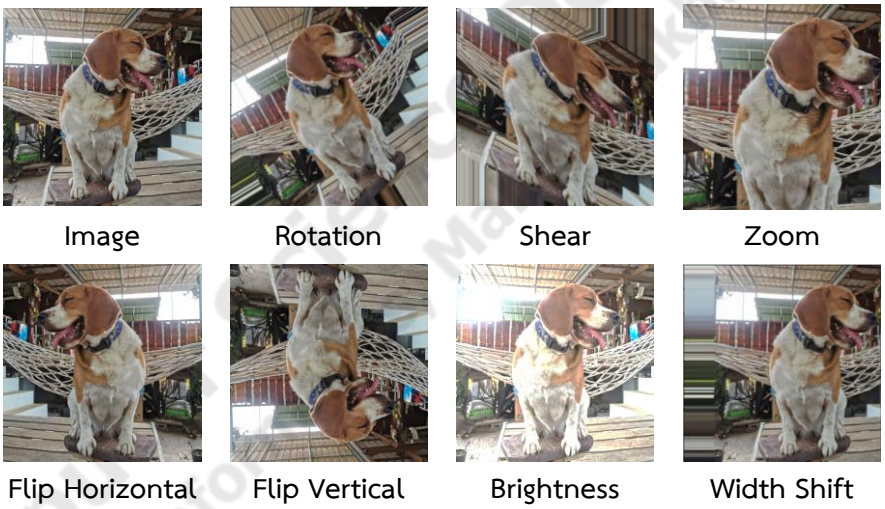
# บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 การเสริมข้อมูล (Data Augmentation)

การเสริมข้อมูล (Data Augmentation) เป็นการเพิ่มข้อมูลรูปภาพ หรือการแปลงภาพในข้อมูลการฝึกอบรม (Training Data) ให้เป็นภาพที่เปลี่ยนแปลง การซูม พลิก เลื่อน ตัด หมุน หรือเปลี่ยนความสว่างของภาพ จะทำให้ได้ภาพรูปแบบต่างๆไม่จำกัด เนื่องจากความแม่นยำของโมเดลขึ้นอยู่กับปริมาณข้อมูลเป็นปัจจัยสำคัญหลัก เพื่อเพิ่มความน่าเชื่อถือให้กับชุดข้อมูล




**ภาพประกอบที่ 2.1** ตัวอย่าง Data Augmentation

### 2.1.2 การเรียนรู้เชิงลึก (Deep Learning) [20]

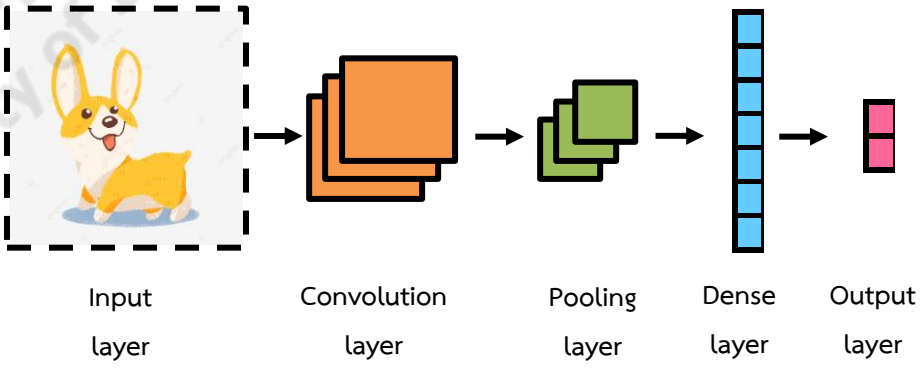
Deep Learning คือวิธีการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาท (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่างที่ป้อนเข้ามา และทำการประมวลอัตโนมัติเพื่อหาข้อมูลตัวอย่างที่จำเป็นในการตรวจจับรูปแบบหรือจัดหมวดหมู่ข้อมูล ความสามารถในการเรียนรู้ คุณลักษณะอัตโนมัติ จากนั้นข้อมูลจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัดหมวดหมู่ข้อมูล (Classify the Data)

การเรียนรู้เชิงลึกใช้ปริมาณข้อมูลที่เพิ่มขึ้นที่เหมาะสม ข้อมูลทั้งหมดที่รวบรวมใช้เพื่อให้ได้ผลลัพธ์ที่ถูกต้องผ่านรูปแบบการเรียนรู้แบบวนซ้ำ การวิเคราะห์ชุดข้อมูลขนาดใหญ่ซ้ำแล้วซ้ำอีกช่วยลดข้อผิดพลาดและความคลาดเคลื่อนในการค้นพบและได้ผลลัพธ์ที่เชื่อถือได้ในที่สุด มีการนำมาใช้งานอย่างแพร่หลายในทางคอมพิวเตอร์ การรู้จำภาพ ทั่วไป การ ประมวลผลภาษาธรรมชาติ (NLP) และการรู้จำเสียงพูด

