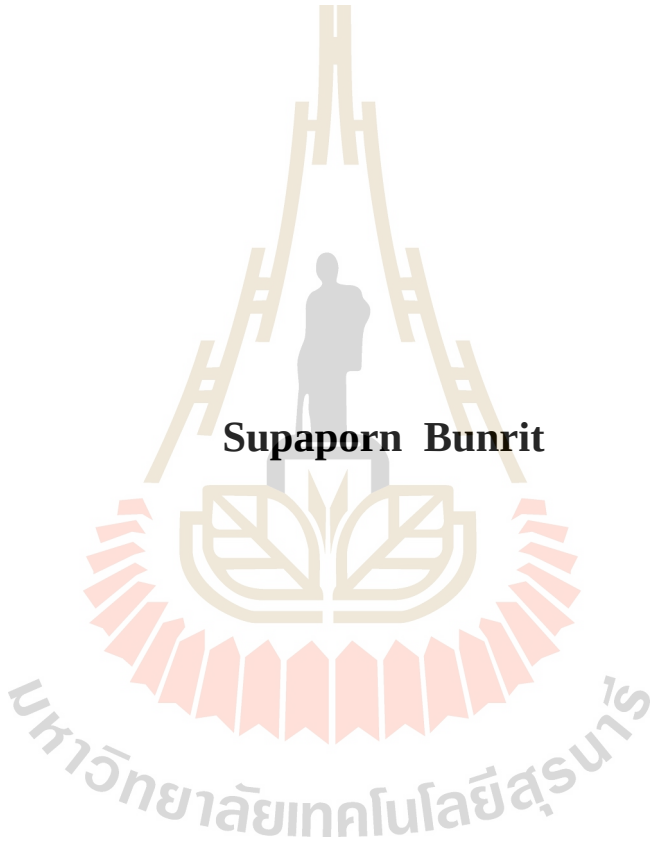
การจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึกของโครงข่ายประสาทแบบคอนโวลูชัน

นางสาวสุภาพร บุญฤทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2562

AUTOMATIC CONSTRUCTION MATERIAL IMAGE CLASSIFICATION WITH DEEP LEARNING TECHNIQUE OF CONVOLUTION NEURAL NETWORK

Supaporn Bunrit



A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Telecommunication and Computer Engineering

Suranaree University of Technology

Academic Year 2019

การจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึกของโครงข่ายประสาทแบบคอนโวลูชัน

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(อ. ดร. กีระชาติ สุขสุทธิ์)
ประธานกรรมการ

(รศ. ดร. กิตติศักดิ์ เกิดประสพ)
อาจารย์ที่ปรึกษา

(รศ. ดร. นิตยา เกิดประสพ)
กรรมการ

(ผศ. ร.อ. ดร. ประโยชน์ คำสวัสดิ์)
กรรมการ

(ผศ. ดร. วิภาวี อุสาหะ)
กรรมการ

(ศ. ดร. สันติ แม้นศิริ)
รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

(รศ. ร.อ. ดร. กนต์ธร ชำนิประศาสน์)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

สุภาพร บุญฤทธิ์ : การจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึกของโครงข่ายประสาทแบบคอนโวลูชัน (AUTOMATIC CONSTRUCTION MATERIAL IMAGE CLASSIFICATION WITH DEEP LEARNING TECHNIQUE OF CONVOLUTION NEURAL NETWORK) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. กิตติศักดิ์ เกิดประสพ, 189 หน้า.

ระบบบริหารจัดการงานก่อสร้างสมัยใหม่นั้นประกอบด้วยหลายส่วนงานย่อยที่จำเป็นต้องใช้กระบวนการทำงานแบบอัตโนมัติหรือแบบกึ่งอัตโนมัติ โดยเฉพาะอย่างยิ่งงานในส่วนของการตรวจติดตามความคืบหน้าการก่อสร้างที่จำเป็นต้องมีขั้นตอนเริ่มต้นเกี่ยวกับการจำแนกความแตกต่างของแต่ละวัสดุที่ใช้จากข้อมูลภาพ ยิ่งผลลัพธ์ที่ได้จากการจำแนกมีสูงก็จะยิ่งประเมินการใช้งานในแต่ละวัสดุได้ถูกต้องมากยิ่งขึ้น ส่งผลให้การประเมินความคืบหน้าของงานมีความน่าเชื่อถือ วิธีการที่มีอยู่เกือบทั้งหมดทำการศึกษาบนพื้นฐานของการนำคุณลักษณะที่ออกแบบด้วยมือ (Hand-Designed Feature) มาใช้ ซึ่งความถูกต้องจากการจำแนกยังไม่น่าพอใจ งานวิจัยนี้จึงนำเสนอวิธีการสำหรับการสกัดคุณลักษณะแบบอัตโนมัติด้วยเทคนิคที่โดดเด่นทางด้านการเรียนรู้เชิงลึกนั่นคือ โครงข่ายประสาทแบบคอนโวลูชัน (CNN) โดยเป็นการนำโมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วมาใช้ในแนวคิดของการเรียนรู้แบบถ่ายโอน (Transfer Learning) ในแบบยึดคุณลักษณะจากตัวสกัด (Fixed Feature Extractor) จากนั้นเครื่องเข้ารหัสอัตโนมัติจึงนำมาใช้สำหรับการเข้ารหัสให้กับคุณลักษณะที่สกัดมาได้ และใช้เครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มสำหรับการจำแนก งานวิจัยนี้ยังนำเสนอการศึกษาวิธีการในรูปแบบอื่น ๆ เพื่อเป็นการศึกษาเปรียบเทียบ โดยนำสถาปัตยกรรมที่ผ่านการฝึกสอนมาก่อนของโมเดล AlexNet และ GoogleNet มาศึกษาเปรียบเทียบกับโมเดล ResNet101 นอกจากนี้ได้นำวิธีการของ PCA (Principal Component Analysis) มาศึกษาร่วมด้วยเพื่อเป็นการเปรียบเทียบกับเครื่องเข้ารหัสอัตโนมัติ จากผลการทดลองพบว่า วิธีการที่นำเสนอด้วยการใช้เครื่องเข้ารหัสอัตโนมัติจะสามารถช่วยเพิ่มประสิทธิภาพในการจำแนกได้ดีกว่าการใช้ PCA ในทุกกรณีที่ศึกษา โดยเฉพาะอย่างยิ่งเมื่อคุณลักษณะที่สกัดมาจากโมเดล ResNet101 ถูกใช้เป็นอินพุตของเครื่องเข้ารหัสอัตโนมัตินั้นผลลัพธ์ที่ได้จะดีที่สุด นั่นคือมี Accuracy 97.8% สำหรับชุดข้อมูลที่ 1 และมี Accuracy 98.0% สำหรับชุดข้อมูลที่ 2 เป็นการบ่งบอกว่าการนำเครื่องเข้ารหัสอัตโนมัติมาใช้ร่วมกับคุณลักษณะที่สกัดมาจากโมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนนั้นมีประสิทธิภาพมากกว่าการฝึกสอนโดยตรงหรือการปรับแต่งการเรียนรู้ให้กับโครงข่ายที่ซับซ้อนของ CNN

