บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1. ทฤษฎีการประมวลผลภาพ (Image Processing)

ภาพ

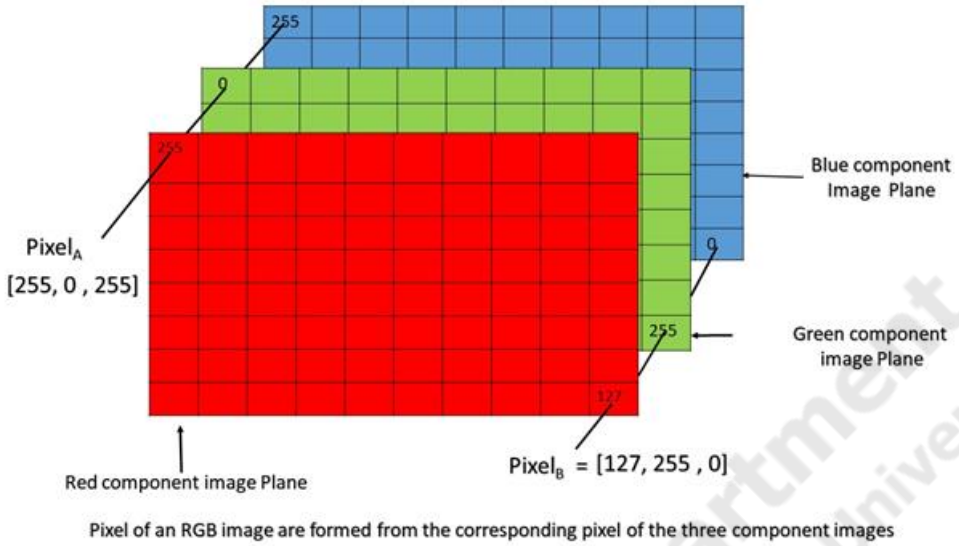
เป็นฟังก์ชันสองมิติ F (x, y)โดยที่ x และ y เป็นพิกัดเชิงพื้นที่และความกว้างของFที่พิกัดคู่ใด ๆ (x, y) เรียกว่าความเข้มของภาพนั้น เมื่อ x, y และค่าความกว้างของFมีจำกัด เราเรียกมันว่าภาพดิจิทัล กล่าวอีกนัยหนึ่งว่ารูปภาพสามารถถูกกำหนดโดยอาร์เรย์สองมิติที่จัดเรียงเป็นพิเศษในแถวและคอลัมน์ ภาพดิจิทัลประกอบด้วยจำนวนจำกัด ซึ่งภาพในระบบดิจิทัลจะมีข้อมูลขนาดเล็ก ๆ ที่เรียงกันในภาพ เรียกว่า พิกเซล ดังนั้น พิกเซลที่มีค่าส่องสว่างมาเรียงต่อ ๆ กัน เป็นจำนวนมาก จะทำให้เราสามารถมองเห็นภาพได้ในระบบคอมพิวเตอร์

ภาพดิจิทัล

เป็นตัวแทนของภาพเสมือนจริงเป็นชุดของตัวเลขที่สามารถจัดเก็บและจัดการโดยเป็นดิจิทัลคอมพิวเตอร์ ในการแปลภาพเป็นตัวเลขและจะถูกแบ่งออกเป็นพื้นที่เล็ก ๆ ที่เรียกว่า พิกเซล (องค์ประกอบภาพ) สำหรับแต่ละพิกเซลอุปกรณ์ถ่ายภาพจะบันทึกหมายเลขหรือชุดตัวเลขขนาดเล็กซึ่งอธิบายคุณสมบัติบางอย่างของพิกเซลนี้เช่นความสว่าง (ความเข้มของแสง) หรือสีของภาพ ตัวเลขจะถูกจัดเรียงในอาร์เรย์ของแถวและคอลัมน์ที่สอดคล้องกับตำแหน่งแนวตั้งและแนวนอนของพิกเซลในภาพ

ภาพแบบ RGB

ภาพ RGB บางครั้งเรียกว่า ภาพสีจริง (TrueColor) ถูกเก็บไว้ใน MATLAB เป็นอาร์เรย์ข้อมูล M * N * 3 ที่กำหนดองค์ประกอบสีแดงสีเขียวและสีน้ำเงินสำหรับแต่ละพิกเซลแต่ละพิกเซล ภาพ RGB ไม่ได้ใช้จานสี สีของแต่ละพิกเซลถูกกำหนดโดยการรวมกันของความเข้มสีแดง, สีเขียวและสีน้ำเงินที่เก็บไว้ในระนาบสีแต่ละจุดที่ตำแหน่งของพิกเซล รูปแบบไฟล์กราฟฟิกจัดเก็บภาพ RGB เป็นภาพ 24 บิตโดยที่องค์ประกอบสีแดงสีเขียวและสีน้ำเงินแต่ละชิ้นจะมี 8 บิต สิ่งนี้ทำให้ศักยภาพของ 16 ล้านสี มีความแม่นยำที่สามารถจำลองภาพในชีวิตจริงได้ สามารถดูได้สามภาพที่แตกต่างกัน (ภาพสเกลสีแดง ภาพสเกลสีเขียว และภาพสเกลสีฟ้า) ซ้อนกันอยู่ด้านบนของกันและกัน (GeeksforGeeks, 2562)



ภาพประกอบที่ 2.1 ระนาบสีแต่ละอันมีอาร์เรย์ M * N

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

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

R = ระดับของแสงสีแดง

G = ระดับของแสงสีเขียว

B = ระดับของแสงสีน้ำเงิน

ดังนั้นสีของพิกเซลใด ๆ จะถูกกำหนดโดยการรวมกันของความเข้ม สีแดง สีเขียว และสีน้ำเงินที่เก็บไว้ในระนาบแต่ละสีที่ตำแหน่งของพิกเซล และระนาบสีแต่ละอันมีอาร์เรย์ M * N (Keim, 2561)

เป็นการแปลงภาพระบบสี RGB ที่แสดงถึงความเข้มสีในแต่ละองค์ประกอบสีให้เป็นภาพที่แสดงถึงค่าความสว่างของแสงเพียงอย่างเดียวโดยทั่วไปภาพระดับเทาสามารถแบ่งระดับของความสว่างได้แตกต่างกัน 256 ระดับ โดยพิกเซลที่มืดที่สุดมีค่าเท่า 0 และพิกเซลที่สว่างที่สุดมีค่าเท่ากับ 255 นอกจากนี้การแปลงภาพสี RGB เป็นภาพระดับเทาสามารถลดเวลาในการประมวลผลภาพดิจิตอลจากการลดจำนวนมิติของข้อมูลในภาพสี RGB ที่จัดเก็บด้วยเมทริกซ์สองมิติซ้อนทับกันสามชั้นให้ลดลงเหลือเพียงเมทริกซ์สองมิติเพียงชั้นเดียว