### 2.1.3 Convolutional Neural Network (CNN) [1] [2]

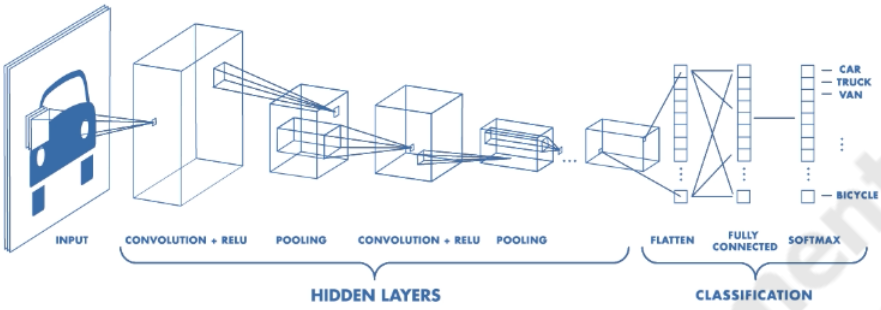
Convolutional Neural Network (CNN) หรือโครงข่ายประสาทเทียม จัดเป็นการเรียนรู้เชิงลึก (Deep Learning) มีการทำงานเริ่มจากการรับข้อมูลเข้ามาผ่านชั้นนำเข้าข้อมูล (Input layer) เพื่อส่งเข้าไปทำการประมวลผลโดยหน่วยย่อยของแต่ละชั้น ที่ชั้นประมวลผล (Hidden Layers) เมื่อประมวลผลเสร็จชั้นสุดท้ายจะส่งไปยังชั้นแสดงผล (Output layer) ซึ่ง CNN สามารถจับคุณสมบัติที่เด่นของข้อมูล โดยใช้ตัวกรองที่เกี่ยวข้อง ในการลดจำนวนพารามิเตอร์ของข้อมูล โดยไม่ลดประสิทธิภาพของคุณสมบัติลง

แต่ถ้าเป็น Deep Learning จะรับข้อมูลดิบเข้าทันที และทำการ ประมวลอัตโนมัติเพื่อหาข้อมูลตัวอย่างที่จำเป็นในการตรวจจับ รูปแบบหรือจัดหมวดหมู่ข้อมูล ความสามารถในการเรียนรู้คุณลักษณะอัตโนมัติทำให้ Deep Learning เป็นประโยชน์อย่างยิ่ง สำหรับการใช้งานในสถานการณ์ต่าง ๆ สิ่งท้าทายที่ยังต้องเผชิญ คือการหาโครงข่ายระบบประสาท ที่เหมาะสมและการค้นหาตัวแปรที่มีผลต่อสมรรถนะในการสอน (Training Performance) ของโครงข่าย ยังคงเป็นเรื่องยากที่จะ รู้ได้ว่า Deep Learning สามารถเรียนรู้คุณลักษณะใดบ้าง นอกจากนี้ Deep Learning ยังมีลักษณะไม่ต่างจาก Machine Learning นั่นคือ ยังไม่สามารถจัดการข้อมูลรับเข้าที่มีความละเอียดเฉพาะทาง (Carefully Crafted Input) จึงอาจทำให้โมเดล เกิดการอนุมานผิดพลาด (Wrong Inferences)




**ภาพประกอบที่ 2.2** Convolutional Neural Network

Computer Science Department
Faculty of Informatics, Mahasarakham University




**ภาพประกอบที่ 2.3** สถาปัตยกรรม Convolutional Neural Network [2]

ขั้นตอนการหลักของ Convolutional Neural Network มีทั้งหมด 3 ชั้น คือ

1. Convolution layer

เป็นการค้นหาคุณลักษณะที่สำคัญของภาพ โดยขั้นตอนนี้จะใช้ตัวกรอง (Filter หรือ Kernel) เพื่อแยกองค์ประกอบต่างๆของภาพ เช่น ขอบ สี เป็นต้น โดยปกติภาพจะมีสีหลัก 3 สี (RGB) คือ สีแดง สีน้ำเงิน และสีเขียว แบ่งเป็น 3 Channel และภาพแสดงเป็นเมทริกซ์ 3 มิติ คือ ความกว้าง ความสูง และความลึก (Width, Height, Depth) โดยความลึกสอดคล้องกับช่องสี (RGB) ซึ่งแต่ละพิกเซลบอกความเข้มของสี ตั้งแต่ 0 ถึง 255 และทำการ Convolution เพื่อเก็บค่าไว้ในเมทริกซ์ชุดใหม่ที่เรียกว่า Feature Map

Stride เป็นตัวกำหนดว่าเราจะเลื่อนตัวกรอง (Filter) ไปด้วย Step เท่าไร สามารถกำหนดค่าของ Stride ให้มากขึ้นได้ ถ้าต้องการให้การคำนวณหา คุณลักษณะมีพื้นที่ทับซ้อนกันน้อยลง เมื่อทำการ Convolution จะทำให้ภาพมีขนาดเล็กลง จึงทำการ Padding ให้ Output มีขนาดใหญ่ขึ้นและทำให้เล็กลงในขั้นตอนของ Pooling layer แทน

การ Convolution ในการส่งข้อมูล โดยแต่ละครั้งใช้ตัวกรองที่แตกต่างกัน ทำให้มีการรวม Feature Map ที่ได้ทั้งหมดเข้าด้วยกัน และแสดงผลข้อมูลสุดท้ายใน Convolution layer ด้วยการค้นหาองค์ประกอบของภาพด้วยการจากการทำงานของ CNN ทำได้ด้วยสมการดังนี้

