# บทที่ 2

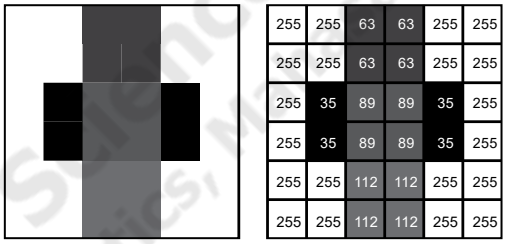
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 การประมวลผลภาพดิจิทัล (Digital Image Processing)

#### 2.1.1.1 ภาพดิจิทัล (Digital Image)

ภาพดิจิทัล [1] ในชีวิตประจำวันของมนุษย์มองเห็นภาพจากการที่แสงตกกระทบสิ่งต่าง ๆ สะท้อนเข้าสู่ดวงตาแล้วส่งข้อมูลภาพที่มองเห็นไปที่สมอง มนุษย์จึงรับรู้ว่าวัตถุที่ดวงตาเห็นคืออะไร มีขนาด รูปร่าง รูปทรง และพื้นผิว อย่างไร หากนำเครื่องมือวัดแสงมาวัดปริมาณแสงจากการที่แสงตกกระทบสิ่งต่าง ๆ จะได้เป็นเลขระบบตัวเลขที่มีค่าแตกต่างกันหลากหลายโดยค่าในแต่ละช่องจะแสดงถึงคุณสมบัติ ของจุดภาพ (Pixel) หรือค่าความเข้มของสี (Intensity) และตำแหน่งของช่องอาร์เรย์เป็นตัวกำหนดตำของจุดภาพ




**ภาพประกอบที่ 2.1.1** แสดงการเก็บค่าของแสงในแต่ละจุดภาพ

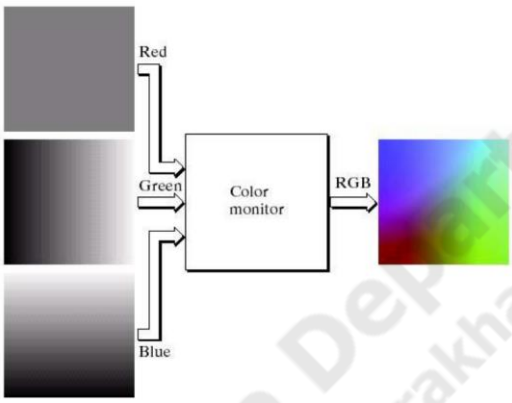
การจัดเก็บรูปภาพจะจัดเก็บข้อมูลของภาพไว้ในรูปของเมทริกซ์ สมมติว่าภาพมีขนาด m x n (แถว x คอลัมน์) ค่าของจุดภาพของภาพที่แทนด้วยเลขจำนวนเต็ม จะสามารถอ้างอิง กับเมทริกซ์ ได้ดังเช่น จุดภาพที่อยู่ ณ ตำแหน่งจุดกำเนิดมีค่า (x, y) = (0, 0) จะเท่ากับเมทริกซ์แถวที่ 0 คอลัมน์ที่ 0 และพิกัดที่อยู่แถวแรกมีค่า (x, y) = (0, 1) จะเท่ากับเมทริกซ์แถวที่ 0 คอลัมน์ ที่ 1 จะแสดงให้เห็นการอ้างอิงจุดภาพของภาพ ตัวเลขเหล่านี้จัดเรียงและเก็บไว้ในหน่วยความจำคอมพิวเตอร์ในรูปแบบอาร์เรย์ จะได้ข้อมูลภาพดิจิทัล (Digital Image) ซึ่งจะมีการจัดเก็บภาพแต่ละชนิดต่างกัน ขึ้นอยู่กับระบบสีของภาพ

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,n-1) \\ \vdots & \ddots & \vdots \\ f(m-0,0) & \cdots & f(m-0,n-1) \end{bmatrix}$$

**ภาพประกอบที่ 2.1.2** พิกัดที่ใช้อ้างอิงถึงภาพดิจิทัล

2.1.1.2 ภาพสี (Color Image)

ในแต่ละพิกเซลจะมีการเก็บความเข้มของแสงแต่ละแถบแสงของแม่สีหลัก 3 สีซ้อนกันคือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) โดยหลักการของระบบพิกัดคาร์ทีเซียน (Cartesian Coordinate System) มากำหนดพื้นที่ของแม่สีแต่ละสีในลักษณะของลูกบาศก์ [1] เวกเตอร์ที่แสดงค่าสีแดง เขียว และ น้ำเงิน แต่ละสีที่ซ้อนกันจะมีความละเอียด (Resolution) อย่างละ 8 บิต รวมทั้ง 3 สีเข้าด้วยกันแล้วได้ 24 บิต สำหรับภาพสี สามารถแสดงสีได้ถึง 16,777,216 สี




**ภาพประกอบที่ 2.1.3** แสดงการประกอบกันของภาพสีแดง เขียว และน้ำเงิน
(ที่มา : Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing)

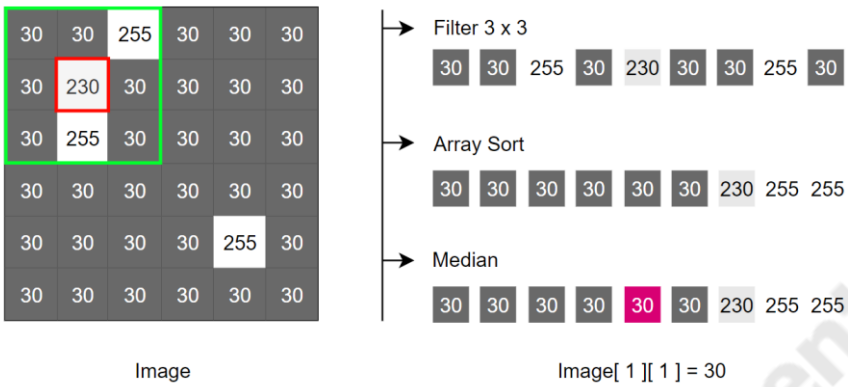
2.1.1.3 การประมวลผลภาพ (Image Processing)