2.1.2.การประมวลผลภาพ (Image Processing)

เป็นการนำภาพมาประมวลผลหรือคำนวณด้วยเครื่องคอมพิวเตอร์เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงปริมาณและในเชิงคุณภาพโดยมีขั้นตอนต่างๆที่สำคัญคือการทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในระยะเวลาอันสั้น จึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบ ตัวอย่างการนำการประมวลผลภาพไปใช้งาน เช่น ระบบรู้จำลายนิ้วมือเพื่อตรวจสอบว่าภาพลายนิ้วมือที่มีอยู่นั้นเป็นของผู้ใด ระบบตรวจกระดาษคำตอบ โดยมีการเปรียบเทียบภาพกระดาษคำตอบที่ถูกต้องกับกระดาษคำตอบที่จะตรวจว่ามีตำแหน่งตรงกันหรือไม่

2.1.3. การปรับปรุงคุณภาพของภาพ

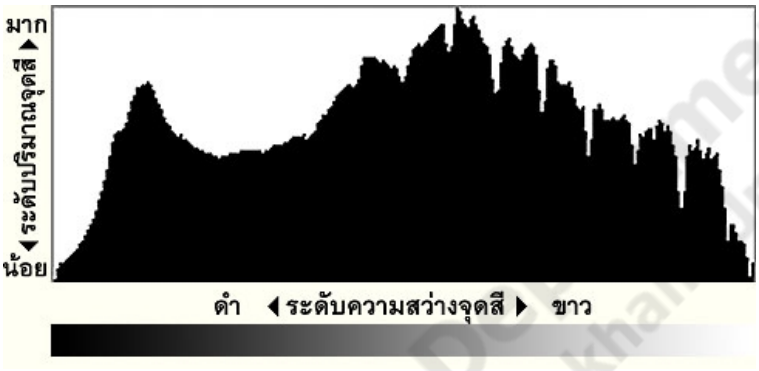
ในการประมวลภาพนั้นการปรับปรุงคุณภาพของภาพได้กลายมาเป็นอีกปัจจัยหนึ่งที่มีความสำคัญและช่วยให้การประมวลผลภาพในลำดับต่อไปนั้นทำได้ง่ายขึ้น ภาพดิจิทัลโดยทั่วไปนั้นเกิดจากดิจิไตส์จากข้อมูลที่อยู่ในรูปแบบคลื่นให้กลายมาเป็นข้อมูลแบบจำกัด ดังนั้นบ่อยครั้งที่สภาพแวดล้อมต่างๆ ทำให้ภาพที่เราได้มานั้นไม่ได้เหมาะในการนำมาประมวลผลเพื่อให้ได้ผลลัพธ์ที่ดี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

การปรับปรุงคุณภาพของภาพเป็นเทคนิคในการประมวลผลภาพในระดับล่างที่มักจะถูกนำไปใช ในการประมวลผลภาพก่อนการประมวลผลจริง (pre-processing) โดยการปรับปรุงคุณภาพของ ภาพนั้นจะเป็นการเปลี่ยนแปลงคุณสมบัติของภาพไม่ว่าจะเป็นเรื่องของสี ความสว่าง หรือ แม้แต่ ความคมชัดของภาพ รวมถึงการกำจัดสิ่งที่ไม่จำเป็นและอาจจะทำให้การประมวลผลในขั้นตอนถัดไปยากขึ้น

2.1.4. Image Histogram

เป็นกราฟที่แสดงจำนวน pixels ในแต่ละความสว่างต่างๆหรือข้อมูลค่าสี R,G,B ของรูปภาพ digital ในภาพ gray scale ในแกนนอนจะแสดงความสว่างดังกล่าว ซึ่งมีความสว่างตั้งแต่ 0-255 (แบ่งเป็น 256 ระดับความแตกต่างสี) โดยทางด้านซ้ายของกราฟจะมีค่าความสว่างน้อย ภาพจะมีสีเข้มเข้าใกล้สีดำ ส่วนทางด้านขวามือจะมีความสว่างสูง ภาพจะสว่างเข้าใกล้สีขาว ส่วนบริเวณตรงกลางกราฟแสดงส่วนน้ำหนักสีกลาง ส่วนในแนวแกนตั้งจะแสดงจำนวน pixels ในแต่ละความสว่างซึ่งในแกนตั้งนี้ ไม่มีขอบเขตจำกัด ถ้าหากภาพมีความมืดมาก กราฟจะไปกองรวมกันทางด้านซ้ายมืด โดยที่ไม่มีขอบเขตจำกัด

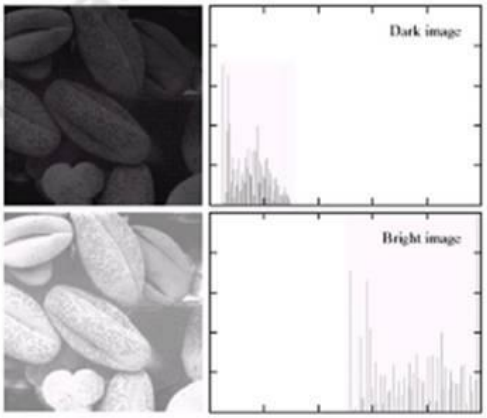
ฮิสโตรแกรมภาพ (image histogram) ภาพประกอบไปด้วยพิกเซลเล็กๆ ซึ่งแต่ละพิกเซลจะมีค่าส่องสว่างหรือว่าค่าสี ค่าสีของพิกเซลจะมีค่าที่แตกต่างกันออกไปขึ้นอยู่กับระบบสีที่ใช้ สำหรับภาพสีแบบ RGB ค่าของพิกเซล 1 พิกเซล จะประกอบไปด้วยค่าสี 3 ค่าด้วยกัน โดยค่าแต่ละค่าจะมีช่วงอยู่ระหว่าง 0-255 นอกจากเรา จะสามารถพิจารณาภาพในรูปแบบปกติแล้ว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



