน Model ในการทำนายตัวอักษร โดยรับ input เป็นรูป เรียบร้อยแล้ว

3.3 Flow Chart การทำงานของระบบ





รูปที่ 3-19 Flowchart การทำงานของระบบ

จากรูปจะเป็นการตรวจสอบทะเบียนรถจากกล้อง ถ้าเป็นทะเบียนรถจะมีการตรวจสอบกับฐานข้อมูลทะเบียนรถ ถ้าไม่มีในฐานข้อมูลทะเบียนรถให้ระบบไม่ต้องเปิดการทำงานของประตู แล้วไปตรวจจับทะเบียนรถจากกล้องอีกครั้ง กรณีที่ไม่ใช่ทะเบียนรถ ก็จะไม่ต้องดำเนินการตรวจสอบกับฐานข้อมูลทะเบียนรถ

3.4 ออกแบบฐานข้อมูล ตารางทะเบียนรถ (License_plate)

ชื่อ field	ชนิดของข้อมูล	คำอธิบาย
ID	Int NOT NULL AUTO_INCREMENT	ลำดับของข้อมูล
Char_plate	Char (20)	หมวดตัวอักษร
Number_plate	Int	หมวดตัวเลข

บทที่ 4

ผลการวิจัย

4.1 ผลการทดสอบจากรูป

เมื่อเข้าโปรแกรมในการตรวจจับทะเบียนรถจากรูปต้นแบบ 4-1, 4-3 จะได้ดังรูปที่ 4-2, 4-4

รูปที่ 4-1 รถต้นแบบก่อนจะตัดเหลือทะเบียนรถ (1)

รูปที่ 4-2 รูปทะเบียนรถหลังจากการรันโปรแกรม (1)

รูปที่ 4-3 รถต้นแบบก่อนจะตัดเหลือทะเบียนรถ (2)

รูปที่ 4-4 รูปทะเบียนรถหลังจากการรันโปรแกรม (2)

บทที่ 5

สรุป อภิปรายผล ข้อเสนอแนะ

จากการวิจัยระบบตรวจจับป้ายทะเบียนรถยนต์ เพื่อสร้างแบบจำลองเปิดประตูกั้นรถยนต์อัตโนมัติ ด้วย Deep Learning พบว่าในส่วนของการหาตำแหน่งของป้ายทะเบียน ระบบสามารถระบุตำแหน่งของป้ายทะเบียนได้ถูกต้องคิดเป็นร้อยละ 90 ในส่วนของป้ายทะเบียนรถที่อยู่ในลักษณะแนวตรงเท่านั้น ความผิดพลาดที่พบในการระบุตำแหน่งเนื่องจากเป็นการถ่ายรูปในลักษณะที่ไม่ได้อยู่ในแนวระดับสายตาทำให้ไม่สามารถจับป้ายทะเบียน และความคมชัด ส่งผลต่อการตรวจจับป้ายทะเบียนอย่างชัดเจน ส่วนของการแยกตัวเลขและตัวอักษร มีความถูกต้องคิดเป็นร้อยละ 98 และส่วนการรู้จำสามารถระบุตัวเลยและตัวอักษรได้ถูกต้องร้อยละ 95 โดยขบวนการวิจัยทั้งหมดใช้การประมวลผล

บรรณานุกรม

[1] Anonymous. 2023. MATLAB | RGB image representation. *Available*: https://www.geeksforgeeks.org/matlab-rgb-image-representation/. 20 September 2024

[2] ผศ. ปัณรสี ฤทธิประวัติ. 2014. Image Histogram. *Available*: https://imageprocessinggr3.wordpress.com/2014/11/22/image-histogram/. 20 September 2024

[3] ผศ.พนมขวัญ ริยะมงคล. 2017. Digital Image Processing. *Available*: http://www.ecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch03-part2.pdf. 20 September 2024

[4] เกษศิรินทร์ โถวสกุล. 2550. MySQL. [ระบบออนไลน]. แหล่งที่มา http://learners.in.th/blog/Donsak73/37469 (13 สิงหาคม 2550)

[5] สมปอง, ธีรศักดิ์ และณัฏฐ์ . 2567. การรู้จำลายมือเขียนภาษาไทยด้วยการเรียนรู้เชิงลึก. วารสารวิชาการพระจอมเกล้าพระนครเหนือ ปีที่ 34 ฉบับที่ 2. หน้า 1-12

[6] U.Pal, R. K. Roy, and F. Kimura, "Handwritten street name recognition for indian postal automation," in Proceedings of International Conference on Document Analysis and Recognition, 2011, pp. 483-487.