$$Output\ of\ size = Output\ Feature\ Map = \frac{W-F+2P}{S} + 1 \qquad (1)$$

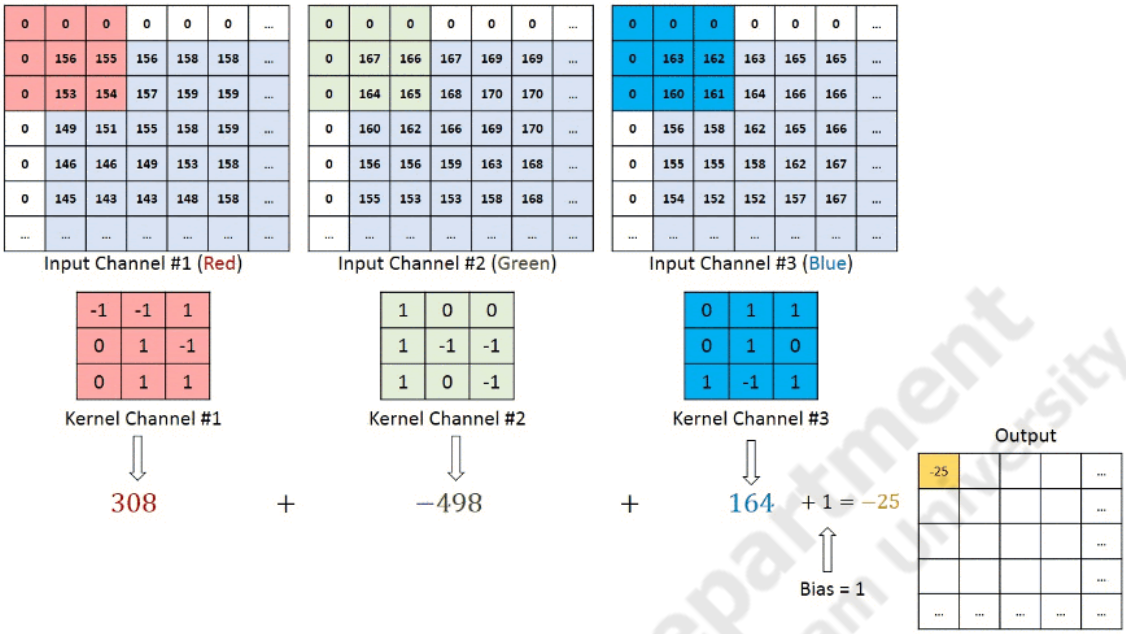
โดยที่ W คือ ขนาดของภาพ

F คือ ขนาดของ Filter

P คือ ขนาดของ Padding

S คือ จำนวนการ Stride

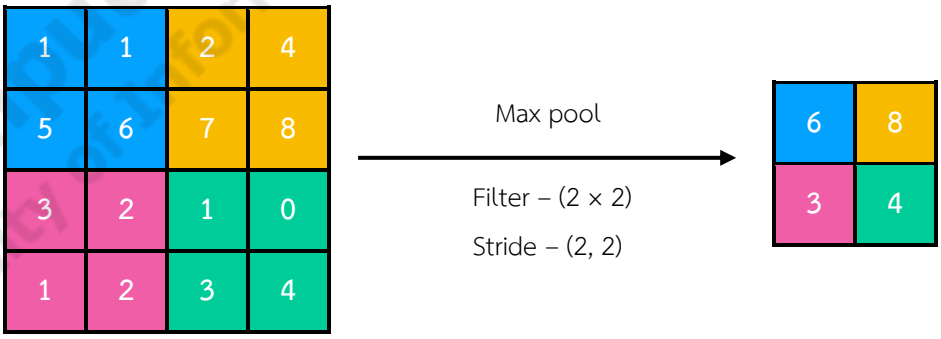
Computer Science Department
Faculty of Informatics, Mahasarakham University




**ภาพประกอบที่ 2.4** การ Convolution บนเมทริกซ์ MxNx3 ที่มีเคอร์เนล 3x3x3 [3]

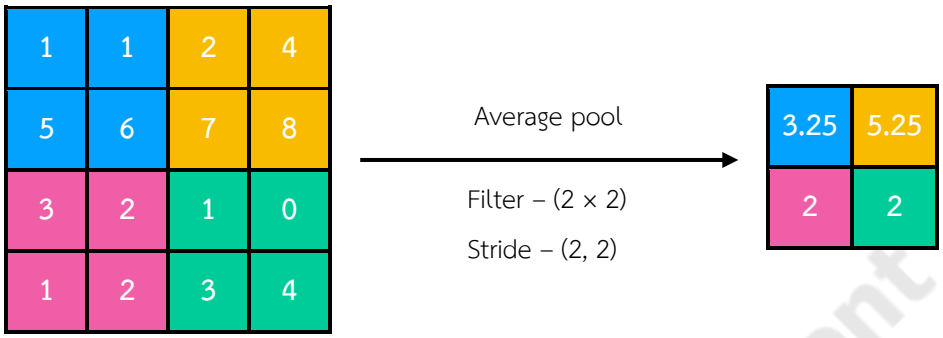
2. Pooling layer

เป็นการลดขนาดของ Feature Map ให้มีขนาดเล็กลงแต่ยังคงรักษาคุณสมบัติของข้อมูลที่สำคัญไว้ โดยมีการหาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลรวดเร็วมากขึ้น ด้วยการหาค่าที่สูงที่สุด (Max Pooling), การหาค่าเฉลี่ย (Average Pooling) ของส่วนที่ครอบคลุมด้วยตัวกรองเก็บไว้ หรือการหาค่าเฉลี่ยรวมของ Feature Map 1 รายการ (Global Average Pooling : GAP)