ภาพประกอบที่ 2. 2 กราฟที่แสดงจำนวน pixels ในแต่ละความสว่างต่างๆ

ที่มา: http://www.fotofile.net/learning/histogram

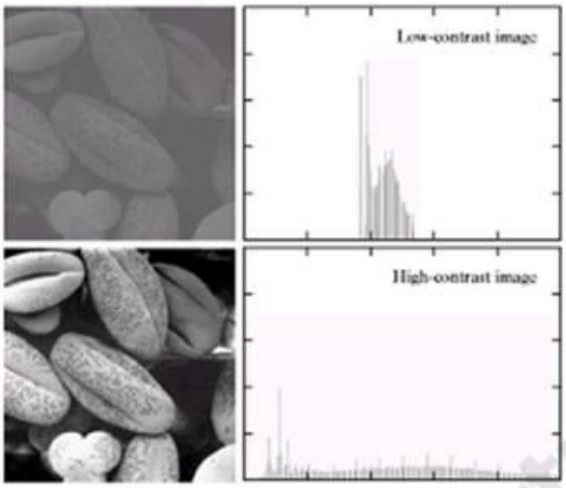
คุณสมบัติของ Histogram หากการกระจายส่วนใหญ่อยู่ทางด้านซ้ายของกราฟ แสดงว่าภาพนั้นมีความสว่างของภาพน้อย ในทางกลับกัน หากการกระจายส่วนใหญ่อยู่ทางด้านขวาของกราฟ แสดงว่าภาพนั้นมีความสว่างของภาพมาก



ภาพประกอบที่ 2.3 Low - contrast

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

หากการกระจายของกราฟเป็นกลุ่มแคบ ๆ แสดงว่าภาพนั้นเป็นภาพที่ Low-contrast และหากการกระจายของกราฟมีการกระจายอย่างสม่ำเสมอทั่วทั้งกราฟ แสดงว่าภาพนั้นเป็นภาพที่ High-contrast



ภาพประกอบที่ 2.4 High-contrast

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

ประโยชน์ของ Histogram

1. ภาพที่มองจากจอภาพอาจมีการคลาดเคลื่อนได้ แต่ข้อมูลจาก histogram จะบอกความสว่างความเข้มของรูปภาพได้อย่างแท้จริง

2. ข้อมูลนี้จะช่วยให้เราเลือกโหมดในการถ่ายภาพได้ดียิ่งขึ้น โดยช่วยในการเลือกการชดเชยแสงของภาพเมื่อต้องถ่ายภาพในที่ที่มีความสว่างของภาพสูงหรือต่ำมากได้

3. สามารถนำข้อมูลมาใช้ประกอบในการประมวลผลและปรับแต่งภาพได้

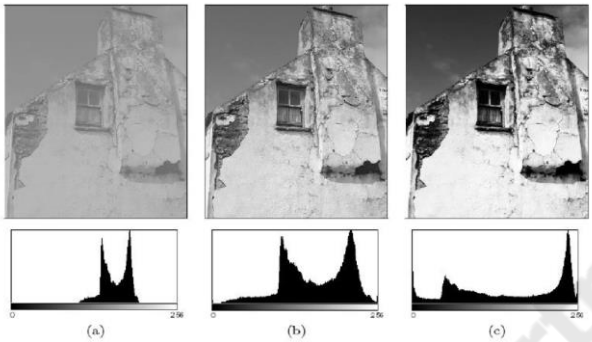
2.1.5. การเพิ่มคอนทราสต์ของภาพ (contrast enhancement)

เป็นอีกกระบวนการหนึ่งที่สามารถ ทำได้ด้วยการดำเนินการกับฮีสโตรแกรมของภาพ ซึ่งคอนทราสต์ของภาพนั้นจะทำให้เราสามารถ เห็นความแตกต่างระหว่างค่าสีที่มีความต่างกันได้ชัดเจนขึ้น โดยทั่วไปการเพิ่มคอนทราสต์ของภาพ นั้นจะดำเนินกับภาพที่มีค่าสีในภาพนั้นใกล้เคียงกัน และในบางครั้งการที่มีคอนทราสต์ที่ไม่เหมาะ สมจะให้เราไม่สามารถเห็นรายละเอียดบางส่วนในภาพอย่างชัดเจน เช่น ตัวอย่างของการปรับปรุง คอนทราสต์ภาพแสดงดัง**ภาพประกอบที่ 2.5**


ภาพที่มีคอนทราสต์ต่ำ ภาพหลังจากการปรับคอนทราสต์

ภาพประกอบที่ 2.5 Contrast

Contrast หมายถึง ค่าความแตกต่างสี โดยในทาง histogram เมื่อกราฟที่ได้มีการกระจายความเข้มของสีมาก ก็จะได้ภาพที่คมชัดมากขึ้น หรือภาพที่มีความแตกต่างกันของระดับ สีในภาพน้อย จะได้ภาพที่มีสีซีด



(a) คอนทราสต์ต่ำ (b) คอนทราสต์ปกติ (c) คอนทราสต์สูง

ภาพประกอบที่ 2.6 การปรับคอนทราสต์

2.1.6.การทรานสฟอร์มภาพ (Image transformation)

การทรานฟอร์มภาพเป็นอีกกระบวนการทางด้านการประมวลผลภาพหนึ่งที่มีการนำมาปรับปรุงคุณภาพ ของภาพเพื่อให้การประมวลผลในขั้นตอนถัดไปนั้นง่ายขึ้น หรือ ลดความหลากหลาย ของภาพเพื่อให้การประมวลผลสามารถดำเนินการได้อย่างมีประสิทธิภาพ ในหัวข้อนี้จะอธิบาย หลักการและวิธีการการทรานสฟอร์มภาพด้วยการปรับภาพให้อยู่ในมาตรฐานหรือรูปแบบเดียวกัน เพื่อให้ง่ายต่อการประมวลผล ขั้นตอนในการทรานฟอร์ภาพจากคุณสมบัติของภาพ มีดัง ต่อไปนี้ คือ

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

ภาพประกอบที่ 2.7 การทรานสฟอร์มภาพ (Image transformation)

จากภาพจะเห็นได้ว่าภาพประมวลผลหรือภาพนำเข้า เมื่อมีการทรานฟอร์สภาพ แล้วจะมีโทนสีหรือ look and feel ของภาพที่คล้ายกับภาพต้นแบบ ในการประมวลภาพเมื่อภาพมีความหลากหลายมากๆ เทคนิคของการทรานฟอร์มภาพนั้นสามารถ นำมาใช้สำหรับการปรับภาพให้อยู่ในโทนสีเดียวกัน เพื่อให้การออกแบบอัลกอริทึมในการประมวล ผลภาพในขั้นตอนต่อไปนั้นทำได้ง่ายและ ยังสามารถเจาะจงกับกลุ่มภาพที่มีลักษณะโทนสีเดียวกัน

