แบบจำลองการเรียนรู้เชิงลึกสำหรับการจำแนกประเภทภาพแบบละเอียด

นายสรนันท์ พยัตศุภร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

DEEP LEARNING MODEL FOR FINE-GRAINED VISUAL CLASSIFICATION

Mr. Soranan Payatsuporn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

หัวข้อวิทยานิพนธ์	แบบจำลองการเรียนรู้เชิงลึกสำหรับการจำแนกประเภทภาพแบบละเอียด
โดย	นายสรนันท์ พยัตศุภร
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
()

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พีรพล เวทีกูล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... กรรมการ
(อาจารย์ ดร.พิตติพล คันธวัฒน์)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.นวลวรรณ สุนทรภิษัช)

สรนันท์ พยัตศุภร : แบบจำลองการเรียนรู้เชิงลึกสำหรับการจำแนกประเภทภาพแบบละเอียด. (DEEP LEARNING MODEL FOR FINE-GRAINED VISUAL CLASSIFICATION) อ.ที่ปรึกษาหลัก : ศ. ดร.บุญเสริม กิจศิริกุล

การจำแนกประเภทภาพแบบละเอียดเป็นปัญหาการจำแนกประเภทภาพที่อยู่ในหมวดหมู่หลักเดียวกัน เช่น ชนิดของนก, รุ่นของรถยนต์และรุ่นของเครื่องบิน โดยปัญหาหลักของการจำแนกประเภทภาพแบบละเอียดคือมีความผันผวนภายในประเภทและความเหมือนระหว่างประเภทสูง ทำให้งานวิจัยส่วนใหญ่มุ่งเน้นไปที่การระบุตำแหน่งของวัตถุหรือชิ้นส่วนสำคัญของภาพด้วยการออกแบบโครงสร้างแบบจำลองที่มีความซับซ้อนเพื่อแก้ปัญหาดังกล่าว ในงานวิจัยนี้ได้นำเสนอวิธีการเพิ่มประสิทธิภาพของความแม่นยำในการจำแนกประเภทซึ่งประกอบด้วยแบบจำลองสองระดับที่ทำหน้าที่แยกกันในการระบุตำแหน่งและจำแนกประเภท โดยการระบุตำแหน่งวัตถุทำหน้าที่หาพื้นที่ในรูปภาพที่มีวัตถุอยู่ด้วยสมมติฐานพื้นที่ต่อเนื่องที่มีขนาดใหญ่ที่สุดบนการรวมของผังฟีเจอร์ ซึ่งสกัดมาจากหลังจากคอนโวลูชันนิวรอลเน็ตเวิร์ค หลังจากนั้นในขั้นตอนการจำแนกประเภท ได้ปรับปรุงฟังก์ชันสูญเสียค่าสูงสุดอย่างอ่อนด้วยการเพิ่มมาจินเชิงมุมปรับค่าได้ในค่ามุมระหว่างฟีเจอร์เวกเตอร์และเวกเตอร์ศูนย์กลางประจำแต่ละประเภทในระหว่างการฝึกสอนแบบจำลอง วิธีการในงานวิจัยนี้สามารถฝึกสอนแบบจำลองได้แบบเอ็นทูเอ็นโดยไม่ต้องใช้กล่องขอบเขตในการฝึกสอนเพิ่มเติม ทั้งนี้ผลการทดลองแสดงให้เห็นว่า เทคนิคที่งานวิจัยนี้นำมาใช้มีประสิทธภาพที่ดีบนชุดข้อมูลสามชุดที่มีการใช้อย่างกว้างขวางในการทดลองเกี่ยวกับการจำประแนกประเภทภาพแบบละเอียด

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6272088521 : MAJOR COMPUTER ENGINEERING

KEYWORD: Fine-grained visual classification, deep learning, convolutional neural network, localization, loss function, Embedding

Soranan Payatsuporn : DEEP LEARNING MODEL FOR FINE-GRAINED VISUAL CLASSIFICATION. Advisor: Prof. BOONSERM KIJSIRIKUL, Ph.D.

Fine-grained visual classification (FGVC) is image categorization task belonging to multiple sub-categories within a same category. It is a challenge task due to high intra-class variance and inter-class similarity. Most exiting methods pay attention to capturing discriminative semantic parts by generate complex model structure. In this research, we propose new methods for improve the classification performance called Efficient Image Embedding, which is integration of two steps model as a localization-classification sub-network, which included localization approach and loss function. The localization approach is used to identify the object region from fine-grained image using concept of the largest component of the feature channel aggregation in an unsupervised fashion. Then classification sub-network following with the loss function, which enhance the discriminative power of the softmax loss by added adaptive penalize to the ground-truth of image in the training state. Our approach can be trained in an end-to-end manner, without the need for any bounding-box/part annotations. Experiment results show our Efficient Image Embedding when implement with base deep convolutional neural architecture can achieve competitive performance on three fine-grained classification datasets.

Field of Study: Computer Engineering Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จไปด้วยความกรุณาเป็นอยางสูงจากศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาที่ให้คำปรึกษา ข้อชี้แนะและความช่วยเหลือในหลายสิ่งหลายอยางจนกระทั่งลุล่วงไปได้ด้วยดี ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและขอขอบคุณอาจารย์เป็นอยางสูงไว้ ณ ที่นี้

ขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร.พีรพล เวทีกูล ประธานกรรมการวิทยานิพนธ์ ดร. พิตติพล คันธวัฒน์ กรรมการวิทยานิพนธ์ และรองศาสตราจารย์ นวลวรรณ สุนทรภิษัช กรรมการภายนอกมหาวิทยาลัย ที่กรุณาให้คำปรึกษาและให้คำแนะนำตลอดจนแก้ไขข้อบกพร่องในการทำวิทยานิพนธ์

ขอขอบคุณ นักศึกษาสาขาวิชาวิศวกรรมคอมพิวเตอร์ ทุกคน ที่คอยเป็นกาลังใจ ร่วมทุกข์ร่วมสุข และให้ความช่วยเหลือเกื้อกูลตลอดมา

และที่ขาดเสียไม่ได้ ขอขอบพระคุณเป็นพิเศษสำหรับความห่วงใยและกาลังใจจาก ครอบครัว ซึ่งเป็นที่รักยิ่ง ที่คอยห่วงใย สนับสนุนการศึกษาเพื่อรอความสำเร็จของผู้วิจัยและเป็นแรงใจสำคัญจนทำให้งานวิจัยครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

สรนันท์ พยัตศุภร

สารบัญ

หน้า

สารบัญภาพ

หน้า

สารบัญตาราง

หน้า

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันแบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) เป็นที่นิยมและได้รับการพัฒนาจนกลายเป็นแบบจำลองที่มีประสิทธิภาพดีที่สุดสำหรับปัญหาหลายรูปแบบ ซึ่งหนึ่งในนั้นคือการจำแนกประเภทภาพ (Image Classification) อย่างไรก็ตามการจำแนกประเภทภาพด้วยแบบจำลองการเรียนรู้เชิงลึกในปัจจุบันส่วนใหญ่ถูกพัฒนาขึ้นโดยมุ่งเน้นไปที่การออกแบบสถาปัตยกรรมการเรียนรู้เชิงลึก (Deep Learning Architectures) สำหรับการจำแนกชุดข้อมูลขนาดใหญ่ (Large Scale Datasets) หรือพัฒนาฟังก์ชันสูญเสีย (Loss Function) เพื่อแก้ปัญหาหรือเพิ่มคุณสมบัติในการฝึกสอนแบบจำลองด้วยเทคนิคต่างๆ เพื่อแบ่งประเภทภาพแบบทั่วไป ซึ่งมีความแตกต่างระหว่างประเภท (Classes) อย่างชัดเจน เช่น แยกประเภทยานพาหนะ (รถ เรือหรือเครื่องบิน) และชนิดของสัตว์ (สุนัข แมวหรือชนิดอื่นๆ)

การจำแนกประเภทภาพแบบละเอียด (Fine-Grained Visual Classification) เป็นหนึ่งในรูปแบบของการจำแนกประเภทภาพ ซึ่งเป็นการจำแนกประเภทของหมวดหมู่ย่อย (Sub-Category) ความแตกต่างระหว่างภาพในแต่ละประเภท (Classes) นั้นมีความละเอียดอ่อนมาก เช่น การจำแนกชนิดของนก (Caltech-UCSD birds[1]) รุ่นของรถยนต์ (Stanford cars[2]) และรุ่นของเครื่องบิน (FGVC aircraft[3]) โดยปัญหาหลักของการจำแนกประเภทภาพแบบละเอียดคือ ความเหมือนระหว่างประเภทสูง (High Inter-class Similarity) เช่น นกทุกชนิดต้องมีจงอยปาก มีสองขาและสองปีก หรือรถยนต์ทุกคันต้องมี 4 ล้อ และมีความผันผวนภายในประเภทสูง (Intra-class Variation) เช่น รูปภาพนก 2 รูปที่อยู่ต่างประเภทที่แตกต่างกันแต่เป็นนกที่มีลักษณะคล้ายกันอย่างมาก ซึ่งถ่ายด้วยพื้นหลัง (Background) เดียวกันและมีลักษณะของวัตถุเหมือนกัน ทำให้ปัญหาการจำแนกประเภทภาพแบบละเอียดต้องอาศัยเทคนิคการเรียนรู้ที่มีประสิทธิภาพมากกว่าการจำแนกประเภทภาพทั่วไป (General Image Classification) การพัฒนาเทคนิคเพื่อช่วยประสิทธิภาพแก่งานวิจัยการจำแนกประเภทภาพแบบละเอียด สามารถไปต่อยอดเพื่อใช้ประโยชน์ได้มากมาย เช่น พัฒนาเทคโนโลยีสำหรับอุตสาหกรรมการแพทย์ การเกษตร ค้าปลีกแบบอัจฉริยะและธุรกิจอีคอมเมิร์ซ[4]

งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทภาพแบบละเอียดในช่วงหลายปีที่ผ่านมาได้การต่อยอดมาจากงานวิจัยที่ใช้เทคนิคแบบจำลองการเรียนรู้เชิงลึก[5-8] ที่ใช้กับชุดข้อมูลรูปภาพขนาดใหญ่ (Large Scale Datasets) โดยแบ่งงานวิจัยได้เป็น 2 ประเภท ได้แก่ งานวิจัยเกี่ยวกับการระบุตำแหน่งและแบ่งประเภทแบบซับเน็ตเวิร์ค (Localization-Classification Sub-Network) ที่จะเน้นไปที่การออกแบบโครงสร้างแบบจำลอง โดยแบ่งเป็นซับเน็ตเวิร์คสองส่วน โดยในคอนโวลูชันนิวรอลเน็ตเวิร์คส่วนแรกจะใช้เพื่อระบุตำแหน่งของชิ้นส่วนที่สำคัญหรือพื้นที่วัตถุภายในรูปภาพ ซึ่งจะต่อกับคอนโวลูชันนิวรอลเน็ตเวิร์คส่วนที่สองเพื่อจำแนกประเภท ซึ่งช่วยให้แบบจำลองสกัดฟีเจอร์ซึ่งเรียนรู้จากตำแหน่งวัตถุที่ถูกต้องและช่วยเพิ่มความแม่นยำ ในการจำแนกประเภท ตัวอย่างงานวิจัย [9-13] อย่างไรก็ตามการจำแนกซับเน็ตเวิร์คจะเป็นการเพิ่มขนาดของแบบจำลองและจำเป็นต้องใช้กล่องขอบเขต (Boundary Box) เป็นข้อมูลเพิ่มเติมเพื่อใช้ในการฝึกสอนแบบจำลอง แต่มีงานวิจัยที่พยายามออกแบบให้สามารถใช้คำตอบแบบหมวดหมู่ (Categorical Labels) ในการฝึกสอนเท่านั้น เช่น งานวิจัยของ Heliang Zheng และคณะ[14] พยายามฝึกสอนแบบจำลองให้สามารถจำแนกประเภทจากฟีเจอร์ที่สกัดมาโดยพิจารณาชิ้นส่วนสำคัญที่แตกต่างกันภายในรูป และงานวิจัยของ Jianlong Fu และคณะ[15] ที่แบ่งการฝึกสอนแบบจำลองเป็นสามระดับและหาทั้งพื้นที่สำคัญและชิ้นส่วนที่เป็นจุดเด่นของวัตถุภายในรูปในระหว่างฝึกสอนแบบจำลองและครอบตัดรูปภาพ เพื่อนำไปฝึกสอนแบบจำลองในระดับถัดไปเพื่อเพิ่มความแม่นยำ

งานวิจัยรูปแบบที่สองคือ แบบจำลองแบบการเข้ารหัสฟีเจอร์แบบเอ็นทูเอ็น (End-to-End Features Encoding Model) เป็นการเรียนรู้และสกัดฟีเจอร์เวกเตอร์โดยตรงจากข้อมูลรูปภาพโดยผ่านแบบจำลองการเรียนรู้เชิงลึก โดยการปรับปรุงโครงสร้างของแบบจำลองหรือเพิ่มชั้น (Layers) หรือฟังก์ชันต่างๆ เช่น Tsung-Yu Lin และคณะ[16] ที่ใช้ผังฟีเจอร์ (Feature maps) ที่สกัดมาจากคอนโวลูชันนิวรอลเน็ตเวิร์คสองเน็ตเวิร์ค และนำผังฟีเจอร์ทั้งสองมาหาผลคูณภายนอก (Outer Product) เพื่อเรียนรู้สถิติและรูปแบบของการเข้ารหัสที่ดีขึ้นและยังช่วยเพิ่มความแม่นยำแต่มีข้อจำกัดที่ขนาดของผลคูณของฟีเจอร์มีขนาดใหญ่อย่างมาก ซึ่ง Yang Gao และคณะ[17] พยายามต่อยอดงานของ Tsung-Yu Lin และคณะ[16] โดยลดขนาดฟีเจอร์ดังกล่าว ด้วยวิธี tensor sketch ซึ่งนอกจากจะช่วยลดขนาดของฟีเจอร์แล้วยังช่วยเพิ่มประสิทธิภาพให้แบบจำลองอีกด้วย และ ตัวอย่างงานวิจัยเพิ่มเติม[18-23] อีกวิธีหนึ่งของงานวิจัยในรูปแบบนี้คือ การออกแบบฟังก์ชันสูญเสีย (Loss

Function) ที่มีวัตถุประสงค์ในการฝึกสอนแตกต่างกัน เพื่อเพิ่มประสิทธิภาพไปจากฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax loss) ซึ่งช่วยในการเรียนรู้จากชุดข้อมูลรูปภาพที่ความเหมือนกันมากของวัตถุในรูปภาพ เช่น Yandong Wen และคณะ[24] ที่ออกแบบฟังก์ชันสูญเสียเพื่อลดความผันผวนภายในคลาส (Intra-class Variance) ของฟีเจอร์ด้วยการพยายามลดระยะห่าง (Distance) ระหว่างฟีเจอร์เวกเตอร์และเวกเตอร์ศูนย์กลางประจำแต่ละคลาส Abhimanyu Dubey และคณะ[25] ที่พยายามลดปัญหาการปรับเหมาะเกินไป (Overfitting) ด้วยการเพิ่มค่าความสับสน (Confusion) บนเวกเตอร์ความน่าจะเป็นที่ใช้ทำนายประเภท Ming Sun และคณะ[26] ออกแบบให้ค่าสูญเสียคำนวณจากการพิจารณาฟีเจอร์บนชิ้นส่วนสำคัญของวัตถุ ซึ่งช่วยลดต้นทุนในการคำนวณ (Computational Cost) และช่วยเพิ่มความแม่นยำ Dongliang Chang และคณะ[27] ออกแบบให้ฟังก์ชันสูญเสียพิจารณามาจากการแบ่งแยกลักษณะเด่นของผังฟีเจอร์และการระบุตำแหน่งชิ้นสำคัญต่างๆได้

วิทยานิพนธ์นี้จะมุ่งเน้นไปที่การพัฒนาแบบจำลองเพื่อแยกแยะฟีเจอร์ด้วยการระบุตำแหน่งและแบ่งประเภทแบบซับเน็ตเวิร์ค โดยใช้ข้อมูลรูปภาพและคำตอบแบบหมวดหมู่ของรูปภาพ (Categorical Label) ในการฝึกสอนเพียงอย่างเดียวและไม่เพิ่มจำนวนพารามิเตอร์ของสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คอย่างมีนัยสำคัญ โดยได้แนวคิดมาจากงานวิจัยที่ใช้สำหรับการจดจำใบหน้า (Face Recognition) ซึ่งเป็นงานวิจัยที่มีปัญหาที่ต้องพิจารณาคล้ายกับการจำแนกประเภทภาพแบบละเอียดคือ ความเหมือนระหว่างประเภทสูง (High Inter-class Similarity) และความผันผวนภายในประเภทสูง (High Intra-class Variation) จึงเกิดแนวคิดในการนำฟังก์ชันสูญเสียที่ใช้ในแบบจำลองการเรียนรู้เชิงลึกสำหรับการจดจำใบหน้า ซึ่งในงานวิจัยของ Jiankang Deng et และคณะ[28] ได้นำเสนอไว้โดยการนำฟังก์ชันสุญเสียค่าสูงสุดอย่างอ่อน (Softmax loss) ที่ใช้การฝึกสอนแบบเอ็นทูเอ็น มาปรับปรุงต่อยอดเพื่อเพิ่มความแม่นยำและยังสามารถประยุกต์ใช้กับคอนโวลูชันนิวรอลเน็ตเวิร์คได้หลากหลายโดยใช้ชื่อว่า ฟังก์ชันสูญเสียมาจินเชิงมุมปรับค่าได้ (Adaptive Angular Margin loss or AAM loss) อีกทั้งยังใช้เทคนิคการระบุตำแหน่งวัตถุ (Localization) ซึ่งไม่ต้องใช้กล่องขอบเขต (Boundary Box) ในการฝึกสอนแบบจำลองอีกด้วยโดยได้แนวคิดมาจากงานวิจัยของ Xiu-Shen Wei และคณะ[29] ระบุตำแหน่งวัตถุจากการหาพื้นที่เชื่อมต่อ (Connected Component) ที่มีขนาดใหญ่ที่สุดบนผังฟีเจอร์ที่ผ่านการปรับรวม (Aggregation) ซึ่งจากสมมติฐานพบว่า พื้นที่เชื่อมต่อที่ใหญ่ที่สุดจะเป็นพื้นที่ที่แบบจำลองเรียนรู้ได้ว่าเป็นตำแหน่งบนรูปภาพที่มีวัตถุ

อยู่ โดยแบบจำลองที่ใช้ในวิทยานิพนธนี้ประยุกต์ใช้เทคนิคทั้งสอง และออกแบบขั้นตอนการฝึกสอนเป็นสองระดับซึ่งทำให้สามารถฝึกสอนได้แบบเอ็นทูเอ็น รวมถึงได้ทำการทดลองเพื่อยืนยันประสิทธิภาพของแบบจำลองด้วยการวัดความแม่นยำและเปรียบเทียบกับงานวิจัยอื่นๆ ซึ่งเทคนิคที่นำเสนอในวิทยานิพนธ์นี้จะช่วยเพิ่มความแม่นยำให้แบบจำลองที่ใช้กับชุดข้อมูลที่มีความคล้ายกันสูง

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อนำเสนอเทคนิคที่ประยุกต์ใช้กับสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์ค ซึ่งสามารถเพิ่มความแม่นยำ (Accuracy) ในปัญหาการจำแนกประเภทภาพแบบละเอียด โดยมุ่งเน้นไปที่การปรับปรุงฟังก์ชันสูญเสียให้แบ่งแยกและสกัดฟีเจอร์จากชุดข้อมูลรูปภาพที่คล้ายกันได้อย่างมีประสิทธิภาพ โดยไม่ทำให้แบบจำลองมีจำนวนพารามิเตอร์เพิ่มขึ้นมากอย่างมีนัยสำคัญเมื่อเทียบกับสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คแบบดั้งเดิม ซึ่งง่ายต่อการใช้งานและยังสามารถประยุกต์ใช้กับสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คได้หลากหลาย

1.3 ขอบเขตการดำเนินงาน

1) ชุดข้อมูลรูปภาพแบบละเอียด (Fine-grained Visual Dataset) ที่ใช้ในงานวิจัยนี้ได้แก่ Caltech-UCSD birds[1], Stanford cars[2] และ FGVC aircraft[3]
2) สถาปัตยกรรมการเรียนรู้เชิงลึกที่ใช้เป็นฐาน คือ วีจีจีเน็ต[5] (VGG16) และเรสเน็ต[6] (ResNet18, ResNet50 and ResNet101)
3) การวัดประสิทธิภาพของแบบจำลอง ใช้การวัดความแม่นยำ (Accuracy)

1.4 ขั้นตอนการดำเนินงาน

1) ศึกษาทฤษฎีเกี่ยวกับแบบจำลองการเรียนรู้เชิงลึก และสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คและงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทภาพแบบละเอียดด้วยเทคนิคต่างๆ
2) ออกแบบเทคนิคที่ใช้เพิ่มควาแม่นยำให้กับการจำแนกประเภทภาพแบบละเอียด
3) กำหนดรูปแบบการทดลอง พัฒนาแบบจำลอง และเก็บผลการทดลอง
4) ทำการทดลอง และปรับปรุงค่าพารามิเตอร์ต่างๆ
5) สรุปผลการทดลอง
6) เขียนบทความเพื่อตีพิมพ์ผลงานทางวิชาการ

7) สรุปผลการวิจัยและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) เพื่อเพิ่มประสิทธิภาพของการจำแนกประเภทภาพภาพแบบละเอียดที่สร้างโดยใช้แบบจำลองการเรียนรู้เชิงลึกที่สามารถนำไปปรับใช้กับสถาปัตยกรรมแบบต่างๆได้
2) นักวิจัยสามารถนำแบบจำลองจากวิทยานิพนธ์นี้ไปใช้แก้ปัญหาในเชิงอุตสาหกรรมต่างๆได้
3) นักวิจัยสามารถนำฟังก์ชันสูญเสียจากวิทยานิพนธ์นี้ไปต่อยอดและใช้ในงานวิจัยหรือชุดข้อมูลอื่นๆที่มีปัญหาใกล้เคียงกันได้

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

การจัดลำดับเนื้อหาในวิทยานิพนธ์นี้แบ่งเป็น 5 บท ประกอบไปด้วย บทที่ 1 บทนำ จะกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขต ขั้นตอนสำหรับการงานวิจัยนี้ บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อเป็นข้อมูลและอ้างอิงความรู้สำหรับการทำงานวิจัยนี้ บทที่ 3 จะกล่าวถึงรายละเอียด และขั้นตอนวิธีการทั้งหมดที่ใช้ในการทำงานวิจัยนี้ บทที่ 4 กล่าวถึงรายละเอียดขั้นตอนและรูปแบบการทดลองและผลการทดลองในงานวิจัยนี้ บทที่ 5 คือข้อสรุปและข้อเสนอแนะจากงานวิจัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้อธิบายถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง ในส่วนของทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ประกอบด้วยทฤษฎีพื้นฐานพื้นฐานของแบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) ฟังก์ชันกระตุ้น (Activation Function) ฟังก์ชันสูญเสีย (Loss Function) การหาค่าที่เหมาะสมที่สุด (Optimization) และทฤษฎีการฝึกสอนแบบจำลองและการแพร่กระจายย้อนกลับ (Back propagation) สถาปัตยกรรมการเรียนรู้เชิงลึก (Deep Learning Architectures) ที่มีประสิทธิภาพและเป็นที่นิยมในปัจจุบัน รวมถึงการวัดประสิทธิภาพของการจำแนกประเภท (Classification Performance Evaluation) และงานวิจัยที่เกี่ยวข้องที่ได้ทำการศึกษาเพื่อนำแนวคิดและวิธีการมาต่อยอดในงานวิจัยนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 แบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model)