**ภาพประกอบที่ 2.1.4** การประมวลผลภาพ

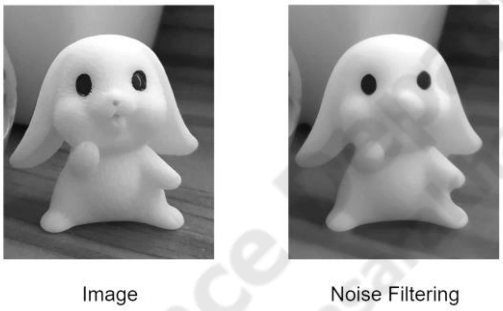
การประมวลผลภาพ (Image Processing) [2] เป็นกระบวนการจัดการและวิเคราะห์รูปภาพในรูปแบบดิจิทัลด้วยคอมพิวเตอร์โดยมีขั้นตอนหลายขั้นตอนที่สำคัญเช่น การปรับให้ภาพมีคุณภาพที่ดีขึ้น การกำจัดสัญญาณรบกวนของภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณโดยสามารถอธิบายขั้นตอนได้ดังต่อไปนี้

(1) การลดสัญญาณรบกวน (Noise filtering)

เป็นกระบวนการการลดค่าของสัญญาณรบกวนจากภาพโดยการนำภาพมาผ่านตัวกรอง (Filter) ซึ่งภาพที่ได้จะมีลักษณะที่เรียบขึ้น ยกตัวอย่างด้วยการลบสัญญาณรบกวนด้วยวิธี Median Filter เป็นวิธีการหาค่ากลาง ณ จุด ๆ หนึ่งของภาพแสดงวิธีการทำงานดังนี้




**ภาพประกอบที่ 2.1.5** วิธีการลดสัญญาณรบกวนด้วย Median Filter




**ภาพประกอบที่ 2.1.6** ผลลัพธ์จาก Median Filter

(2) การดึงค่าคุณลักษณะของภาพ (Image features extraction)

ในการจำแนกประเภท ของวัตถุที่สนใจจากภาพจะต้องดึงค่าคุณลักษณะต่าง ๆ เช่น ขนาดพื้นที่ของวัตถุค่าสีของวัตถุ ขอบภาพของวัตถุ ความกว้างวัตถุ ความยาววัตถุ ตำแหน่งของวัตถุ จำนวนวัตถุจากภาพ เป็นต้น โดยที่คุณลักษณะเหล่านี้จะต้องเป็นคุณลักษณะที่มีความชัดเจนและจำนวนคุณสมบัติเพียงพอต่อการจำแนกประเภทในขั้นตอนต่อไป

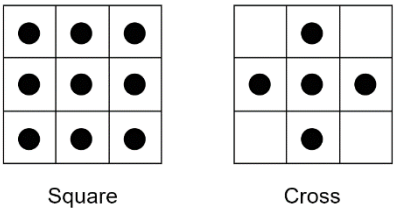

**ภาพประกอบที่ 2.1.7** Image features extraction

(3) การประมวลผลภาพด้วยมอร์โฟโลยี (Morphological)

การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological) [3] โดยมีความสามารถในการย่อขยายจุดพิกเซลของภาพได้ โดยตัวดำเนินการจะมีความสัมพันธ์กับตัวดำเนินการทางตรรกะ

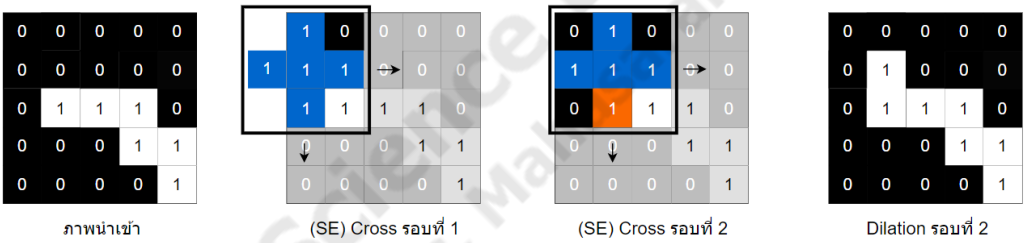
ได้แก่ AND, OR, NOT, XOR, NAND โดยวิธีการทำงานมีสองวิธีหลัก ๆ มี Dilation Operation และ Erosion Operation

การนำการประมวลผลภาพกับรูปร่างและโครงร่างของภาพมาใช้ประมวลผลกับ Structural Element (SE) โดยที่ Structural Element จะมีขนาดโครงสร้างเล็กกว่าขนาดของภาพสามารถเพิ่มขนาดของโครงสร้างได้ขนาด n x n ที่ประมวลผลซึ่งมีหลัก ๆ 2 ประเภทดังนี้

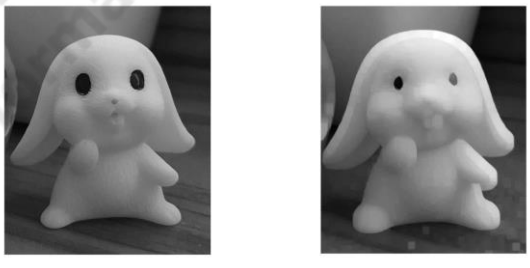



**ภาพประกอบที่ 2.1.8** Structural Element

Dilation Operation คือการใช้เพิ่มขนาดรูปร่างของภาพนำเข้า เมื่อภาพถูกดำเนินการจะทำให้จุดพิกเซลขยายขนาดใหญ่ขึ้น รวมทั้งใช้ในการเชื่อมต่อจุดที่ขาดหายไปได้

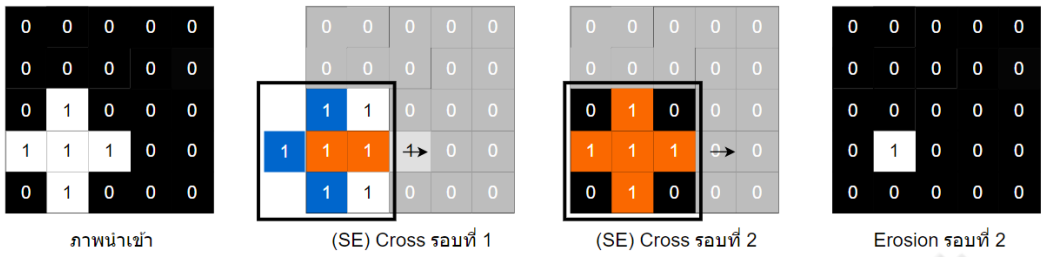



**ภาพประกอบที่ 2.1.9** การทำงานของ Dilation




**ภาพประกอบที่ 2.1.10** Dilation Operation

Erosion Operation คือการนำเนินการย่อขนาดจุดของพิกเซล หรือลดขนาดของวัตถุเมื่อนำมาประยุตก์ใช้กับการกำจัดข้อมูลขนาดเล็ก ๆ ออกจากภาพได้