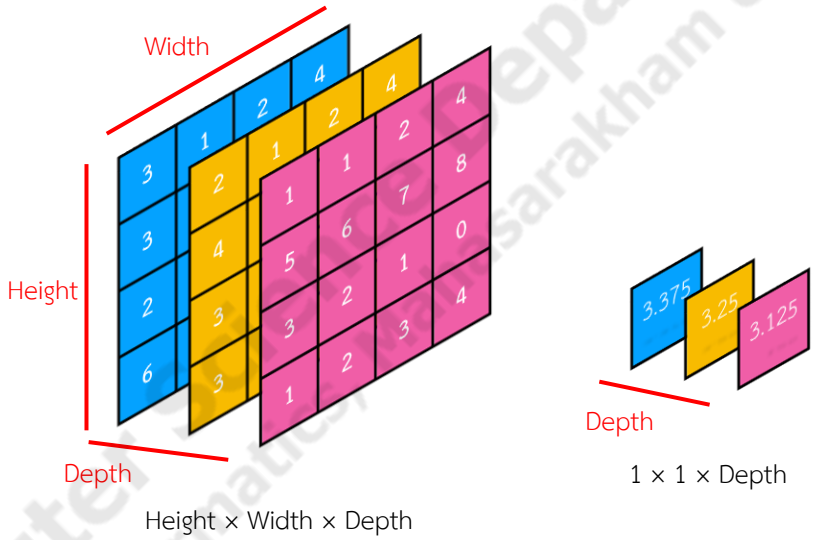



**ภาพประกอบที่ 2.5** การทำ Max Pooling

Computer Science Department
Faculty of Informatics, Mahasarakham University




**ภาพประกอบที่ 2.6** การทำ Average Pooling

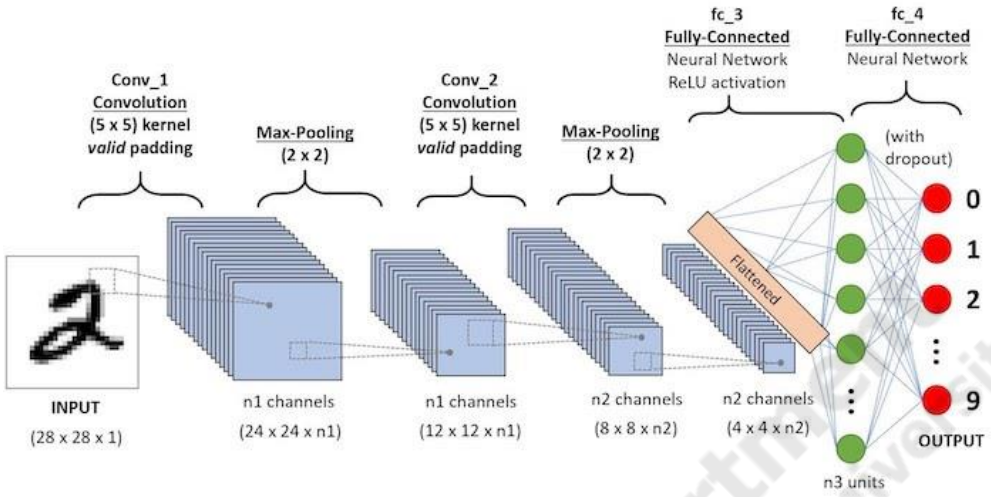



**ภาพประกอบที่ 2.7** การทำ Global Average Pooling

3. Fully Connected Layer

การเชื่อมต่อกันของแต่ละชั้นอย่างสมบูรณ์ จากกระบวนการ 2 ขั้นตอน คือ Convolution layer ทำการสกัดคุณลักษณะที่สำคัญของภาพ และ Pooling layer ทำการลดมิติของ Feature Map ให้มีขนาดเล็กลงแต่ยังคงรักษาคุณสมบัติของข้อมูลที่สำคัญไว้ ซึ่งในขั้นตอนนี้จะเป็นการทำซ้ำตั้งแต่ Convolution layer ไปจนถึง Pooling layer จนกว่าจะเกิดการเชื่อมต่อกันอย่างสมบูรณ์ และทำกระบวนการ 2 ขั้นตอน ด้วย Activation Functions และ Loss Function เพื่อนำลักษณะเด่นที่สำคัญของภาพที่ได้มาทำการสร้างเป็น Neural Network สำหรับการเรียนรู้และทำนายประเภทของภาพ

Computer Science Department
Faculty of Informatics, Mahasarakham University




**ภาพประกอบที่ 2.8** CNN เพื่อจำแนกตัวเลขที่เขียนด้วยลายมือ [2]

ก่อนจะได้ผลลัพธ์การทำนายต้องนำค่าตัวเลขผ่านขั้นตอนรับผลรวมการประมวลผลทั้งหมดออกมาเป็นค่าความน่าจะเป็นด้วยฟังก์ชัน Activation Functions ด้วยสมการดังนี้

- Softmax

$$Softmax(x) = S(x) = \frac{e^{x_i}}{\sum_{i=1}^{k} e^{x_k}}, \quad i = 1, \dots, k \qquad (2)$$

โดยที่ $\boldsymbol{x_i}$ คือ ค่าที่ Input

เมื่อคำนวณครบทุกค่าผลลัพธ์จะได้ตั้งแต่ 0 ถึง 1 ถ้าทั้งหมดรวมกันจะเท่ากับ 1 เสมอ

- Sigmoid

$$Sigmoid = S(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

โดยที่ $\boldsymbol{x}$ คือ ค่าที่ Input

เมื่อคำนวณครบทุกค่าผลลัพธ์จะได้ตั้งแต่ 0 ถึง 1

**Cross Entropy Loss** เป็น Loss Function สำหรับทดสอบ Model แบบ Classification เป็นการนำความน่าจะเป็นของผลลัพธ์ (Output probabilities : P) และประมาณค่าจากค่าความจริง (Actual) เป็นการเรียนรู้และทำนายผลลัพธ์ที่ได้ออกมา ค่าที่คำนวณจะแทนค่าดังสมการ

$$Cross\ Entropy\ Loss = CE = -\sum_{i=1}^{N} \boldsymbol{y_i} \cdot log\ \boldsymbol{\hat{y}_i} \qquad (4)$$