เป็นรูปแบบหนึ่งของแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning Model) ที่สร้างขึ้นโดยเลียนแบบกระบวนการคิดและลักษณะเซลล์สมองของมนุษย์ โดยแบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) หรือนิวรอลเน็ตเวิร์คเชิงลึก (Deep Neural Network) เป็นแบบจำลองที่พัฒนามาจากนิวรอลเน็ตเวิร์ค ซึ่งเพิ่มประสิทธิภาพได้ด้วยการเพิ่มชั้น (Layers) ซึ่งในแต่ชั้นสามารถเรียนรู้ฟีเจอร์ที่แตกต่างกันได้โดยทั่วไปประกอบไปด้วย ชั้นการเชื่อมโยงเต็มรูปแบบ (Fully Connected Layer or FC) นิวรอลเน็ตเวิร์คคอนโวลูชัน (Convolutional Neural Network or CNN) นิวรอลเน็ตเวิร์คแบบวกกลับ (Recurrent Neural Network or RNN) ทำให้นิวรอลเน็ตเวิร์คเชิงลึกสามารถเรียนรู้ข้อมูลที่มีความซับซ้อน (Complexity) สูงได้อย่างมีประสิทธิภาพเมื่อเทียบกับนิวรอลเน็ตเวิร์คแบบดั้งเดิม (Neural Network)

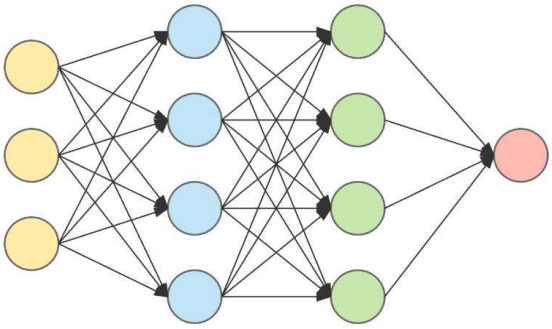
2.1.1.1 นิวรอลเน็ตเวิร์ค (Neural Network)

เพอร์เซปตรอน (Perceptron) หากเปรียบเทียบกับโครงสร้างของเซลล์สมองก็คือหน่วยที่เล็กที่สุดที่เรียกว่านิวรอน (Neuron) โดยนิวรอลเน็ตเวิร์ค 1 ชั้นประกอบไปด้วยเพอร์เซปตรอน

จำนวนหนึ่งซึ่งเชื่อมต่อกับเพอร์เซปตรอนทุกตัวของทั้งชั้นก่อนหน้าและชั้นถัดไปดังรูปที่ 1 โดยเมื่อรับข้อมูลขาเข้าจะสามารถคำนวณผลลัพธ์ขาออกในแต่ละชั้นได้ตามสมการที่ (1) และ (2)

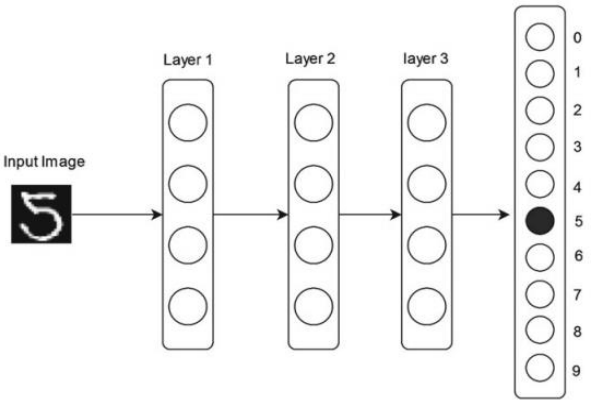
$$z_j^l = \sum_{k=1}^{n} w_{jk}^l a_k^{l-1} + b_j^l \quad (1)$$

$$a_j^l = g(z_j^l) \quad (2)$$



รูปที่ 1 ตัวอย่างการเชื่อมต่อของเพอร์เซปตรอน

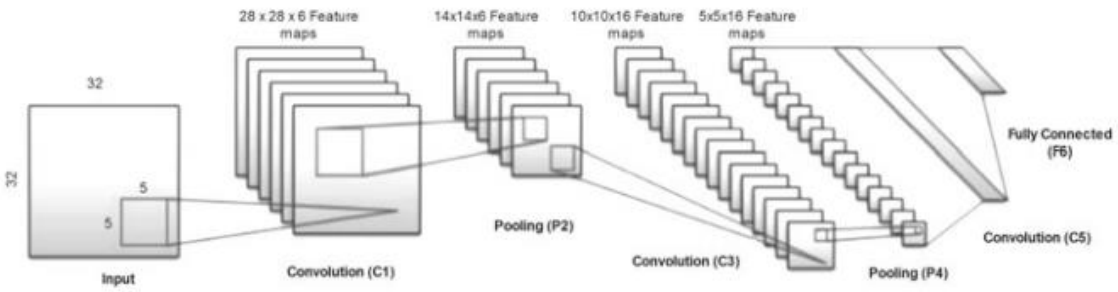
โดยเราเรียกวิธีการคำนวณเพื่อหาผลลัพธ์ขาออกของนิวรอลเน็ตเวิร์คในทุกๆชั้นว่า การป้อนไปข้างหน้า (Feedforward) เมื่อกำหนดให้ w_{jk}^l คือน้ำหนัก (weights) ของเพอร์เซปตรอนตัวที่ j ของชั้นที่ l ที่รับมาจากเพอร์เซปตรอนตัวที่ k ในชั้นที่ l-1 และ b_j^l คือไบแอส (bias) และ n คือจำนวนข้อมูลรับเข้าโดย รับมาจาก a_k^{l-1} หรือก็คือผลลัพธ์ของเพอร์เซปตรอนในชั้นที่ l-1 และ $g(.)$ คือฟังก์ชันกระตุ้น (Activation Function) ที่ทำให้การคำนวณของเพอร์เซปตรอนแตกต่างจากฟังก์ชันเชิงเส้น (Linear) ซึ่งข้อมูลส่งออกของเพอร์เซปตรอนของชั้นก่อนหน้าจะเป็นข้อมูลรับเข้าของเพอร์เซปตรอนในชั้นถัดไป



รูปที่ 2 ตัวอย่างแบบจำลองนิวรอลเน็ตเวิร์คสำหรับจำแนกภาพตัวเลข

2.1.1.2 นิวรอลเน็ตเวิร์คคอนโวลูชัน (Convolutional Neural Network or CNN)

ในนิวรอลเน็ตเวิร์คแบบดั้งเดิม ซึ่งประกอบไปด้วย เพอร์เซปตรอนหลายชั้นระหว่างชั้นข้อมูลรับเข้าและผลลัพธ์หรือที่เรียกว่า ชั้นซ่อน (Hidden Layers) ที่นิวรอลทุกหน่วยจากชั้นนั้นจะถูกเชื่อมเข้ากับทั้งชั้นก่อนหน้า และชั้นถัดไปทำให้จำนวนพารามิเตอร์ของแบบจำลองมีจำนวนมากตามรูปแบบของข้อมูลที่ใช้กับแบบจำลอง โดยเฉพาะอย่างยิ่งข้อมูลประเภทรูปภาพ โดยนิวรอลเน็ตเวิร์คคอนโวลูชันถูกออกแบบมาเพื่อแก้ปัญหานี้ โดยการใช้ตัวกรอง (Filter) เพื่อนำไปสกัดผังฟีเจอร์ (Features Map) ซึ่งจะถูกนำไปใช้เป็นข้อมูลรับเข้าของชั้นถัดไปเช่นเดียวกับเพอร์เซปตรอน แสดงดังรูปที่ 3 ซึ่งเกิดจากหลายๆชั้นมาประกอบกันดังนี้



รูปที่ 3 โครงสร้างของนิวรอลเน็ตเวิร์คคอนโวลูชัน

1) ชั้นคอนโวลูชัน (Convolutional Layer)

เป็นชั้นที่สำคัญชั้นหนึ่งของนิวรอลเน็ตเวิร์คคอนโวลูชัน ซึ่งใช้ตัวกรองดอทกับเมทริกซ์ข้อมูลรับเข้า เปรียบเสมือนว่าตรวจหาฟีเจอร์เป็นพื้นที่จากข้อมูล เช่น ตรวจวัตถุจากรูปภาพ โดยพารามิเตอร์ของตัวกรองจะใช้ร่วมกันสำหรับทุกการทำคอนโวลูชัน (Weight Sharing) ทำให้จำนวนพารามิเตอร์ลดลงอย่างมาก เมื่อเทียบกับเพอร์เซปตรอน กำหนดให้ข้อมูลรับเข้าแทนด้วย เมทริกซ์ A และแทนตัวกรองด้วยเมทริกซ์ K ซึ่งมีขนาด $h\ x\ w$ การทำคอนโวลูชันจะทำได้ตามสมการ *(3)*

$$(A * K)(x, y) = \sum_{i=1}^{h} \sum_{j=1}^{w} A(x + i,\ y + j)K(i, j) \quad (3)$$

โดยในการทำคอนโวลูชัน จะต้องมีการกำหนดค่าพารามิเตอร์งนี้

- ขนาดของตัวกรอง (Filter Size)

  คือความกว้างและความสูงของตัวกรองนำมาใช้ในการทำคอนโวลูชัน โดยปกติแล้วจะมีค่าไม่เกินความกว้างและสูงของข้อมูลรับเข้า ในสมการที่ (3) ก็คือค่า h x w โดยรูปที่ *4* คือตัวอย่างการทำคอนโวลูชันด้วยตัวกรองขนาด *3x3*

- จำนวนตัวกรอง (Number of Filter)

  การทำคอนโวลูชันในแต่ละชั้นสามารถมีตัวกรองได้มากกว่าหนึ่งตัว เพื่อให้แบบจำลองสามารถค้นหาฟีเจอร์ได้หลากหลายมากขึ้น เนื่องจากตัวกรองแต่ละตัวจะมีค่าน้ำหนักที่ต่างกัน

- จำนวนช่องสัญญาณ (Channel)

  คือค่าแสดงความลึกของข้อมูล ซึ่งมีได้มากกว่าหนึ่ง เช่น รูปภาพโดยปกติจะมีจำนวนช่องสัญญาณเท่ากับ 3 ช่องซึ่งแทนด้วยสีแดง เขียว และน้ำเงิน (RGB)

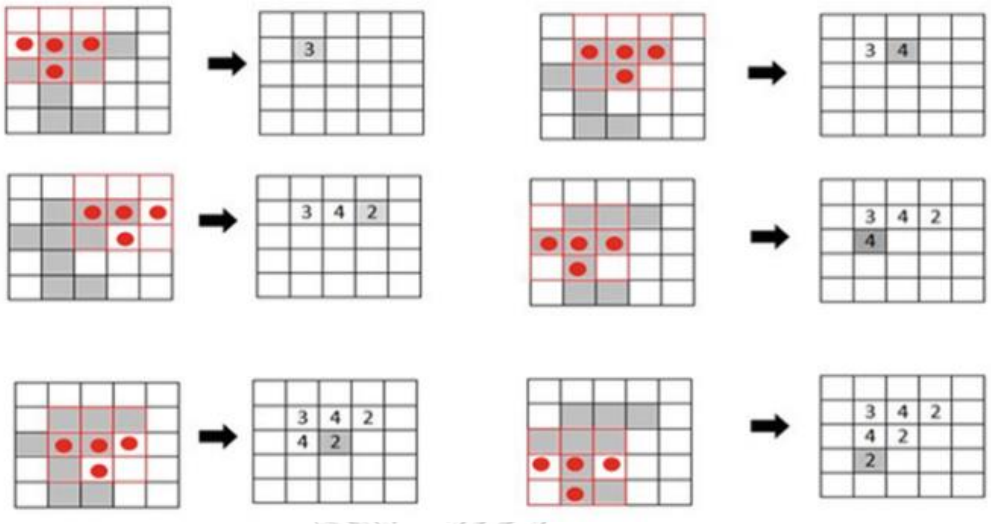
- ขนาดการก้าวข้าม (Stride Size)

  คือจำนวนช่องของข้อมูลรับเข้า ที่จะทำการเลื่อนไปเมื่อทำการหาผลลัพธ์ของการคอนโวลูชันในแต่ละช่อง โดยทั่วไปเมื่อขนาดของการก้าวข้ามเป็น 1 จะเป็นคอนโวลูชันแบบปกติ

- การเสริมเติม (Padding)

  คือการเติมพื้นที่ เพื่อให้สามารถทำคอนโวลูชันได้ตามขนาดที่ต้องการ โดยที่พื้นที่ส่วนที่เกิน ออกไปนั้นจะมีการแทนค่าของข้อมูล ณ ช่องนั้นด้วย 0

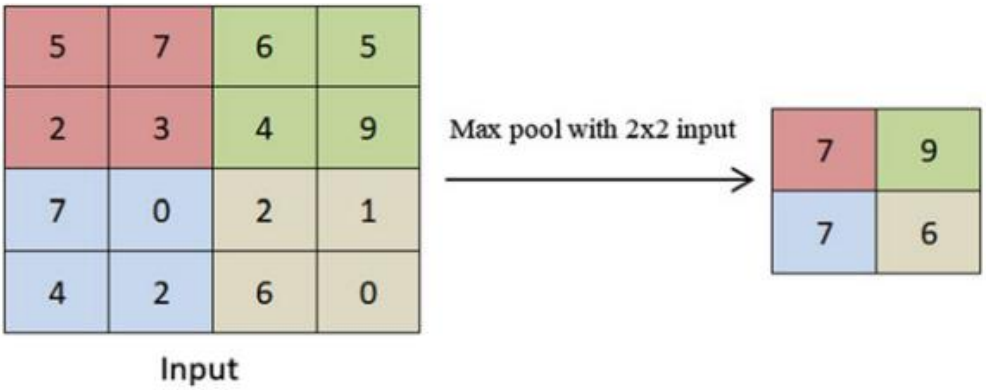
ซึ่งขนาดของผังฟีเจอร์ที่เป็นผลลัพธ์ขาออก จะเท่ากับ $1 + (A - K + 2P)/S$ เมื่อกำหนดให้ A คือ ขนาดข้อมูลค่าเข้า K คือขนาดของตัวกรอง P คือค่าของการเสริมเติม S คือขนาดการก้าวข้าม ถ้าสมมติให้ข้อมูลรับเข้าเป็นรูปภาพขนาด 128x128 กำหนดให้ตัวกรองมีจำนวน 5 ตัวโดยมีขนาด 5x5 ขนาดของการก้าวข้ามเป็นหนึ่ง และการเสริมเติมคือศูนย์ จะสามารถคำนวณขนาดของผังฟีเจอร์ได้ว่า 1 + (128 – 5 + 0)/1 = 124 ซึ่งก็คือ 124x124x5 เพราะกำหนดจำนวนตัวกรองไว้ที่ 5 ตัว



รูปที่ 4 ตัวอย่างการทำคอนโวลูชันด้วยตัวกรองขนาด 3x3

2) ชั้นการรวม (Pooling Layer)

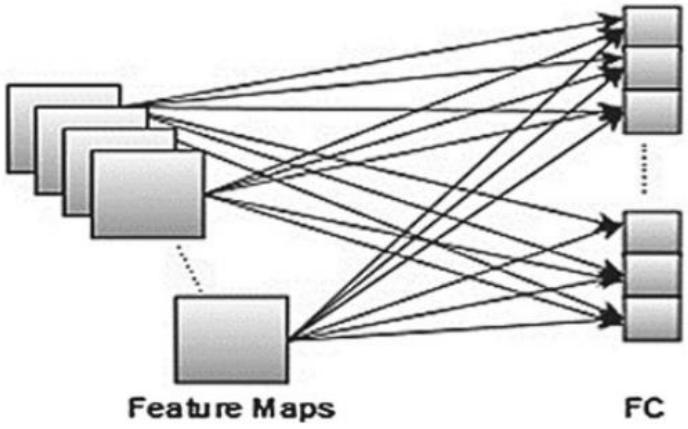
คือ ชั้นที่ทำหน้าที่ลดขนาดข้อมูลเพื่อให้เหลือเฉพาะข้อมูลที่สำคัญเท่านั้น และเป็นการลดจำนวนพารามิเตอร์ของแบบจำลองอีกด้วย โดยนิยมนำมาต่อกับชั้นคอนโวลูชัน ทั่วไปแล้วมักจะเลือกค่ามากสุด (Max Pooling) หรือค่าเฉลี่ย (Average Pooling) มาจากแต่ละพื้นที่ของเมทริกซ์ โดยใช้ค่าการก้าวข้ามเป็นตัวบอกขอบเขตการรวม ดังรูปที่ 5



รูปที่ 5 ตัวอย่างชั้นการรวมโดยใช้ค่ามากสุดด้วยขอบเขต 2x2

3) ชั้นการเชื่อมโยงเต็มรูปแบบ (Fully Connected Layer or FC)

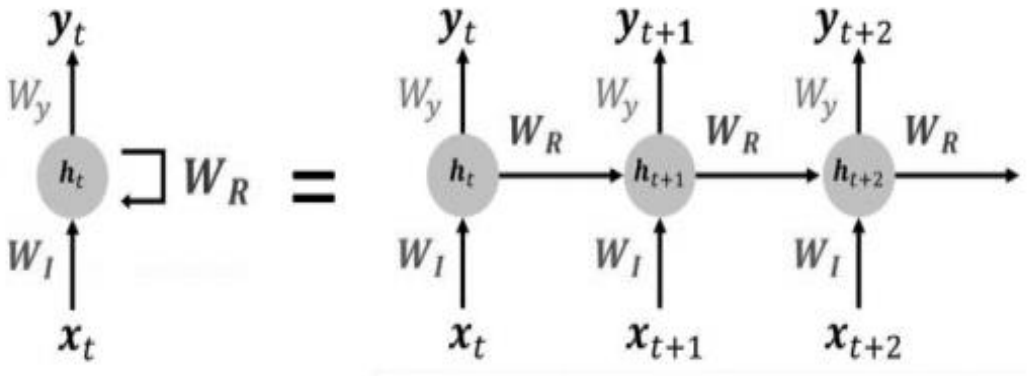
สำหรับนิวรอลเน็ตเวิร์คคอนโวลูชัน โดยทั่วไปมักจะแบ่งเป็น 2 ขั้น คือขั้นการสกัดฟีเจอร์ (Feature Extraction Stage) ซึ่งจะเป็นการป้อนข้อมูลขาเข้า เช่น รูปภาพ เข้าไปในคำนวณในชั้นคอนโวลูชันและชั้นการรวมจำนวนหนึ่ง ซึ่งจะได้ข้อมูลขาออกเป็นผังฟีเจอร์ หลังจากนั้นจะเป็นการคำนวณในขั้นของการจำแนกประเภท (Classification Stage) โดยจะคำนวณในชั้นการเชื่อมโยงเต็มรูปแบบ โดยในชั้นนี้ ก็คือนิวรอลเน็ตเวิร์คแบบดั้งเดิม ที่ประกอบไปด้วยเพอร์เซปตรอน โดยจะมีฟังก์ชันกระตุ้นที่ใช้สำหรับปัญหาการจำแนกประเภท ที่นิยมอยู่แล้ว ได้แก่ ฟังก์ชันซิกมอยสำหรับการจำแนกประเภทแบบทวิภาค (Binary-class Classification) และ ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function) สำหรับปัญหาการจำแนกประเภทแบบหลายคลาส (Multi-class Classification) โดยการเชื่อมต่อของนิวรอลเน็ตเวิร์คคอนโวลูชันทั้ง 2 แสดงดังรูปที่ 6



รูปที่ 6 การเชื่อมต่อระหว่างชั้นคอนโวลูชันและชั้นการเชื่อมโยงเต็มรูปแบบ

2.1.1.3 นิวรอลเน็ตเวิร์คแบบวกกลับ (Recurrent Neural Network or RNN)

เป็นรูปแบบหนึ่งของนิวรอลเน็ตเวิร์ค ที่ถูกออกแบบมาใช้สำหรับปัญหาและใช้กับข้อมูลแบบลำดับ (Sequence Data) โดยใช้การเรียนรู้จากข้อมูลที่เก็บสะสมและเรียนรู้มาจากอดีตหรือลำดับก่อนหน้าเพื่อใช้สกัดฟีเจอร์ของข้อมูลในปัจจุบันหรือลำดับปัจจุบัน โครงสร้างของนิวรอลเน็ตเวิร์คแบบวกกลับแสดงดังรูปที่ 7



รูปที่ 7 โครงสร้างของนิวรอลเน็ตเวิร์คแบบวกกลับ

เมื่อกำหนดให้ x_t แทนด้วยข้อมูลรับเข้าเป็นลำดับที่ t ของชุดข้อมูล W_t แทนด้วยค่าน้ำหนักสำหรับข้อมูลรับเข้า W_R แทนค่าน้ำหนักวกกลับ (Recurrent Weight) ซึ่งใช้ร่วมกันสำหรับทุกลำดับของข้อมูล W_y แทนน้ำหนักของข้อมูลออก h_t แทนสถานะซ่อน (Hidden State) และ y_t สำหรับข้อมูลลำดับที่ t จะสามารถคำนวณผลลัพธ์สำหรับลำดับข้อมูล t ได้ดังสมการที่ (4) - (5) เมื่อ g_h และ g_y คือฟังก์ชันกระตุ้นสำหรับการคำนวณในสถานะซ่อนและ ผลลัพธ์ขาออก

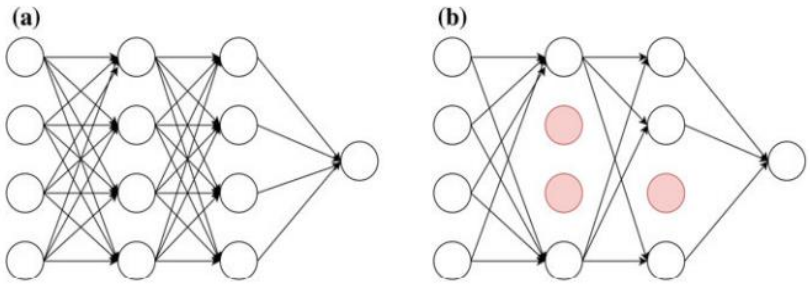
$$h_t = g_h(W_t x_t + W_R h_{t-1}) \quad (4)$$

$$y_t = g_y(W_y h_t + b_y) \quad (5)$$

การฝึกสอนนิวรอลเน็ตเวิร์คแบบวกกลับ จะใช้วิธีการแพร่กระจายย้อนกลับตามเวลา (Back Propagation Through Time) เพื่อปรับน้ำหนักต่าง ๆ ของนิวรอลเน็ตเวิร์ค ซึ่งวิธีการนี้อาจจะเกิดปัญหา หากความ ยาวของข้อมูลรับเข้ามีมากเกินไป เนื่องจาก W_R ซึ่งแทนค่าน้ำหนักวกกลับ จะถูกส่งต่อไปยังชั้นสถานะซ่อนถัด ๆ ไป การแพร่กระจายย้อนกลับซึ่งอาศัยกฎลูกโซ่ (Chain Rule) เพื่อใช้ในการปรับน้ำหนัก อาจจะทำให้เกิดปัญหา เนื่องจากเกรเดียนของ W_R ซึ่งเกิดจากการคูณกันของลำดับก่อนหน้า ส่งผลให้ค่าที่ได้มีค่าเป็นศูนย์ (Vanishing Gradient) เมื่อน้ำหนักอยู่ในระหว่างช่วงศูนย์ถึงหนึ่ง