2.1.7. Deblur

เป็นอีกกระบวนการทางด้านการประมวลผลภาพหนึ่งที่มีการนำมาปรับปรุง คุณภาพ ของภาพ เพื่อให้การประมวลผลในขั้นตอนถัดไปนั้นง่ายขึ้น หรือ ลดเบลอของภาพเพื่อให้การประมวลผลสามารถดำเนินการได้อย่างมีประสิทธิภาพ ในหัวข้อนี้จะอธิบาย หลักการและวิธีการการดีเบลอภาพด้วยการปรับให้มีความคมชัดมากขึ้น เพื่อให้ง่ายต่อการประมวลผล ขั้นตอนในดีเบลอภาพที่ใช้มีดัง ต่อไปนี้ คือ

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

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

Global Thresholding

เป็นการหาจุดตัดจากภาพรวมทั้งหมดของภาพ แบ่งเป็นวิธีหลักๆ 2 วิธี ได้แก่ Single thresholding และ Double thresholding

- Single thresholding เป็นการพิจารณาจากฮิสโตแกรมของระดับความเข้มของภาพ เพื่อหาจุดตัดที่ เหมาะสม วิธีนี้ใช้ได้ดีกับการแยกภาพที่วัตถุกับพื้นหลังแยกกันอย่างชัดเจน
- Double Thresholding สำหรับกรณีที่ต้องการหาค่า 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 เป็นวิธีที่ประยุกต์จากอนุพันธ์อันดับที่ 1 โดยการประยุกต์กับ Guassian เพื่อประมาณค่า Operator โดยมีขั้นตอน ดังนี้

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

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

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

กำหนดให้ $g(x,y)$ Guassian Function

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

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

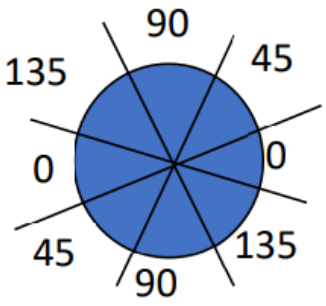
$$g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{\sigma^2} g(x,y) \tag{4}$$

2. หาค่าขนาด (Magnitude) และทิศทาง (Direction) ของแกรเดียนท์

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} \,, \tag{5}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{6}$$

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

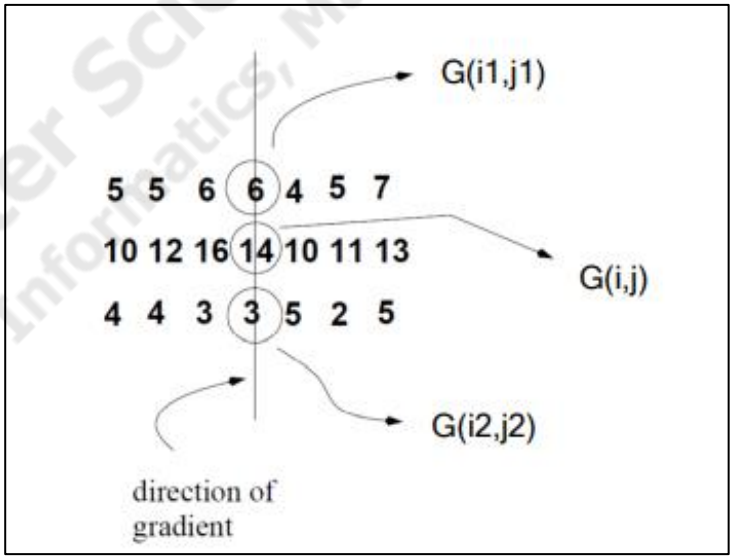


ภาพประกอบที่ 2.8 มุม θ ของแกรเดียนท์

3. ประยุกต์การทำ Non-maximum suppression เพื่อให้ขอบวัตถุบางเหลือเพียงของวัตถุเพียงจุดเดียว

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

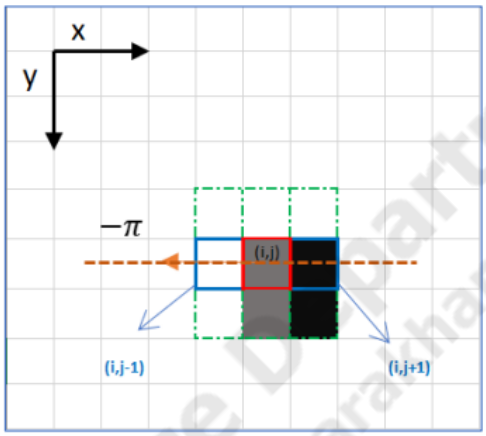
- ตรวจสอบถ้าจุดภาพที่มีทิศทางเดียวกันโดยเช็คว่า ถ้าจุดภาพใดมีค่ามากที่สุดให้พิจารณาค่านั้นเป็นจุดภาพที่น่าจะป็นขอบภาพนำไปประมวลผลต่อ นอกนั้นก็ให้เป็น 0



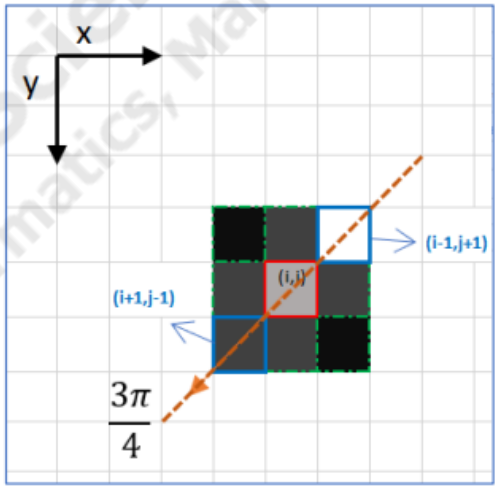
ภาพประกอบที่ 2.9 เปรียบเทียบแกรเดียนท์ตามทิศทางที่คำนวณได้

```
for each pixel f(x,y) do
  check direction and choose pair of G(i,j)
  if G(i, j) < G(i1, j1) or G(i, j) < G(i2, j2)
    then out(i, j) = 0
    else out(i, j) = G(i, j)
```

ภาพประกอบที่ 2.10 อัลกอริทึม เปรียบเทียบแกรเดียนท์ตามทิศทางที่คำนวณได้



ภาพประกอบที่ 2.11 ทิศตามแนวนอน 0 หรือ 180 องศา



ภาพประกอบที่ 2.12 ทิศตามแนวนอน 45 หรือ 225 องศา

ที่มา : https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

4. ประยุกต์การทำ Double thresholding โดยการนำค่า threshold 2 ค่า (Low และ high) ให้ค่า Low threshold (T_{lo}) ในการกำหดจุดเริ่มต้นของเส้นขอบและจุดปลายอยู่ที่ High

threshold (T_{hi}) ซึ่งโดยปกติแล้ว $T_{\text{ร}} \approx 2(T_{lo})$ มาเปรียบเทียบกับ Gradient magnitude เพื่อระบุค่าจุดภาพ 3 ชนิด (Weak , Strong และ Non-relevant) ดังต่อไปนี้