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา สุภาพร บุญฤทธิ์

ปีการศึกษา 2562 ลายมือชื่ออาจารย์ที่ปรึกษา

SUPAPORN BUNRIT : AUTOMATIC CONSTRUCTION MATERIAL IMAGE CLASSIFICATION WITH DEEP LEARNING TECHNIQUE OF CONVOLUTION NEURAL NETWORK. THESIS ADVISOR : ASSOC. PROF. KITTISAK KERDPRASOP, Ph.D., 189 PP.

CONVOLUTION NEURAL NETWORK (CNN)/TRANSFER LEARNING/AUTOENCODER/CONSTRUCTION MATERIAL IMAGE CLASSIFICATION.

Various sub-tasks on modern construction management system require automatic or semi-automatic processes in handling the operation inside. Especially for construction progress monitoring task, the automatic process in classifying the difference of each construction material from an image is necessary in the preliminary stage. The more the preciseness in automatic classifying, the more the exactness in assessment of each material had been used. Subsequently, the progress of the construction can be evaluated with the highest degree of reliability. Almost all existing related works have been studied based on hand-designed features of which the classified accuracy still not much appreciated. In this research, automatic feature extracted method from the prominent technique in deep learning, convolution neural network (CNN), is studied. The pre-trained of ResNet101 model is adopted in the concept of transfer learning in the scheme of fixed feature extractor. Data representation learning based on autoencoder model is then employed to encode such the CNN extracted feature. Finally, multi-class support vector machine (SVM) is used for the classification stage. This research also studied other diversified methods in applying CNN models. The pre-trained architectures of AlexNet and GoogleNet are

also explored compare to ResNet101 model. Whereas, principal component analysis (PCA) is investigated to be compared to autoencoder. Experimental results reveal that with the autoencoder-based method employed in the proposed work, the classification performance can improve more than the performance obtained from PCA in all cases. Especially, when the fixed feature extractor of ResNet101 is used as the input to an autoencoder, the classified result outperforms the others with an accuracy of 97.8% for dataset 1 and 98.0% for dataset 2. It can be inferred that by applying autoencoder on top of the pre-trained transferred features from a suitable transferred model, the performance can be improved without the need to train or fine-tune the complex CNN model.

School of Computer Engineering Student's Signature _____

Academic Year 2019 Advisor's Signature _____

กิตติกรรมประกาศ

การที่วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยได้รับความกรุณาเกี่ยวกับคำปรึกษา คำแนะนำ กำลังใจ การสนับสนุน และความช่วยเหลืออย่างดียิ่ง ทั้งทางด้านวิชาการและด้านการดำเนินงานวิจัยจากบุคคลและกลุ่มบุคคล ผู้วิจัยขอกราบขอบพระคุณบุคคลต่อไปนี้เป็นอย่างสูง ที่มีส่วนช่วยให้วิทยานิพนธ์นี้ลุล่วงด้วยดี

รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์ และอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้โอกาสและเมตตาให้การอบรม สั่งสอน ชี้แนะ ให้กำลังใจและช่วยเหลือในการทำการศึกษาวิจัย รวมถึงการให้คำแนะนำเกี่ยวกับรูปแบบในการนำเสนอผลการศึกษา และ การวิเคราะห์ผลการศึกษา

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ และกรรมการสอบวิทยานิพนธ์ที่ให้โอกาสและเมตตาสั่งสอน ชี้แนะ ให้กำลังใจและช่วยเหลือในการทำวิจัยรวมถึงการให้คำแนะนำในการเขียนและตรวจแก้ไขวิทยานิพนธ์จนเสร็จสมบูรณ์

รองศาสตราจารย์ เรืออากาศเอก ดร. กนต์ธร ชำนิประศาสน์ คณบดีสำนักวิชาวิศวกรรมศาสตร์ที่ให้โอกาสสำหรับการลาศึกษาต่อ

มหาวิทยาลัยเทคโนโลยีสุรนารีที่ได้เปิดโอกาสให้สำหรับการลาศึกษาต่อ รวมถึงการให้การสนับสนุนทุนการศึกษา และทุนอุดหนุนการนำเสนอและเผยแพร่ผลงานวิจัย

ผู้ช่วยศาสตราจารย์ ดร. วิภาวี อุสาหะ และผู้ช่วยศาสตราจารย์ เรืออากาศเอก ดร. ประโยชน์ คำสวัสดิ์ กรรมการที่ให้การแนะนำและชี้แนะให้วิทยานิพนธ์ฉบับนี้สมบูรณ์มากยิ่งขึ้น รวมถึงอาจารย์ ดร. กีระชาติ สุขสุทธิ์ ผู้ทรงคุณวุฒิที่เสียสละเวลาทำหน้าที่กรรมการสอบ

ผู้ช่วยศาสตราจารย์ ดร.ธีรวัฒน์ สินศิริ หัวหน้าโครงการวิจัยวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า ที่ให้คำแนะนำ อำนวยความสะดวกและให้การอนุเคราะห์เกี่ยวกับสถานที่ถ่ายภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าเพื่อใช้เป็นส่วนหนึ่งของชุดข้อมูลภาพในงานวิจัยนี้

เพื่อนบัณฑิตศึกษา สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ทุกคนที่คอยช่วยเหลือกันในการดำเนินงานวิจัย การให้คำปรึกษา รวมถึงกำลังใจ

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา ที่ให้การอุปการะอบรมเลี้ยงดู ส่งเสริมการศึกษา รวมทั้งครอบครัวที่ให้กำลังใจเป็นอย่างดีเสมอมา จนกระทั่งวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