**ภาพประกอบที่ 2.1.11** การทำงานของ Erosion


**ภาพประกอบที่ 2.1.12** Erosion Operation

(4) การขยายข้อมูล (Data augmentation)

เนื่องจากประสิทธิภาพความแม่นยำของโมเดลในตระกูล Deep Learning นั้นขึ้นกับปริมาณข้อมูลเป็นปัจจัยสำคัญอันดับหนึ่งการ Training Data มาเพิ่มได้ใช้เทคนิคที่สำคัญมากสำหรับ Machine Learning โดยเฉพาะคอมพิวเตอร์วิทัศน์ (Computer Vision) นั่นคือ การขยายข้อมูล (Data Augmentation) [4] หรือ การสร้างภาพใหม่ โดยการดัดแปลงภาพเดิมที่มี เช่น บิด ตัด หมุน เปลี่ยนสี ทำภาพให้มืดหรือสว่างขึ้น หรือใส่สัญญาณรบกวน (Noise) ลงไป จะทำให้ได้รูปภาพแบบต่าง ๆ ไม่จำกัด


**ภาพประกอบที่ 2.1.13** Data Augmentation

การทำ Augmentation นั้นจะต้องไม่เหมือนข้อมูลเดิมจากต้นฉบับมากเกินไป เพราะโมเดลนั้นอาจจะไปเรียนรู้สิ่งที่ไม่สำคัญบนภาพและอาจจะจำสิ่งที่ไม่จำเป็นไป ซึ่งจะทำให้การทำนายผลนั้นใช้ไม่ได้ จะใช้กันอย่างแพร่หลายในข้อมูลรูปภาพ แต่ก็ปัจจุบันมีการศึกษาเกี่ยวกับ Data Augmentation ในข้อมูลแบบอื่น ๆ เช่น ตาราง เสียงพูด และ ข้อความ NLP เช่น เปลี่ยนชื่อตัวละครจากชื่อหนึ่งเป็นอีกชื่อหนึ่งเปลี่ยนสลับคำศัพท์ที่มีความหมายเหมือนกัน

2.1.2 การเรียนรู้ของเครื่อง (Machine Learning)

การเรียนรู้ของเครื่อง (Machine Learning) [5] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ สามารถเรียนรู้ข้อมูลและทำนายข้อมูลได้โดยอัลกอริทึมหรือสามารถเรียนรู้ได้ด้วยตนเอง จะทำงานโดยอาศัยแบบจำลอง (Model) ที่สร้างขึ้นจากชุดข้อมูลตัวอย่างเพื่อการทำนายหรือตัดสินใจในภายหลัง อัลกอริทึมที่ใช้สอนจะถูกแบ่งออกเป็นหมวดหมู่ได้ดังนี้

2.1.2.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

การเรียนรู้แบบมีผู้สอนจะมีการเตรียมข้อมูลตัวอย่างและผลลัพธ์ที่ผู้สอนต้องการ และนำข้อมูลสอนให้คอมพิวเตอร์เรียนรู้ด้วยผลลัพธ์ที่นำเข้าไป คอมพิวเตอร์จะเชื่อมโยงข้อมูลและสร้างเป็นโมเดลไว้ทำนายผลลัพธ์

2.1.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

การเรียนรู้แบบไม่มีผู้สอนจะถูกพัฒนาให้ใกล้เคียงกับการทำงานของสมองมนุษย์มากยิ่งขึ้น โดยคอมพิวเตอร์จะได้รับข้อมูลนำเข้าจะไม่มีผลลัพธ์ให้ จากนั้นกระบวนการเรียนรู้จะใช้หลักทางสถิติหาค่าทางสถิติของชุดข้อมูลที่ฝึกสอน และทำการจัดกลุ่มข้อมูลออกเป็นระดับต่าง ๆ

2.1.2.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

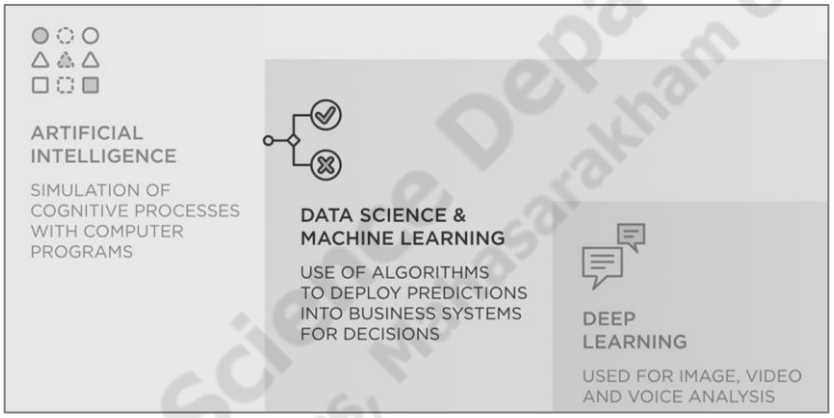
การเรียนรู้แบบเสริมกำลังเป็นการเรียนรู้ที่ลองผิดลองถูกและเรียนรู้ว่าเส้นทางการทำงานแบบใดที่ให้ผลตอบแทนดีที่สุด ตัวอย่างเช่นการเรียนรู้เพื่อเล่นเกม

ตัวอย่างการใช้งานการเรียนรู้ของเครื่อง (Machine Learning) ได้แก่ การประเมินความต้องการของสินค้า คาดการณ์พฤติกรรมของลูกค้า แนะนำผลิตภัณฑ์ที่เหมาะสมให้ลูกค้า ซึ่งการเรียนรู้ของเครื่อง (Machine Learning) จะทำงานได้ดีกับข้อมูลที่เป็นระบบเช่น แบบสอบถาม, สถิติย้อนหลังผู้ป่วยโรคต่าง ๆ เป็นต้น และการทำงานจะให้ผลลัพธ์ที่ดีกับข้อมูลที่ไม่ซับซ้อนแต่เมื่อข้อมูลที่มีความซับซ้อนสูงมากและไม่เป็นระบบ จำเป็นต้องใช้การเรียนรู้เชิงลึก (Deep Learning) เข้ามาช่วยเพราะสามารถเรียนรู้ข้อมูลไปพร้อม ๆ กับพัฒนาตัวเองได้โดยไม่จำเป็นต้องให้การช่วยเหลือจากมนุษย์