โดยที่ $\boldsymbol{N}$ คือ จำนวนคลาสทั้งหมด

$\boldsymbol{y_i}$ คือ ค่าความน่าจะเป็นสำหรับคลาสที่ i ในค่าความจริง (Actual)

$\boldsymbol{\hat{y}_i}$ คือ ค่าความน่าจะเป็นสำหรับคลาสที่ i ในค่าที่ทำนาย (Prediction)

$\boldsymbol{log}$ คือ ค่า log ฐาน 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.1.4 สถาปัตยกรรม CNN (CNN Architecture)

**ตารางที่ 2.1** ความแม่นยำ 1 อันดับแรกของเครือข่ายต่างๆ [21]

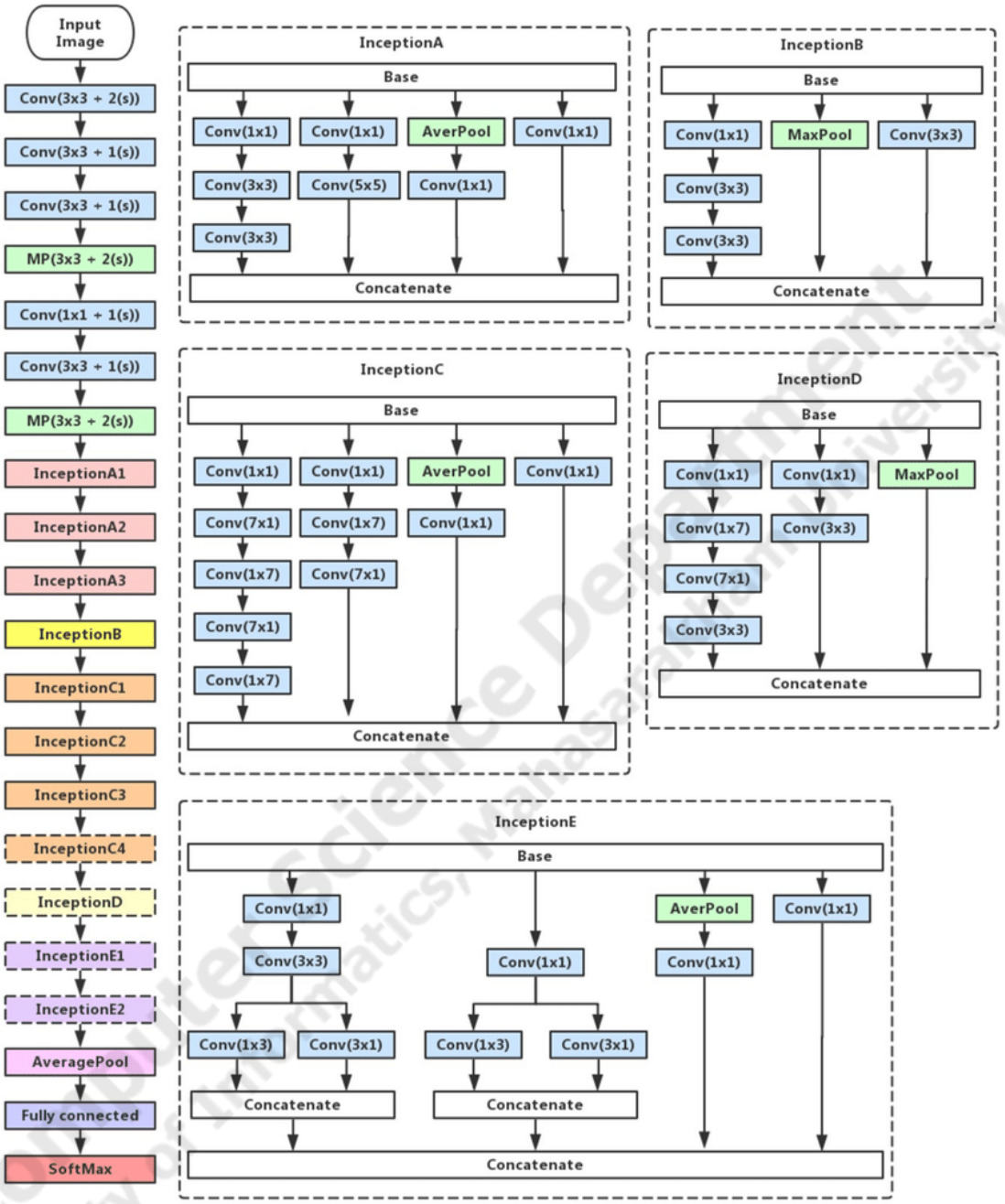
| Model | Proposed year | Number of parameters | Top-1 accuracy (%) |
|---|---|---|---|
| AlexNet | 2012 | 60M | 63.3 |
| VGG-16 | 2014 | 138M | 74.5 |
| VGG-19 | 2014 | 143M | 71.3 |
| ResNet-50 | 2015 | 25M | 77.15 |
| Inception V3 | 2015 | 24M | 78.8 |
| Xception | 2016 | 22.9M | 79.0 |
| InceptionResNetV2 | 2016 | 55.9M | 80.3 |
| EfficientNet-B0 | 2019 | 5.3M | 76.3 |
| EfficientNet-B7 | 2019 | 66M | 84.4 |

2.1.4.1 สถาปัตยกรรม Inception V3 [5]