2.1.1.4 ดรอปเอ้าท์ (Dropout)

นิวรอลเน็ตเวิร์คเชิงลึกเป็นแบบจำลองขนาดใหญ่ มีความซับซ้อนเนื่องจากประกอบไปด้วยชั้นซ่อน (Hidden Layers) จำนวนมาก ซึ่งสามาถเรียนรู้ฟีเจอร์ที่ซับซ้อนจากข้อมูลได้ ซึ่งอาจเกิดปัญหาการปรับเหมาะเกินไป (Overfitting) ทำให้แบบจำลองมีประสิทธิภาพแค่ในการใช้กับข้อมูลที่นำมาฝึกสอน แต่เมื่อนำไปทดสอบกับชุดข้อมูลอื่นๆ แล้วประสิทธิภาพจะลดลงอย่างมีนัยสำคัญ เพื่อแก้ไขปัญหานี้จึงต้องมีการเพิ่มชั้นดรอปเอ้าท์ เข้าไปในแบบจำลองกาเรียนรู้เชิงลึกซึ่งจะปิดกั้นการเชื่อมต่อจากนิวรอลจำนวนหนึ่งแบบสุ่ม แสดงดังรูปที่ 8 เปรียบเสมือนว่าให้แบบจำลองเรียนรู้จากข้อมูลที่ไม่สมบูรณ์บางส่วน ทำให้ข้อมูลรับเข้ามีความหลากหลายและทำให้แบบจำลองเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้น



รูปที่ 8 (a) นิวรอลเน็ตเวิร์คแบบปกติ, (b) นิวรอลเน็ตเวิร์คที่ใช้ดรอปเอ้าท์

2.1.1.5 ฟังก์ชันกระตุ้น (Activation Function)

หลังจากคำนวณค่าต่างๆภายในชั้นแล้ว ข้อมูลส่งออกของแต่ละชั้นจะผ่านการคำนวณด้วยฟังก์ชันกระตุ้น $g(z)$ เพื่อให้นิวรอลเน็ตเวิร์คมีความซับซ้อนและลดความเป็นเชิงเส้น (Linear) ซึ่งช่วยให้นิวรอลเน็ตเวิร์คสามารถแก้ปัญหาจากข้อมูลขนาดใหญ่ที่มีความซับซ้อนสูงได้ดียิ่งขึ้น และหลากหลายมากขึ้น ตัวอย่างของฟังก์ชันกระตุ้นที่ได้รับความนิยมมีดังต่อไปนี้

1) ฟังก์ชันซิกมอยด์ (Sigmoid Function)

เป็นฟังก์ชันที่ทำให้ผลลัพธ์ที่คำนวณได้มีค่าอยู่ในช่วง [0, 1] ซึ่งนิยมใช้สำหรับปัญหาการจำแนกประเภทหรือการทำนายแบบทวิภาค (Binary Classification) เช่น การทำนายว่าพรุ่งนี้ฝนจะตกใช่หรือไม่ ซึ่งสามารถคำนวณได้จากสมการ (6)

$$\sigma(z) = \frac{1}{1+ e^{-z}} \tag{6}$$

2) ฟังก์ชันแทนเจต์ไฮเพอร์โบลิก (Hyperbolic Tangent Function)

เป็นฟังก์ชันที่ทำให้ผลลัพธ์ที่คำนวณได้มีค่าอยู่ในช่วง [-1, 1] โดยทั่วไปแล้วมักจะเขียนแทนด้วย $tanh$ ซึ่งคำนวณได้จากสมการ (7)

$$tanh(z) = \frac{e^{z}- e^{-z}}{e^{z}+ e^{-z}} \tag{7}$$

3) ฟังก์ชันเรกติไฟเชิงเส้น (Rectified Linear Unit Function or ReLU)

เป็นฟังก์ชันที่ทำให้ผลลัพธ์ได้คำนวณได้ไม่ติดลบ ซึ่งหากค่ารับเข้ามีค่าเป็นบวกอยู่แล้วก็จะให้ค่าขาออกมีค่าเท่าเดิม และเป็นศูนย์ในกรณีที่ค่ารับเข้าน้อยกว่าศูนย์ โดยคำนวณได้ตามสมการ (8)

$$f(z) = \max_{z}(0,\ z) \tag{8}$$

4) ฟังก์ชันเอกซ์โพเนนเชียลเชิงเส้น (Exponential Linear Unit or ELU)

เป็นฟังก์ชันที่ลักษณะการคำนวณใกล้เคียงกับ ฟังก์ชันเรกติไฟเชิงเส้น แต่เพิ่มขอบเขตส่วนที่มีค่าติดลบเพื่อให้ผลลัพธ์ขาออกมีความราบเรียบ (Smooth) มากขึ้นโดยคำนวณได้ตามสมการ (9)

$$f(z) = \begin{cases} \alpha(e^{z} - 1), & x < 0 \\ z, & x \geq 0 \end{cases} \tag{9}$$

5) ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function)

เป็นฟังก์ชันที่ค่ารับเข้าโดยส่วนใหญ่จะอยู่ในรูปแบบเวกเตอร์ ซึ่งผลลัพธ์ที่คำนวณออกมาจะอยู่ในรูปค่าความน่าจะเป็น (Probability) สำหรับแต่ละค่าของเวกเตอร์นั้นโดยจะมีผลลัพธ์อยู่ในช่วง [0, 1] และผลลัพธ์ค่าที่ i หรือ f_i จากทั้งหมด k ค่าจะสามารถคำนวณได้จากสมการ (10)

$$f(z)_i = \frac{e^{z_i}}{\sum_{i=1}^{k} e^{z_i}} \tag{10}$$

2.1.1.6 ฟังก์ชันสูญเสีย (Cost Function) หรือฟังก์ชันสูญเสีย (Loss Function)

เป็นฟังก์ชันที่แสดงถึงความผิดพลาดของนิวรอลเน็ตเวิร์ค เปรียบเสมือนกับวัตถุประสงค์ที่จะใช้ในการปรับค่าน้ำหนักของแบบจำลองในกระบวนการเรียนรู้ ซึ่งคำนวณได้จากผลลัพธ์จริงจากชุดข้อมูลตัวที่ i คือ y_i เทียบกับผลลัพธ์ที่แบบจำลองทำนาย $\hat{y}_i$ โดยกำหนดให้ แทนฟังก์ชันสูญเสีย และ k คือจำนวนข้อมูลทั้งหมด ฟังก์ชันสูญเสียพื้นฐานที่นิยมใช้มีดังนี้

1) ค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean Squared Error or MSE)

$$J = \frac{1}{k}\sum_{i=1}^{k}(\hat{y}_i - y_i)^2 \tag{11}$$

2) ครอสเอนโทรปีแบบทวิภาค (Binary Cross-entropy)

$$J = -\frac{1}{k}\sum_{i=1}^{k} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{12}$$

3) ค่าลบลอการิทึมของความเป็นไปได้ (Negative Log-Likelihood)

$$J = -\frac{1}{k}\sum_{i=1}^{k} y_i \log(\hat{y}_i) \tag{13}$$

ซึ่งค่าเฉลี่ยความผิดพลาดแบบกำลังสองจะเหมาะที่จะใช้กับปัญหาประเภทการทดทอย (Regression) ซึ่งเป็นปัญหาในรูปแบบการทำนายหรือคาดการร์ผลลัพธ์ที่เป็นค่าตัวเลข ส่วนครอสเอนโทรปีแบบทวิภาค และค่าลบลอการิทึมของความเป็นไปได้จะใช้กับปัญหาประเภทการทำนายหรือจำแนกประเภท ทั้งแบบทวิภาค (Binary-class) และแบบหลายประเภท (Multi-class) ซึ่งค่า $\hat{y}_i$ ที่ได้จากแบบจำลอง คือค่าความน่าจะเป็นที่ได้มาจากฟังก์ชันซิกมอย สำหรับแบบทวิภาค และฟังก์ชันค่าสูงสุดอย่างอ่อนสำหรับแบบหลายคลาส

2.1.1.7 การหาค่าเหมาะสมที่สุด (Optimization)

เพื่อให้แบบจำลองสามารถทำนายค่าออกมาได้อย่างแม่นยำ จึงต้องมีการปรับค่าน้ำหนักของแบบจำลองให้ค่าที่ได้จากฟังก์ชันสูญเสียมีค่าน้อยที่สุดหรือหมายถึงมีค่าความผิดพลาดน้อยที่สุด ซึ่งการหาค่าที่เหมาะสมที่สุด (Optimization) ก็คือ กระบวนการหรือวิธีในการปรับแต่งค่าน้ำหนัก โดยวิธีที่ได้รับความนิยมมีดังนี้

1) สโตแคสติกเกรเดียนเดสเซนท์ (Stochastic Gradient Descent or SGD)

เมื่อกำหนดให้ พารามิเตอร์ของแบบจำลองกนิวรอลเน็ตเวิร์ค คือค่าน้ำหนัก (w) การปรับค่าพารามิเตอร์จะขึ้นอยู่กับเกรเดียนของฟังก์ชันสูญเสียเทียบกับค่าน้ำหนักหรือ $\frac{\partial J_t}{\partial w}$ โดยมีอัตราการเรียนรู้ (Learning rate) แทนด้วย α ตามสมการที่ (14)

$$W_t = W_{t-1} - \alpha \frac{\partial J_t}{\partial w} \tag{14}$$

โดยการใช้สโตแคสติกเกรเดียนเดสเซนท์ อาจจะเกิดปัญหาระหว่างการเรียนรู้ คือการที่แบบจำลองไม่สามารถไปสู่จุดที่ดีที่สุดได้ จึงได้มีการนำโมเมนตัม (Momentum) เข้าช่วย เมื่อกำหนดให้ v แทนด้วยความเร็วที่จะปรับค่าพร้อมกันกับ w และ γ แทนค่าสัมประสิทธิ์ของโมเมนตัม (Momentum Coefficient) สามารถคำนวณได้ตามสมการที่ (15) และ (16)

$$v_t = \gamma v_{t-1} + \alpha \frac{\partial J_t}{\partial w} \tag{15}$$

$$w_t = w_{t-1} - v_t \tag{16}$$

2) วิธีเกรเดียนปรับตัวได้ (Adaptive Gradient Method or AdaGrad)

ในการทำสโตแคสติกเกรเดียนเดสเซนท์ อัตราการเรียนรู้จะถูกกำหนดขึ้นมาโดยไม่ได้มีความเกี่ยวข้องกับเกรเดียน ซึ่งในบางครั้งอาจเป็นปัญหา เช่น เมื่อกำหนดให้อัตราการเรียนรู้มีค่ามาก และเกรเดียนในบางลำดับมีค่ามากเช่นกัน จะทำให้แบบจำลองมีการปรับค่าน้ำหนักที่มากเกินไป ซึ่งอาจทำให้ข้ามจุดที่ดีที่สุดได้ โดยวิธีเกรเดียนปรับตัวได้จะมีการปรบอัตราการเกรียนรู้ในแต่ละครั้ง โดยการนำค่าเกรเดียนจากลำดับก่อนหน้ามาใช้ เมื่อกำหนดให้ g_t แทนเกรเดียนที่เวลา t จะคำนวณได้ตามสมการที่ (17) และ (18)

$$g_t = \frac{\partial J_t}{\partial w} \tag{17}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{\sum_{k=1}^{t} g_k^2}} g_t \tag{18}$$

3) อาเอ็มเอสพรอพ (RMSProp)

ในวิธีเกรเดียนปรับตัวได้ จะใช้ค่าเกรเดียนตั้งแต่ค่าแรกจนถึงค่าปัจจุบันทำให้อาจจะเกิดปัญหาเกรเดียนไม่สัมพันธ์กัน ซึ่งอาเอ็มเอสพรอพถูกออกแบบมาเพื่อแก้ปัญหานี้ โดยจะใช้วิธีการเก็บค่าเกรเดียนด้วยค่าเฉลี่ยถ่วงน้ำหนักแบบเอกซ์โพเนนเชียล (Exponentially weighted moving average) และนำไปปรับปรุงอัตราส่วนของอัตราการเรียนรู้ โดยนอกเหนือจากการใช้ g_t แล้วยังมีการใช้ $\mathrm{MeanSquare}_t$ สำหรับการเก็บค่าเฉลี่ยของเกรเดียน และให้ γ แทนอัตราการใช้เกรเดียนของอดีตในการเรียนรู้ ซึ่งปกติจะใช้ค่านี้ที่ 0.9 อาร์เอ็มเอสพรอพสามารถคำนวณได้ดังสมการ (19) – (21)

$$g_t = \frac{\partial J_t}{\partial w} \tag{19}$$

$$\mathrm{MeanSquare}_t = \gamma\, \mathrm{MeanSquare}_{t-1} + (1-\gamma) g_t^2 \tag{20}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{\mathrm{MeanSquare}_t}} g_t \tag{21}$$

4) อดัม (Adaptive Moment Estimation or Adam)

เป็นการหาค่าที่เหมาะสมที่สุดแบบปรับตัวได้วิธีหนึ่ง ที่ใช้ข้อดีของทั้ง วิธีเกรเดียนแบบปรับตัวได้ และอาเอ็มเอสพรอพ โดยจะใช้ทั้งเกรเดียนแบบยกกำลังสอง และค่าเกรเดียนปกติเพื่อเพิ่มประสิทธิภาพให้การปรับค่าอัตราการเรียนรู้ โดยจะถูกกำกับด้วยค่าพารามิเตอร์ β_1 และ β_2 โดยการคำนวณอดัมแสดงดังสมการที่ (22) – (26)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{22}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{23}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{24}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{25}$$

$$w_{t+1} = w_t - \frac{\mu}{\sqrt{\hat{v}_t}} \cdot \hat{m}_t \tag{26}$$

2.1.1.8 การแพร่กระจายย้อนกลับและการเรียนรู้ (Back propagation and Training)

หลังจากมีการคำนวณผ่านแบบจำลองนิวรอลเน็ตเวิร์ค หรือที่เรียกว่าการป้อนไปข้างหน้า (Feedforward) แล้วจึงนำไปหาค่าความผิดพลาด โดยใช้ฟังก์ชันสูญเสียเทียบผลลัพธ์จริงและการทำนายในชั้นลำดับสุดท้าย สำหรับกระบวนการหาค่าที่เหมาะสมที่สุด (Optimization) การหาค่าความผิดพลาดของนิวรอลเน็ตเวิร์คเพื่อใช้ในการเรียนรู้ในชั้นก่อนหน้านั้นไม่สามารถทำได้โดยตรงเหมือนกับ ชั้นสุดท้ายจึงต้องอาศัยกระบวนการแพร่กระจายย้อนกลับ

กำหนดให้ j แทนฟังก์ชันสูญเสีย z คือผลลัพธ์จากการป้อนไปข้างหน้า และ g คือฟังก์ชันกระตุ้น จะสามารถเขียนสมการของ δ_j^k หรือค่าความผิดพลาดของเพอร์เซปตรอนตัวที่ j ในลำดับชั้นที่ k ได้ดังสมการที่ (27)

$$\delta_j^k = \frac{\partial J}{\partial z_j^k} = \frac{\partial J}{\partial a_j^k}\frac{\partial a_j^k}{\partial z_j^k} = \frac{\partial J}{\partial a_j^k} g'(z_j^k) \tag{27}$$

สำหรับค่าของ $\frac{\partial J}{\partial a_j^k}$ ในชั้นสุดท้ายสามารถคำนวณได้โดยตรงจากฟังก์ชันสูญเสีย แต่สำหรับในชั้นก่อนหน้า ต้องใช้การแพร่กระจายย้อนกลับในการคำนวณซึ่งคำนวณได้จากสมการที่ (28)

$$\frac{\partial \mathrm{J}}{\partial \mathrm{a}_\mathrm{j}^\mathrm{k}} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \frac{\partial \mathrm{J}}{\partial \mathrm{z}_\mathrm{j}^{\mathrm{l}+1}} \frac{\partial \mathrm{z}_\mathrm{j}^{\mathrm{l}+1}}{\partial \mathrm{a}_\mathrm{j}^\mathrm{k}} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \delta_\mathrm{j}^{\mathrm{l}+1} \mathrm{w}_\mathrm{jk}^{\mathrm{l}+1} \tag{28}$$

โดย m คือจำนวนเพอร์เซปตรอนในลำดับชั้นที่ $\mathrm{l}+1$ และเมื่อคำนวณค่าความผิดพลาดของแต่ละชั้นได้ จะสามารถหาค่าเกรเดียน หรือค่าความผิดพลาดเทียบกับค่าน้ำหนักของแต่ละชั้นได้ ตามสมการที่ (29) ซึ่งหลังจากนั้นจะเข้าสู่การหาค่าที่ดีที่สุด (Optimization)

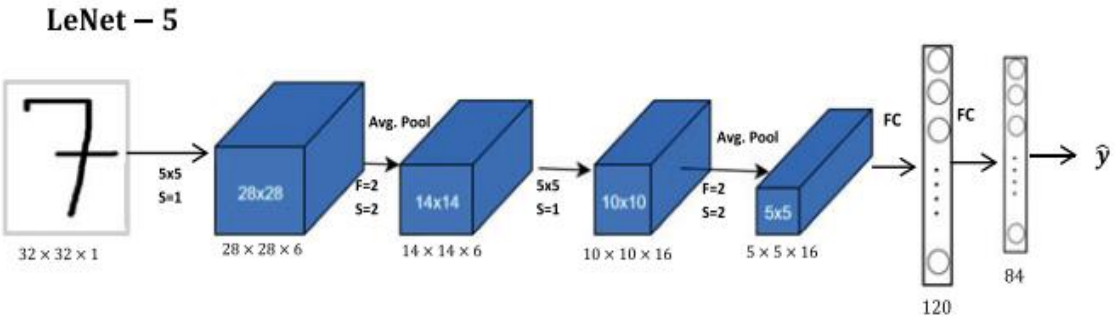
$$\frac{\partial J}{\partial w_{jk}^l} = \frac{\partial J}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l a_k^{l-1} \tag{29}$$

2.1.2 สถาปัตยกรรมการเรียนรู้เชิงลึก (Deep Learning Architectures)

สถาปัตยกรรมการเรียนรู้เชิงลึก คือการนำเอาชั้นต่างๆ ของนิวรอลเน็ตเวิร์คเชิงลึกมาประกอบกันเป็นแบบจำลองการเรียนรู้เชิงลึก ที่มีความซับซ้อนและสามารถเรียนรู้จากข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งในเวลาหลายปีที่ผ่านมา มีสถาปัตยกรรมการเรียนรู้เชิงลึกได้รับการพัฒนาออกมามากมาย โดยสถาปัตยกรรมที่ได้รับความนิยมมีดังต่อไปนี้

2.1.2.1 เลอเน็ต (LeNet-5)

เป็นสถาปัตยกรรมการเรียนรู้เชิงลึกที่ประกอบด้วยนิวรอลเน็ตเวิร์คเชิงลึก 7 ชั้น โดยมาจากชั้นคอนชันและชั้นการเชื่อมโยงเต็มรูปแบบ ข้อมูลรับเข้าคือรูปภาพที่มีขนาด 32x32 พิกเซล (pixels) โดยแสดงแผนภาพของลีเน็ตดังรูปที่ 9 และตารางรายละเอียดของแต่ละชั้นที่ ตารางที่ 1



รูปที่ 9 แผนภาพของเลอเน็ต

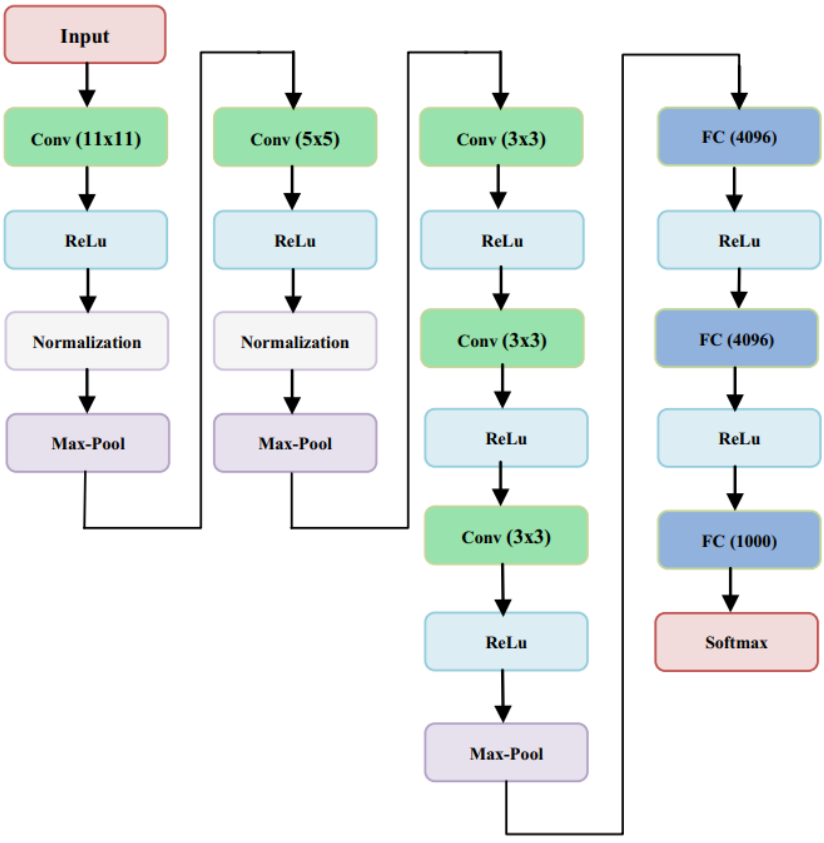
ตาราง 1 แสดงรายละเอียดในแต่ละชั้นของเลอเน็ต

Layer name	Input size	Filter size	Window size	# Filters	Stride	Padding	Output size	# Channels
Conv 1	32 x 32	5 x 5	-	6	1	0	28 x 28	6
Subsampling 1	28 x 28	-	2 x 2	-	2	0	14 x 14	6
Conv 2	14 x 14	5 x 5	-	16	1	0	10 x 10	16
Subsampling 2	10 x 10	-	2 x 2	16	2	0	5 x 5	16
Conv 3	5 x 5	5 x 5	-	120	1	0	1 x 1	120
Fully connected	120	-	-	-	-	-	1 x 1	84
Softmax	84	-	-	-	-	-	1 x 1	10

2.1.2.2 อเล็กซ์เน็ต (AlexNet)

ปัญหาหนึ่งสำหรับการฝึกสอนแบบจำลองการเรียนรู้เชิงลึก คือการที่เกรเดียนมีค่าน้อยมากเกินไป (Vanishing Gradient) เพื่อแก้ปัญหานี้จึงมีการนำฟังก์ชันกระตุ้นเรคติไฟต์เชิงเส้น (Rectified Linear unit or ReLU) อเล็กซ์เน็ตเป็นสถาปัตยกรรมแรกที่มีการนำฟังก์ชันเรคติไฟต์เชิงเส้นมาใช้