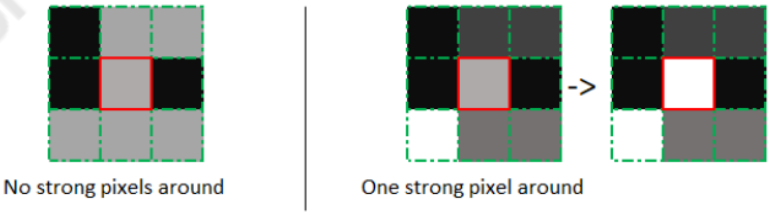
- จุดภาพที่มีระดับความเข้มสูง พิจารณาให้เป็น Strong pixel คือจุดที่คาดว่าจะเป็นของภาพ
- จุดที่มีระดับความเข้มไม่สูงละต่ำจัด จะพิจารณาเป็น Weak pixel
- นอกนั้นพิจารณาให้เป็น non-relevant

อัลกอริทึม Double thresholding

```
for each pixel f(x,y) do
  grad_mag = magnitude[i,j]
  if grad_mag<weak_th: #mark as non-relevant
    magnitude [i, j]= 0
  else if strong_th>grad_mag>= weak_th: #mark as weak ones
    magnitude[i, j]= weak
  else: #mark as strong ones
    magnitude[i, j]= strong
```

ภาพประกอบที่ 2.13 ประยุกต์การทำ Double thresholding

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



ภาพประกอบที่ 2.14 ตัวอย่างการพิจารณาขอบภาพจาก Weak pixel

ที่มา : https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

ข้อดี

1. เป็นวิธีที่ดีในการระบุขอบวัตถุ
2. เป็นการจำแนกคุณลักษณะ() โดยไม่เปลี่ยนแปลงคุณลักษณะของภาพ
3. ไวต่อสัญญาณรบกวนน้อยมาก

ข้อเสีย

1. อาจทำให้กรณี Zero Crossing เป็นเท็จได้
2. ใช้เวลาในการประมวลผลมากและขั้นตอนในการคำนวณซับซ้อน

คำสั่ง Canny ใน OpenCV

คำสั่งการหาขอบภาพด้วย Canny ดำเนินการได้ดังนี้

edges = cv2.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_16SC3)

dy : ค่าอนุพันธ์ตามแนวแกน y (CV_16SC1 หรือ CV_16SC3)

threshold1 : ค่า Low threshold ของ Hysteresis

threshold2 : ค่า High threshold ของ Hysteresis

apertureSize : ขนาดของการคำนวณหาค่า Sobel ซึ่ง default=3

L2gradient : การหาค่า magnitude ด้วยรากของกำลังสอง ซึ่งมีค่าเป็น True หรือหา absolute ซึ่งมีค่าเป็น default=False

Result:

edges : ภาพผลลัพธ์ (edge map) ขนาด 8 bit หรือเท่ากับภาพนำเข้า

ตัวอย่างการเขียนโปรแกรมประมวลผลการหาขอบภาพด้วย Canny

```python
# Non-maximum suppression
def non_max_suppression(magnitude, angle):
    # getting the dimensions of the input image
    height, width = magnitude.shape
    out = np.zeros(magnitude.shape)

    for row in range(1, height - 1):
        for col in range(1, width - 1):
            direction = angle[row, col]

            # (0 degree)
            if (0 <= direction<22.5) or (337.5 <= direction <= 360):
                before_pixel = magnitude[row, col - 1]
                after_pixel = magnitude[row, col + 1]
            # (45 degree)
            elif (22.5 <= direction < 67.5) or (180 <= direction < 247.5):
                before_pixel = magnitude[row + 1, col - 1]
                after_pixel = magnitude[row - 1, col + 1]
            # (90 degree)
            elif (67.5 <= direction < 112.5) or (247.5 <= direction < 292.5):
                before_pixel = magnitude[row - 1, col]
                after_pixel = magnitude[row + 1, col]
            # (135 degree)
            else:
                before_pixel = magnitude[row - 1, col - 1]
                after_pixel = magnitude[row + 1, col + 1]

            if magnitude[row, col] >= before_pixel and \
               magnitude[row, col] >= after_pixel:
                out[row, col] = magnitude[row, col]
    return out
```

```python
# double thresholding step
def double_threshold(mag,weak_th,strong_th):
    weak_ids = np.zeros(mag.shape,np.uint8)
    strong_ids = np.zeros(mag.shape,np.uint8)
    flag = np.zeros_like(mag.shape,np.uint8)
    height,width = mag.shape
    weak, strong=25,255
    for i in range(height):
        for j in range(width):
            grad_mag = mag[i, j]

            if grad_mag<weak_th: # Non-relevant
                mag[i, j]= 0
            elif strong_th>grad_mag>= weak_th: #weak
                mag[i, j] = weak
            else: #strong
                mag[i, j] = strong
    return mag, weak

# Hysteresis step
def hysteresis(img, weak, strong=255):
    M, N = img.shape
    for i in range(1, M-1):
        for j in range(1, N-1):
            if (img[i,j] == weak):
                if ((img[i+1, j-1] == strong) or (img[i+1, j] == strong)
                    or (img[i+1, j+1] == strong) or (img[i, j-1] == strong)
                    or (img[i, j+1] == strong) or (img[i-1, j-1] == strong)
                    or (img[i-1, j] == strong) or (img[i-1, j+1] == strong)):
                        img[i, j] = strong
                else:
                        img[i, j] = 0
    return img
```

ภาพประกอบที่ 2.15 โปรแกรมประมวลผลการหาขอบภาพด้วย Canny

```python
# defining the canny detector function
def Canny_detector(img, weak_th = None, strong_th = None):

    # conversion of image to grayscale
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Noise reduction step
    img = cv2.GaussianBlur(img, (5, 5), 1.4)

    # Calculating the gradients
    gx = cv2.Sobel(np.float32(img), cv2.CV_64F, 1, 0, 3)
    gy = cv2.Sobel(np.float32(img), cv2.CV_64F, 0, 1, 3)

    # Conversion of Cartesian coordinates to polar
    magnitude, angle = cv2.cartToPolar(gx, gy, angleInDegrees = True)

  # setting the min and max thresholds for double thresholding
    mag_max = np.max(magnitude)
    if not weak_th:weak_th = mag_max * 0.1
    if not strong_th:strong_th = mag_max * 0.5

    mag=non_max_suppression(magnitude, angle)
    mag_thr,weak=double_threshold(mag, weak_th, strong_th)
    out=hysteresis(mag_thr, weak)

    return out

img = cv2.imread('../lena512color.tiff')
# calling the designed function for finding edges
img_canny1 = Canny_detector(img, 20, 40)
cv2.imshow('original',img)
cv2.imshow('canny1',img_canny1)
cv2.waitKey(5)
```