สุภาพร บุญฤทธิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

การบริหารจัดการและการควบคุมโครงการก่อสร้างสมัยใหม่ให้สามารถดำเนินโครงการได้ประสบความสำเร็จนั้นจำเป็นอย่างยิ่งที่ต้องมีการวางแผนและการจัดการเป็นอย่างดีเพื่อให้โครงการเป็นไปตามแผนงานที่วางไว้ เพราะโครงการก่อสร้างทุก ๆ โครงการมีข้อจำกัดในหลาย ๆ ด้าน ไม่ว่าจะเป็นด้านระยะเวลาดำเนินการ งบประมาณที่จะใช้ รวมถึงคุณภาพของผลงานที่ต้องการ ดังนั้นทีมที่ทำหน้าที่บริหารโครงการจะต้องมีการติดตาม ตรวจสอบ และประเมินโครงการอยู่เสมอ เพื่อให้ผู้เกี่ยวข้องหลักของโครงการได้รับข้อมูลเกี่ยวกับความคืบหน้าของงาน หรือปัญหาที่เกิดขึ้นอย่างรวดเร็วที่สุด เพื่อจะได้แก้ไขปัญหาได้ทันท่วงที (วิสูตร จิระดำเกิง, 2554) การตรวจติดตามความคืบหน้างานก่อสร้าง (Construction Progress Monitoring) จึงจัดว่าเป็นขั้นตอนที่สำคัญมากในการบริหารงานก่อสร้าง (Construction Management) เพราะการที่สามารถทราบได้อย่างรวดเร็วและทันเวลาเกี่ยวกับสถานะ (Status) ต่าง ๆ ของโครงการนั้น นอกจากจะสามารถประเมินงานในส่วนที่แล้วเสร็จและยังไม่เสร็จได้อย่างถูกต้องแม่นยำแล้ว ยังสามารถจัดสรรการใช้ทรัพยากรเกี่ยวกับกำลังแรงงาน (Workforce) อุปกรณ์ (Equipment) หรือวัสดุ (Material) ต่าง ๆ ได้อย่างเหมาะสมอีกด้วย (Teizer, 2015) ซึ่งในการบริหารจัดการเพื่อให้โครงการเป็นไปตามกำหนดเวลาที่วางแผนไว้นั้นจำเป็นต้องให้ความสำคัญในการจัดการแต่ละส่วนของงานย่อย เช่น การจัดหาวัสดุต่าง ๆ ที่ต้องใช้ การควบคุมสถานะในการทำงาน และการตรวจติดตามในเรื่องของความปลอดภัยและคุณภาพ

การตรวจติดตามความคืบหน้างานก่อสร้างจึงถือเป็นหัวใจของระบบควบคุมโครงการ (Project Control System) เพื่อจัดการกับปัญหาที่เกิดขึ้นในระหว่างการดำเนินโครงการ เพื่อให้โครงการสามารถดำเนินการไปได้ตามแผนที่วางไว้ ทั้งด้านคุณภาพ เวลาและต้นทุน โดยกระบวนการพื้นฐานของระบบควบคุมโครงการ ได้แก่ การกำหนดแผนงานฐาน การวัดความคืบหน้าของงานก่อสร้างที่ทำได้ขณะดำเนินโครงการแบบเป็นทางการ หรือไม่เป็นทางการ การประเมินผลงานที่ทำได้เทียบกับแผนงานฐาน เพื่อดูว่ามีการเบี่ยงเบนจากแผนงานฐานหรือไม่ และการแก้ไขในกรณีที่การประเมินพบว่ามีสิ่งที่ต้องปรับปรุงแก้ไข เพื่อให้การดำเนินโครงการกลับมาอยู่ในแผนงานฐานที่วางไว้ (วิสูตร จิระดำเกิง, 2554) ดังนั้นความรวดเร็วและทันเวลาในการวัดประเมินและแก้ไขเป็นสิ่งที่สำคัญมากในการควบคุมโครงการ ทั้งนี้เพราะโครงการเป็น

กระบวนการที่ดำเนินการอย่างต่อเนื่อง ซึ่งความล่าช้าเพียงเล็กน้อย อาจก่อให้เกิดผลเสียหายตามมาอีกมากมายก็ได้

ปัจจุบันนี้งานในส่วนของการตรวจติดตามความคืบหน้าของโครงการ รวมถึงการวัดประเมินเปอร์เซ็นต์ของงานที่แล้วเสร็จ ยังใช้การประเมินด้วยมือ (Manual Assessment) ถึงแม้ว่าจะเป็นโครงการขนาดใหญ่ก็เช่นเดียวกัน นั่นคือต้องใช้กำลังคนเป็นหลักในการตรวจติดตามและประเมินความคืบหน้าของโครงการ ซึ่งโอกาสจะเกิดความคลาดเคลื่อนมีสูงและอาจไม่ทันต่อเวลา (Teizer, 2015) นอกจากนี้จากงานวิจัยของ Mccullouch (1997) พบว่า จากวิธีการตรวจติดตามและควบคุมงานแบบทำด้วยมือนั้น ผู้จัดการโครงการต้องใช้เวลาโดยเฉลี่ย 30-50% ของเวลาทั้งหมดไปกับการบันทึกและวิเคราะห์ข้อมูลเกี่ยวกับสถานที่ก่อสร้าง (Site Data) ซึ่งเป็นการเบียดเบียนเวลาไปจากงานอื่นที่สำคัญกว่าเป็นอย่างมาก ปัจจุบันจึงมีความพยายามอย่างกว้างขวางจากกลุ่มงานวิจัยในสถาบันการศึกษารวมถึงกลุ่มธุรกิจที่เกี่ยวข้องกับโครงการก่อสร้างในการหาวิธีแบบกึ่งอัตโนมัติ (Semi-Automatic) หรือแบบอัตโนมัติ (Automatic) ในการตรวจติดตามและประเมินความคืบหน้าของโครงการก่อสร้าง