อเล็กซ์เน็ตประกอบไปด้วยชั้นคอนโวลูชัน ชั้นรวมและชั้นการเชื่อมโยงเต็มรูปแบบ ที่ข้อมูลรับเข้าเป็นรูปภาพขนาด 224x224 พิกเซล ซึ่งแสดงแผนภาพดังรูปที่ 10 และรายละเอียดแต่ละชั้นในตารางที่ 2 โดยอเล็กซ์เน็ตถูกนำไปประเมินประสิทธิภาพด้วย ฐานข้อมูลอิมเมจเน็ต (ImageNet database) ซึ่งเป็นฐานข้อมูลรูปภาพขนาดใหญ่ ประมาน 15 ล้านรูป โดยนำมาฝึกสอนการจำแนกประเภทภาพด้วยชุดข้อมูลรูปภาพ 1.2 ล้านรูป 1000 ประเภท



รูปที่ 10 แผนภาพของอเล็กซ์เน็ต

ตาราง 2 แสดงรายละเอียดในแต่ละชั้นของอเล็กซ์เน็ต

Layer name	Input size	Filter size	Window size	# Filters	Stride	Padding	Output size	# Channels
Conv 1	224 x 224	11x11	-	96	4	1	55 x 55	96
Max pooling1	55 x 55	-	3 x 3	-	2	0	27 x 27	96
Conv 2	27 x 27	5 x 5	-	256	1	2	27 x 27	256
Max pooling2	27 x 27	-	3 x 3	-	2	0	13 x 13	256
Conv 3	13 x 13	3 x 3	-	384	1	1	13 x 13	384
Conv 4	13 x 13	3 x 3	-	384	1	1	13 x 13	384
Conv 5	13 x 13	3 x 3	-	256	1	1	13 x 13	256
Max pooling3	13 x 13	-	3 x 3	-	2	0	6 x 6	256
Fullyconnected1	4096 neurons							
Fullyconnected2	4096 neurons							
Fullyconnected3	1000 neurons							
Softmax	1000 Classes							

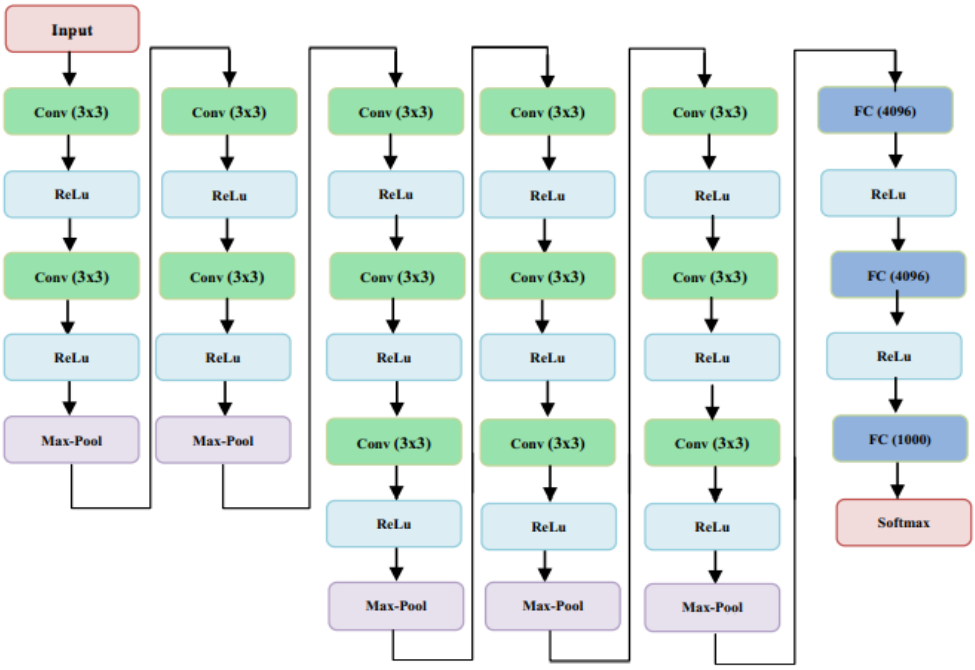
2.1.2.3 วีจีจีเน็ต (VGGNet)

วีจีจีเน็ตเป็นสถาปัตยกรรมการเรียนรู้เชิงลึกที่พัฒนาต่อจาก อเล็กซ์เน็ตโดยการเพิ่มจำนวนชั้น แต่ลดขนาดของตัวกรองลง โดยจะใช้ตัวกรองในชั้นคอนโวลูชัน ที่มีขนาด 3x3 ทั้งหมดและลดขนาดของ ผังฟีเจอร์ลงด้วยชั้นการรวมด้วยค่ามากสุด โดยสถาปัตยกรรมวีจีจี จะถูกกำกับไว้ด้วยเลขตามขนาดของแบบจำลอง เช่น วีจีจีเน็ต16 ซึ่งแสดงตามรูปที่ 11 และรายละเอียดแต่ละชั้น แสดง ณ ตารางที่ 3

ตาราง 3 แสดงรายละเอียดในแต่ละชั้นของ วีจีจีเน็ต16

Layer name	Input size	Filter size	Window size	# Filters	Stride	Padding	Output size	# Channels
Conv 1	224x224	3 x 3	-	64	1	1	224x224	64
Conv 2	224x224	3 x 3	-	64	1	1	224x224	64
Max pooling 1	224x224	-	2 x 2	-	2	0	112x112	64
Conv 3	112x112	3 x 3	-	128	1	1	112x112	128

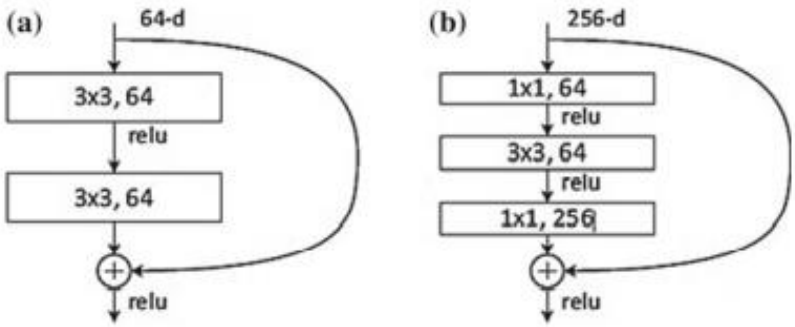
Conv 4	112x112	3 x 3	-	128	1	1	112x112	128
Max pooling 2	112x112	-	2 x 2	-	2	0	56x56	128
Conv 5	56x56	3 x 3	-	256	1	1	56x56	256
Conv 6	56x56	3 x 3	-	256	1	1	56x56	256
Conv 7	56x56	3 x 3	-	256	1	1	56x56	256
Max pooling 3	56x56	-	2 x 2	-	2	0	28x28	256
Conv 8	28x28	3 x 3	-	512	1	1	28x28	512
Conv 9	28x28	3 x 3	-	512	1	1	28x28	512
Conv 10	28x28	3 x 3	-	512	1	1	28x28	512
Max pooling 4	28x28	-	2 x 2	-	2	0	14x14	512
Conv 11	14x14	3 x 3	-	512	1	1	14x14	512
Conv 12	14x14	3 x 3	-	512	1	1	14x14	512
Conv 13	14x14	3 x 3	-	512	1	1	14x14	512
Max pooling 5	14x14	-	2 x 2	-	2	0	7x7	512
Fullyconnected1	4096 neurons							
Fullyconnected2	4096 neurons							
Fullyconnected3	1000 neurons							
Softmax	1000 Classes							



รูปที่ 11 แผนภาพของวีจีจีเน็ต16

2.1.2.4 เรสเน็ต (ResNet)

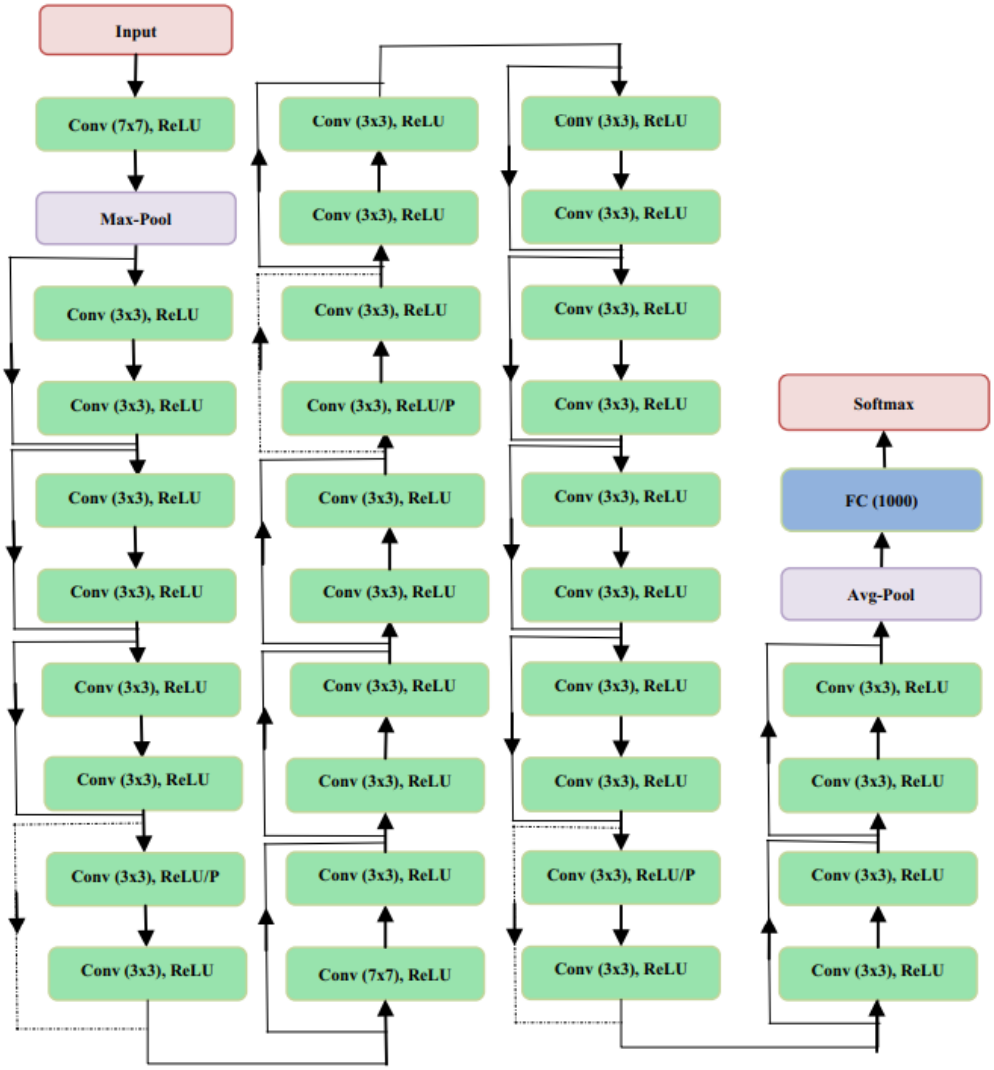
ประสิทธิภาพของสถาปัตยกรรมการเรียนรู้เชิงลึก ส่วนใหญ่แล้วจะเพิ่มขึ้นตามจำนวนชั้นและความซับซ้อนของแบบจำลอง อย่างไรก็ตามเมื่อขนาดหรือจำนวนชั้นของสถาปัตยกรรมเพิ่มขึ้นถึงจุดหนึ่ง ประสิทธิภาพที่ได้ก็จะเริ่มคงที่ซึ่งเป็นจุดที่ดีที่สุด ที่เกิดความสมดุลระหว่างความซับซ้อนของแบบจำลองและประสิทธิภาพ ซึ่งมีแนวโน้มที่จะความผิดพลาดจะมากขึ้นหากพยามยามที่จะเพิ่มจำนวนชั้นให้มากขึ้นเกินกว่าจุดสมดุลนั้น ซึ่ง การเรียนรู้แบบเรสสิดวล (Residual Learning) ถูกออกแบบมาเพื่อพัฒนาจุดที่ดีที่สุดของสถาปัตยกรรมการเรียนรู้เชิงลึก โดยการใช้ เรสสิดวลบล็อค (Residual Block) แสดงดังรูปที่ 12 แทนที่การต่อกันด้วยชั้นคอนโวลูชันแบบปกติ



รูปที่ 12 ตัวอย่างเรสสิดวลบล็อก

เรสสิดวลบล็อก คือการนำข้อมูลรับเข้า มารวมกันกับข้อมูลขาออกจากชั้นคอนโวลูชันที่ต่อกันจำนวนหนึ่ง เมื่อกำหนดให้ Y คือข้อมูลขาออกของเรสสิดวลบล็อก และ X คือข้อมูลขาเข้า $F(x,\{w_i\})$ คือการคำนวณด้วยชั้นคอนลูชันภายในเรสสิดวลบล๊อค และ w_S คือค่าน้ำหนักสำหรับการรวมข้อมูลขาเข้ากับคอนโวลูชันของข้อมูลขาเข้า ซึ่งสามารถคำนวณได้ดังสมการที่ (30)

$$y = F(x,\{w_i\}) + w_s x \tag{30}$$



รูปที่ 13 แผนภาพของเรสเน็ต34

ตาราง 4 แสดงรายละเอียดในแต่ละชั้นของเรสเน็ต 34

Layer name	Input size	Filter size	Window size	# Filters	Stride	Padding	Output size	# Channels
Conv 1	224x224	7 x 7	-	64	2	2	112x112	64
Conv 2_x	112x112	-	3 x 3	-	2	0	56x56	64
	56x56	3 x 3	-	64	1	1	56x56	64
	56x56	3 x 3	-	64	1	1	56x56	64
	56x56	3 x 3	-	64	1	1	56x56	64

Conv 2_x	56x56	3 x 3	-	64	1	1	56x56	64
	56x56	3 x 3	-	64	1	1	56x56	64
	56x56	3 x 3	-	64	1	1	56x56	64
Conv 3_x	56x56	3 x 3	3 x 3	128	2	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
	28x28	3 x 3	-	128	1	1	28x28	128
Conv 4_x	28x28	3 x 3	3 x 3	256	2	0	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
	14x14	3 x 3	-	256	1	1	14x14	256
Conv 5_x	14x14	3 x 3	3 x 3	512	2	0	7x7	512
	7x7	3 x 3	-	512	1	1	7x7	512
	7x7	3 x 3	-	512	1	1	7x7	512
	7x7	3 x 3	-	512	1	1	7x7	512
	7x7	3 x 3	-	512	1	1	7x7	512
	7x7	3 x 3	-	512	1	1	7x7	512
Avg pooling	7x7	-	7 x 7	-	-	-	1 x 1	1000
Fully connected	1000 Classes							

2.1.3 การวัดประสิทธิภาพของการจำแนกประเภท (Classification Performance Evaluation)

2.1.3.1 คอนฟิวชันเมทริกซ์ (Confusion Matrix)

คอนฟิวชันเมทริกซ์ คือ เมทริกซ์ที่แสดงผลของการจำแนกโดยแจกแจงจำนวนที่จำแนกได้ตามคลาส ดังตัวอย่างในตารางที่ 5 ซึ่งแสดงการจำแนกข้อมูลเป็น 2 ประเภท โดยค่าแต่ละแถวแสดงจำนวนข้อมูลที่มีคลาสนั้นเป็นคำตอบที่ถูกต้อง ส่วน ค่าในแต่ละหลักแสดงจำนวนข้อมูลที่ทำนายได้คลาสนั้น กำหนดให้สำหรับคลาสใด ๆ

(1) TP คือ จำนวนข้อมูลที่ทำนายได้คลาสที่หนึ่งและทำนายถูก (True Positive)

(2) FP คือ จำนวนข้อมูลที่ทำนายได้คลาสที่หนึ่งและทำนายผิด (False Positive)

(3) TN คือ จำนวนข้อมูลที่ทำนายได้คลาสที่สองและทำนายถูก (True Negative)

(4) FN คือ จำนวนข้อมูลที่ทำนายได้คลาสที่สองและทำนายผิด (True Negative)

ตาราง 5 คอนฟิวชันเมทริกซ์แบบทวิภาค

		คลาสที่ทำนาย	
		โพสิทีฟ (1)	**เนกาทีฟ (0)**
คลาสจริง	**โพสิทีฟ (1)**	TP	FN
	เนกาทีฟ (0)	FP	TN

2.1.3.2 ตัววัดประสิทธิภาพจำแนกตามคลาส

โดยทั่วไปตัววัดประสิทธิภาพที่นิยมใช้กันในงานวิจัยมีอยู่ 4 ค่า ดังนี้

(1) ค่าความเที่ยง (Precision) เป็นการวัดความแม่นยำของแบบจำลองโดยการพิจารณาแยกทีละคลาส ตัวอย่างเช่น การวัดว่าแบบจำลองทำนายว่าคำตอบที่เป็นบวกถูกต้องเท่าไหร่จากผลการทำนายคลาสบวกทั้งหมดเท่าไหร่ แสดงดังสมการที่ (31)

$$Precision = \frac{TP}{TP+FP} \tag{31}$$

(2) ค่าความระลึก (Recall) เป็นการวัดความถูกต้องของแบบจำลองโดยการพิจารณาแยกทีละคลาส ตัวอย่างเช่น การวัดว่าผลการทำนายคลาสบวกความถูกต้องเท่าไหร่เมื่อเทียบกับคลาสบวกจริงทั้งหมด แสดงดังสมการที่ (32)

$$Recall = \frac{TP}{TP+FN} \tag{32}$$

(3) คะแนนเอฟวัน (F1-score) เป็นการวัดความเที่ยงและความระลึกของแบบจำลองไปพร้อม ๆ กันโดยคำนวณได้ดังสมการที่ (33)

$$F1 = 2 \cdot (\frac{precision*recall}{precision+recall}) \tag{33}$$

(4) ค่าความแม่นยำ (Accuracy) เป็นการวัดความแม่นยำของแบบจำลองโดยรวม กล่าวคือ แบบจำลอง ทำนายถูกกี่ครั้งจากจำนวนการทำนายทั้งหมด แสดงดังสมการที่ (34)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{34}$$

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ คือ งานวิจัยแบบจำลองเพื่อแบ่งประเภทภาพแบบละเอียดด้วยการเรียนรู้เชิงลึก โดยแบ่งงานวิจัยได้เป็น 3 กลุ่มได้แก่ 1) การเข้ารหัสฟีเจอร์แบบเอ็นทูเอ็น (End-to-End Features Encoding) 2) การระบุตำแหน่งและแบ่งประเภทแบบซับเน็ตเวิร์ค (Localization-Classification Sub-Network) และ3) ฝึกสอนด้วยข้อมูลเพิ่มเติม (Training with External information) โดยในวิทยานิพนธ์นี้จะให้ความสำคัญไปที่งานวิจัยกลุ่มที่หนึ่งและสอง ซึ่งจะเป็นการพัฒนาในส่วนของสถาปัตยกรรมการเรียนรู้เชิงลึก หรือฟังก์ชันต่างๆที่ช่วยเพิ่มความแม่นยำในการจำแนกประเภท เช่น ฟังก์ชันสูญเสียหรือฟังก์ชันกระตุ้น แต่งานวิจัยในกลุ่มที่สามจะใช้ข้อมูลเพิ่มเติมเข้าช่วยในการฝึกสอนแบบจำลอง และในบทนี้จะกล่าวถึงงานวิจัยเกี่ยวกับฟังก์ชันสูญเสียอื่นๆที่ช่วยเพิ่มประสิทธิภาพในการฝึกสอนแบบจำลองและมักนำมาใช้กับปัญหาที่เกี่ยวกับจำแนกประเภทภาพแบบต่างๆ โดยมีตัวอย่างงานวิจัยดังนี้

2.2.1 การเข้ารหัสฟีเจอร์แบบเอ็นทูเอ็น (End-to-End features encoding)

งานวิจัยในกลุ่มนี้จะเน้นไปที่การพัฒนาแบบจำลองการเรียนรู้เชิงลึกเพื่อแบ่งแยกฟีเจอร์เวกเตอร์โดยตรงจากรูปภาพ และมีขั้นตอนการฝึกสอนแบบเอ็นทูเอ็น โดยใช้เพียงคลาสหรือคำตอบแบบหมวดหมู่ (Categorical Labels) เท่านั้นซึ่งส่วนใหญ่จะเป็นการออกแบบเพื่อเพิ่มชั้นหรือเพิ่มเทคนิคในชั้น หรือออกแบบฟังก์ชันสูญเสียสำหรับฝึกสอนแบบจำลอง โดยไม่เพิ่มจำนวนพารามิเตอร์ของคอนโวลูชันนิวรอลเน็ตเวิร์คที่ใช้ทำหน้าที่สกัดฟีเจอร์อย่างมีนัยสำคัญและง่ายต่อการใช้งาน ซึ่งสามารถนำไปปรับใช้กับสถาปัตยกรรมได้หลากหลาย

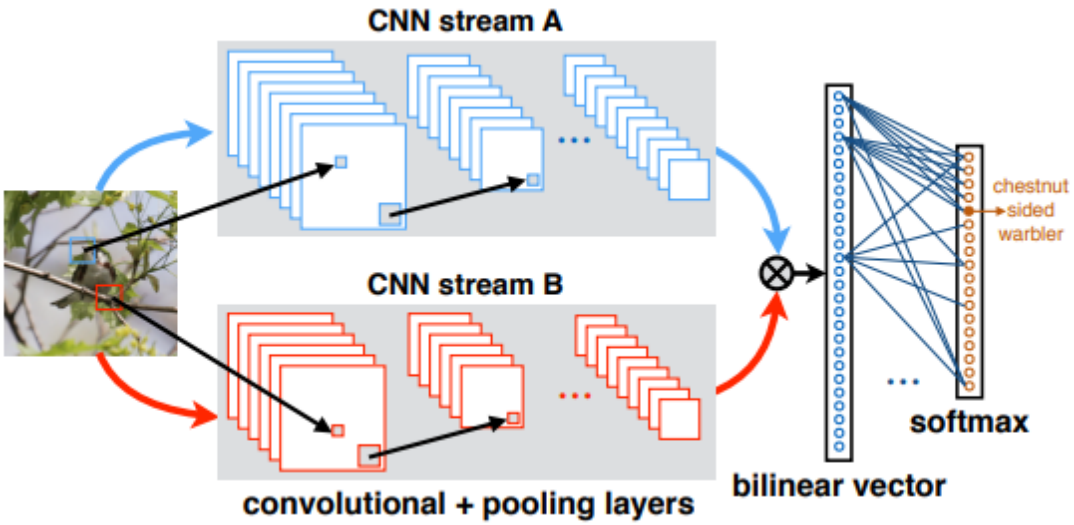
2.2.1.1 งานวิจัยของ Tsung-Yu Lin และคณะ[16]

เป็นงานวิจัยในปี 2015 โดยงานวิจัยนี้นำเสนอแบบจำลองเรียนรู้เชิงลึก ไบลิเนียร์คอนโวลูชันนิวรอลเน็ตเวิร์ค (Bilinear Convolutional Neural Networks or B-CNN) ซึ่งคำนวณฟีเจอร์มาจากผลคูณภายนอก (Outer Product) ของผังฟีเจอร์ที่สกัดมาจากคอนโวลูชันนิวรอลเน็ตเวิร์คสองเน็ตเวิร์ค ตามสมการที่ (35) เรียนรู้มาจากข้อมูลเดียวกัน เมื่อกำหนดให้ ข้อมูลรับเข้า I ตำแหน่งที่ l ของฟีเจอร์ f_A และ f_B ซึ่งคำนวณมาจากคอนโวลูชันนิวรอลทั้งสองตามลำดับ

$$bilinear(l, I, f_A, f_B) = f_A(l, I)^T f_B(l, I) \tag{35}$$

ในงานวิจัยนี้พยายามหาความสัมพันธ์ของตำแหน่งและลักษณะของวัตถุที่ทับซ้อนกันของเน็ตเวิร์คทั้งสอง โดยไบลิเนียร์คอนโวลูชันนิวรอลเน็ตเวิร์ค ถือเป็นตัวอธิบายแบบไม่กำหนดลำดับ (Orderless Descriptor) ที่ช่วยแบ่งแยกฟีเจอร์จากข้อมูลที่มีความแตกต่างเล็กน้อยได้ดี และสามารถฝึกสอนข้อมูลแบบเอ็นทูเอ็นได้ตามรูปที่ 14 โดยใช้การจำแนกประเภทแบบทั่วไปด้วยฟังก์ชันค่าสูงสุดอย่างอ่อน และค่าความผิดพลาดด้วยค่าลบลอการิทึมของความเป็นได้ ซึ่งความแม่นยำของแบบจำลองนี้อยู่ที่ 84.1%, 86.9% and 91.3% สำหรับชุดข้อมูล Caltech-UCSD birds, FGVC aircraft, และ Stanford cars