2.1.2.4 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก (Deep Learning) เป็นส่วนหนึ่งใน Machine Learning โดยที่การเรียนรู้ของเครื่องใช้โครงข่ายคล้ายกับเซลล์ประสาทแบบสมองของมนุษย์ (Neural Network) เมื่อแบบจำลองสมองมนุษย์ถูกสร้างด้วยคอมพิวเตอร์จึงเรียกว่าโครงข่ายประสาทเทียม (Artificial Neural Network)

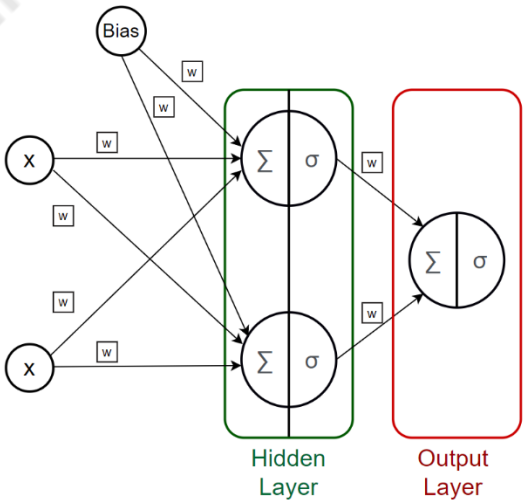
การนำไปใช้การเรียนรู้เชิงลึก (Deep Learning) จะมีประสิทธิภาพมากขึ้นเมื่อนำไปใช้กับข้อมูลที่เป็นรูปภาพ (RGB Image) เพราะจะมีการนำภาพไปค้นหาส่วนที่สำคัญจริง ๆ เท่านั้นออกมาก่อนหรือเรียกขั้นตอนนี้ว่า Convolutional จากนั้นจะลดขนาดภาพให้เล็กลงเรื่อย ๆ เรียกขั้นตอนนี้ว่า Pooling Layer ทำให้ได้ทั้งความเร็วในการประมวลผลและภาพก็ยังคงเหลือส่วนที่สำคัญไว้ แล้วนำส่วนที่สำคัญในภาพเข้าไปทำงานในส่วนโครงข่ายประสาทเทียม (Artificial Neural Network) และจึงได้ผลลัพธ์การทำนายออกมา




**ภาพประกอบที่ 2.1.14** การเรียนรู้เชิงลึก Deep Learning

(ที่มา : www.tibco.com/fr/reference-center/what-is-machine-learning)

2.1.2.5 โครงข่ายประสาทเทียม (Artificial Neural Network, ANN)




**ภาพประกอบที่ 2.1.15** โครงสร้างของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมมีคุณลักษณะที่คล้ายกับการส่งสัญญานของสมองมนุษย์ สามารถเรียนรู้และเก็บความรู้ให้อยู่ในรูปแบบค่าน้ำหนัก (Weight) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่ ๆ เข้าไป ค่าน้ำหนักของความรู้เปรียบเสมือนความรู้ที่รวบรวมไว้เพื่อแก้ปัญหาเฉพาะแบบในสมองของมนุษย์ การทำงานของโครงข่ายประสาทเทียมมีส่วนการทำงานหลัก ๆ ดังนี้

(1) Input

มีหน้าที่ในการรับข้อมูลเข้ามาในโครงข่ายประสาทโดย Input Layer จะเพียงชั้นเดียวเท่านั้น และมีหน้าส่งข้อมูลไปยังชั้นถัดไป (Hidden Layer) สามารถมีได้หลาย Input โดยทั่วไปมักจะมีเท่ากับจำนวนของ Class ข้อมูลที่รับเข้ามาอาจเป็นข้อมูลประวัติผู้ป่วยเบาหวานเช่น น้ำหนัก ส่วนสูง ความดัน น้ำตาลในเลือด เป็นต้น

(2) Weight

เป็นการให้นำหนักของแต่ละ Input ที่ส่งเข้ามาแต่ละเส้นก่อนที่จะเข้าสู่โหนดต่อ ๆ ไป มีหน้าที่บ่งบอกความสำคัญของเส้นแต่ละเส้น

(3) Bias

จะเป็นตัวแปรที่เข้าไป (+) หรือ (-) ค่าที่ได้หลังจากรวมรวมค่าทุก ๆ ค่าก่อนจะเข้า Activation Function ค่า Output ที่ได้หลังจากผ่าน Activation Function จะเปลี่ยนแปลงไปตาม Bias ด้วย เช่น Activation Function แบบ ReLu จะพยายามปรับค่าไม่ให้น้อยกว่า 0 ซึ่งถ้าผลลัพธ์จากการทำงาน Input และ Weight แล้วได้น้อยกว่า 0 นั่นก็หมายความว่าโหนดที่เป็น 0 จะไม่มีการทำงาน อาจส่งผลเสียต่อการเรียนรู้

(4) Output

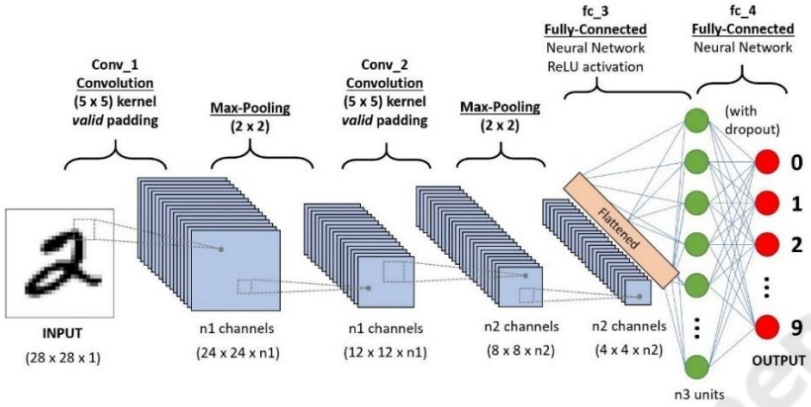
ผลลัพธ์สุดท้ายที่โครงข่ายประสาทเทียมจะให้คำตอบออกมา

(5) Back Propagation

คือการทำงานย้อนกลับเพื่อปรับโครงสร้าง Weight ใหม่เพื่อให้เกิดผลลัพธ์ที่ดีขึ้นกว่าเดิม จะนำค่า Error จาก Output ที่ได้และที่เรียนรู้ย้อนกลับไปปรับ Weight และ Bias ใหม่