ภาพประกอบที่ 2.15 โปรแกรมประมวลผลการหาขอบภาพด้วย Canny

2.1.8. การเรียนรู้เชิงลึก (Deep Learning)

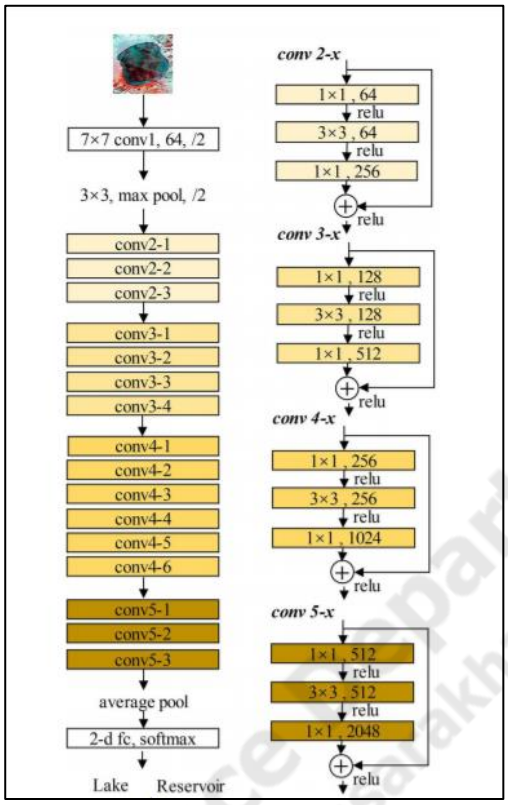
การเรียนรู้อย่างลึกซึ้งเป็นฟังก์ชันของ ปัญญาประดิษฐ์ที่เลียนแบบการทำงานของสมองมนุษย์ในการประมวลผลข้อมูลและสร้างรูปแบบเพื่อใช้ในการตัดสินใจ การเรียนรู้อย่างลึกซึ้งเป็นส่วนหนึ่งของการเรียนรู้ของเครื่องในปัญญาประดิษฐ์ (AI) เป็นเครือข่ายที่สามารถเรียนรู้ได้โดยที่ไม่ได้รับการสนับสนุนจากข้อมูลที่ไม่มีโครงสร้างหรือไม่มีป้ายกำกับ ยังเป็นที่รู้จักกันในนามการเรียนรู้ระบบประสาทลึกหรือเครือข่ายประสาทลึก

การเรียนรู้อย่างลึกซึ้งได้พัฒนาไปพร้อม ๆ กับยุคดิจิทัลซึ่งทำให้เกิดการระเบิดของข้อมูลในทุกรูปแบบและจากทุกภูมิภาคของโลก ข้อมูลนี้เรียกได้ว่าเป็นข้อมูลขนาดใหญ่นั้นนำมาจากแหล่งต่าง ๆเช่น โซเชียลมีเดียเครื่องมือค้นหาอินเทอร์เน็ตแพลตฟอร์มอีคอมเมิร์ซและโรงภาพยนตร์ออนไลน์เป็นต้น ข้อมูลจำนวนมหาศาลนี้สามารถเข้าถึงได้อย่างง่ายดายและสามารถแบ่งปันผ่านแอพพลิเคชั่น fintech เช่น cloud computing การเรียนรู้เชิงลึกกับการเรียนรู้ของเครื่องในเทคนิค AI ที่ใช้กันมากที่สุดในการประมวลผลข้อมูลขนาดใหญ่คือการเรียนรู้ด้วยเครื่องซึ่งเป็นอัลกอริทึมการปรับตัวเองที่ได้รับการวิเคราะห์และรูปแบบที่ดีขึ้นด้วยประสบการณ์หรือด้วยข้อมูลที่เพิ่มเข้ามาใหม่

การเรียนรู้เชิงลึกหรือที่เรียกว่าการเรียนรู้แบบลำดับชั้นหรือการเรียนรู้เชิงโครงสร้างเป็นประเภทของการเรียนรู้ของเครื่องที่ใช้สถาปัตยกรรมอัลกอริทึมแบบแบ่งชั้นเพื่อวิเคราะห์ข้อมูลในโมเดลการเรียนรู้เชิงลึกข้อมูลจะถูกกรองผ่านการเรียงซ้อนของหลายเลเยอร์โดยแต่ละเลเยอร์ที่ต่อเนื่องจะใช้เอาต์พุตจากเลเยอร์ก่อนหน้าเพื่อแจ้งผลลัพธ์ แบบจำลองการเรียนรู้แบบลึกสามารถมีความแม่นยำมากขึ้นเมื่อประมวลผลข้อมูลมากขึ้นโดยการเรียนรู้จากผลลัพธ์ก่อนหน้าเพื่อปรับแต่งความสามารถในการสร้างความสัมพันธ์และการเชื่อมต่อการเรียนรู้อย่างลึกซึ้งนั้นขึ้นอยู่กับวิธีการที่เซลล์ประสาททางชีวภาพเชื่อมโยงถึงกันเพื่อประมวลผลข้อมูลในสมองของสัตว์ เช่นเดียวกับที่สัญญาณไฟฟ้าเคลื่อนที่ผ่านเซลล์ของสิ่งมีชีวิตแต่ละเลเยอร์ต่อมาจะถูกเปิดใช้งานเมื่อได้รับสิ่งเร้าจากเซลล์ประสาทข้างเคียงในเครือข่ายประสาทเทียม (ANNs) พื้นฐานสำหรับโมเดลการเรียนรู้เชิงลึกแต่ละชั้นอาจได้รับมอบหมายส่วนเฉพาะของงานการแปลงและข้อมูลอาจเคลื่อนที่เลเยอร์หลายครั้งเพื่อปรับแต่งและเพิ่มประสิทธิภาพของเอาต์พุตขั้นสูงสุด 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) โดย ใช้เทคนิคการออกแบบ module ที่มีลักษณะทางลัดลงใน network ตัวโครงข่ายนี้ประกอบด้วยกัน 4 block โดยจำนวนที่มีพารามิเตอร์สำหรับฝึกทั้งหมดคือชั้นที่เราใช้เรียกชื่อ เช่น ResNet50 จะหมายถึงจำนวน 50 เลเยอร์ซึ่งจะอธิบายขนาดว่า [3, 4, 6, 3] ซึ่งคือ (3 + 4 + 6 + 3) x 3 = 48 ชั้น + 2 ชั้น = 50 ซึ่ง ResNet ที่นิยมใช้จะเป็น ResNet18, ResNet34, ResNet50, ResNet101 และ ResNet152 ซึ่งในงานวิจัยนี้ได้ใช้ ResNet50 อ้างอิงงานวิจัย