จากการที่เทคโนโลยีเกี่ยวกับการรวบรวมข้อมูล (Data Acquisition Technologies) รวมถึงเทคโนโลยีเกี่ยวกับคอมพิวเตอร์และอินเทอร์เน็ตมีการพัฒนาไปมากในปัจจุบัน ในวงการงานก่อสร้างจึงได้รับประโยชน์จากการนำเทคโนโลยีดังกล่าวมาใช้ในแต่ละขั้นตอนของการบริหารจัดการโครงการก่อสร้าง ตัวอย่างของเทคโนโลยีเกี่ยวกับการรวบรวมข้อมูลที่นำมาใช้จากการสำรวจในงานของ Chen และคณะ (2015) คือ เซนเซอร์ (Sensor) จีพีเอส (GPS, Global Positioning System) จีไอเอส (GIS, Geographic Information System) เออาร์ (AR, Augmented Reality) กล้องถ่ายรูป (Camera) อาร์เอฟไอดี (RFID, Radio Frequency Identification) และ เลเซอร์สแกนนิง (Laser Scanning) จากการสำรวจในงานประยุกต์เกี่ยวกับ BBB (Application of Bridging BIM and Building, BIM : Building Information Modeling) พบว่า เลเซอร์สแกนนิ่ง มีการนำมาใช้ในงานประยุกต์ต่าง ๆ มากที่สุด รองลงมาคือ อาร์เอฟไอดี และ กล้องถ่ายรูป ตามลำดับ ซึ่งอาร์เอฟไอดีเหมาะที่จะใช้กับวัสดุที่ทำจากเหล็ก (Steel) หรือวัสดุที่เป็นส่วนประกอบสำเร็จรูป (Prefabricated Component) ในขณะที่กล้องถ่ายรูปจัดว่าเป็นเทคโนโลยีที่ใช้ต้นทุนต่ำและใช้เวลาในการคำนวณน้อยกว่ามากเมื่อเทียบกับเลเซอร์สแกนนิง นอกจากนี้ กล้องถ่ายรูปยังเป็นเทคโนโลยีที่สามารถนำไปใช้ได้โดยทั่วไป ที่อาจอยู่ในรูปแบบของ กล้องวงจรปิด สมาร์ทโฟน หรือกล้องที่ติดตั้งกับยานพาหนะไร้คนขับ (Unmanned Aerial Vehicles) เช่นโดรน ซึ่งข้อมูลที่จัดเก็บมาได้จากกล้องถ่ายรูปอาจอยู่ในรูปของภาพ (Image) หรือภาพเคลื่อนไหว (Video)

สืบเนื่องจากความก้าวหน้าทางเทคโนโลยีของกล้องถ่ายรูปแบบดิจิทัลประกอบกับความสามารถในการประมวลผลของคอมพิวเตอร์ ทำให้เกิดความต้องการอย่างกว้างขวางที่จะพัฒนากระบวนการที่มีประสิทธิภาพในการสกัดข้อมูลออกมาจากภาพและวิดีโอที่จัดเก็บมา ปัจจุบันนี้เทคนิคต่าง ๆ เกี่ยวกับการประมวลผลภาพ (Image Processing) และเทคนิคการมองเห็นของครื่อง (Computer Vision) จึงเป็นทิศทางงานวิจัยที่เติบโตขึ้นเป็นอย่างมากในอุตสาหกรรมเกี่ยวกับ สถาปัตยกรรม วิศวกรรม การก่อสร้าง และ การจัดการสิ่งอำนวยความสะดวก (Architecture, Engineering, Construction, and Facilities Management : AEC/FM) (Rashidi et al., 2016) โดยเฉพาะอย่างยิ่งในการบริหารจัดการงานก่อสร้างนั้น เทคนิคเกี่ยวกับการประมวลผลภาพและเทคนิคการมองเห็นของเครื่องสามารถนำไปประยุกต์ใช้ในหลาย ๆ ส่วนของงาน

ในส่วนของงานเกี่ยวกับการตรวจติดตามความคืบหน้าของโครงการก่อสร้างแบบอัตโนมัติหรือแบบกึ่งอัตโนมัตินั้นจำเป็นต้องไปมีการวัดประเมินส่วนของงานที่เกี่ยวข้องกับการนำวัสดุต่าง ๆ สำหรับงานก่อสร้างไปใช้ ดังนั้นขั้นตอนแบบอัตโนมัติสำหรับการจำแนกความแตกต่างระหว่างแต่ละวัสดุจึงเป็นขั้นตอนเริ่มต้นที่สำคัญมากที่จำเป็นต้องมี ซึ่งการรวบรวมข้อมูลที่เกี่ยวกับรายละเอียดของวัสดุจากโครงการที่กำลังก่อสร้างนั้นจำเป็นต้องเป็นข้อมูลที่มาจากภาพนิ่งหรือภาพเคลื่อนไหวเท่านั้น เพราะเทคโนโลยีการรวบรวมข้อมูลแบบเลเซอร์สแกนนิงไม่สามารถให้รายละเอียดในเรื่องความแตกต่างของแต่ละวัสดุได้ (Dimitrov and Golparvar-Fard, 2014) ส่วนเทคโนโลยีอาร์เอฟไอดีนั้นก็ยังมีข้อจำกัดมาก นั่นคือสามารถใช้ได้เฉพาะกับวัสดุบางประเภทเท่านั้น รวมถึงจำเป็นต้องมีการติดตั้งตำแหน่งของแท็ก (Tag) บนวัสดุแต่ละชิ้นและต้องมีการดูแลรักษาแท็กดังกล่าว ที่ต้องมีการลงทุนเพิ่มและยุ่งยากสำหรับสภาพแวดล้อมของงานก่อสร้างที่มีการเปลี่ยนแปลงไปในทุก ๆ วัน (Kopsida et al., 2015)

ในงานวิจัยนี้จึงมีความสนใจในการนำข้อมูลจากภาพดิจิทัล (Digital Image) มาใช้สำหรับเป็นส่วนหนึ่งของงานการตรวจติดตามความคืบหน้าของโครงการก่อสร้างแบบอัตโนมัติ นั่นคือเป็นการนำเสนอวิธีการแบบอัตโนมัติสำหรับการจำแนกวัสดุในงานก่อสร้าง (Construction Materials) จากภาพดิจิทัล ซึ่งจากการสำรวจพบว่างานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้างในอดีตที่ผ่านมาไม่ได้มีการนำเสนอไว้มากนักเมื่อเทียบกับการจำแนกภาพในงานประยุกต์อื่น แต่ปัจจุบันการจำแนกภาพวัสดุในงานก่อสร้างกำลังเป็นที่สนใจและมีการนำเสนอแนวคิดที่หลากหลายมากขึ้นจากการที่มีเทคโนโลยีที่ทันสมัยมากยิ่งขึ้นมารองรับ รวมถึงมีความพยายามที่จะหาวิธีการแบบอัตโนมัติหรือกึ่งอัตโนมัติมาช่วยในอุตสาหกรรมการก่อสร้างที่เกี่ยวกับการบริหารจัดการ โดยเฉพาะเทคโนโลยีที่เกี่ยวข้องกับการพัฒนาระบบ BIM (Building Information Modeling) ที่กำลังมีความพยายามอย่างกว้างขวางในอุตสาหกรรมงานก่อสร้างที่ต้องการพัฒนาและ