2.1.2.6 Convolutional Neural Network (CNN)

เมื่อการทำงานของโครงข่ายประสาทเทียม (Artificial Neural Network) อย่างเดียวยังไม่สามารถทำงานได้ดีกับข้อมูลที่มีความสับซ้อนเช่นรูปภาพ RGB ได้ดีมากนัก Convolutional Neural Network (CNN) จึงเกิดมาแก้ปัญหาด้วยการใช้ Convolution Layer ซึ่งทำหน้าที่สกัดเอาส่วนต่าง ๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่าง ๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำ



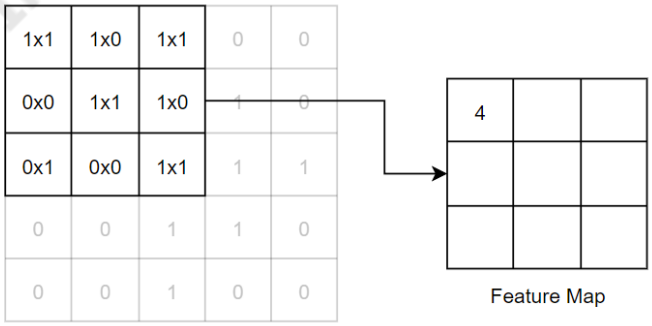

**ภาพประกอบที่ 2.1.16** Convolutional Neural Network

(ที่มา : www.sciencedirect.com/science/article/pii/S1319157821001683)

ใน CNN จะใช้ Convolution Layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling Layer แล้วนำกลุ่ม Layer มาซ้อนต่อ ๆ กันแล้วย้อนกลับไปใช้ Convolution Layer อีกจะทำให้ภาพเล็กลงและเหลือเพียงคุณลักษณะเด่น ๆ ไว้ แล้วสุดท้ายถึงจะนำไปใช้กับโครงข่ายประสาทเทียม วิธีการนำเอาส่วนต่าง ๆ มาประกอบกันนี้เรียกว่าสถาปัตยกรรม (Architecture) ของ CNN ซึ่งมีหลายแบบ เช่น Lenet, Alexnet, VGG, Resnet, Inception Network เป็นต้น สถาปัตยกรรมของ Convolutional Neural Network ประกอบได้ดังนี้

(1) Convolutional

เป็น Layer หลักของ CNN ทำหน้าที่รับ Input เข้ามา จากนั้นจะใช้การดำเนินการทางคณิตศาสตร์เพื่อหาคุณสมบัติที่สำคัญจากรูปภาพการคำนวณจะเริ่มจากการกำหนดค่าใน ตัวกรอง (Filter) หรือ เคอร์เนล (Kernel) ที่ช่วยดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออกมา หรือที่เรียกว่า Feature Map




**ภาพประกอบที่ 2.1.17** การทำ Feature Map

การทำงานของ CNN จะทำการ Sliding Windows (Filter) เพื่อค้นหาองค์ประกอบของภาพ เช่น สี หรือรูปร่าง ทำได้ด้วยสมการดังนี้

$$output\ of\ size = \frac{N - F + 2P}{S} + 1 \quad (1)$$

โดยที่ $N$ คือ ขนาดของภาพ

$F$ คือ ขนาดของ Filter

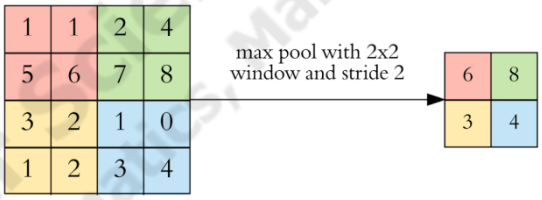
$P$ คือ จำนวนของ Padding

$S$ คือ จำนวนของ Stride (จำนวนของการขยับ Filter)

ปกติแล้วการคำนวณ Convolution จะต้องทำให้ภาพเล็กลงแต่เมื่อมีการตั้งค่า Padding จึงทำให้ Output มีขนาดใหญ่ขึ้นและทำให้เล็กลงในขั้นตอนของ Max Pooling หรือ Pooling Layer แทน ถ้าหากภาพ Input ไม่ได้สนใจว่าขอบภาพที่ไม่ได้นำไปคำนวณในขั้นตอน Convolution มีผลทำให้ผลลัพธ์ออกมาดีขึ้น ก็ไม่จำเป็นต้องตั้งค่า Padding ดังนั้นสามารถใส่เป็น 0

(2) Pooling Layer

Pooling Layer เป็นชั้นที่เชื่อมจาก Convolutional Layer โดยมีเป้าหมายคือทำให้ขนาดของ Feature Map ลดลงด้วยการหาค่าเฉลี่ย (Average Pooling) หรือหาค่าที่สูงที่สุด (Max Pooling) และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนดไว้ โดยขนาดตัวกรองของการหาค่าที่สูงที่สุด (Max Pooling) นิยมเรียกกันว่า Pool Size




**ภาพประกอบที่ 2.1.18** การทำ Pooling Layer

(ที่มา : cs231n.github.io/convolutional-networks/#pool)

(3) Fully Connected Layer

โดยขั้นตอนการหาค่าแต่ละโหนด ในขั้นตอน Fully Connected Layer สามารถทำได้ด้วยสมการดังนี้

$$H_i = \sum_{i=0}^{n-1} (x_i \cdot W_i) \quad (2)$$

โดยที่ $H_i$ คือ ผลลัพธ์ Hidden Layer โหนดที่ $i$

$n$ คือ จำนวน Input ของโหนดก่อนหน้า

$x_i$ คือ ข้อมูลของโหนด Input

$W_i$ คือ ค่าน้ำหนัก

ก่อนจะได้ผลลัพธ์การทำนายต้องนำค่าตัวเลขผ่านขั้นตอนรับผลรวมการประมวลผลทั้งหมดออกมาเป็นค่าความน่าจะเป็นด้วยฟังก์ชัน Softmax Function

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (3)$$

โดยที่ $S$ คือ ผลลัพธ์ Softmax Function มีค่าระหว่าง 0 ถึง 1

2.1.2.7 สถาปัตยกรรม CNN (CNN Architecture)