เป็น Image Classifier Neural Network คือ Convolutional Neural Network และรุ่นที่สามของ Inception Model ซึ่งพัฒนาขึ้นครั้งแรกโดยโมเดล GoogLeNet เป็นโครงข่ายประสาทเทียมขนาดใหญ่และมีระดับความแม่นยำที่ดีตามการจัดประเภทภาพ คุณลักษณะบางอย่างของโมเดลนี้รวมถึงน้ำหนัก (weights), include_top, input_shape เป็นต้น เป็นการจำแนกประเภทที่เป็นลักษณะเฉพาะที่สำคัญที่สุดบางประการสำหรับโมเดลนี้ Inception V3 เป็นโครงข่ายประสาทเทียมที่มีประสิทธิภาพ เนื่องจากช่วยให้เรียนรู้ความซับซ้อนหลายอย่าง และคุณลักษณะที่ยากของแบบจำลองหรือชุดข้อมูล มีขนาดตัวกรองหลายขนาด เช่น 1×1, 3×3, 5×5 ซึ่งสามารถใช้ได้ในระดับเดียวกัน

สำหรับโมเดลที่ได้รับการฝึกแล้ว การเรียนรู้การถ่ายโอน (Transfer learning) จะทำการโอนค่าน้ำหนักสำหรับโมเดลอื่นเพื่อเพิ่มระดับความแม่นยำของโมเดลในแบบจำลอง ชั้นสุดท้ายของ inception V3 จะถูกลบออก และ Deep layer จะถูกเพิ่มเข้าไปที่ส่วนท้ายด้วย Dense layer ซึ่งรับน้ำหนักจาก ImageNet และ Activation Function เป็น ReLU รวมถึง Dense Activation Function คือ Softmax

Computer Science Department
Faculty of Informatics, Mahasarakham University



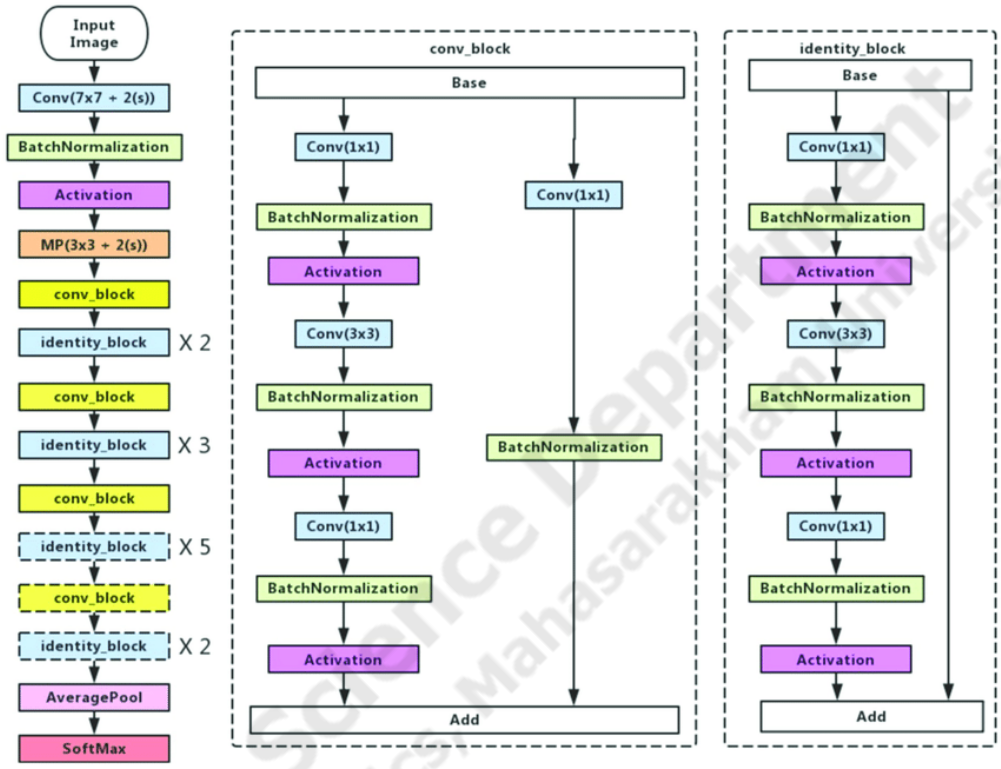

**ภาพประกอบที่ 2.9** โครงสร้าง Inception V3 [6]

2.1.4.2 สถาปัตยกรรม ResNet50 [6] [7]

เป็น Residual Networks แบบ Deep Convolutional Networks มีแนวคิดพื้นฐานคือการเชื่อมต่อแบบ skipblock (หรือการเชื่อมต่อทางลัด) เพื่อให้พอดีกับ Input จากชั้นก่อนหน้าไปยังชั้นถัดไปโดยไม่ต้องปรับเปลี่ยน Input ใดๆ ด้วยสถาปัตยกรรมของ ResNet50 มีโมดูลทางลัดสองประเภทในการใช้งาน ResNet อันแรกคือ identity block ซึ่งไม่มี Convolution layer ที่ทางลัด ในกรณีนี้ Input มีขนาดเท่ากับ Output อีกอันคือ convolution block ซึ่งมี Convolution layer ที่ทางลัด ในกรณีนี้ขนาด Input จะเล็กกว่าขนาด Output ในทั้งสอง block จะมีการเพิ่ม 1×1

Computer Science Department
Faculty of Informatics, Mahasarakham University

Convolution layer ที่จุดเริ่มต้นและจุดสิ้นสุดของโครงข่าย นี่คือเทคนิคที่เรียกว่าการออกแบบ bottleneck ซึ่งลดจำนวน parameters ในขณะที่ไม่ทำให้ประสิทธิภาพของโครงข่ายลดลงมากนัก ในการทดลองของงานวิจัย จะทำการลบโมดูลทางลัดแบบลึกบางส่วนออกและเพิ่ม Classification layer ใหม่ที่ปรับให้เข้ากับชุดข้อมูลของงาน