ผลักดันให้มีการนำระบบ BIM มาใช้เพื่อความสะดวกในการบริหารจัดการงานก่อสร้าง ซึ่งงานในส่วนของการตรวจติดตามความคืบหน้างานก่อสร้างแบบอัตโนมัตินั้นจัดเป็นกระบวนการย่อยหนึ่งที่จำเป็นต้องมีการพัฒนาในระบบของ BIM ด้วย เพื่อการพัฒนาให้กระบวนการดังกล่าวเป็นแบบอัตโนมัติได้นั้น จำเป็นอย่างยิ่งที่ต้องมีส่วนของงานย่อยหลักที่เกี่ยวกับการจำแนกวัสดุในงานก่อสร้างแบบอัตโนมัติ เพื่อจะได้ประเมินงานส่วนที่แล้วเสร็จและส่วนที่ยังไม่เสร็จได้อย่างถูกต้องแม่นยำและลดเวลาให้ได้มากที่สุด

วิธีการโดยส่วนใหญ่ที่มีการนำเสนอไว้ในงานวิจัยที่ผ่านมาสำหรับการจำแนกภาพวัสดุในงานก่อสร้างจะเป็นวิธีการที่อยู่บนพื้นฐานของการหาคุณลักษณะ (Feature Based Methods) ที่เป็นการนำเทคนิคทางการประมวลผลภาพ เทคนิคการมองเห็นของเครื่อง หรือเทคนิคอื่น ๆ มาช่วยสำหรับการสกัดคุณลักษณะ (Features) สำคัญบางอย่างออกมาจากภาพ แล้วนำคุณลักษณะเหล่านั้นมาใช้สำหรับการจำแนกความแตกต่างของวัสดุแต่ละชนิด โดยอาจใช้เทคนิคการเปรียบเทียบความแตกต่างหรือเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) มาใช้ในขั้นตอนการจำแนก โดยวิธีการที่เคยนำเสนอไว้ในงานวิจัยส่วนใหญ่ แทบทั้งหมดใช้ข้อมูลภาพจากฐานข้อมูลที่สร้างขึ้นเองสำหรับงานวิจัยนั้น ๆ และไม่มีการเผยแพร่ข้อมูลภาพดังกล่าว ถึงแม้ว่าในบางงานวิจัย เช่นงานวิจัยของ DeGol และคณะ (2016) จะมีการสร้างฐานข้อมูลที่ค่อนข้างหลากหลายเกี่ยวกับภาพวัสดุในงานก่อสร้างและมีการนำฐานข้อมูลดังกล่าวมาเผยแพร่ แต่ผลลัพธ์ที่ได้จากการจำแนกในงานวิจัยดังกล่าวก็ยังถือว่าไม่สูง นอกจากนี้เทคนิคใหม่ ๆ ทางด้านการเรียนรู้ของเครื่องที่กำลังเป็นที่สนใจอย่างกว้างขวาง เช่น เทคนิคการเรียนรู้เชิงลึก (Deep Learning Technique) ก็ยังไม่ได้มีนำเสนอเพื่อศึกษาอย่างครอบคลุมสำหรับการประยุกต์ใช้เพื่อจำแนกภาพวัสดุในงานก่อสร้าง ดังนั้นงานวิจัยนี้จึงต้องการนำเสนอการศึกษาเพื่อนำเทคนิคการเรียนรู้เชิงลึกที่เหมาะสมกับข้อมูลรูปภาพมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ นั่นคือคือโครงข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Network, CNN) ซึ่งเป็นสถาปัตยกรรมของโครงข่ายที่เป็นความก้าวหน้าใหม่ทางด้านการเรียนรู้ของเครื่อง ที่กำลังเป็นที่สนใจในการนำไปประยุกต์ใช้ในงานต่าง ๆ อย่างกว้างขวาง โดยเฉพาะงานที่เกี่ยวข้องกับข้อมูลภาพ แต่จากการสำรวจงานวิจัยที่เกี่ยวข้องพบว่า โครงข่ายประสาทแบบคอนโวลูชันยังไม่มีการนำมาประยุกต์ใช้โดยตรงสำหรับการจำแนกข้อมูลภาพเกี่ยวกับวัสดุในงานก่อสร้าง ดังนั้นงานวิจัยนี้จึงมุ่งศึกษาในส่วนของวิธีการและแนวคิดต่าง ๆ ในการประยุกต์ใช้ดังกล่าว เพื่อต้องการนำเสนอวิธีการหรือแนวคิดที่มีประสิทธิภาพสำหรับการนำเทคนิคการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ

1.2 วัตถุประสงค์การวิจัย

1. เพื่อนำเสนอการนำเทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ

2. เพื่อนำเสนอการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อน คือโมเดล ResNet101 (ResNet101 Pre-Trained Model) มาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติในลักษณะของการเรียนรู้แบบถ่ายโอน (Transfer Learning) แบบยึดคุณลักษณะจากตัวสกัด (Fixed Feature Extractor)

3. เพื่อนำคุณลักษณะ (Features) ที่ได้จากการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสแบบอัตโนมัติ (Autoencoder Network) และเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม (Multi-Class SVM) สำหรับการเพิ่มประสิทธิภาพในการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติจากชุดข้อมูลที่ศึกษา

4. เพื่อศึกษาเปรียบเทียบวิธีการที่นำเสนอในข้อ 2 และข้อ 3 กับวิธีการในรูปแบบอื่น ๆ สำหรับการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ

1.3 ขอบเขตของการวิจัย

ในการพัฒนางานวิจัยเพื่อนำเทคนิคการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ ในส่วนของการดำเนินงานกำหนดขอบเขตการวิจัยไว้ดังต่อไปนี้

1. ชุดข้อมูลที่ศึกษา ใช้ข้อมูลภาพวัสดุในงานก่อสร้างที่ประกอบด้วยข้อมูลสี่ชนิด (4 Classes) ของวัสดุในงานก่อสร้าง โดยสามชนิดของภาพวัสดุในงานก่อสร้างมาจากข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะ (2016) และอีกหนึ่งชนิดของวัสดุเป็นข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า (Lightweight Brick Cellular Concrete) ที่สร้างขึ้นเองสำหรับงานวิจัยนี้ ซึ่งได้มาจากการถ่ายภาพจริงของวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าที่เป็นผลงานวิจัยของ ผศ. ดร. ธีรวัฒน์ สินศิริ สาขาวิชาวิศวกรรมโยธา มหาวิทยาลัยเทคโนโลยีสุรนารี