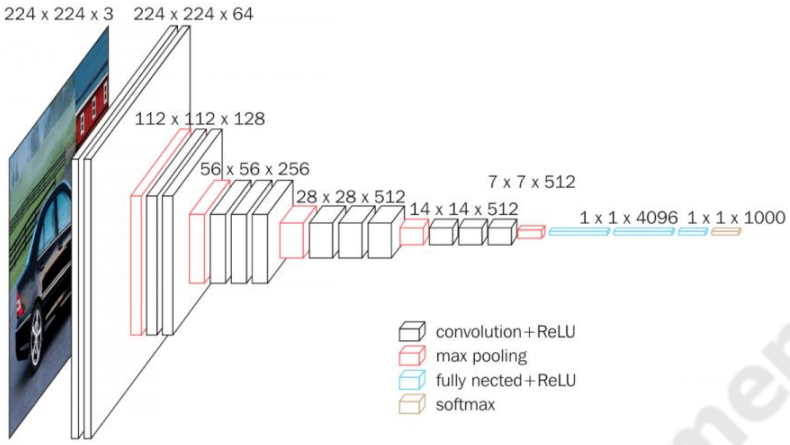
สถาปัตยกรรม CNN (CNN Architecture) ที่มีชื่อเสียงในการแข่งขันเรียนรู้และตรวจสอบความถูกต้องด้วยชุดข้อมูล ImageNet และมีการแจกโมเดลที่ทำการฝึกไว้ล่วงหน้าแล้ว (Pre-Trained Weights) ให้นักพัฒนานำโมเดลไปต่อยอดงานใหม่ ๆ ได้ง่าย โดยห้าอันดับที่ดีที่สุดแสดงตามตารางนี้

**ตารางที่ 2.1** ห้าอันดับที่ดีที่สุดของสถาปัตยกรรม CNN (ปี 2564)

| Model | Size (MB) | Top-1 Accuracy | Top-5 Accuracy | Parameters | Depth |
|---|---|---|---|---|---|
| Xception | 88 | 0.790 | 0.945 | 22,910,480 | 126 |
| VGG16 | 528 | 0.713 | 0.901 | 138,357,544 | 23 |
| VGG19 | 549 | 0.713 | 0.900 | 143,667,240 | 26 |
| ResNet50 | 98 | 0.749 | 0.921 | 25,636,712 | - |
| ResNet101 | 171 | 0.764 | 0.928 | 44,707,176 | - |

โมเดลที่ทำการฝึกไว้ล่วงหน้า (Pre-Trained Weights) ที่จำเป็นต้องฝึกโมเดลไว้ล่วงหน้าเพราะการทำงานของการเรียนรู้เชิงลึกมีความซับซ้อนสูงมากและมีปัญหาการใช้ทรัพยากรจึงใช้เวลานานในการฝึกตั้งแต่ต้นจนจบกระบวนการ และโมเดลจะมีตัวแปรค่าถ่วงน้ำหนัก (Weight) จำนวนมากที่จะต้องทำการสุ่มและปรับใหม่จากการฝึกด้วยข้อมูลมหาศาล โมเดลถึงจะอยู่ในระดับที่เสถียรในการนำไปใช้งานต่อ เมื่อโมเดลอยู่ในระดับที่เสถียรก็จะทำให้นักพัฒนานำไปใช้งานกับข้อมูล Dataset ขนาดเล็กได้และใช้เวลาในการฝึกไม่นาน

ในงานวิจัยนี้ใช้สถาปัตยกรรม CNN (CNN Architecture) แบบ VGG-16 โดยการทำงานของ VGG-16 มีขั้นตอนดังรูปต่อไปนี้




**ภาพประกอบที่ 2.1.19** VGG-16 Model

(ที่มา : www.kaggle.com/blurredmachine/vggnet-16-architecture-a-complete-guide)

VGG-16 เป็นรูปแบบโครงข่ายประสาทเทียมที่นำเสนอโดย Karen Simonyan และ Andrew Zisserman จาก University Of Oxford รูปแบบโครงข่ายประสาทเทียมมีโครงสร้างดังนี้

| Layer | | Feature Map | Size | Kernel Size | Stride | Activation |
|---|---|---|---|---|---|---|
| Input | Image | 1 | 224 x 224 x 3 | - | - | - |
| 1 | 2 X Convolution | 64 | 224 x 224 x 64 | 3x3 | 1 | relu |
| | Max Pooling | 64 | 112 x 112 x 64 | 3x3 | 2 | relu |
| 3 | 2 X Convolution | 128 | 112 x 112 x 128 | 3x3 | 1 | relu |
| | Max Pooling | 128 | 56 x 56 x 128 | 3x3 | 2 | relu |
| 5 | 2 X Convolution | 256 | 56 x 56 x 256 | 3x3 | 1 | relu |
| | Max Pooling | 256 | 28 x 28 x 256 | 3x3 | 2 | relu |
| 7 | 3 X Convolution | 512 | 28 x 28 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 14 x 14 x 512 | 3x3 | 2 | relu |
| 10 | 3 X Convolution | 512 | 14 x 14 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 7 x 7 x 512 | 3x3 | 2 | relu |
| 13 | FC | - | 25088 | - | - | relu |
| 14 | FC | - | 4096 | - | - | relu |
| 15 | FC | - | 4096 | - | - | relu |
| Output | FC | - | 1000 | - | - | Softmax |

**ภาพประกอบที่ 2.1.20** โครงสร้างของ VGG-16

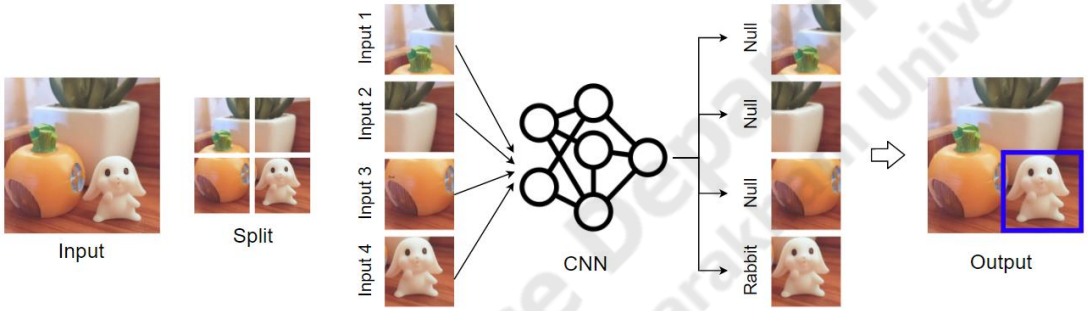
(ที่มา : www.kaggle.com/blurredmachine/vggnet-16-architecture-a-complete-guide)