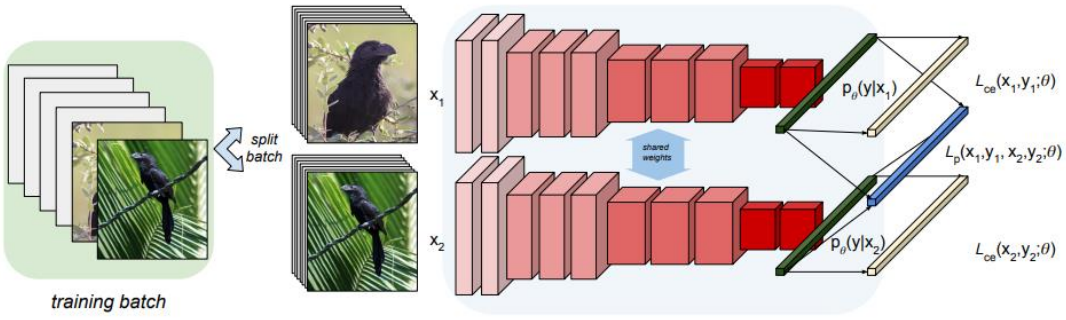
อย่างไรก็ตามการนำข้อมูลขาเข้าเพื่อผ่านคอนโวลูชันนิวรอลเน็ตเวิร์คถึงสองเน็ตเวิร์คเพื่อสกัดและนำผังฟีเจอร์ไปคำนวณผลคูณภายนอก ซึ่งเป็นการคูณเมทริกซ์ที่มีมิติขนาดใหญ่ ทำให้ขนาดของแบบจำลองใหญ่ขึ้นตามไปด้วยอีกทั้งยังใช้ต้นทุนในการคำนวณ (Computational Cost) สูงอีกด้วย เพื่อแก้ปัญหาดังกล่าว Yang Gao และคณะ[17] พยายามเพื่อลดขนาดของเมทริกซ์ผังฟีเจอร์ด้วยวิธี Tensor Sketching ซึ่งช่วยลดต้นทุนในการการคำนวณและยังช่วยเพิ่มประสิทธิภาพและความแม่นยำของการจำแนกประเภทอีกด้วย



รูปที่ 14 โครสร้างแบบจำลอง ไบลิเนียร์คอโวลูชันนิวรอลเน็ตเวิร์ค (B-CNN)

2.2.1.2 งานวิจัยของ Abhimanyu Dubey และคณะ[25]

เป็นงานวิจัยในปี 2018 โดยงานวิจัยนี้นำเสนอฟังก์ชันสูญเสียเพื่อเพิ่มวัตถุประสงค์ในการฝึกฝนแบบจำลอง (Multi-task Learning) ซึ่งปกติแล้วสำหรับการจำแนกประเภทแบบหลายคลาสจะคำนวณจากฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function) และคำนวณค่าสูญเสียด้วยค่าลบลอการิทึมของความเป็นได้ โดยงานวิจัยนี้จะเพิ่มค่าสูญเสียด้วย ค่าความแตกต่างของเวกเตอร์ความเป็นไปได้ เมื่อกำหนดให้ ข้อมูลรูปภาพทั้งสองมีคลาสที่แตกต่างกันเพื่อแก้ปัญหา ความมั่นใจมากเกินไป (Over Confidential) เมื่อแบบจำลองทำนายความเป็นไปได้ของคลาสใดคลาสหนึ่งสูงกว่าคลาสอื่นๆมาก อย่างเช่นว่า 0.99 สำหรับรูปภาพชนิดนกที่คล้ายกันมาก ซึ่งหากเปรียบเทียบกับกระบวนการคิดวิเคราะห์ของมนุษย์ที่ไม่สามารถระบุความแตกต่างจากข้อมูลที่มีความละเอียดสูงได้อย่างชัดเจน ซึ่งเป็นสาเหตุหนึ่งของปัญหาการปรับเหมาะเกินไป (Overfitting) และทำให้ประสิทธิภาพของแบบจำลองลดน้อยลง



รูปที่ 15 อัลกอริทึมของคอนฟิวชั่นแบบคู่

เมื่อกำหนดให้ (X_1, y_1) และ (X_2, y_2) คือข้อมูลรับเข้าและคลาสที่เป็นคำตอบของข้อมูลตัวที่หนึ่งและสองสำหรับการคำนวณเป็นคู่ $L_{CE}(P_\theta(y|X_i), y_i)$ คือค่าลบลอการิทึมของความเป็นไปได้สำหรับข้อมูลแต่ละตัวบน ชุดพารามิเตอร์ θ ซึ่งจะเพิ่ม D_{EC} คือความแตกต่างของความน่าจะเป็นจากการทำนายด้วยแบบจำลองของข้อมูลทั้งสอง เมื่อ $\gamma(y_1, y_2)$ จะมีค่าเท่ากับ 1 เมื่อข้อมูลทั้งสองมีคลาสแตกต่างกัน และเท่ากับ 0 เมื่อมีคลาสเดียวกัน และ λ เป็นค่าสัมประสิทธิ์ของ D_{EC} ซึ่งแสดงอัลกอริทึมของการคำนวณแสดงดังขั้นตอนวิธีที่ 1 และสมการสำหรับค่าสูญเสีย แสดงดังสมการที่ (36)

$$L_{pair}(X_1, X_2, y_1, y_2; \theta) = \sum_{i=1}^{2} [L_{CE}(P_\theta(y|X_i), y_i)] + \lambda\gamma(y_1, y_2)\ \mathrm{D}_{EC}(P_\theta(y|X_1), P_\theta(y|X_2)) \quad (36)$$

Algorithm 1 Training Using Euclidean Confusion

Training data D, Test data $\hat{D}$, parameters θ, hyperparameters $\hat{\theta}$
for *epoch* $\in$ [0,max_epochs]) **do**
  $D_1 \Leftarrow$ shuffle(D)
  $D_2 \Leftarrow$ shuffle(D)
  for $i \in$ [0,num_batches] **do**
    $\mathcal{L}_{\text{batch}} = 0$
    for $(d_1, d_2) \in$ batch i of (D_1, D_2) **do**
      $\gamma \Leftarrow 1$ if label(d_1) $\neq$ label(d_2), 0 otherwise
      $\mathcal{L}_{\text{pair}} \Leftarrow \mathcal{L}_{\text{CE}}(d_1; \theta) + \mathcal{L}_{\text{CE}}(d_2; \theta) + \lambda \cdot \gamma \cdot \mathbb{D}_{\text{EC}}(d_1, d_2; \theta)$
      $\mathcal{L}_{\text{batch}} \Leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}_{\text{pair}}$
    end for
    $\theta \Leftarrow$ Backprop($\mathcal{L}_{\text{batch}}, \theta, \hat{\theta}$)
  end for
  $\hat{\theta} \Leftarrow$ ParameterUpdate(epoch, $\hat{\theta}$)
end for

ขั้นตอนวิธีที่ 1 อัลกอริทึมของคอนฟิวชั่นแบบคู่ (Pairwise Confusion)

2.2.1.3 งานวิจัยของ Dongliang Chang และคณะ[26]

เป็นงานวิจัยในปี 2020 โดยงานวิจัยนี้นำเสนอค่าสูญเสียช่องสัญญาณสอดคล้อง (Mutual-Channel loss or MC loss) ที่มีวัตถุประสงค์เพื่อให้แบบจำลองนิวรอลเน็ตเวิร์คเชิงลึกสามารถสกัดฟีเจอร์โดยเรียนรู้มาจากแบ่งแยกฟีเจอร์จากรูปภาพและระบุตำแหน่งชิ้นส่วนสำคัญของวัตถุได้พร้อมกัน โดยเสนอสมมติฐานที่ว่า ฟีเจอร์จากชิ้นส่วนสำคัญสำหรับประเภทใดจะสามารถเรียนรู้ได้จากช่องสัญญาณจำนวนหนึ่งซึ่งเป็นค่าคงที่ ตัวอย่างเช่น ช่องสัญญาณของเรสเน็ต 101 (ResNet101) สำหรับผังฟีเจอร์คือ 2048 และชุดข้อมูล Caltech-UCSD birds (CUB200-2011) มีทั้งหมด 200 คลาส หมายความว่า สำหรับคลาสใดก็ตามจะมีฟีเจอร์ของชิ้นส่วนสำคัญของวัตถุในรูปภาพที่แบบจำลองจะสามารถเรียนรู้ได้ประมาน 10 ตำแหน่งซึ่งมากเพียงพอสำหรับรูปภาพนกหรือรูปภาพอื่นๆแล้ว วิธีการคำนวณของค่าสูญเสียช่องสัญญาณสอดคล้อง (Mutual-Channel loss or MC loss) แสดงดังรูปที่ 16 และแสดงสมการที่ (37)

$$L_{MC} = L_{dis} - \lambda \times L_{div} \quad (37)$$

ซึ่ง L_{dis} คือค่าสูญเสียที่วัตถุประสงค์ในการแบ่งแยกความแตกต่างของผังฟีเจอร์ให้เป็นระเบียบและแบ่งแยกชิ้นส่วนสำคัญให้สอดคล้องตามแต่ละช่องสัญญาณเฉพาะที่เหมาะสมกับฟีเจอร์นั้นๆ ซึ่งสามารถคำนวณได้ตามมการที่ (38)

$$L_{dis} = L_{softmax}\left(\frac{\left[e^{g(F_0)}, e^{g(F_1)}, e^{g(F_2)}, \ldots, e^{g(F_{C-1})}\right]^T}{\sum_{i=0}^{C-1} e^{g(F_i)}}\right) \tag{38}$$

เมื่อฟังก์ชัน $g(.)$ แสดงดังสมการที่ (39)

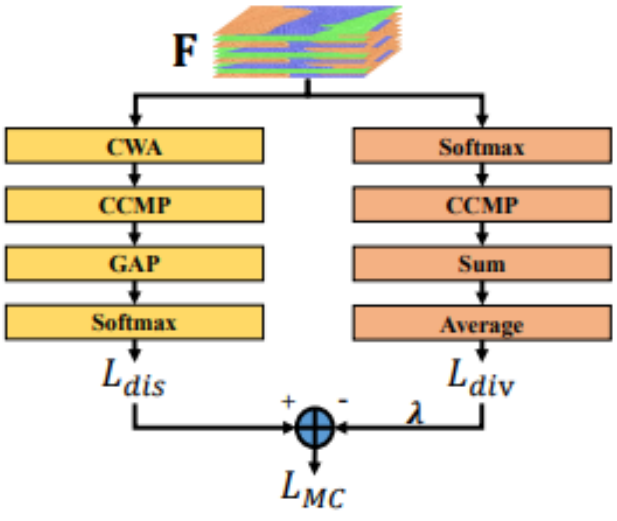
$$g(F_i) = \frac{1}{WH}\sum_{k=1}^{WH} \max_{j=1,2,..,\xi}\left[M_i \cdot F_{i,j,k}\right] \tag{39}$$

และส่วนของค่าสูญเสีย L_{div} ที่ใช้เพื่อเพิ่มความหลากหลายของฟีเจอร์ในแต่ละช่องสัญญาณสามารถคำนวณได้ตามสมการที่ (40)

$$L_{div} = \frac{1}{C}\sum_{i=0}^{C-1} h(F_i) \tag{40}$$

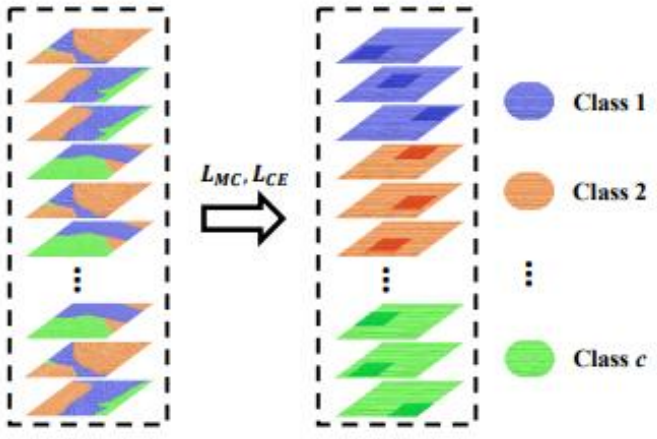
เมื่อฟังก์ชัน $h(.)$ แสดงดังสมการที่ (41)

$$h(F_i) = \sum_{k=1}^{WH} \max_{j=1,2,..,\xi}\left[\frac{e^{F_{i,j,k}}}{\sum_{k'=1}^{WH} e^{F_{i,j,k'}}}\right] \tag{41}$$

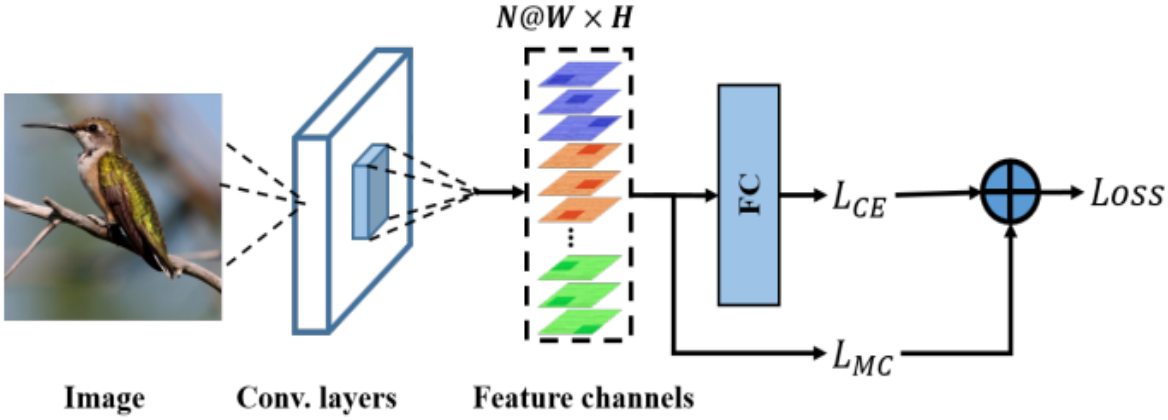


รูปที่ 16 แผนภาพการคำนวณค่าสูญเสียช่องสัญญาณสอดคล้อง

ผังฟีเจอร์ที่สกัดจากคอนโวลูชันนิวรอลเน็ตเวิร์คที่ฝึกสอนด้วยค่าสูญเสียช่องสัญญาณสอดคล้องจะมีลักษณะแบ่งแยกฟีเจอร์จากแต่ละตำแหน่งไปในช่องสัญญาณต่างๆกันอย่างชัดเจนดังรูปที่ 17 โดยฝึกฝนแบบจำลองแบบหลายวัตถุประสงค์ ร่วมกับค่าลบลอการิทึมของความเป็นไปได้แสดงดังรูปที่ 18



รูปที่ 17 ตัวอย่างผังฟีเจอร์ก่อนและหลังการฝึกสอนด้วยฟังก์ชันสูญเสีย



รูปที่ 18 โครงสร้างการของฝึกสอนแบบจำลอง

ผลการทดลองของงานวิจัยนี้แสดงให้เห็นว่า การฝึกแบบจำลองแบบหลายวัตถุประสงค์ด้วยกันกับฟังก์ชันสูญเสียช่องสัญญาณสอดคล้องที่นำเสนอกับฟังก์ชันค่าสูญเสียสูงสุดอย่างอ่อน นอกจากจะช่วยเพิ่มประสิทธิภาพในการจัดการ การสกัดและการแยกแยะฟีเจอร์บนช่องสัญญาณของผังฟีเจอร์แล้ว ยังช่วยให้แบบจำลองระบุตำแหน่งวัตถุได้ดีขึ้นอีกด้วย ซึ่งความแม่นยำของแบบจำลองนี้

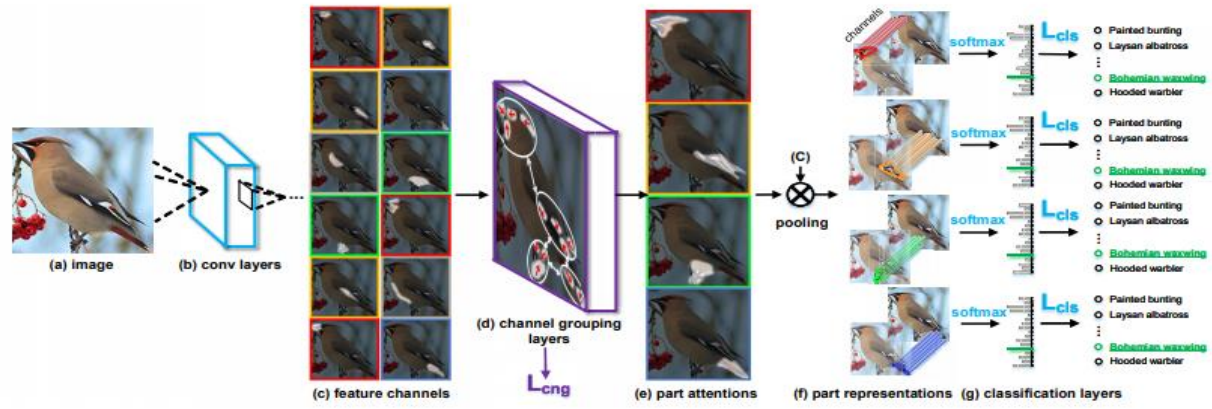
อยู่ที่ 87.3%, 92.6% and 93.7% สำหรับชุดข้อมูล Caltech-UCSD birds, FGVC aircraft, และ Stanford cars ซึ่งดีกว่างานวิจัยของ Tsung-Yu Lin และคณะ[16] และของ งานวิจัยของ Abhimanyu Dubey และคณะ[25]

2.2.2 ระบุตำแหน่งและแบ่งประเภทแบบซับเน็ตเวิร์ค (Localization-Classification Sub-network)

เนื่องด้วยปัญหาความเหมือนระหว่างประเภทสูง (High Intra-class similarity) ของการจำแนกประเภทแบบละเอียด ทำให้การสกัดฟีเจอร์จากรูปภาพต้องให้ความสำคัญกับการระบุตำแหน่งวัตถุเพื่อที่แบบจำลองจะสามารถเข้าใจถึงความสำคัญของพื้นที่ต่างๆได้อย่างถูกต้อง รวมไปถึงระบุถึงตำแหน่งของชิ้นส่วน (Parts Level) ต่างๆของภาพ เช่น ความแตกต่างของจงอยปากนก หรือส่วนปีก และสกัดฟีเจอร์ออกมาจากการวิเคราะห์ชิ้นส่วนเหล่านั้นโดยตรงอย่างถูกต้อง ซึ่งเป็นหน้าที่ของเน็ตเวิร์คหนึ่ง หลังจากนั้นจะนำฟีเจอร์ที่สกัดมาได้มาทำการจำแนกประเภทบนเน็ตเวิร์คอีกส่วนหนึ่ง โดยมีตัวอย่างงานวิจัยดังนี้

2.2.2.1 งานวิจัยของ Heliang Zheng และคณะ[14]

เป็นงานวิจัยในปี 2017 โดยงานวิจัยนี้นำเสนอแบบจำลองการเรียนรู้เชิงลึกเพื่อระบุตำแหน่งชิ้นส่วนสำคัญในแต่ละรูปภาพ และแบ่งประเภทจากชิ้นส่วนเหล่านั้นในหลายตำแหน่ง ซึ่งเป็นชิ้นส่วนที่แสดงเอกลักษณะสำหรับแต่ละประเภทแบบ 2 ชิ้นและ 4 ชิ้น เป็นการเรียนรู้แบบหลายวัตถุประสงค์ อีกทั้งยังนำเสนอฟังก์ชันสูญเสียเพื่อระบุตำแหน่งชิ้นส่วนอีกด้วย ดังรูปที่ 19

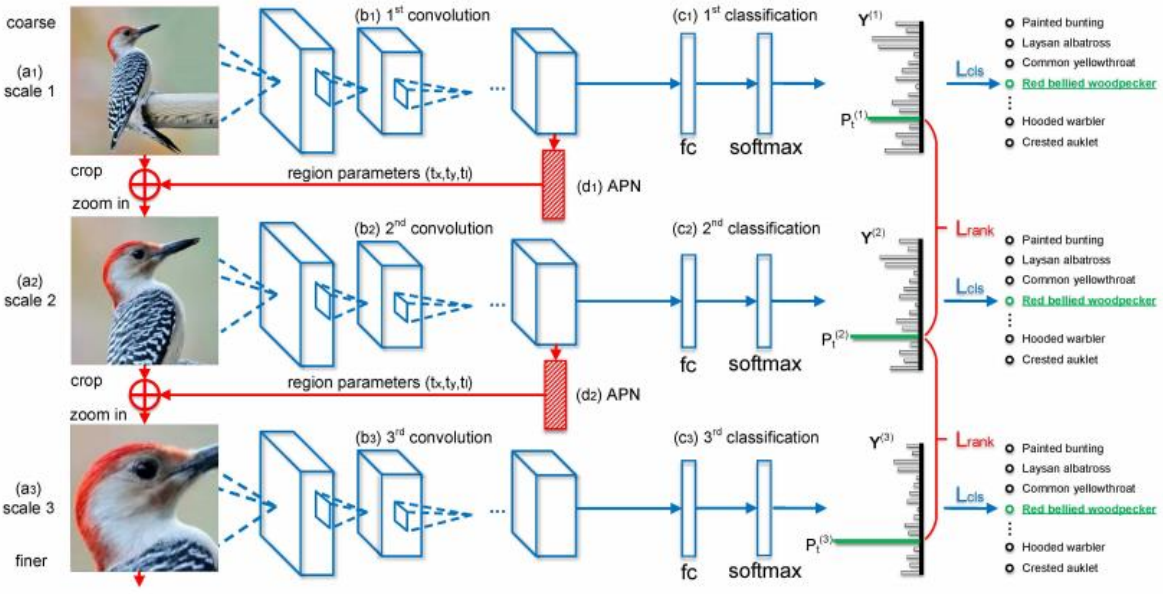


รูปที่ 19 โครงสร้างการของแบบจำลอง

ผลการทดลองสำหรับงานวิจัยนี้ แสดงให้เห็นว่า จำนวนชิ้นส่วน และตำแหน่งมีส่วนช่วยในการแยกแยะฟีเจอร์ ความแม่นยำของแบบจำลองเมื่อใช้ชิ้นส่วน 4 ชิ้นในการฝึกดีกว่าแบบจำลองที่ใช้ 2 ชิ้น 1.1% โดยเฉลี่ย ซึ่งความแม่นยำของแบบจำลองนี้เมื่อใช้ฟีเจอร์ของชิ้นส่วน 4 ชิ้นอยู่ที่ 86.5%, 89.9% and 92.8% สำหรับชุดข้อมูล Caltech-UCSD birds, FGVC aircraft, และ Stanford cars