2. การศึกษาเปรียบเทียบประสิทธิภาพของวิธีการที่นำเสนอกับวิธีการในรูปแบบอื่น ๆ จะทำการเปรียบเทียบกันด้วยผลลัพธ์ที่ได้จากการจำแนกกับเฉพาะชุดข้อมูลที่เผยแพร่ในงานวิจัยของ DeGol และคณะ ที่ประกอบด้วยข้อมูลสามชนิด (3 Classes) ของวัสดุในงานก่อสร้าง โดยทำการเปรียบเทียบในสามรูปแบบคือ

1) เปรียบเทียบกับผลลัพธ์ที่ได้จากการใช้สถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนด้วยโมเดลอื่น ซึ่งคือโมเดล AlexNet และโมเดล GoogleNet ที่เป็นการประยุกต์ใช้ในแบบการเรียนรู้แบบถ่ายโอน
2) เปรียบเทียบกับผลลัพธ์ที่ได้จากการใช้เทคนิคการเข้ารหัสข้อมูลด้วยวิธีการอื่น ซึ่งคือ Principal Component Analysis (PCA)
3) เปรียบเทียบกับผลลัพธ์ที่ได้จากวิธีการสร้างสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันสำหรับการฝึกสอนด้วยชุดข้อมูลที่ศึกษาขึ้นมาเอง (Learning from Scratch)

3. การศึกษาเปรียบเทียบประสิทธิภาพของวิธีการที่นำเสนอกับวิธีการในรูปแบบอื่น ๆ จะเปรียบเทียบด้วยมาตรวัดที่ประกอบด้วย ค่า Accuracy, Precision, Recall และ F-Measure

1.4 ประโยชน์ที่ได้รับ

1. ได้วิธีการแบบอัตโนมัติสำหรับการจำแนกภาพวัสดุในงานก่อสร้างที่มีประสิทธิภาพ
2. ได้วิธีการที่มีประสิทธิภาพในการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันสำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ
3. ได้โมเดลการเข้ารหัสอัตโนมัติที่เหมาะสมสำหรับสร้างตัวแทนข้อมูลจากคุณลักษณะที่ได้จากการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันกับภาพวัสดุในงานก่อสร้าง
4. ได้โมเดลการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มที่เหมาะสมสำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ
5. ได้ทราบถึงแนวทางที่หลากหลายในการนำคุณลักษณะที่ได้จากการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันมาใช้ร่วมกับเทคนิคอื่น ๆ ทางด้านการเรียนรู้ด้วยเครื่องเพื่อให้ได้วิธีการสำหรับการจำแนกข้อมูลภาพที่มีประสิทธิภาพ
6. ผลลัพธ์ที่ได้จากการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติที่มีประสิทธิภาพสามารถนำไปประยุกต์ใช้ในเชิงปฏิบัติในส่วนงานต่าง ๆ ของอุตสาหกรรมการก่อสร้างที่เกี่ยวข้องกับการนำวัสดุสำหรับงานก่อสร้างไปใช้ เช่นในส่วนของงานการตรวจติดตามความคืบหน้าของโครงการก่อสร้างแบบอัตโนมัติ

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้เป็นการกล่าวถึง การศึกษาในหลักการ ทฤษฎี และงานวิจัยที่เกี่ยวข้องสำหรับนำมาประยุกต์ใช้เพื่อพัฒนางานวิจัย ซึ่งประกอบด้วยเนื้อหาเกี่ยวกับ พื้นฐานโครงข่ายประสาทเทียม โครงข่ายเพอร์เซปตรอนแบบหลายชั้น การเรียนรู้เชิงลึก การเรียนรู้เชิงลึกของโครงข่ายประสาทแบบคอนโวลูชัน การประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในทางปฏิบัติ งานประยุกต์ที่ใช้โครงข่ายประสาทแบบคอนโวลูชัน Frameworks ของโครงข่ายประสาทแบบคอนโวลูชัน โครงข่ายเครื่องเข้ารหัสอัตโนมัติ การวิเคราะองค์ประกอบหลัก เครื่องเวกเตอร์เกื้อหนุนและงานวิจัยที่เกี่ยวข้อง ตามลำดับ

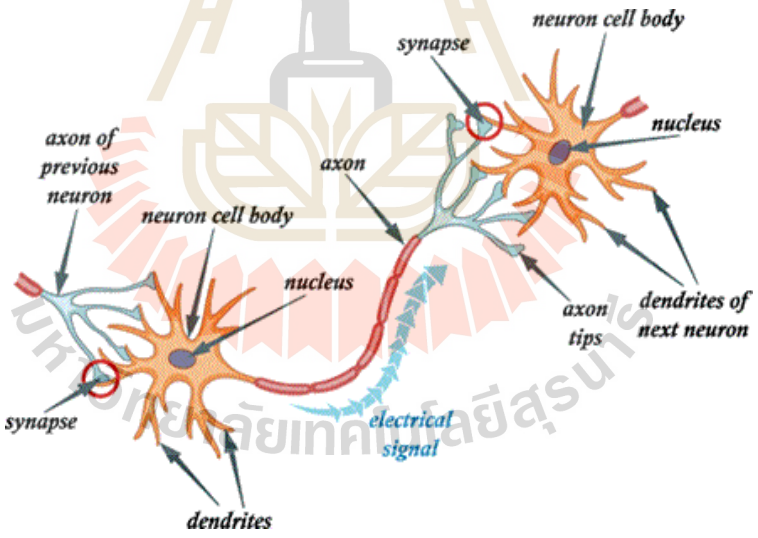
2.1 พื้นฐานโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial Neural Network, ANN) เป็นศาสตร์ว่าด้วยการคำนวณโดยอาศัยโครงข่าย (Network) ที่มีรูปแบบโครงสร้างและการทำงานของการประมวลผลเลียนแบบระบบการทำงานในสมองของมนุษย์ ที่มีการปรับเปลี่ยนตัวเองต่อการตอบสนองของอินพุต (Input) ตามกฎการเรียนรู้ (Learning Rule) หลังจากที่โครงข่ายได้เรียนรู้สิ่งที่ต้องการแล้วโครงข่ายนั้นก็จะสามารถทำงานตามที่กำหนดไว้ได้ (อาทิตย์ ศรีแก้ว, 2558)