ลักษณะการทำงาน Input เป็นภาพ RGB 244 x 224 x 3 ซึ่งส่งต่อไปที่เลเยอร์ Convolutional ไปเรื่อย ๆ ทำจำนวน 2 ถึง 3 ครั้ง จะได้ 64 ถึง 512 Feature Map และ Filter ที่มีขนาด 3 x 3 และใช้วิธีการขยับ (Stride) ทีละ 1 ก่อนจะย้ายไปทำขั้นตอน Convolutional จะทำการหาคุณลักษณะด้วย Max Pooling ขนาด Filter 3 x 3 และใช้วิธีการขยับ (Stride) ทีละ 2 ซึ่งจะทำให้ภาพมีขนาดที่เล็กลงวัดตามขนาดของ Size ที่จะมีตัวเลขที่เล็กลงเรื่อย ๆ สุดท้ายจะเหลือเพียง Feature Map จากนั้นก่อนจะส่งไป Fully Connected จะนำภาพที่เหลือแผ่ออกเป็นแนวตั้ง (Flatter Layer) ได้จำนวน 25,088

และสุดท้าย Output จะเหลือ 1,000 คำตอบที่เป็นไปได้ ด้วยการทำ Activation Function แบบ Softmax

2.1.3 การตรวจจับวัตถุด้วยการเรียนรู้เชิงลึก (Object Detection)

การตรวจจับวัตถุเป็นเทคโนโลยีในทางคอมพิวเตอร์ที่ใช้ Deep Learning และ Image Processing โดยระยะแรกสุดในการพยายามทำให้คอมพิวเตอร์สามารถตรวจจับวัตถุได้ จะใช้วิธีแบ่งภาพออกเป็นหลาย ๆ ส่วนแล้วนำภาพทั้งหมดตรวจสอบด้วย Convolutional Neural Network (CNN) ทีละภาพ [6]

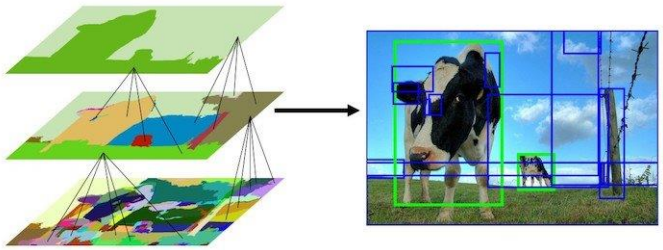



**ภาพประกอบที่ 2.1.21** ระยะแรกของการตรวจจับวัตถุ

ปัญหาในการใช้วิธีนี้คือ ขนาดของการตัดภาพ (Split) เมื่อต้องการความละเอียดสูงในการตรวจจับวัตถุ ต้องตัดภาพ (Split) ที่เยอะมากขึ้น และการตรวจจับจะเกิดขึ้นในทุก ๆ ภาพ ซึ่งอาจส่งผลทำให้เกิดการใช้ทรัพยากรอย่างเกินความจำเป็น

2.1.3.1 R-CNN

การตรวจจับวัตถุด้วยวิธี Selective Search เป็นวิธีการแยกองค์ประกอบของภาพโดยดูจากตำแหน่งของพิกเซลและความเหมือนกันของคุณสมบัติของพิกเซล ถ้าพิกเซลที่อยู่ติดกันและมีคุณสมบัติเหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน จากนั้นนำภาพที่หา Selective Search ทุก ๆ กลุ่มที่เกิดขึ้นไปทำการจำแนกประเภทด้วย CNN เพื่อแก้ปัญหาการกินทรัพยากรณ์แบบระยะแรกของการตรวจจับวัตถุ

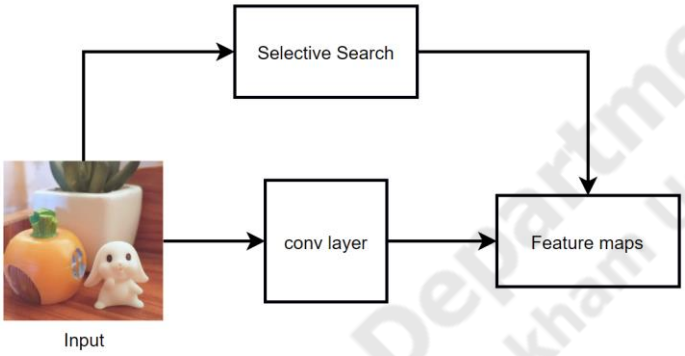


**ภาพประกอบที่ 2.1.22** การทำงาน Selective Search

(ที่มา : learnopencv.com/selective-search-for-object-detection-cpp-python)

2.1.3.2 Fast R-CNN

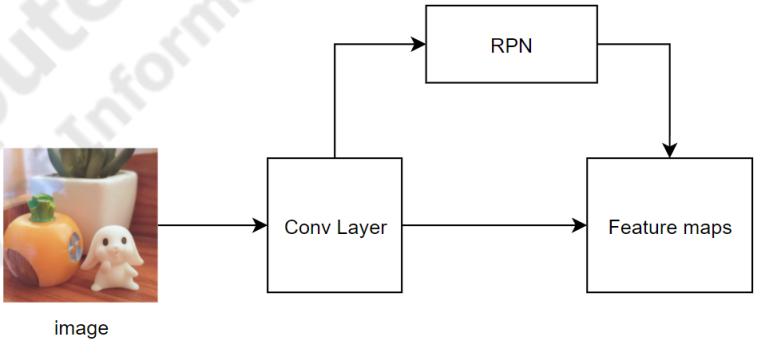
เมื่อจุดอ่อนของ R-CNN ยังคงใช้ทรัพยากรณ์อย่างเกินความจำเป็น จึงมีการเสนอวิธีใหม่ที่เร็วกว่าด้วยการนำภาพนำเข้า (Input Image) ไปประมวลผลด้วย Convolutional Layer ของ CNN ก่อน จะได้ผลลัพธ์เป็นภาพที่มีเพียง Feature Map เท่านั้น ทำให้ภาพมีขนาดที่เล็กลงและทำงานได้เร็วขึ้น จากนั้นนำกรอบสี่เหลี่ยมจากการหา Selective Search รวมเข้ากับภาพที่เป็น Feature Map แล้วจึงนำไปทำการจำแนกประเภทโดยที่ไม่ต้องผ่านขั้นตอน Convolutional Layer อีกครั้ง




**ภาพประกอบที่ 2.1.23** การทำงานของ Fast R-CNN

2.1.3.3 Faster R-CNN

เมื่อการทำงาน Fast R-CNN ยังคงเสียเวลาไปกับการหา Selective Search ซึ่งยังคงเกิดการกินทรัพยากรณ์อยู่ จึงทำให้ Faster R-CNN เกิดขึ้น และขั้นตอนการทำงานจะไม่ใช้วิธี Selective Search แล้ว เปลี่ยนไปใช้โครงข่ายเสนอพื้นที่ (Region Proposal Network, RPN) แทน ซึ่งเป็นการทำงานใน GPU โดยเฉพาะ ทำให้ความล่าช้าในการส่งข้อมูลไปมาระหว่าง CPU กับ GPU น้อยลง