2.2.2.2 งานวิจัยของ Jianlong Fu และคณะ[15]

เป็นงานวิจัยในปี 2017 งานวิจัยนี้นำเสนอวิธีการฝึกฝนแบบจำลองการเรียนรู้เชิงลึก เพื่อให้สามารถแยกแยะความแตกต่างที่ละเอียดอ่อนของรูปภาพได้ โดยระบุตำแหน่งของชิ้นส่วนพิจารณาผังฟีเจอร์เพื่อเน้นไปที่บริเวณของวัตถุ และนำเสนอฟังก์ชันสูญเสียที่ช่วย ให้แบบจำลองรับรู้ได้ว่า ทุกๆ การระบุตำแหน่งวัตถุฟีเจอร์จะมีความสำคัญมากขึ้น แสดงดังรูปที่ 20



รูปที่ 20 โครงสร้างการของแบบจำลอง

ผลการทดลองสำหรับงานวิจัยนี้ แสดงให้เห็นว่าการระบุตำแหน่งอย่างแม่นยำมีส่วนช่วยให้การจำแนกประเภทมีประสิทธิภาพมากขึ้นด้วย ซึ่งความแม่นยำของแบบจำลองนี้อยู่ที่ 86.5%, 89.9% and 92.8% สำหรับชุดข้อมูล Caltech-UCSD birds, FGVC aircraft, และ Stanford cars

2.2.3 ฝึกสอนด้วยข้อมูลเพิ่มเติม (Training with External information)

นอกเหนือจากงานวิจัยที่เกี่ยวข้องกับการพัฒนาแบบจำลองการเรียนรู้เชิงลึกเพื่อให้นิวรอลเน็ตเวิร์คมีประสิทธิภาพมากขึ้นทั้งในการสกัดฟีเจอร์ การแบ่งแยกฟีเจอร์และการระบุตำแหน่งแล้ว อีกปัจจัยหนึ่งที่ทำให้แบบจำลองมีประสิทธิภาพดีขึ้นได้ เพื่อที่จะแบ่งแยกประเภทของภาพที่มีความคล้ายกันมากของการจำแนกประเภทภาพแบบละเอียด ต้องใช้ชุดข้อมูลรูปภาพที่ระบุคลาสคำตอบอย่างถูกต้อง อย่างไรก็ตามการรวบรวมชุดข้อมูลรูปภาพ เช่น ชนิดของนก (Species of Bird) และระบุชนิดได้อย่างถูกต้องจำเป็นต้องใช้ผู้เชี่ยวชาญเฉพาะด้าน ซึ่งอาศัยทรัพยากรและเวลามาก จึงมีงานวิจัยที่นำข้อมูลรูปภาพจากเว็บไซต์ (Web Data) ซึ่งเกี่ยวข้องกับชุดข้อมูลเพิ่มเข้าไปในชุดข้อมูลสำหรับฝึกฝนแบบจำลอง ซึ่งอาจจะไม่จำเป็นต้องระบุคลาสอย่างถูกต้อง จะสามารถจัดการกับข้อมูลประเภทนี้ได้ด้วยเทคนิค ข้อมูลรบกวน (Noisy Data) ตัวอย่างงานวิจัยของ Yin Cui et al. และคณะ [30] และงานวิจัยของ Jonathan Krause และคณะ[31]

นอกจากนี้ ยังมีงานวิจัยที่นำข้อมูลหลายรูปแบบ (Multi-modality Data) เช่น คำอธิบายที่เกี่ยวข้องกับรูปนั้นๆ (Text Descriptions) หรือฐานความรู้เกี่ยวกับรูปภาพ (Knowledge Base) เป็นตัวช่วยในการฝึกฝนแบบจำลองคู่กันกับรูปภาพเพื่อช่วยเพิ่มความแม่นยำ ตัวอย่างงานวิจัยของ Scott Reed และคณะ[32] และงานวิจัยของ Xiangteng He และ Yuxin Peng[33]

2.2.4 งานวิจัยเกี่ยวกับฟังก์ชันสูญเสีย (Loss Function)

งานวิจัยเกี่ยวกับฟังก์ชันสูญเสีย (Loss Function) เป็นงานวิจัยอีกรูปแบบหนึ่งที่ได้รับความนิยมเนื่องจากมีความหลากหลายในการออกแบบ ซึ่งการออกแบบฟังก์ชันสูญเสียให้มีวัตถุประสงค์เฉพาะในการฝึกสอนเพื่อแก้ปัญหาบางประการ จะสามารถช่วยให้แบบจำลองสามารถเรียนรู้จากชุดข้อมูลที่มีความยากและปัญหาเฉพาะได้อย่างเหมาะสม โดยมีตัวอย่างงานวิจัยที่นำมาประยุกต์ใช้เพื่อแก้ปัญหาที่คล้ายกับปัญหาของการจำแนกประเภทภาพแบบละเอียดดังนี้

2.2.4.1 งานวิจัยของ Yandong Wen และคณะ[24]

เป็นงานวิจัยนี้ปี 2016 เพื่อใช้ในงานวิจัยที่ออกแบบฟังก์ชันสูญเสียศูนย์กลาง (Center loss) เพื่อใช้ในงานกับงานวิจัยการจดจำใบหน้า ซึ่งใบหน้าของมนุษย์ส่วนใหญ่มีโครงสร้างและรูปแบบของอวัยวะที่คล้ายคลึงกันมาก ทำให้เป็นงานที่มีปัญหาความผันผวนภายในประเภทสูง (High Intra-class

Variance) และความเหมือนระหว่างประเภทสูง (High Inter-class Similarity) ซึ่งเป็นปัญหาที่เหมือนกับการจำแนกประเภทแบบละเอียด

งานวิจัยนี้นำเสนอฟังก์ชันเพื่อคำนวณค่าสูญเสียศูนย์กลาง (Center loss) เพื่อเพิ่มประสิทธิภาพในการเรียนรู้และสกัดฟีเจอร์โดย มีวัตถุประสงค์ในการลดความผันผวนภายในประเภทโดยยังคงให้ฟีเจอร์เวกเตอร์ที่สกัดได้จากรูปในแต่ละประเภทแยกออกจากกันโดยพยายามลดค่าความห่าง (Distance) ระหว่างฟีเจอร์เวกเตอร์ของแต่ละประเภทกับเวกเตอร์ศูนย์กลางของประเภทนั้นๆ ซึ่งสามารถคำนวณได้จากสมการที่ (42)

$$L_c = \frac{1}{2}\sum_{i=1}^{m}\left\|x_i - c_{y_i}\right\|_2^2 \tag{42}$$

เมื่อกำหนดให้ c_{y_i} คือค่าศูนย์กลางของฟีเจอร์ที่สกัดมาจากคอนโวลูชันนิวรอลเน็ตเวิร์คในประเภท y_i โดยงานวิจัยนี้นำเสนอวิธีการใช้งานฟังก์ชันสูญเสียศูนย์กลาง (Center loss) ไว้ว่าควรใช้งานร่วมกับฟังก์ชันค่าสูญเสียค่าสูงสุดอย่างอ่อน เพื่อให้ได้วัตถุประสงค์ในการฝึกสอนครบถ้วน

2.2.4.2 งานวิจัยของ Weiyang Liu และคณะ[34]

เป็นงานวิจัยนี้ปี 2017 เพื่อใช้ในงานวิจัยที่ออกแบบฟังก์ชันสูญเสียมาจินเชิงมุม (Angular Margin Softmax loss or AM loss) เพื่อใช้ในงานเกี่ยวกับกับการจดจำใบหน้าเช่นเดียวกับงานวิจัยที่กล่าวไปข้างต้น ซึ่งเป็นปัญหาที่เหมือนกับการจำแนกประเภทแบบละเอียด โดยนำเสนอวิธีการปรับแต่งค่าสูญเสียค่าสูงสุดอย่างอ่อน (Softmax loss) ซึ่งสามารถคำนวณได้จากสมการที่ (43)

$$L_{softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{43}$$

เมื่อกำหนดให้ $W_j^T x_i + b_j$ คือสมการคำนวณเชิงเส้นบนชั้นการเชื่อมโยงเต็มรูปแบบ W_j^T และ b_j คือค่าน้ำหนักและค่าไบแอสบนตำแหน่งที่ j ตามลำดับและ x_i คือฟีเจอร์เวกเตอร์ที่สกัดมาจากคอนโวลูชันนิวรอลเน็ตเวิร์ค ด้วยสมการคำนวณความเหมือนของโคไซน์ (Cosine Similarity) ทั้งนี้ในขั้นตอนการออกแบบได้กำหนดให้ $\left\|W_j^T\right\|$ หรือขนาดของเวกเตอร์ค่าน้ำหนักบนตำแหน่งที่ j ใดๆมีค่าเท่ากับ 1 และให้ค่าไบแอสเท่ากับ 0 แสดงดังสมการที่ (44) และ (45)

$$\cos\theta_j = \frac{W_j^T x_i}{\|W_j^T\|\|x_i\|} \tag{44}$$

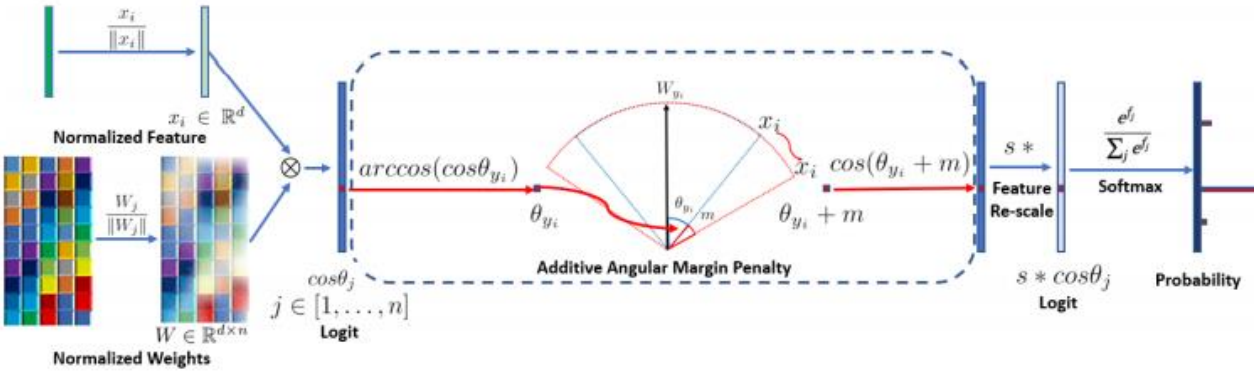
$$L_{AM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\|x_i\|\cos m\theta_{y_i}}}{e^{\|x_i\|\cos m\theta_{y_i}} + \sum_{j=1,j\neq y_i}^{n} e^{\|x_i\|\cos\theta_j}} \tag{45}$$

เมื่อกำหนดให้ x_i คือฟีเจอร์เวกเตอร์บนชั้นเชื่อมโยงเต็มรูปแบบซึ่งต่อมาจากคอนโวลูชันนิวรอลเน็ตเวิร์ค และ $\cos\theta_j$ คือมุมระหว่าง ฟีเจอร์เวกเตอร์ x_i และค่าน้ำหนักของชั้นเชื่อมโยงเต็มรูปแบบ และเพิ่มค่ามาจินเชิงมุม m เข้าบนมุมระหว่าง ฟีเจอร์และเวกเตอร์ค่าน้ำหนัก ซึ่งจะมีค่าเท่ากับ $\cos(m\theta)_{y_i}$ เพื่อเป็นการบังคับให้ คอนโวลูชันนิวรอลเน็ตเวิร์คพยายามเรียนรู้เพื่อให้จำแนกประเภทได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

2.2.4.3 งานวิจัยของ Jiankang Deng และคณะ[28]

เป็นงานวิจัยในปี 2019 ที่ออกแบบเพื่องานวิจัยเกี่ยวกับการจำจดใบหน้าที่ได้รับความนิยมมากในปัจจุบันเนื่องจากสามารถนำไปประยุกต์ใช้ได้ไม่ว่าจะเป็นระบบรักษาความปลอดภัยชีวภาพ (Biometric Security) หรือแอปพลิเคชันบนโทรศัพท์มือถือ (Mobile Applications) ซึ่งใบหน้าของมนุษย์ส่วนใหญ่มีโครงสร้างที่คล้ายคลึงกันมาก ทำให้เกิดปัญหาความเหมือนระหว่างคลาสสูง (High Inter-class similarity) และความผันผวนภายในคลาสก็สูงเช่นกัน (High Intra-class variance) ซึ่งปัญหาดังกล่าวคือ ปัญหาเดียวกันกับของการจำแนกประเภทภาพแบบละเอียด

งานวิจัยนี้ออกแบบฟังก์ชันสูญเสียอาร์คเฟซ (ArcFace) ที่แบ่งแยกฟีเจอร์จากรูปภาพ ให้มีความแตกต่างระหว่างคลาสที่ชัดเจนมากขึ้น โดยต่อยอดมาจากงานวิจัยของ *Weiyang Liu และคณะ* หรือฟังก์ชันสูญเสียมาจินเชิงมุม (Angular Margin Softmax loss or AM loss) ซึ่งทำการเพิ่มมาจินเชิงมุมเข้าไปคุณกับค่ามุมที่คำนวณมาจากชั้นการเชื่อมโยงเต็มรูปแบบ แผนภาพการคำนวณของฟังก์ชันสูญเสียอาร์คเฟซ (ArcFace) แสดงดังรูปที่ 21



รูปที่ 21 ผังการคำนวณของฟังก์ชันสูญเสียอาร์คเฟซ (ArcFace)

เมื่อเปรียบเทียบกับค่าลบลอการิทึมของความเป็นไปได้ซึ่งคำนวณจากความน่าจะเป็นจากฟังก์ชันค่าสูญเสียมาจินเชิงมุม (Angular Margin Softmax loss or AM loss) ตามสมการที่ (45) แล้วเมื่อนอร์มัลไลเซชัน (Normalization) ค่าน้ำหนักของชั้นการเชื่อมโยงเต็มรูปแบบ และค่ารับเข้าแล้วผลคูณเชิงสเกลา (Dot Product) ของเวกเตอร์ทั้งสอง จะมีค่าเท่ากับค่าความเหมือนของโคไซน์ $\cos\theta_{y_i}$ หรือมุมระหว่างเวกเตอร์ทั้งสองนั้น ในงานวิจัยนี้นำเสนอวิธีการขยายขอบเขตเชิงมุม (Additive Angular Margin Penalty) หรือวิธีการเพิ่มมาจินเชิงมุม m เป็นค่าคงที่บนตำแหน่งของประเภทของเวกเตอร์ฟีเจอร์ในแต่ละรูปภาพทำให้ค่าความเหมือนของโคไซน์มีค่าลดลงซึ่งเท่ากับ $\cos(\theta+m)_{y_i}$ เพื่อเป็นการบังคับให้แบบจำลองเรียนรู้ที่จะแบ่งแยกฟีเจอร์ของแต่ละตัวตนหรือของรูปภาพของแต่ละคนซึ่งมีความคล้ายกันมากได้ดีมากยิ่งขึ้นเพื่อปรับให้ค่าความเหมือนความของโคไซน์กลับไปเป็นค่าที่มาก หรือใกล้เคียงเดิม ตามสมการที่ (46) เมื่อวัดผลการทดลองด้วยการทดสอบความแม่นยำจากการจดจำใบหน้าแล้วพบว่า มีความแม่นยำมากกว่าค่าสูญเสียมาจินเชิงมุม (AM loss)

$$L_{Arcface} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \quad (46)$$

บทที่ 3

การระบุตำแหน่งวัตถุ ฟังก์ชันค่าสูญเสียมาจินเชิงมุมปรับค่าได้ และแบบจำลองรูปภาพฝังตัวแบบมีประสิทธิภาพ

ในบทนี้จะอธิบายถึงการออกแบบและรายละเอียดวิธีการคำนวณของเทคนิควิธีที่ใช้ในงานวิจัยนี้โดยแบ่งเทคนิคออกเป็น 2 ส่วนคือเทคนิคการระบุตำแหน่งวัตถุ (Localization Method) ภายในรูปภาพ เพื่อให้แบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์คสามารถเรียนรู้จากพื้นที่ของวัตถุในรูปภาพได้อย่างมีประสิทธิภาพ โดยไม่ใช้กล่องขอบเขต (Boundary Box) ซึ่งเป็นข้อมูลเพิ่มเติม และฟังก์ชันค่าสูญเสียมาจินเชิงมุมแบบปรับค่าได้ (Adaptive Angular Margin Loss or AAM loss) ที่พัฒนาและต่อยอดมาจากฟังก์ชันสูญเสียค่าสูงสุดอย่างอ่อน (Softmax loss) ซึ่งช่วยให้แบบจำลองแก้ปัญหาความเหมือนระหว่างประเภทสูง (Inter-class Similarity) และความผันผวนภายในประเภทสูง (Intra-class Variation) ได้และเพิ่มความแม่นยำในการจำแนกประเภท ซึ่งมีรายละเอียดดังนี้

3.1 การระบุตำแหน่งวัตถุ (Localization Method)

ในงานวิจัยนี้ได้มีการออกแบบเทคนิคการระบุตำแหน่งวัตถุที่ต่อยอดมาจากกระบวนการระบุตำแหน่งวัตถุ (Localization Method) จากงานวิจัยของ Xiu-Shen Wei และคณะ[29] ซึ่งสามารถระบุตำแหน่งวัตถุได้จากผังฟีเจอร์ (Feature maps) ที่ได้มาระหว่างการฝึกสอนแบบจำลองด้วยแบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์คโดยไม่ต้องใช้กล่องขอบเขต (Boundary Box) ในการฝึกสอน

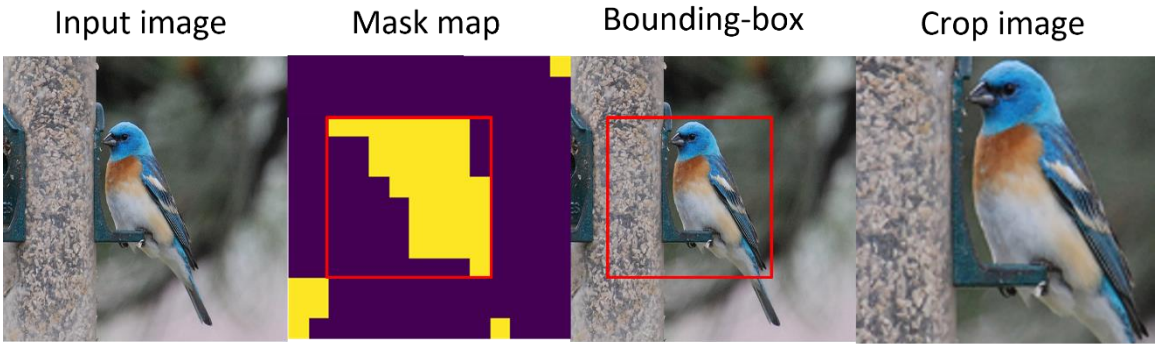
โดยกำหนดให้ข้อมูลรูปภาพขาเข้า (Input) ส่งผ่านขั้นตอนการป้อนไปข้างหน้าบนแบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์ค ซึ่งจะสกัดผลลัพธ์เป็นผังฟีเจอร์ (Feature maps) F ซึ่งเป็นเมทริกซ์ที่มีมิติเท่ากับ $k \times h \times w$ โดยกำหนดให้ k แทนจำนวนช่องสัญญาณ (Channels) ของผังฟีเจอร์ h แทนความสูงของผังฟีเจอร์ และ w แทนความกว้างของผังฟีเจอร์ หลักจากนั้นนำ ผังฟีเจอร์ (Feature maps) F ที่มีจำนวนช่องสัญญาณ k มาทำการปรับรวม (Aggregation) โดยรวมค่าบนตำแหน่ง $h \times w$ เดียวกันสำหรับทุกๆช่องสัญญาณ ตามสมการ (47) ซึ่งจะช่วยให้เห็นแผนผังความร้อน (Heat map) ของผังฟีเจอร์โดยภาพรวมและเข้าใจได้ว่าแบบจำลองให้ความสำคัญและพิจารณาพื้นที่ส่วนไหนในระหว่างขั้นตอนการฝึกสอน

$$\boldsymbol{F}_{(x,y)} = \sum_{i=1}^{k} F_{(x,y)} \tag{47}$$

หลังจากนั้นนำผังฟีเจอร์ชาแนล $\boldsymbol{F}_{(x,y)}$ ที่ได้หลังจากทำการปรับรวม มาทำผังคัดกรอง (Mask map) $M_{(x,y)}$ ที่ใช้เพื่อเลือกเฉพาะพื้นที่ที่มีความสำคัญและคาดว่าจะเป็นตำแหน่งของวัตถุภายในรูป ด้วยเงื่อนไขตามสมการที่ (48) เมื่อค่า â คือค่าเฉลี่ยของผลรวมทุกตำแหน่ง $h \times w$ บนผังฟีเจอร์ชาแนล $\boldsymbol{F}_{(x,y)}$

$$M_{(x,y)} = \begin{cases} 1, & if\ \boldsymbol{F}_{(x,y)} > \hat{a} \\ 0, & else \end{cases} \tag{48}$$

สมมติฐานสำหรับการกำหนดกล่องขอบเขตของรูปภาพแต่ละรูปนั้น คือพื้นที่ที่เชื่อมต่อกันและมีขนาดใหญ่ที่สุด แต่เนื่องจากผังคัดกรอง (Mask map) $M_{(x,y)}$ มีขนาดความกว้างและความสูงไม่เท่ากันกับขนาดของรูปภาพที่ใช้เป็นข้อมูลขาเข้า เพื่อให้สามารถนำกล่องขอบเขตที่มาจาก การพิจารณาผังคัดกรอง ไปใช้ครอบตัดรูปภาพได้ตามสัดส่วนที่ถูกต้องจึงต้องมีการทำการประมาณค่าช่วง (Interpolation) เพื่อขยายสัดส่วนให้ขนาดของผังคัดกรอง (Mask map) $M_{(x,y)}$ มีขนาดใหญ่ขึ้นให้เท่ากันกับขนาดของรูปภาพที่จะทำการครอบตัดรูปภาพ เช่น เมื่อขั้นตอนการฝึกสอนนี้ใช้ขนาดของรูปภาพที่ 448 พิกเซล และคอนโวลูชันนิวรอลเน็ตเวิร์คเป็นเรสเน็ตห้าสิบ (ResNet50) ขนาดของผังคัดกรองจะเท่ากับ 14 พิกเซล จะต้องมีการประมาณค่าช่วง จากขนาด 14 ให้เท่ากันกับ 448 โดยขั้นตอนกระบวนการระบุตำแหน่งวัตถุแสดงดังรูปที่ 22



รูปที่ 22 กระบวนการระบุตำแหน่ง และผลการตัดรูปภาพ

3.2 ฟังก์ชันค่าสูญเสียมาจินเชิงมุมปรับค่าได้ (Adaptive Angular Margin or AAM Loss)