สมองของมนุษย์ประกอบไปด้วยหน่วยประมวลผลที่เรียกว่า “นิวรอน” (Neuron) หรือเซลล์ประสาท จำนวนนิวรอนในสมองมนุษย์มีอยู่ประมาณ 10^{11} นิวรอนและมีการเชื่อมต่อกันอย่างมากมาย สมองมนุษย์จึงสามารถกล่าวได้ว่าเป็นคอมพิวเตอร์ที่มีการปรับตัวเอง (Adaptive) ในลักษณะแบบไม่เป็นเชิงเส้น (Nonlinear) และทำงานเป็นแบบขนาน (Parallel) ในการดูแลจัดการการทำงานร่วมกันของนิวรอนในสมอง (Hagan et al., 1996) ซึ่งคอมพิวเตอร์ในปัจจุบันถึงแม้มีความสามารถในการคำนวณสูงมาก ยังไม่สามารถเทียบกับความสามารถของสมองมนุษย์ในงานง่าย ๆ บางอย่าง เช่น การจดจำใบหน้า การฟังและการตีความหมาย การแปลภาษา เป็นต้น สมองของมนุษย์มีประสิทธิภาพและมั่นคงมาก ทุกวันมีเซลล์ประสาทในสมองตายโดยที่ไม่ส่งผลกระทบต่อประสิทธิภาพของสมองโดยรวม ระบบสมองของมนุษย์ยืดหยุ่นมากสามารถปรับตัวเข้ากับสิ่งแวดล้อมใหม่โดยการเรียนรู้ ต่างจากคอมพิวเตอร์ที่จะต้องโปรแกรมใหม่ สมองมนุษย์สามารถจัดการกับข้อมูลที่มีความไม่แน่นอน มีสัญญาณรบกวนและไม่สม่ำเสมอได้ดี สามารถประมวลผล

ข้อมูลขนาดมหาศาล เช่นรูปภาพ ในลักษณะการประมวลผลแบบขนานได้ดี สมองมีขนาดเล็กและใช้พลังงานน้อย นอกจากนี้โครงสร้างของสมองมนุษย์ได้วิวัฒนาการมาเป็นเวลาหลายล้านปีและได้รับการพิสูจน์จากธรรมชาติตราบจนกระทั่งทุกวันนี้

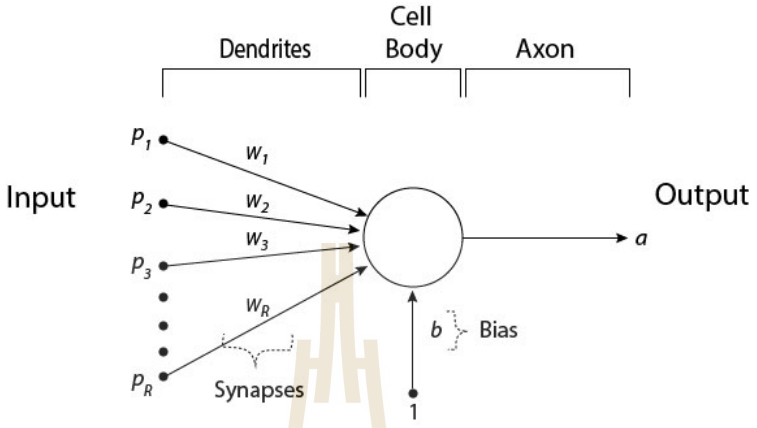
จากคุณลักษณะเด่น ๆ ดังที่กล่าวข้างต้นเกี่ยวกับสมองของมนุษย์จึงเกิดความต้องการที่จะสร้างการประมวลผลเพื่อเลียนแบบระบบการทำงานดังกล่าว ศาสตร์ทางด้านโครงข่ายประสาทเทียมจึงเป็นผลมาจากการศึกษาการเชื่อมต่อกันของโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมองซึ่งประกอบด้วยเซลล์ประสาทหรือนิวรอนและจุดประสานประสาท (Synapses) ดังแสดงในรูปที่ 2-1 ซึ่งแต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาทที่เรียกว่า “เดนไดรท์” (Dendrite) ซึ่งเป็นอินพุตเข้าสู่เซลล์และมีส่วนปลายในการส่งกระแสประสาทที่เรียกว่า “แอคซอน” (Axon) ซึ่งเป็นเหมือนเอาต์พุต (Output) ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอนิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอน



รูปที่ 2-1 ส่วนประกอบของเซลล์ประสาทในสมองมนุษย์ที่เป็นการเชื่อมต่อกันของโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง (Anatomylibrary, 2015)

จากการศึกษาการเชื่อมต่อกันของโครงข่ายไฟฟ้าชีวภาพในสมอง รูปแบบโครงสร้างและการทำงานของการประมวลผลเลียนแบบระบบการทำงานในสมองของมนุษย์ที่จำลองด้วยโครงข่ายประสาทเทียมอย่างง่ายจึงถูกจำลอง (Model) ขึ้นมาดังรูปที่ 2-2 ตัวโครงข่ายทำการเก็บข้อมูลความรู้

(Knowledge) ในระหว่างขั้นตอนของการเรียนรู้โดยทำการเก็บค่าที่ได้จากการเรียนรู้ไว้ที่ค่าของน้ำหนักประสาท (Synapses หรือ Synaptic Weights)

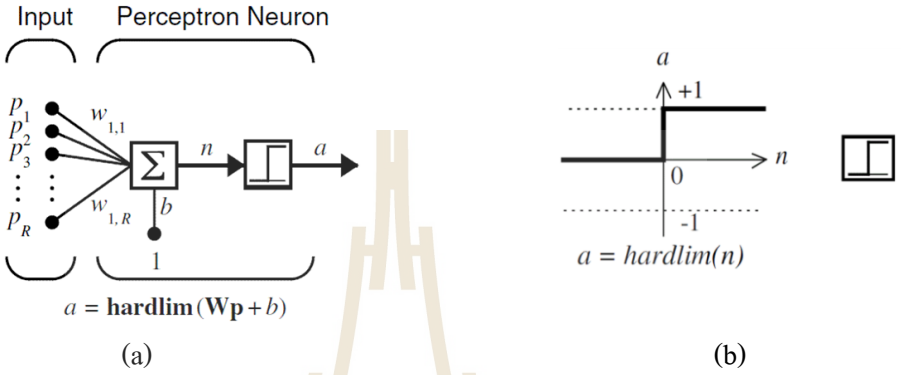


รูปที่ 2-2 โมเดลโครงข่ายประสาทเทียมอย่างง่ายที่เลียนแบบระบบการทำงานในสมองของมนุษย์