**ภาพประกอบที่ 2.1.24** การทำงานของ Faster R-CNN

2.1.4 การประเมินประสิทธิภาพ

การวัดประสิทธิภาพการทำนายของโปรแกรม เปรียบเทียบกับผลลัพธ์จริง ๆ ที่ถูกต้องโดยมีความหมายแต่ละตัวดังนี้

**True Positive (TP)** คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่ามันจริง (IoU มากกว่าหรือเท่ากับเกณฑ์ที่กำหนด)

**True Negative (TN)** คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่ามันไม่จริง (ไม่สามารถใช้ในงานตรวจจับวัตถุได้ เพราะกรอบที่ตรวจจับไม่ได้ถือเป็นเรื่องปกติในงานตรวจจับวัตถุ)

**False Positive (FP)** คือ สิ่งที่โปรแกรมทำนายว่าจริง แต่คนบอกว่าไม่จริง (IoU น้อยกว่าหรือเท่ากับเกณฑ์ที่กำหนด แต่ IoU ต้องมากกว่าศูนย์)

**False Negative (FN)** คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง แต่คนบอกว่าจริง (การทำนายออกนอกค่าความจริง Ground Truth จนไม่สามารถคำนวณหา IoU ได้)

2.1.4.1 Precision and Recall

การหาค่าระลึก (Recall) คือ ค่าที่บอกว่าโปรแกรมทำนายได้จริง เป็นอัตราส่วนเท่าใดของค่าจริงทั้งหมด

$$Recall = \frac{TP}{(TP + FN)} = \frac{TP}{All\ Ground\ Truth} \quad (4)$$

ค่าความแม่นยำ (Precision) คือ ค่าที่โปรแกรมทำนายว่าจริงถูกต้อง

$$Precision = \frac{TP}{(TP + FP)} = \frac{TP}{All\ Predict} \quad (5)$$

2.1.4.2 Average Precision

ความแม่นยำเฉลี่ย (Average Precision) ใช้วัดความแม่นยำของโมเดลตรวจจับวัตถุโดยใช้ค่า Precision และ Recall และเรียงความมั่นใจที่การทำนายสูงสุดตามลำดับ เมื่อกำหนดให้ IoU มากกว่าหรือเท่ากับ 0.5 จากนั้นปรับค่า Precision ด้วยฟังก์ชัน Interpolated Precision เพื่อหาพื้นที่ใต้กราฟ (AUC) แล้วจะแบ่งกราฟออกเป็นส่วน ๆ เพื่อคำนวณความแม่นยำเฉลี่ย (Average Precision) ในขั้นตอนสุดท้ายดังนี้

$$P_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \quad (6)$$

โดยที่ $p$ คือ Precision

$\tilde{r}$ คือ Recall

การแบ่งกราฟออกเป็นส่วน ๆ เพื่อหาพื้นที่ใต้กราฟ (AUC) ทำได้ดังนี้

$$AP = \sum (r_{n+1} - r_n)\, p_{interp}(r_{n+1}) \quad (7)$$

โดยที่ผลลัพธ์ของ AP คือความแม่นยำเฉลี่ย (Average Precision) ของการทำนายที่ถูก (TP) และผิด (FP) หรือไม่ถูกทำนาย (FN)

## 2.2 งานวิจัยที่เกี่ยวข้อง

จากงานวิจัยของ Zhun Fan, Senior Member, IEEE, Yuming Wu, Jiewei Lu, and Wenji Li เรื่อง “Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network” [7] งานวิจัยนี้ต้องการตรวจจับรอยแตกบนพื้นผิวทางอัตโนมัติ ได้รับการวิจัยมาหลายสิบปี เนื่องจากพื้นผิวทางที่ซับซ้อนในโลกความจริง ในบทความนี้ใช้วิธีการตรวจสอบที่อยู่บนพื้นฐานการเรียนรู้ลึก โดยเฉพาะ Convolutional Neural Network (CNN) ใช้เรียนรู้โครงสร้างของรอยแตกจากภาพดิบโดยไม่ต้องประมวลผลล่วงหน้า หรือ Pre-processing แพทช์ขนาดเล็กจะถูกดึงออกมาจากภาพรอยแตกและนำมาเป็นอินพุตเพื่อสร้างฐานข้อมูลการฝึกสอน CNN ได้รับการฝึกสอนและการตรวจจับรอยแตกของการจำแนก มีปัญหาเรื่องของ multi-label โดยทั่วไปพิกเซลที่แตกจะมีน้อยกว่าพิกเซลที่ไม่แตก เพื่อจัดการกับปัญหาข้อมูลที่ไม่สมดุลจึงมีการปรับเปลี่ยนอัตราส่วนของบวกกับตัวอย่างเชิงลบในบทความนี้ วิธีนี้ได้รับการทดสอบบนฐานข้อมูลสาธารณะสองฐานข้อมูล และเปรียบเทียบกับห้าวิธีที่มีอยู่ ผลการทดสอบแสดงให้เห็นว่ามีประสิทธิภาพดีกว่าวิธีอื่น ๆ

จากงานวิจัยของ Jonathan Long, Evan Shelhamer UC Berkeley, Trevor Darrell เรื่อง “Fully Convolutional Networks for Semantic Segmentation” [8] งานวิจัยนี้ต้องการปรับปรุงแก้ไขเครือข่ายการจำแนกประเภท (AlexNet , VGG net และ GoogLeNet) ลงในเครือข่ายแบบ Fully-convolutional และทำการปรับปรุงแก้ไขผลลัพธ์ที่ได้ใช้กับงาน Segmentation จากนั้นกำหนดสถาปัตยกรรมการข้ามไป ซึ่งรวมข้อมูลของเซแมนติก จากชั้นลึกและชั้นใหญ่ กับข้อมูลจากชั้นเล็กและบาง ๆ เพื่อให้มีการแยกกันอย่างถูกต้องและแสดงรายละเอียดของ segmentations เครือข่าย Fully-Convolutional ทำการแบ่ง Segmentation ของระดับ PASCAL VOC (20% ต่อ 62.2% ค่าเฉลี่ยในปี 2012) NYUDV2 และ SIFT Flow ในขณะที่การอนุมานใช้เวลาน้อยกว่า 1/5 วินาที สำหรับภาพตัวอย่าง