ในขั้นตอนของการฝึกสอนแบบจำลองจำเป็นต้องใช้ฟังก์ชันสูญเสียเพื่อปรับปรุงแบบจำลองในทุกๆรอบของการฝึกสอน ซึ่งโดยปกติแล้วสำหรับการจำแนกประเภทภาพจะใช้ฟังก์ชันสุญเสียที่เป็นที่นิยม คือการใช้ ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function) รวมกันกับค่าลบลอการิทึมของความเป็นไปได้ (Negative Log-Likelihood) โดยเรียกว่า ค่าสูญเสียค่าสูงสุดอย่างอ่อน (Softmax Loss) ซึ่งแสดงดังสมการที่ (49)

$$L_{softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{49}$$

โดย x_i คือฟีเจอร์ที่เรียนรู้มาจากแบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์คผ่านการส่งผ่านข้อมูลไปเป็นจำนวน N ตัว และ y_i คือคำตอบประเภทของข้อมูลขาเข้านั้นๆ ซึ่งมีทั้งหมด n ประเภท W_j และ b_j คือน้ำหนัก (weights)และไบแอส (bias) ตามลำดับ

ในงานวิจัยนี้ได้ต่อยอดมาจากงานวิจัยของ Jiankang Deng และคณะ[28] ที่ปรับปรุงค่าสูญเสียค่าสูงสุดอย่างอ่อน (Softmax Loss) เพื่อใช้สำหรับการจดจำใบหน้า (Face Recognition) ซึ่งเป็นงานที่มีปัญหาหลักคล้ายกันกับการจำแนกประเภทภาพแบบละเอียด จึงเกิดแนวคิดในการออกแบบค่าสูญเสียมาจินเชิงมุมปรับค่าได้ (Adaptive Angular Margin Loss) โดยนำฟังก์ชันสูญเสียสูญเสียอาร์คเฟซ (ArcFace) ที่แสดงดังสมการที่ (50) มาพัฒนาเพิ่มเติม

$$L_{ArcFace} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\,(\cos(\theta_{y_i}+m))}}{e^{s\,(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{50}$$

ฟังก์ชันสูญเสียสูญเสียอาร์คเฟซที่ใช้ในแบบจำลองการเรียนรู้เชิงลึกสำหรับการจดจำใบหน้า ซึ่งในงานวิจัย Jiankang Deng และคณะ[28] ได้นำเสนอไว้ด้วยการคำนวณหามุมระหว่างเวกเตอร์ประจำคลาส และฟีเจอร์เวกเตอร์จากข้อมูลขาเข้า และฝึกสอน (Training Stage) แบบจำลองด้วยการขยายขอบเขตเชิงมุม (Additive Angular Margin Penalty) หรือวิธีการเพิ่มมาจินเชิงมุม m เป็นค่าคงที่บนตำแหน่งของประเภทของเวกเตอร์ฟีเจอร์ในแต่ละรูปภาพทำให้ค่าความเหมือนของโคไซน์มีค่าลดลงซึ่งเท่ากับ $\cos(\theta+m)_{y_i}$ สำหรับค่ามาจินเชิงมุม m ในงานวิจัยดั้งเดิมของ Jiankang Deng และคณะ[28] จะใช้ค่าคงที่ส่งผลให้ในตอนต้นของการฝึกสอนแบบจำลองอาร์คเฟซจะให้ค่า

สูญเสียที่ค่อนข้างเยอะ ทำให้แบบจำลองเรียนรู้ได้ช้าได้ช่วงแรก และเข้าถึงจุดที่ดีที่สุดได้ช้า ซึ่งในงานวิจัยนี้ จะเปลี่ยนมาใช้ค่ามาจินเชิงมุมแบบปรับตัวได้ (Adaptive Angular Margin) ซึ่งเป็นฟังก์ชันที่ปรับค่ามาจินเชิงมุมไปตามจำนวนรอบของการฝึกสอน $f(e)$ แสดงดังสมการที่ (51) แต่ในช่วงการทดสอบ (Test Stage) จะใช้ค่ามุมโดยไม่มีการขยายขอบเขต หรือใช้ฟังก์ชันค่าสูงสุดอย่างอ่อนทำการจำแนกประเภทเหมือนปกติ เพราะว่า การที่นำเวกเตอร์ผลลัพธ์ที่มีการเพิ่มมาจินจะทำให้การทำนายคลาดเคลื่อนได้

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\,(\cos(\theta_{y_i}+f(e)))}}{e^{s\,(\cos(\theta_{y_i}+f(e)))}+\sum_{j=1,j\neq y_i}^{n}e^{s\,\cos\theta_j}} \qquad (51)$$

ขั้นตอนการฝึกสอนแบบจำลองด้วยการปรับค่ามาจินเชิงมุมจะเริ่มที่ ค่าไม่มีมาจิน $(m=0)$ และเพิ่มขึ้นโดยเป็นฟังก์ชันของจำนวนรอบการฝึกสอน (epochs) ของการฝึกสอนโดยเพิ่มไปจนถึงค่าหนึ่งแล้วจะใช้ค่าสูงสุดนั้น ในการฝึกสอนไปจนจบรอบทั้งหมด ซึ่งในงานวิจัยนี้ ฟังก์ชันปรับค่าได้จะใช้ฟังก์ชัน 3 ประเภท คือ ฟังก์ชันขั้นบันได (Step Function) ฟังก์ชันเชิงเส้น (Linear Function) และฟังก์ชันเอกซ์โพเนนเชียล (Exponential Function) ซึ่งแสดงดังสมการที่ (52) - (54) ตามลำดับ

$$f(e)=\begin{cases}0 & ;e\ \le\ k_1\\ m_1 & ;k_1<e\ \le\ k_2\\ m_2 & ;\ k_2<e\end{cases} \qquad (52)$$

$$f(e)=\begin{cases}0 & ;e\ \le\ k_1\\ \frac{e-k_1}{k_2-k_1}(m_1) & ;k_1<e\ \le\ k_2\\ m_2 & ;\ k_2<e\end{cases} \qquad (53)$$

$$f(e)=\begin{cases}0 & ;e\ \le\ k_1\\ m_1 e^{\left(\frac{e-k_1}{k_2-k_1}-1\right)} & ;k_1<e\ \le\ k_2\\ m_2 & ;\ k_2<e\end{cases} \qquad (54)$$

3.3 แบบจำลองรูปภาพฝังตัวแบบมีประสิทธิภาพ

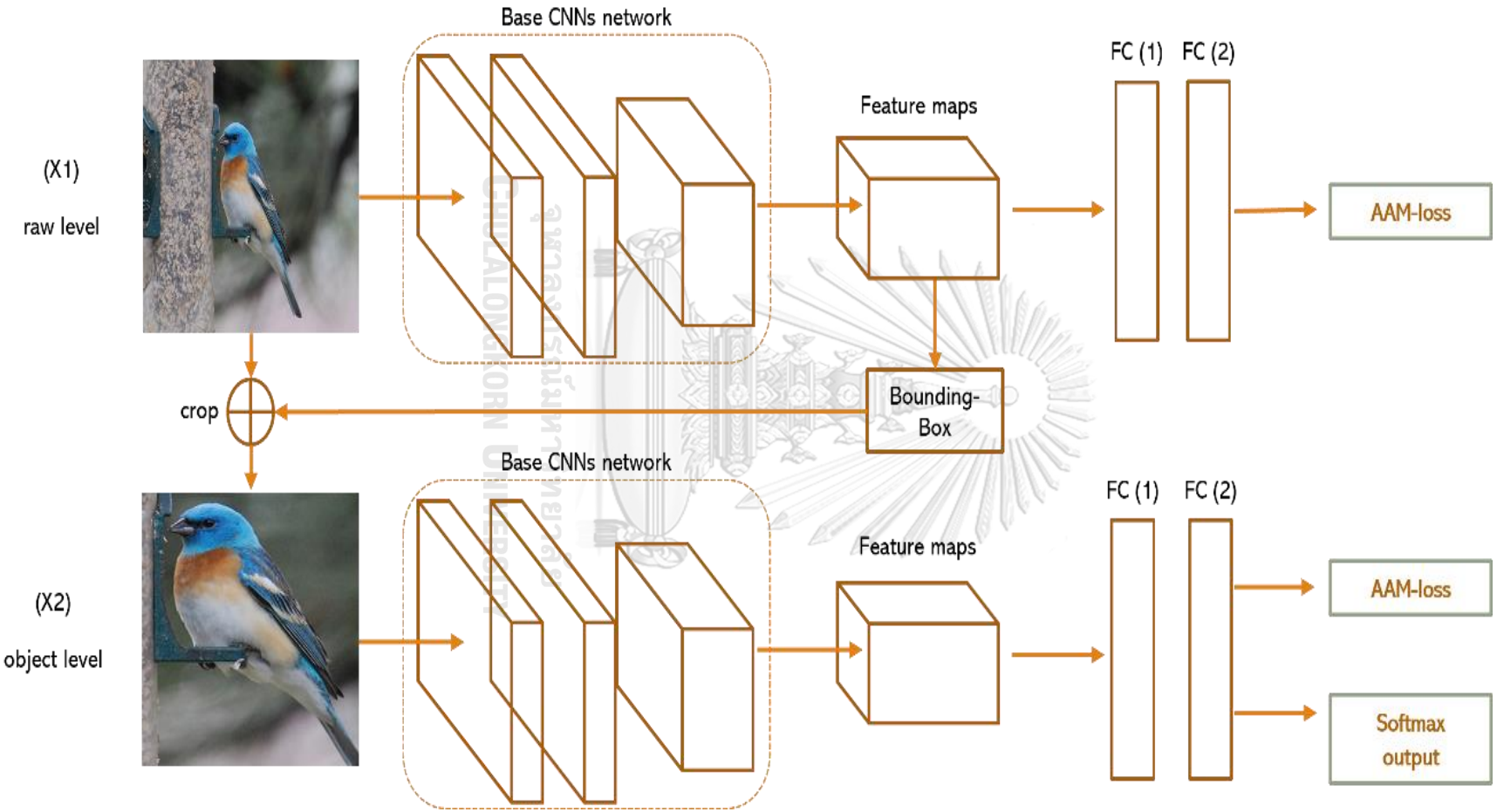
ขั้นตอนการฝึกสอนโดยรวมของแบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์คโดยรวมที่ใช้ในงานวิจัยนี้แสดงดังรูปที่ 23 โดยแบ่งแบบจำลองออกเป็น สองระดับคือระดับข้อมูลดั้งเดิม (Raw Level) ซึ่งทำหน้าที่ระบุตำแหน่งวัตถุและครอบตัดรูปภาพด้วยเทคนิคที่กล่าวไปในหัวข้อที่ 3.1 และระดับของวัตถุ (Object Level) ที่ใช้สำหรับจำแนกประเภทซึ่งทั้งสองระดับฝึกสอนด้วยฟังก์ชันค่าสูญเสียมาจินเชิงมุมปรับค่าได้ (AAM loss) แต่ในการใช้งานจำแนกประเภทหรือการทดสอบ (Test Stage) จะใช้ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function) แทนเนื่องจากการเพิ่มมาจินจะทำให้การทำนายคลาดเคลื่อนได้

กำหนดให้ข้อมูลรูปภาพขาเข้า **X1** ที่ถูกส่งเข้าแบบจำลองในระดับข้อมูลดั้งเดิม (Raw Level) ผ่านขั้นตอนการป้อนไปข้างหน้าด้วยคอนโวลูชันนิวรอลเน็ตเวิร์ค และสกัดออกมาเป็นผังฟีเจอร์ (Feature maps) ซึ่งจะนำไปผ่านกระบวนการระบุตำแหน่งวัตถุ และครอบตัดภาพ ซึ่งในระดับนี้ก็จะมีการคำนวณค่าสูญเสียเพื่อให้แบบจำลองสามารถครอบตัดภาพได้แม่นยำมากขึ้นด้วยค่าสูญเสียมาจินเชิงมุมปรับค่าได้ (AM loss) หลังจากนั้น ภาพที่ถูกตัดแล้วให้ภาพของวัตถุที่ชัดเจนมากยิ่งขึ้น และนำไปจำแนกประเภทในระดับของวัตถุ (Object Level) ด้วยคอนโวลูชันนิวรอลเน็ตเวิร์คที่มีการแบ่งค่าน้ำหนัก (Weight Sharing) มาจากคอนโวลูชันนิวรอลเน็ตเวิร์คในระดับข้อมูลดั้งเดิม

โดยในคอนโวลูชันนิวรอลเน็ตเวิร์คทั้งระดับวัตถุ (Object Level) และระดับข้อมูลดั้งเดิม (Raw Level) จะฝึกสอนแบบจำลองด้วยค่าสูญเสียโดนยรวมของแบบจำลอง $L_{overall}$ ในงานวิจัยนี้แสดงดังสมการที่ (55)

$$L_{overall} = L_{raw-Level} + L_{object-Level} \quad (55)$$

เมื่อ $L_{raw-Level}$ และ $L_{object-Level}$ คือ ค่าสูญเสียมาจินเชิงมุมปรับค่าได้ (AAM loss) ที่คำนวณบนแบบจำลองคอนโวลูชันนิวรอลเน็ตเวิร์ค ในระดับวัตถุ (Object Level) และระดับข้อมูลดั้งเดิม (Raw Level) ตามลำดับ



รูปที่ 23 แผนภาพแสดงโครงสร้างของแบบจำลองโดยรวม

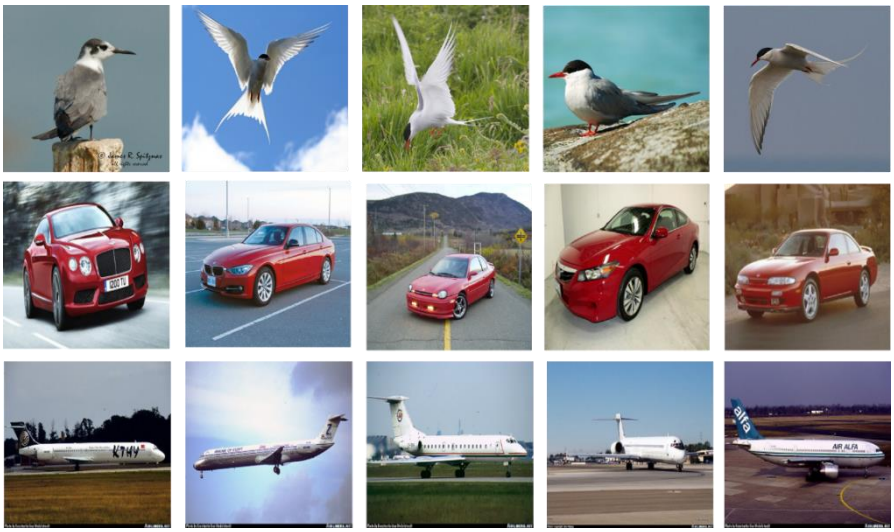
บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงรายละเอียดของการทดลองและผลการทดลองในงานวิจัยนี้ การทดลองนี้เพื่อวัดประสิทธิภาพของแบบจำลอง เทคนิค และฟังก์ชันสูญเสียที่ใช้ในการฝึกสอนแบบจำลองที่ใช้ในงานวิจัยโดยเปรียบเทียบประสิทธิภาพด้วยค่าความแม่นยำ (Accuracy) เทียบกับเทคนิคและงานวิจัยอื่นๆ รายละเอียดในบทนี้ประกอบด้วย 3 ส่วน ได้แก่ ชุดข้อมูลที่ใช้ในการทดลอง รายละเอียดการปรับแต่ง และการตั้งค่าต่างๆ ในการทดลอง และผลลัพธ์ของการทดลอง

4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ในการทดลองนี้ใช้ชุดข้อมูลปัญหาการจำแนกประเภทแบบละเอียดที่ใช้กันอย่างแพร่หลาย 3 ชุดข้อมูลได้แก่ CUB200-2011[1], Stanford Cars[2] และ FGVC-Aircraft[3] แสดงตัวอย่างรูปภาพในแต่ละชุดข้อมูลในรูปที่ 24 ซึ่งมีรายละเอียดตามตารางที่ 6 โดยแสดงรายละเอียดของชุดข้อมูล ได้แก่ ชื่อชุดข้อมูล (Datasets) จำนวนประเภทข้อมูล (#Class) จำนวนรูปทั้งหมดในชุดข้อมูล (#Images) โดยในการวัดประสิทธิภาพชุดข้อมูลทั้ง 3 ชุดได้มีการแบ่ง จำนวนรูปทั้งหมดในชุดข้อมูลฝึกสอน (#Train) จำนวนรูปทั้งหมดในชุดข้อมูลทดสอบ (#Test) ไว้เป็นมาตรฐาน ซึ่งในการทดลองจะฝึกสอนแบบจำลองด้วย ชุดข้อมูลฝึกสอน



รูปที่ 24 ตัวอย่างรูปภาพของแต่ละชุดข้อมูล

ตาราง 6 แสดงรายละเอียดของชุดข้อมูลสำหรับทดลอง

ชุดข้อมูล	จำนวนประเภทข้อมูล	จำนวนรูปภาพในชุดข้อมูล	จำนวนรูปภาพสำหรับฝึกสอน	จำนวนรูปภาพสำหรับทดสอบ
CUB200-2011	200	11,877	5,994	5,794
Stanford Cars	100	10,000	6,667	3,333
FGVC-Aircraft	196	16,185	8,144	8,041

4.2 รายละเอียดการตั้งค่าสำหรับการทดลอง

สำหรับการทดลองในงานวิจัยนี้จะใช้ Pytorch ซึ่งเป็นไลบรารี่ในภาษาไพทอน (Python) ที่ออกแบบมาเพื่อใช้เป็นเฟรมเวิร์กของแบบจำลองการเรียนรู้เชิงลึก (Deep Learning Framework) โดยเฉพาะ ซึ่งแบ่งการทดลองออกเป็น 3 การทดลอง ดังนี้

4.2.1 การทดลองเพื่อเปรียบเทียบแบบจำลอง

สำหรับในการทดลองนี้ จะเป็นการทดลองเพื่อทดสอบความแม่นยำของแบบจำลองรูปภาพฝังตัวแบบมีประสิทธิภาพ ซึ่งจะเปรียบเทียบกับงานวิจัยอื่นๆ ในขอบเขตเดียวกันคือปรับปรุงแบบจำลองโดยไม่เพิ่มจำนวนพารามิเตอร์ หรือไม่ใช้ข้อมูลเพิ่มเติมจากคำตอบแบบหมวดหมู่ในการฝึกสอน โดยจะใช้สถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์ค เรสเน็ตห้าสิบ และเรสเน็ตหนึ่งร้อยหนึ่ง (ResNet50 and ResNet101[6]) เป็นฐานของแบบจำลองโดยเปลี่ยนเฉพาะส่วนชั้นการเชื่อมโยงเต็มรูปแบบที่ทำหน้าที่จำแนกประเภท (Classification Layer) เพื่อให้จำนวนประเภทเท่ากับชุดข้อมูลที่จะทำการทดสอบ โดยค่าน้ำหนักของคอนโวลูชันนิวรอลเน็ตเวิร์ค จะใช้จากงานวิจัยของ Jia Deng และคณะ[35] ซึ่งฝึกสอนแบบชุดข้อมูล อิมเมจเน็ต (ImageNet[35]) และจะใช้ค่าน้ำหนักเริ่มต้นแบบสุ่มสำหรับชั้นการเชื่อมโยงที่เพิ่มเข้าไปสำหรับการจำแนกประเภท

ขั้นตอนการฝึกสอนจะมีการปรับขนาดของรูปภาพขาเข้า เป็น 600 x 600 พิกเซล (Pixels) และครอบตัดภาพแบบสุ่ม (Random Cropping) ให้เหลือขนาด 448 x 448 พิกเซล และ การกลับภาพแนวนอน (Horizontal Filp) และการปรับแต่งค่าสี ซึ่งเป็นเป็นเทคนิคการปรับแต่งข้อมูลพื้นฐานเพื่อลดปัญหาการปรับเหมาะเกินไป แต่ในใช้เป็นการครอบตัดภาพจากศูนย์กลาง (Center Cropping) เพียงอย่างเดียวในการทดสอบแทน การฝึกสอนจะทำทั้งหมด 100 รอบโดยใช้ สโตแคสติกเกรเดียนเดสเซนท์ (Stochastic Gradient Descent or SGD) โดยมีค่าน้ำหนักดีเค (Weight Decay) เท่ากับ 10^{-5} และโมเมนตัม (Momentum) เท่ากับ 0.9 ในการหาค่าเหมาะสมที่สุด โดย

อัตราการเรียนรู้จะแยกกันระหว่างคอนโวลูชันนิวรอลเน็ตเวิร์คและชั้นการจำแนกประเภทโดยเริ่มต้นที่ 0.001 และ 0.01 ตามลำดับ ซึ่งจะคูณด้วย 0.1 ในรอบที่ 60 สำหรับคอนโวลูชันนิวรอลเน็ตเวิร์คและคูณด้วย 0.1 ในรอบที่ 50 และ 75 สำหรับชั้นการจำแนกประเภท และใช้ขนาดของ Batch เท่ากับ 16 และฟังก์ชันสำหรับปรับค่ามาจินเชิงมุมแสดงดังสมการที่ (56)

$$f(e) = \begin{cases} 0 & ; e \leq 30 \\ 0.25 & ; 30 < e \leq 60 \\ 0.5 & ; 60 < e \end{cases} \tag{56}$$

4.2.2 การทดลองเพื่อเปรียบเทียบฟังก์ชันสูญเสีย

สำหรับในการทดลองนี้มีวัตถุประสงค์เพื่อเปรียบเทียบเฉพาะประสิทธิภาพของฟังก์ชันสูญเสียมาจินเชิงมุมปรับค่าได้เทียบกับฟังก์ชันสูญเสียอื่นๆ โดยจะใช้สถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คขนาดเล็กโดยไม่ใช้เทคนิคการระบุตำแหน่งช่วยเพื่อทดสอบเพียงแค่ประสิทธิภาพของค่าสูญเสียเท่านั้น โดยใช้วีจีจีสิบหกและ เรสเน็ตสิบแปด (VGG16[5] and ResNet18[6]) เป็นฐานของแบบจำลองโดยเปลี่ยนเฉพาะส่วนชั้นการเชื่อมโยงเต็มรูปแบบที่ทำหน้าที่จำแนกประเภท (Classification Layer) เพื่อให้จำนวนประเภทเท่ากับชุดข้อมูลที่จทำการทดสอบ โดยค่าน้ำหนักของคอนโวลูชันนิวรอลเน็ตเวิร์คและชั้นการจำแนกประเภทจะใช้ค่าน้ำหนักเริ่มต้นแบบสุ่มเนื่องจากจะเป็นการฝึกสอนตั้งแต่ต้น (From Scratch)

ขั้นตอนการฝึกสอนจะมีการปรับขนาดของรูปภาพขาเข้า เป็น 224 x 224 พิกเซล (Pixels) การฝึกสอนจะทำทั้งหมด 300 รอบโดยใช้ สโตแคสติกเกรเดียนเดสเซนท์ (Stochastic Gradient Descent or SGD) โดยมีค่าน้ำหนักดีเค (Weight Decay) เท่ากับ 10^{-5} และโมเมนตัม (Momentum) เท่ากับ 0.9 ในการหาค่าเหมาะสมที่สุด โดยอัตราการเรียนรู้จะแยกกันระหว่างคอนโวลูชันนิวรอลเน็ตเวิร์คและชั้นการจำแนกประเภทโดยเริ่มต้นที่ 0.1 คูณด้วย 0.1 ในรอบที่ 150 และ 225 และใช้ขนาดของ Batch เท่ากับ 32 และฟังก์ชันสำหรับปรับค่ามาจินเชิงมุมแสดงดังสมการที่ (57)

$$f(e) = \begin{cases} 0 & ; e \leq 175 \\ 0.25 & ; 175 < e \leq 250 \\ 0.5 & ; 250 < e \end{cases} \tag{57}$$