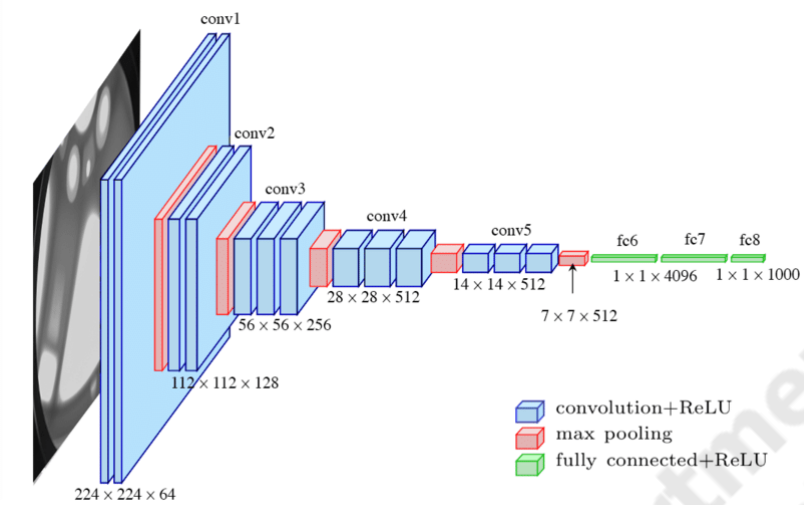


ภาพประกอบที่ 2.16 ResNet50-v2 architecture

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

VGG16

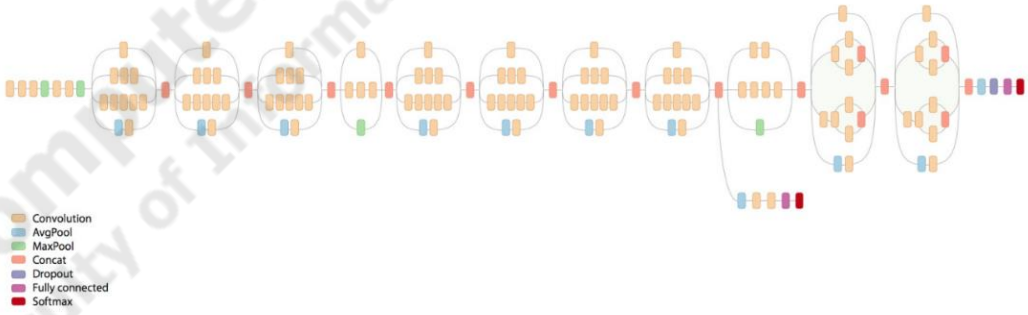
โมเดล VGG ย่อมาจาก Visual Geometry Group ซึ่งเป็นกลุ่มนักวิจัยจาก Oxford ทำการพัฒนาสถาปัตยกรรมนี้ขึ้นมา และที่ได้รับความสนใจมากจากการแข่งขัน ILSVR ปี ค.ศ. 2014 และเป็นที่นิยมจนถึงปัจจุบัน โดยสิ่งที่เป็นจุดเด่นของ VGG16 คือการแทนที่ hyperparameter จำนวนมาก เน้นไปที่การออกแบบ เลเยอร์ conv2D 3x3 pixels, 1 stride และการใช้ same padding และ maxpooling 2x2 pixels, 2 stride แบบเดียวกันตลอดทั้งโครงสร้าง โดยชื่อของ VGG16 หมายถึงมี16 ชั้นที่มีน้ำหนัก เครือข่ายนี้เป็นเครือข่ายที่ใหญ่และมีพารามิเตอร์ประมาณ 138 ล้าน ดังภาพประกอบที่ 2.19



ภาพประกอบที่ 2.17 ภาพโครงสร้าง VGG-16

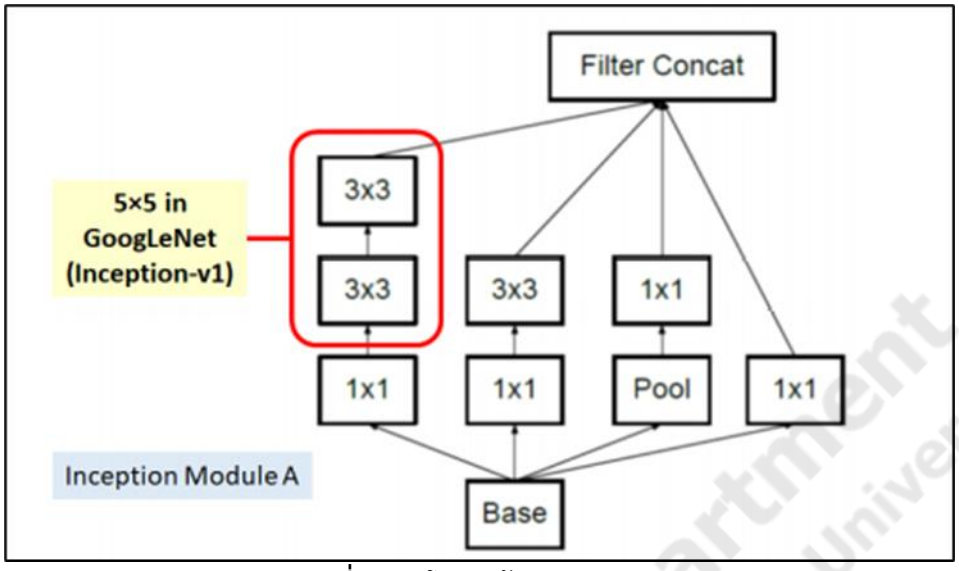
Inception V3

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 จำนวน 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 ดังภาพประกอบที่ 2.20



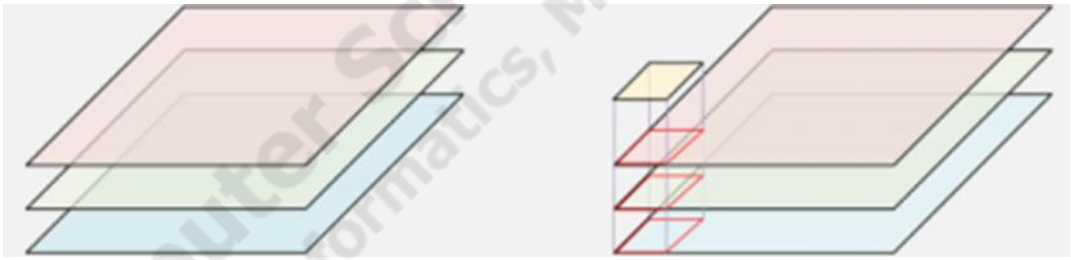
ภาพประกอบที่ 2.18 Inception V3 Architecture