[7] J. L. Mitrpanont and Y. Imprasert, "Thai handwritten character recognition using heuristic rules hybrid with neural network,"in Proceedings of 8th International Joint. Conference on Computer Science and Software Engineering, 2011, pp. 160-165.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the lEEE, vol. 86, no. 11, 1998, pp. 2278-2324.

[9] S. Rathor, (2018, June 3). Simple RNN vs GRU vs LSTM: Difference lies in More Flexible control. [Online]. Available: https//medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

[10] R.C.Staudemeyer and E. R. Morris, Understanding LSMT-a tutorial into Long Short-Term Memory Recurrent Neural Networks. Thuringia,Germany: Schmalkalden University of Applied Sciences, 2019.

[11] NECTEC. (2019, March 20). 68PersonsBmp. [Online]. Available: https://thailang.nectec.or.th/best/best2019-hand writtenrecognition-trainingset

บรรณานุกรม

[12] MathWorks. (2019, January 1). Rgb2Gray. [Online]. Available: https://ww.mathworks.com/help/matlab/ ref/rgb2gray.htm

[13] OpenCV.(2019,January 1). Image Thresholding. [Online]. Available: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

[14] J.Canny, "A Computational Approach to Edge Detection," in lEEE Transactions on Pattern Analysis and Mochine Intelligence, Massachusetts, 1986, pp. 679-698.

[15] NECTEC. (2019, March 20). WD200-1, WD200-2,WD200-3 and WD200-4. [Online]. Available:https://thailang. nectec.or.th/best/best2019-handwrittenrecognition-trainingset

[16] T. Sangsuwan and S. Valuvanathoorn, "Thai handwritten character recognition usingcharacter line level grouping and keeping as-pect ratio with convolutional neural network, "in Proceedings of NCCIT2019, 2019, pp. 383-388 (in Thai).

[17] วิเชษฐ์รจน์ และรัชฎา . 2556. การแยกภาพตัวอักษรลายมือเขียนภาษาไทยแบบอัตโนมัติโดยใช้การวิเคราะห์ถดถอยเชิงเส้นและการเรียนรู้ด้วยต้นไม้ตัดสินใจ. Thai Journal of Science and Technology. ปีที่ 2 ฉบับที่ 2. หน้า 166-174

[18] Phokharatkul, P., Kimpan, C., 2002, Handwritten Thai character recognition using Fourier descriptors and genetic neural networks, Computational Intelligence 18: 270-93.

[19] Nopsuwanchai, R., Biem, A., Clocksin, W.F.,2006, Maximization of mutual information for offline Thai handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 28: 1347-1351.

ประวัติผู้วิจัย

หัวหน้าโครงการวิจัย

1. **ชื่อ-นามสกุล** (ภาษาไทย) .....นายเกียรติศักดิ์ ลาภพาณิชยกุล

   (ภาษาอังกฤษ) Mr. Kreadtisak Lappanitchayakul

2. **หมายเลขบัตรประจำตัวประชาชน** 3609700368962

3. **ตำแหน่งปัจจุบัน**

   ตำแหน่งทางวิชาการ ผู้ช่วยศาสตราจารย์

   ตำแหน่งทางบริหาร .....