**ภาพประกอบที่ 2.10** โครงสร้าง ResNet50 [6]

2.1.5 การจำแนกแบบ Fine-grained classification [8]

การจัดประเภทภาพแบบ Fine-grained classification เป็นการแยกความแตกต่างระหว่างคลาสย่อย (Subclass) ของหมวดหมู่ (Class) เดียวกัน เพื่อลดความแปรปรวนภายในคลาสที่สูงและความแปรปรวนระหว่างคลาสที่ต่ำ เป็นการการออกแบบการแสดงคุณลักษณะและตัวแยกประเภทตามคำอธิบายระดับต่ำขั้นพื้นฐาน (Basic low-level descriptors) โดยตัวอย่างภาพอาจคล้ายกันมากและอาจแตกต่างกันในส่วนที่ละเอียดอ่อนบางคุณลักษณะที่แยกแยะได้

Computer Science Department Faculty of Informatics, Mahasarakham University

| Basic-level Classification | Dog | Cat | Chicken |
|---|---|---|---|
| Input | | | |

| Fine-Grained classification | Doberman | Beagle | Golden |
|---|---|---|---|
| Input | | | |
| | | | |
| | | | |

**ภาพประกอบที่ 2.11** Basic-level (Class) และ Fine-grained classification (Subclass)

จากภาพเป็นการแยกหมวดหมู่ย่อย (Sub Class) จากประเภทหมวดหมู่ (Class) ในระดับเดียวกัน จากตัวอย่างภาพจะมีอยู่ทั้งหมด 3 หมวดหมู่ สุนัข แมว และดอกไม้ ในการจัดประเภทแบบ Fine-Grained จะเป็นการจัดประเภทสายพันธุ์ (Breed) สุนัข แมว และชนิด (Species) ดอกไม้ ตามลำดับ

2.1.6 การจำแนกแบบ Multi-label classification [12]

ความยากในการจำแนกประเภทหลายการแสดงผล คือปัญหาของการจำแนกประเภทคลาสที่มีลาเบลหลายรายการให้กับแต่ละข้อมูล เป็นการสร้างแบบจำลองเชิงคาดการณ์ที่เกี่ยวข้องกับการแสดงลาเบลหลายลาเบลของคลาสที่ได้รับข้อมูลเข้ามา เพื่อจดจำคุณสมบัติเฉพาะของข้อมูล ปัญหาหลักเบื้องต้นคือ

การจำแนกประเภทหลายคลาส (Multi class classification) เมื่อรับข้อมูลเข้ามา จะอยู่ในลาเบลเดียวเท่านั้น เป็นปัญหาในการจัดประเภทข้อมูลแต่ละข้อมูลที่นำเข้ามา ซึ่งมีลาเบลเป็นหนึ่งในสองคลาสหรือมากกว่า ซึ่งหมายความว่าแต่ละข้อมูลสามารถอยู่ในคลาสเดียวเท่านั้น ไม่สามารถเป็นได้มากกว่าหนึ่งคลาสในเวลาเดียวกัน ตัวอย่างทั่วไปเช่น บุคคลสามารถเป็นได้ทั้งชาย หญิง แต่ไม่สามารถเป็นพร้อมกัน

การจำแนกประเภทหลายลาเบล (Multi-label classification) เมื่อรับข้อมูลเข้ามา สามารถอยู่ในลาเบลได้มากกว่าหนึ่ง ซึ่งเป็นตัวแปรของสองคลาสขึ้นไปพร้อมกันได้ แต่ละข้อมูลสามารถเป็นได้มากกว่าหนึ่งคลาสในเวลาเดียวกัน ตัวอย่างเช่น สายพันธุ์สามารถอยู่ในประเภทใดประเภทหนึ่งเหล่านี้ ปี๊ก, ไซบีเรียน, บีเกิ้ล, คอร์กี้ หรืออาจเป็นของทั้งสี่ประเภทพร้อมกัน นอกจากนี้ในปัญหาการจำแนกหลายลาเบลคือไม่จำกัดว่าสามารถกำหนดข้อมูลที่นำเข้ามา ว่าแต่ละข้อมูลจะเป็นได้กี่คลาส

**ภาพประกอบที่ 2.12** ประเภทของการจำแนกคุณสมบัติ [13]

2.1.7 Library ที่นำมาใช้ร่วมกับ Python

การเรียนรู้เชิงลึกเป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่เกี่ยวข้องกับการแก้ปัญหาที่ซับซ้อนสูง โดยการจำลองการทำงานของสมองมนุษย์ ในการเรียนรู้เชิงลึกเราใช้โครงข่ายประสาทเทียม

2.1.7.1 TensorFlow [14] [17]

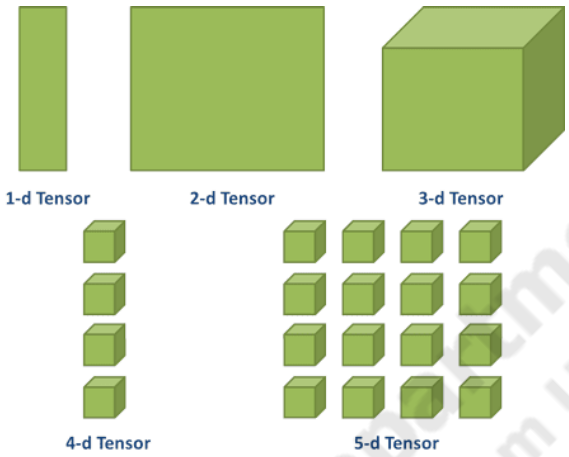
TensorFlow เป็นไลบรารีโอเพนซอร์ซ (Library open source) ได้รับการพัฒนาโดยทีม Google Brain ของบริษัท Google ได้ทำการเปิดตัวเมื่อวันที่ 11 กุมภาพันธ์ 2017 สามารถทำงานบน CPU และ GPUs รองรับระบบปฏิบัติการ Linux, macOS, Windows และ Android