พัฒนาให้ Inception-v3 มี parameter ลดลงจากเดิม แต่ยังคงมีประสิทธิภาพสูงโดยการปรับปรุง convolutions เดิมที่ 5x5 ลงเหลือ 3x3 pixels และ maxpooling เดิมที่ 3x3 ลงเหลือ 2x2 pixels ซึ่งวัตถุประสงค์ของการออกแบบ 1x1 convolution ต้องการให้ shape outputs ขา ออกเป็น tensor เช่นกำหนด (N, F, H, W) ซึ่ง N คือ batch size, F คือ จำนวนของ convolution filters ส่วน H,W คือ spatial ของมิติ ดังภาพประกอบที่ 2.19



ภาพประกอบที่ 2.19 โครงสร้าง inception module

โดยภายในของ inception module กรณี input ข้อมูลเข้ามามี 3 สีคือ RGB ในที่นี้เรา สามารถเปรียบเทียบได้กับการทำ feature map ดังรูปด้านซ้าย ซึ่งสมมุติว่าเราต้องการผลลัพธ์ feature map เพียง 1 ค่านั้นถูกทำได้โดยใช้ convolution ขนาด 1x1 pixels ที่มีการขยับที่ 1 stride **ดังภาพประกอบที่ 2.20** โดยค่า weight ที่คูณในแต่ละชั้นของ RGB นั้นจะมีค่าที่ไม่เท่ากันส่งผลให้ค่าที่ได้ใน แต่ละพิกเซลหลังการทำ feature map นี้จะมีที่เป็นลักษณะเฉพาะ



ภาพประกอบที่ 2.20 การทำ feature map โดยใช้ 1x1 convolution layer

2.1.9. ความรู้ทางการแพทย์ ลักษณะของปอดปกติ

ตำแหน่ง

- ด้านบนสุด อยู่เหนือไหปลาร้า ชิดซี่โครงด้านบน
- ด้านข้างติดกับซี่โครงพอดี ไม่มีช่องว่าง
- ด้านล่างติดกะบังลม
- กะบังลมด้านขวาสูงกว่าด้านซ้ายเล็กน้อย (ไม่เกินซี่โครงหนึ่งช่อง)
- ตรงกลางด้านล่างมีหัวใจอยู่เยื้องไปทางซ้ายเล็กน้อย

ลักษณะสี

- โดยทั่วไปของปอดจะมีสีขาวด้านในเป็นเส้นเล็กๆ กระจายตัวกันเสมอกัน ทั่วปอด
- สีขาวโดยทั่วไป ดูไม่ราบ ชัดจนเกินไป
- ปอดสีไม่ดำสนิท
- ไม่มีสีขาวเป็นกลุ่มก้อน หรือขาวมากเกินไป
- การเดินเส้นของสีขาว ไม่เป็นเส้นตรงในแนวระนาบ หรือแนวขนาน
- เส้นสีขาว จะเดินจากบนลงล่าง แล้วกระจายตัวไปทั่วปอด จากเส้นใหญ่ ไปเล็ก
- ปอดทั้งสองข้างจะมี ขนาด และสีใกล้เคียงกัน

2.2 งานวิจัยที่เกี่ยวข้อง

1.) A. Paisal และ T. Kasetkasem (อนล ไพศาล และ ธีรสิทธิ์ เกษตรเกษม) ได้ศึกษาการคัดแยกการปนของเมล็ดพันธุ์ถั่วเขียว โดยการวิเคราะห์จากภาพถ่าย ผลของงานวิจัยแสดงให้เห็นว่าการคัดแยกเมล็ดพันธุ์จากภาพถ่ายเมล็ด จำนวน 200 เมล็ด เป็นพันธุ์ชัยนาท 72 และพันธุ์กำแพงแสน 2 ระบบสามารถจำแนกภาพของเมล็ดพันธุ์ถั่วเขียวพันธุ์ชัยนาท 72 และพันธุ์กำแพงแสน 2 ที่ปนกัน ซึ่งสรุปผลการคัดแยกได้ ถูกต้องมากกว่า 90 %

2.) T. Tathawee และคณะ ได้ศึกษาวิธีระบุชนิดของ กล้วยไม้แล้วพัฒนาระบบการมองเห็ นด้วยคอมพิวเตอร์ ผลการวิจัยพบว่าการพัฒนาเทคโนโลยีการมองเห็นด้วยระบบ คอมพิวเตอร์เพื่อระบุชนิดกล้วยไม้ทั้งหมดสี่ชนิดจากสี่ สกุล ด้วยวิธีการเปรียบเทียบพื้นที่ contour ของสีปรากฏบนภาพดอกกล้วยไม้ การมองเห็นของคอมพิวเตอร์บ่งชี้ว่าพื้นที่ ที่ความยาว คลื่นแบบต่อเนื่อง (λ =400-700 nm) มีประสิทธิภาพสำหรับระบุชนิดกล้วยไม้ทั้งสี่ ชนิดอย่างชัดเจน แต่ในช่วงความยาว คลื่นแบบไม่ต่อเนื่องที่ช่วงสีน้ำเงิน (λ = 475 nm) มีประสิทธิภาพสำหรับระบุกล้วยไม้ทั้งสี่ชนิดได้ดีที่สุด

3.) N. Masunee ได้เสนองานวิจัยการพัฒนาเทคนิค การตรวจสอบคุณภาพหมึกโดยใช้ปริมาณพื้นที่สีที่ปรากฏบน ตัวหมึกเป็นลักษณะเด่นในการจำแนกระดับคุณภาพได้แก่ สี ขาว สีชมพู สีแดง และสีดำซึ่งปริมาณพื้นที่สีที่ปรากฏบนตัว หมึกถูกคำนวณด้วยเทคนิคการประมวลผลภาพเพื่อจำแนก คุณภาพของหมึกโดยใช้การปริมาณข้อมูลพื้นที่สี

4.) S. Aunkaew และคณะ ได้เสนองานวิจัยการตรวจสอบราขาวบนผิวเนื้อยางแผ่น โดยระบบที่พัฒนาขึ้นด้วย เทคนิคการประมวลผลภาพสามารถตรวจสอบและจำแนกยาง แผ่นที่มีราขาวออกจากยางแผ่นดีได้ผิดพลาดน้อยมากและมี ความรวดเร็วเมื่อเทียบกับการตรวจสอบและจำแนกด้วยตา