4.2.3 การทดลองเพื่อเปรียบเทียบฟังก์ชันปรับค่าสำหรับค่าสูญเสียมาจินปรับค่าได้

เป็นการทดลองเพื่อทดสอบว่าการใช้ฟังก์ชันปรับค่าได้สำหรับค่ามาจินเชิงมุม นั้นมีประสิทธิภาพมากกว่าค่าคงที่ที่งานวิจัยของ Jiankang Deng และคณะ[28] นำเสนอซึ่งจะทำบนชุดข้อมูล CUB200-2011 บนคอนโวลูชันนิวรอลเน็ตเวิร์ค เรสเน็ตห้าสิบ (ResNet50) โดยค่าน้ำหนักของคอนโวลูชันนิวรอลเน็ตเวิร์ค จะใช้จากงานวิจัยของ Jia Deng และคณะซึ่งฝึกสอนแบบชุดข้อมูลอิมเมจเน็ต (ImageNet[35]) และจะใช้ค่าน้ำหนักเริ่มต้นแบบสุ่มสำหรับชั้นการเชื่อมโยงที่เพิ่มเข้าไปสำหรับการจำแนกประเภท โดยฟังก์ชันปรับค่าได้ 3 ประเภทจะเริ่มต้นฝึกสอนที่ค่ามาจินเชิงมุม $m = 0$ จนถึง $m = 0.5$

4.3 ผลการทดลอง

ผลการทดลองทั้ง 3 การทดลองบนชุดขอมูล 3 ชุด ได้แก่ CUB200-2011, Stanford Cars และ FGVC-Aircraft ซึ่งเปรียบเทียบกับงานวิจัยที่ใช้อ้างอิง แสดงอยู่ในตารางที่ 7 ถึงตารางที่ 9 โดยตารางที่ 7 แสดงผลการทดลองแบบจำลองโดยรวมเปรียบเทียบกับงานวิจัยอ้างอิงบนชุดข้อมูลทั้ง 3 ชุด ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%) โดยตัวเลขหนาสีเขียวแสดงถึงความแม่นยำสูงสุดในการเปรียบเทียบด้วยชุดข้อมูลแต่ละชุดข้อมูล และตัวเลขที่ถูกขีดเส้นใต้สีเหลือง แสดงถึงความแม่นยำอันดับสองในการเปรียบเทียบด้วยชุดข้อมูลแต่ละชุดข้อมูล โดยแบบจำลองที่งานวิจัยนี้นำเสนอให้ความแม่นยำสูงสุดบนชุดข้อมูล 2 ชุด คือ CUB200-2011 และ FGVC-Aircraft ที่มีความต่างของแม่นยำที่เอาชนะได้ที่ 0.9% และ 0.5% ตามลำดับ เมื่อเทียบกับงานวิจัยอื่นๆ ซึ่งในชุดข้อมูล Stanford Cars แบบจำลองของเราให้ความแม่นยำเป็นอันดับที่ 2 ซึ่งน้อยกว่างานวิจัยที่ให้ความแม่นยำสูงสุดเพียง 0.1%

ในตารางที่ 8 แสดงผลการทดลองฝึกสอนแบบจำลองด้วยฟังก์ชันค่าสูญเสียเปรียบเทียบกับฟังก์ชันค่าสูญเสียอ้างอิงบนชุดข้อมูลทั้ง 3 ชุด ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%) ซึ่งใช้สถาปัตยกรรมการเรียนรู้เชิงลึก 2 ชนิดคือ วีจีจีและเรสเน็ตถึงแสดงค่าอยู่หน้าและหลัง เครื่องหมายทับ (Slash) ตามลำดับโดยตัวเลขหนาสีเขียวแสดงถึงความแม่นยำสูงสุดในการเปรียบเทียบด้วยชุดข้อมูลแต่ละชุดข้อมูล และตัวเลขที่ถูกขีดเส้นใต้สีเหลือง แสดงถึงความแม่นยำอันดับสองในการเปรียบเทียบด้วยชุดข้อมูลแต่ละชุดข้อมูล โดยค่าสูญเสียมาจินเชิงมุมปรับค่าได้ ให้ความแม่นยำสูงสุดบนชุดข้อมูล 2 ชุดในทั้งสองสถาปัตยกรรม คือ CUB200-2011 และ Stanford Cars

ตาราง 7 ผลการทดลองแบบจำลองโดยรวมเปรียบเทียบกับงานวิจัยอ้างอิงบนชุดข้อมูลทั้ง 3 ชุด ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%)

งานวิจัย	**สถาปัตยกรรมการเรียนรู้เชิงลึก**	**ชุดข้อมูล CUB200-2011(%)**	**ชุดข้อมูล FGVC-Aircraft (%)**	**ชุดข้อมูล Stanford Cars (%)**
FT VGG[18]	VGG19	77.8	84.8	84.9
B-CNN[16]	VGG16	84.1	84.1	91.3
RA-CNN[15]	VGG19	85.3	-	92.5
PC[25]	ResNet50	80.2	83.4	93.4
FT ResNet[18]	ResNet50	84.1	88.5	91.7
DFL-CNN[18]	ResNet50	87.4	91.7	93.1
NTS-Net[19]	ResNet50	87.5	91.4	93.9
MC-Loss[27]	ResNet50	87.3	92.6	93.7
TASN[21]	ResNet50	87.9	-	93.8
MAMC[26]	ResNet50	86.2	-	92.8
MAMC[26]	ResNet101	86.5	-	93.0
CIN[22]	ResNet50	87.5	92.6	94.1
CIN[22]	ResNet101	88.1	92.8	**94.5**
Ours	ResNet50	88.3	92.8	94.0
Ours	ResNet101	**89.0**	**93.3**	94.4

ตาราง 8 ผลการทดลองฝึกสอนแบบจำลองด้วยฟังก์ชันค่าสูญเสียเปรียบเทียบกับฟังก์ชันค่าสูญเสียอ้างอิงบนชุดข้อมูลทั้ง 3 ชุด ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%)

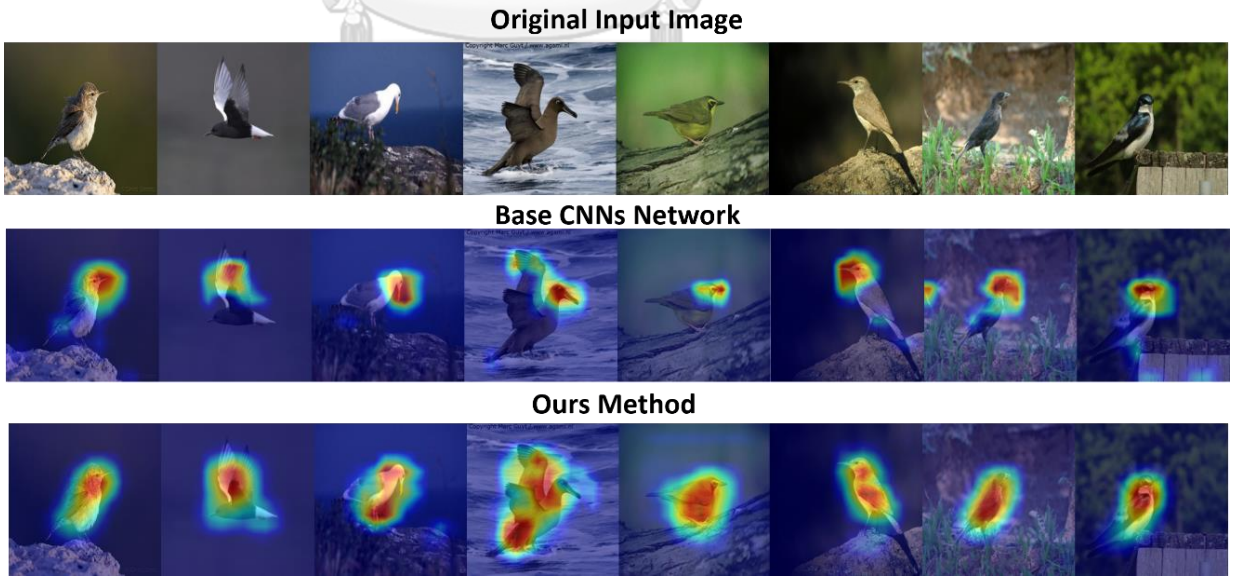
ฟังก์ชันสูญเสีย	**สถาปัตยกรรมการเรียนรู้เชิงลึก**	**ชุดข้อมูล CUB200-2011 (%)**	**ชุดข้อมูล FGVC-Aircraft (%)**	**ชุดข้อมูล Stanford Cars (%)**
Center loss[24]	VGG16 / ResNet18	51.38 / 50.26	88.26 / 83.86	89.27 / 81.84
A-Softmax loss[34]	VGG16 / ResNet18	60.79 / 49.67	88.15 / 82.42	88.71 / 82.15
Focal loss[36]	VGG16 / ResNet18	31.12 / 47.67	80.85 / 80.47	77.02 / 79.75
COCO loss[37]	VGG16 / ResNet18	48.31 / 46.01	86.41 / 80.02	67.27 / 72.38
LGM loss[38]	VGG16 / ResNet18	28.14 / 44.91	87.49 / 80.98	71.27 / 74.37
LMCL[39]	VGG16 / ResNet18	41.11 / 46.01	86.17 / 78.52	49.57 / 71.17
MC loss[27]	VGG16 / ResNet18	65.98 / 59.41	**89.20 / 85.57**	90.85 / 87.47
AAM loss	VGG16 / ResNet18	**71.42 / 68.36**	86.74 / 84.43	**90.91 / 88.51**

ในตารางที่ 9 แสดงผลการทดลองเปรียบเทียบฟังก์ชันปรับค่าของค่ามาจินเชิงมุมบนชุดข้อมูล CUB200-2011ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%) โดยใช้ฟังก์ชัน 3 รูปแบบที่ใช้ในงานวิจัยนี้คือ ฟังก์ชันขั้นบันได (Step Function) ฟังก์ชันเชิงเส้น (Linear Function) และฟังก์ชันเอกซ์โพเนนเชียล (Exponential Function) เปรียบเทียบกับค่ามาจินที่เป็นค่าคงที่ใช้ งานวิจัยของ Jiankang Deng และคณะ[32] ซึ่งตัวเลขหน้าแสดงถึงค่าความแม่นยำสูงสุด ซึ่งมาจินเชิงมุมแบบปรับค่าได้ด้วยฟังก์ชันขั้นบันไดให้ความแม่นสูงสุด

ตาราง 9 ผลการทดลองเปรียบเทียบฟังก์ชันปรับค่าของค่ามาจินเชิงมุมบนชุดข้อมูล CUB200-2011 ด้วยผลความแม่นยำแสดงค่าเป็นเปอร์เซ็นต์ (%)

ฟังก์ชันสูญเสีย	ฟังก์ชันปรับค่าได้ (f€)	ชุดข้อมูล CUB200-2011(%)
ArcFace[28]	Constant m = 0.5	86.8
AAM loss	Step	**87.5**
AAM loss	Linear	87.1
AAM loss	Exponential	87.3

ในรูปที่ 25 แสดงผลการทดลองเพิ่มเติมการนำเสนอแผนภาพความร้อนของเกรเดียนที่ได้มาจากการฝึกสอนแบบจำลองโดยรวม โดยใช้งานวิจัยของ Ramprasaath R. Selvaraju และคณะ[40] ที่ซึ่งแสดงแผนภาพความร้อนแสดงพื้นที่สำคัญที่แบบจำลองพิจารณาในระหว่างการฝึกสอนโดยการหาผลรวมของค่าน้ำหนักของผังฟีเจอร์ ซึ่งทำบนชุดข้อมูล CUB200-2011 จากรูปแสดงให้เห็นว่าเมื่อเปรียบเทียบแบบจำลองของงานวิจัยนี้กับคอนโวลูชันนิวรอลเน็ตเวิร์คแบบดั้งเดิม ซึ่งใช้ เรสเน็ตห้าสิบ (ResNet50) แบบจำลองของเราสามารถพิจารณาครอบคลุมวัตถุภายในรูปได้มากกว่า ซึ่งช่วยให้สามารถจำแนกประเภทได้แม่นยำมากขึ้นเมื่อเปรียบเทียบกัน



รูปที่ 25 ผลการทดลองแผนภาพความร้อนพื้นที่พิจารณาของแบบจำลอง

บทที่ 5

บทสรุปงานวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าววถึงข้อสรุปของแบบจำลองและเทคนิคต่างๆ ที่ใช้ในงานวิจัยนี้รวมถึงผลการทดลองต่างๆ และข้อเสนอแนะเพิ่มเติมสำหรับต่อยอดงานวิจัยนี้

5.1 บทสรุปงานวิจัย

การจำแนกประเภทภาพแบบละเอียด (Fine-Grained Visual Classification) คืองานของการจำแนกประเภทภาพของหมวดหมู่ย่อย (Sub-Category) ซึ่งความแตกต่างระหว่างภาพในแต่ละประเภท (Classes) นั้นมีความละเอียดอ่อนมาก เช่น การจำแนกชนิดของนก ปัญหาหลักของการจำแนกประเภทภาพแบบละเอียดคือ ความเหมือนระหว่างประเภทสูง (Inter-class Similarity) และรายละเอียดของภาพในแต่ละประเภทมีโอกาสผันแปรได้สูง (Intra-class Variation) ทำให้การจำแนกประเภทภาพแบบละเอียดต้องอาศัยเทคนิคการเรียนรู้ที่มีประสิทธิภาพมากกว่าการจำแนกประเภทภาพทั่วไป (General Image Classification) งานวิจัยที่โดยมากเพิ่มความแม่นยำที่ต่อยอดมาจากสถาปัตยกรรมการเรียนรู้เชิงลึกงานวิจัยที่ใช้เทคนิคแบบจำลองการเรียนรู้เชิงลึกที่ใช้กับชุดข้อมูลรูปภาพขนาดใหญ่ (Large Scale dataset) ด้วยรูปแบบงานวิจัยการระบุตำแหน่งและแบ่งประเภทแบบซับเน็ตเวิร์ค (Localization-Classification Sub-Network) เน้นไปที่การออกแบบโครงสร้างแบบจำลอง โดยแบ่งเป็นซับเน็ตเวิร์คสองส่วนเพื่อระบุตำแหน่งของชิ้นส่วนที่สำคัญ ซึ่งจะต่อกับคอนโวลูชันนิวรอลเน็ตเวิร์คเพื่อแบ่งประเภท ซึ่งช่วยให้แบบจำลองเรียนรู้จากตำแหน่งวัตถุที่ถูกต้องและช่วยเพิ่มความแม่นยำ อย่างไรก็ตามการจำแนกซับเน็ตเวิร์คจะเป็นการเพิ่มขนาดของแบบจำลองและจำเป็นต้องใช้ กล่องขอบเขต (Boundary Box) เพื่อใช้ในการฝึกสอนแบบจำลอง และการเข้ารหัสฟีเจอร์แบบเอ็นทูเอ็น (End-to-End Features Encoding) เป็นการเรียนรู้ฟีเจอร์เวกเตอร์โดยตรงจากข้อมูลรูปภาพโดยผ่านแบบจำลองการเรียนรู้เชิงลึก โดยการปรับปรุงสถาปัตยกรรม หรือออกแบบฟังก์ชันสูญเสีย (Loss Function) เพื่อเพิ่มประสิทธิภาพในการเรียนรู้จากชุดข้อมูลรูปภาพที่ความแตกต่างของวัตถุในรูปภาพละเอียดอ่อน

งานวิจัยนี้นำเสนอเทคนิคเพื่อเพิ่มประสิทธิภาพปัญหาการจำแนกประเภทภาพแบบละเอียด และสามารถปรับใช้กับสถาปัตยกรรมคอนโวลูชันนิวรอลเน็ตเวิร์คแบบอื่นๆได้ ซึ่งประกอบด้วยเทคนิคการระบุตำแหน่งโดยไม่ใช้ข้อมูลกล่องขอบเขตเพิ่มเติม และค่าสูญเสียมาจินเชิงมุมปรับค่าได้ที่ต่อยอดและพัฒนามาจากงานวิจัยค่าความสูญเสียที่ช่วยเพิ่มความแม่นยำให้กับงานการจดจำใบหน้า ซึ่งใช้แก้ปัญหาเดียวกันกับการจำแนกประเภทภาพแบบละเอียด โดยนำเทคนิคทั้งสองประยุกต์ใช้กับคอนโวลูชันนิวรอลเน็ตเวิร์ค โดยฝึกสอนแบบ 2 ระดับคือ ระดับของข้อมูลดั้งเดิม และระดับวัตถุ

งานวิจัยนี้ได้ทำการทดลองเพื่อทดสอบประสิทธิภาพของค่าสูญเสียมาจินเชิงมุมปรับค่าได้และแบบจำลองโดยรวม ซึ่งเปรียบเทียบกับงานวิจัยอ้างอิง โดยใช้ชุดข้อมูลที่ใช้อย่างแพร่หลายในงานวิจัยการจำแนกประเภทภาพแบบละเอียด 3 ชุดข้อมูล ได้แก่ CUB200-2011, Stanford Cars และ FGVC-Aircraft จากผลการทดลองสรุปได้ว่า แบบจำลองที่นำเสนอให้ความแม่นยำในการจำแนกสูงกว่างานอ้างอิงสำหรับ 2 ชุดข้อมูล คือ CUB200-2011 และFGVC-Aircraft และให้ความแม่นยำใกล้เคียงกับงานวิจัยอ้างอิง สำหรับชุดข้อมูล Stanford Cars และหากเปรียบเทียบค่าสูญเสียมาจินแบบปรับค่าได้กับค่าสูญเสียอื่นๆ ค่าสูญเสียมาจินแบบปรับค่าได้ ให้ความแม่นยำสูงกว่าโดยรวม

5.2 ข้อเสนอแนะ

1. ในการทดลองของงานวิจัยนี้ใช้พารามิเตอร์ของแบบจำลอง และค่าที่ปรับจูนสำหรับการฝึกสอนเหมือนกันสำหรับข้อมูลทั้ง 3 ชุด ซึ่งสำหรับข้อมูลบางชุดอาจเพิ่มความแม่นยำได้โดย การปรับจูนพารามิเตอร์ให้เหมาะสมสำหรับชุดข้อมูลนั้นๆ

2. ในการทดลองของงานวิจัยนี้ ใช้ฟังก์ชันปรับค่า 3 ประเภท คือ ฟังก์ชันขั้นบันได (Step Function) ฟังก์ชันเชิงเส้น (Linear Function) และฟังก์ชันเอกซ์โพเนนเชียล (Exponential Function) ซึ่งเป็นฟังก์ชันพื้นฐาน โดยแนวคิดการปรับค่ายังมีฟังก์ชันอื่นๆ อีกที่อาจสามารถเพิ่มความแม่นยำในการทำการทดลองได้

บรรณานุกรม

1. Wah, C., et al., *The caltech-ucsd birds-200-2011 dataset.* 2011.
2. Krause, J., et al. *3d object representations for fine-grained categorization*. in *Proceedings of the IEEE international conference on computer vision workshops*. 2013.
3. Maji, S., et al., *Fine-grained visual classification of aircraft.* arXiv preprint arXiv:1306.5151, 2013.
4. Wei, X.-S., J. Wu, and Q. Cui, *Deep learning for fine-grained image analysis: A survey.* arXiv preprint arXiv:1907.03069, 2019.
5. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.
6. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
7. Huang, G., et al. *Densely connected convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
8. Tan, M. and Q. Le. *Efficientnet: Rethinking model scaling for convolutional neural networks*. in *International Conference on Machine Learning*. 2019. PMLR.
9. Zhang, N., et al. *Part-based R-CNNs for fine-grained category detection*. in *European conference on computer vision*. 2014. Springer.
10. Lin, D., et al. *Deep lac: Deep localization, alignment and classification for fine-grained recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
11. Huang, S., et al. *Part-stacked cnn for fine-grained visual categorization*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
12. Wei, X.-S., C.-W. Xie, and J. Wu, *Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition.* arXiv preprint arXiv:1605.06878, 2016.
13. Hanselmann, H. and H. Ney. *ELoPE: Fine-grained visual classification with*

efficient localization, pooling and embedding. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020.

14. Zheng, H., et al. *Learning multi-attention convolutional neural network for fine-grained image recognition*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
15. Fu, J., H. Zheng, and T. Mei. *Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
16. Lin, T.-Y., A. RoyChowdhury, and S. Maji. *Bilinear cnn models for fine-grained visual recognition*. in *Proceedings of the IEEE international conference on computer vision*. 2015.
17. Gao, Y., et al. *Compact bilinear pooling*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
18. Wang, Y., V.I. Morariu, and L.S. Davis. *Learning a discriminative filter bank within a cnn for fine-grained recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
19. Yang, Z., et al. *Learning to navigate for fine-grained classification*. in *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
20. Ding, Y., et al. *Selective sparse sampling for fine-grained image recognition*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.
21. Zheng, H., et al. *Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
22. Gao, Y., et al. *Channel interaction networks for fine-grained image categorization*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020.
23. Luo, W., et al., *Learning semantically enhanced feature for fine-grained image classification.* IEEE Signal Processing Letters, 2020. **27**: p. 1545-1549.
24. Wen, Y., et al. *A discriminative feature learning approach for deep face recognition*. in *European conference on computer vision*. 2016. Springer.

25. Dubey, A., et al. *Pairwise confusion for fine-grained visual classification*. in *Proceedings of the European conference on computer vision (ECCV)*. 2018.
26. Sun, M., et al. *Multi-attention multi-class constraint for fine-grained image recognition*. in *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
27. Chang, D., et al., *The devil is in the channels: Mutual-channel loss for fine-grained image classification.* IEEE Transactions on Image Processing, 2020. **29**: p. 4683-4695.
28. Deng, J., et al. *Arcface: Additive angular margin loss for deep face recognition*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
29. Wei, X.-S., et al., *Selective convolutional descriptor aggregation for fine-grained image retrieval.* IEEE Transactions on Image Processing, 2017. **26**(6): p. 2868-2881.
30. Cui, Y., et al. *Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
31. Krause, J., et al. *The unreasonable effectiveness of noisy data for fine-grained recognition*. in *European Conference on Computer Vision*. 2016. Springer.
32. Reed, S., et al. *Learning deep representations of fine-grained visual descriptions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
33. He, X. and Y. Peng. *Fine-grained image classification via combining vision and language*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
34. Liu, W., et al. *Sphereface: Deep hypersphere embedding for face recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
35. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee.
36. Lin, T.-Y., et al. *Focal loss for dense object detection*. in *Proceedings of the IEEE*

international conference on computer vision. 2017.

37. Liu, Y., H. Li, and X. Wang, *Rethinking feature discrimination and polymerization for large-scale recognition.* arXiv preprint arXiv:1710.00870, 2017.
38. Wan, W., et al. *Rethinking feature distribution for loss functions in image classification*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
39. Wang, H., et al. *Cosface: Large margin cosine loss for deep face recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
40. Selvaraju, R.R., et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. in *Proceedings of the IEEE international conference on computer vision*. 2017.

ประวัติผู้เขียน

ชื่อ-สกุล	นายสรนันท์ พยัตศุภร
วัน เดือน ปี เกิด	16 พฤศจิกายน 2538
สถานที่เกิด	Bangkok, Thailand
วุฒิการศึกษา	B.ENG Mechanical Engineering, Chulalongkorn University
ที่อยู่ปัจจุบัน	Bangkok, Thailand