4. **หน่วยงานที่อยู่ที่สามารถติดต่อได้สะดวก พร้อมหมาย เลขโทรศัพท์ และ e-mail**

   สาขาวิชา ระบบสารสนเทศ

   คณะ บริหารธุรกิจ

   มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

   เลขที่ 399 ถนน สามเสน แขวง วชิรพยาบาล เขต ดุสิต จังหวัด กทม

   โทรศัพท์ 02-665-3777

   e-mail address : kreadtisak.l@rmutp.ac.th

5. **ประวัติการศึกษา**

ระดับปริญญา	คุณวุฒิ/สาขาวิชา	สถาบันอุดมศึกษา	ปีที่สำเร็จ
ปริญญาโท	วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ (หลักสูตรนานาชาติ)	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	พ.ศ. 2551
ปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์	มหาวิทยาลัยนเรศวร	พ.ศ. 2546

6. **สาขาวิชาการที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา) ระบุสาขาวิชาการ**

7. **ประสบการณ์ที่เกี่ยวข้องกับการบริหารงานวิจัยทั้งภายใน และภายนอกประเทศ**

   7.1 ผลงานวิจัย

ชื่อผลงานวิจัย	สถานภาพ	แหล่งทุน/ปี
ศึกษาความพร้อมของเทคโนโลยีสารสนเทศเพื่อรองรับการสื่อสารทางการศึกษาในประชาคม	หัวหน้า โครงการวิจัย	งบประมาณแผ่นดิน พ.ศ. 2560

ชื่อผลงานวิจัย	สถานภาพ	แหล่งทุน/ปี
อาเซียนของมหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร		
ระดับการรับรู้ของบุคลากรคณะบริหารธุรกิจต่อการเผยแพร่ข่าวสารบนโลกอินเตอร์เน็ต กับการกระทำความผิดตาม พ.ร.บ.คอมพิวเตอร์.พ.ศ. 2550	หัวหน้า โครงการวิจัย	งบประมาณรายได้ คณะบริหารธุรกิจ พ.ศ. 2560
การศึกษาความพร้อมด้านการผลิตบัณฑิต ของคณะบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร เพื่อรองรับการก้าวเข้าสู่ประเทศไทย 4.0	หัวหน้า โครงการวิจัย	งบประมาณรายได้ คณะบริหารธุรกิจ พ.ศ. 2561

7.2 การตีพิมพ์เผยแพร่ผลงานวิจัย

การประชุมวิชาการระดับชาติ

สุจิรา ไชยกุสินธุ์ สัจธรรม สุภาจันทร์ รัตนาวลี ไม้สัก พรคิด อั้นขาว และเกียรติศักดิ์ ลาภพาณิชยกุล. (2559). การประยุกต์ใช้ต้นไม้ตัดสินใจสำหรับการรับนักศึกษาเข้าทำงานในสถานประกอบการ. การประชุมวิชาการระดับชาติการจัดการธุรกิจและเทคโนโลยีดิจิทัล. มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร. กรุงเทพมหานคร. 16-17 ธันวาคม 2559. หน้า 877-883.

การประชุมวิชาการระดับนานาชาติ

Kreadtisak Lappanitchayakul, "The readiness of Information technology of Rajamangala University of Technology Phra Nakhon for education communication to ASEAN", 2019 First International Conference on Smart Technology & Urban Development (STUD), 13-14 December 2019, Chiang Mai, Thailand, pp. 1-4

Kreadtisak Lappanitchayakul, "Development of Email and SMS Based Notification System to Detect Abnormal Network Conditions: A Case Study of Faculty of Business Administration, Rajamangala University of Technology Phra Nakhon, Thailand", 2018 International Conference on

Intelligent Informatics and Biomedical Sciences (ICIIBMS), 21-24 October 2018, Bangkok, Thailand, pp. 98-105

Kreadtisak Lappanitchayakul, "Anti-theft device for car : Alert system using radio wave" , 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 21-24 November 2019, Shanghai, China , pp. 351-355

Kreadtisak Lappanitchayakul, "The study of readiness in graduates production of Faculty of Business Administration, Rajamangala University of Technology Phra Nakhon to support the entering into Thailand 4.0", 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE), 20-22 November 2019, Bangkok, Thailand