เป็นการทำงานสำหรับการคำนวณเชิงตัวเลข เป็นการสร้างกราฟกระแสข้อมูลและโครงสร้าง เพื่อกำหนดการทำงานของข้อมูลผ่านกราฟ โดยรับข้อมูลโหนดเข้ามาเป็นอาร์เรย์หลายมิติที่เรียกว่าเทนเซอร์ (tensor) ไหลผ่านการเชื่อมต่อการทำงานและแสดงผลข้อมูลแต่ละครั้ง

Computer Science Department
Faculty of Informatics, Mahasarakham University

ส่วนประกอบเบื้องต้นของ TensorFlow คือ

เทนเซอร์ เป็นเวกเตอร์ (vector) หรือเมตริกซ์ (Metrix) ของมิติ n ที่แสดงข้อมูลประเภทเดียวกัน เป็นการคำนวณทั้งหมดที่เกี่ยวข้องกับเมตริกซ์ ที่มีการดำเนินงานในกราฟ




**ภาพประกอบที่ 2.13** รูปร่างเทนเซอร์ [15]

กราฟ เป็นการคำนวณตัวดำเนินการทางคณิตศาสตร์ (operator) เช่น บวก ลบ คูณ หาร เป็นต้น ทั้งหมดภายในกราฟที่เรียกว่าโหนด (node) เป็นการเชื่อมต่อเทนเซอร์เข้าด้วยกัน

2.1.7.2 Keras [18] [19]

Keras เป็นไลบรารีระดับสูง (High-level API) แบบโอเพนซอร์ซ เขียนด้วยภาษา Python ที่พัฒนาโดย Google เมื่อวันที่ 27 มีนาคม 2015 ออกแบบเพื่อสร้างแบบจำลองให้ใช้งานกับโครงข่ายประสาทเทียมได้ง่ายและรวดเร็ว และสามารถทำงานบน TensorFlow หรือ Theano ได้

Keras มีการรองรับ TensorFlow เป็น backend หลัก ซึ่งมีหน้าที่เป็นอินเทอร์เฟซที่เข้าถึงได้และมีประสิทธิภาพสูงสำหรับการแก้ปัญหาการเรียนรู้ของเครื่อง ใช้ในการสร้างแบบจำลองการเรียนรู้เชิงลึก ซึ่งมีการทำงานเพื่อลดความซับซ้อนที่จำเป็นสำหรับการดำเนินงานการประมวล (code) มีประสิทธิภาพและความสามารถในการปรับขนาดโครงสร้างการทำงานให้มีความเรียบง่าย

2.1.8 การประเมินประสิทธิภาพ (Evaluation)

**(1) Confusion Matrix**

ในการประเมินประสิทธิภาพการทำนาย หรือการทำนาย (Prediction) กับโมเดลที่สร้างขึ้นในโปรแกรมแอปพลิเคชั่น เปรียบเทียบกับผลลัพธ์จริงที่ถูกต้อง โดยมีความหมายแต่ละตัวดังนี้

Computer Science Department
Faculty of Informatics, Mahasarakham University

**True Positive (TP)** คือ ค่าทำนายว่าจริง และผลการทำนายบอกว่าจริง

**False Positive (FP)** คือ ค่าทำนายว่าจริง และผลการทำนายบอกว่าไม่จริง

**True Negative (TN)** คือ ค่าทำนายว่าไม่จริง และผลการทำนายบอกว่าไม่จริง

**False Negative (FN)** คือ ค่าทำนายว่าไม่จริง และผลการทำนายบอกว่าจริง

| | | Predicted | | |
|---|---|---|---|---|
| | | Positive (1) | Negative (0) | |
| Actual | Positive (1) | TP | FN | Recall |
| | Negative (0) | FP | TN | |
| | | Precision | | Accuracy |

**ภาพประกอบที่ 2.14** ตาราง Confusion Matrix

**(2) ค่าระลึก (Recall)**

ค่าที่โมเดลทำนายเป็นคลาสที่กำลังพิจารณาถูก (TP) หารค่าเหตุการณ์จริงเป็นคลาสที่กำลังพิจารณาทั้งถูกและผิด (TP+FN) คำนวณโดยสมการดังนี้

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

**(3) ค่าความแม่นยำ (Precision)**

ค่าที่โมเดลทำนายเป็นคลาสที่กำลังพิจารณาถูก (TP) หารค่าที่โมเดลทำนายว่าเป็นคลาสที่กำลังพิจารณาทั้งถูกและผิด (TP+FP) คำนวณโดยสมการดังนี้

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

**(4) ค่าความถูกต้อง (Accuracy)**

การคำนวณความถูกต้องเพื่อเปรียบเทียบประสิทธิภาพการทำนาย จากค่าที่โมเดลทำนายถูกทั้งหมด (TP+TN) หารค่าทั้งหมด (TP+TN+FP+FN) คำนวณโดยสมการดังนี้

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (7)$$

Computer Science Department
Faculty of Informatics, Mahasarakham University