ในเวลาต่อมา ได้มีการนำเสนอรูปแบบโครงสร้างของตัวนิวรอนภายในโครงข่ายนั้นซึ่งมีอยู่มากมายหลายชนิด โครงสร้างดังกล่าวเป็นองค์ประกอบสำคัญที่ทำให้คุณลักษณะต่าง ๆ ของโครงข่ายแตกต่างกันออกไป ไม่ว่าจะเป็นการจัดวางเรียงตัวของนิวรอน กฎการเรียนรู้ที่ทำให้เกิดการปรับเปลี่ยนค่าของน้ำหนักประสาทหรือแม้กระทั่งเงื่อนไขในการฝึกสอนของโครงข่าย รูปที่ 2-3(a) แสดงโครงสร้างทางสถาปัตยกรรม (Architecture) ของโมเดลเพอร์เซปตรอน (Perceptron) ที่เป็นโครงข่ายแบบนิวรอนเดียว เมื่อ R คือ จำนวนองค์ประกอบ (Element) หรือจำนวนมิติ (Dimension) ของอินพุต นั่นคืออินพุตแต่ละตัวจะเป็นเวกเตอร์ (Vector) ที่ประกอบด้วย R องค์ประกอบ โดยเพอร์เซปตรอนนั้นเป็นโมเดลที่กำหนดให้ฟังก์ชันถ่ายโอน (Transfer Function/Activation Function/Signal Function) ที่ใช้ในโครงข่ายเป็นแบบฮาร์ดลิมิต (Hard Limit) ลักษณะของฟังก์ชันฮาร์ดลิมิตแสดงดังรูปที่ 2-3(b) ซึ่งฟังก์ชันถ่ายโอนนั้นใช้สำหรับการแปลงค่าของ n ไปเป็นเอาต์พุต (a) ของโครงข่าย

นอกเหนือจากเพอร์เซปตรอนแล้วมีการนำเสนอคุณลักษณะต่าง ๆ ของโครงข่ายที่แตกต่างกันออกไป ในเวลาต่อมาจึงเกิดเป็นโมเดลแบบต่าง ๆ ของโครงข่ายประสาทเทียม โดยถ้าแบ่งกลุ่มของโมเดลโครงข่ายประสาทเทียมดังกล่าวสามารถแบ่งได้ในสองรูปแบบคือแบ่งตามคุณลักษณะทางสถาปัตยกรรมของนิวรอน (Architecture Neuron Characteristic) และแบ่งตามอัลกอริทึมสำหรับการเรียนรู้ (Learning Algorithm) ซึ่งการแบ่งตามคุณลักษณะทางสถาปัตยกรรมของนิวรอน

นั้นสามารถแยกออกเป็นสถาปัตยกรรมแบบป้อนไปข้างหน้า (Feedforward) แบบวนซ้ำ (Recurrent) และแบบป้อนกลับ (Feedback) ส่วนการแบ่งตามอัลกอริทึมสำหรับการเรียนรู้นั้นสามารถแบ่งออกเป็น การเรียนรู้แบบมีผู้ฝึกสอน (Supervised Learning) การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) และการเรียนรู้แบบไม่มีผู้ฝึกสอน (Unsupervised Learning)



รูปที่ 2-3 โครงข่ายเพอร์เซปตรอนแบบนิวรอนเดียว (Demuth and Beale, 2004)

(a) สถาปัตยกรรมของโครงข่าย

(b) ฟังก์ชันถ่ายโอนแบบฮาร์ดลิมิตที่ใช้ในโครงข่าย

2.1.1 การเรียนรู้ของโครงข่ายประสาทเทียม

ตัวอย่างแสดงในรูปที่ 2-4 เป็นแนวคิดการนำโครงข่ายประสาทเทียมมาใช้เพื่อเรียนรู้สำหรับการคัดแยกแอปเปิ้ล (Apples) และ ส้ม (Oranges) ซึ่งเมื่อแอปเปิ้ลและส้มถูกนำเข้ามาคละกันทีละผลผ่านเซนเซอร์ (Sensors) เซนเซอร์ก็จะทำการวัดค่าต่าง ๆ ของแอปเปิ้ลหรือส้มผลนั้นออกเป็นสามค่า คือ ค่าของ shape ค่าของ texture และค่าของ weight ตามลำดับ ซึ่งทั้งสามค่านั้นนำมาประกอบเข้าด้วยกันเป็นค่าของเวกเตอร์อินพุต **p** นั่นคือ

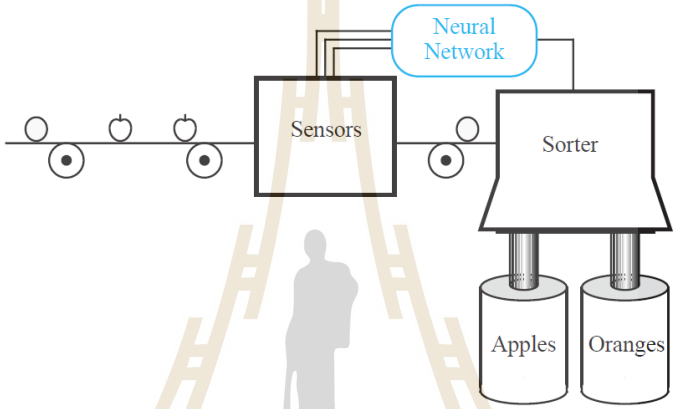
$$\mathbf{p} = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix}$$

โดยค่าของ shape บอกรูปร่างของผลไม้แต่ละผลว่ามีความกลมแค่ไหน (เกือบกลมให้มีค่าเป็น 1 แต่ถ้าเกือบเป็นวงรีมีค่าเป็น -1) ค่าของ texture บอกลักษณะของพื้นผิวว่าเรียบหรือขรุขระ (ถ้าเรียบจะเป็น 1 แต่ขรุขระจะเป็น -1) และค่าของ weight ที่บอกน้ำหนัก (ถ้ามากกว่า 1 ปอนด์ค่าเป็น 1 แต่น้อยกว่า 1 ปอนด์จะเป็น -1) นั่นคืออินพุตของผลไม้แต่ละผลที่นำเข้าไปใน

โมเดลของโครงข่ายประสาทเป็นเวกเตอร์ที่ประกอบด้วย 3 องค์ประกอบ ดังตัวอย่าง $\mathbf{p}_1$ และ $\mathbf{p}_2$ เมื่อ

$$\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \qquad \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

ด้วยโมเดลของโครงข่ายประสาท (Neural Network ในภาพ) เมื่อรับอินพุตของผลไม้แต่ละผลเข้ามาก็จะทำการตัดสินใจเพื่อคัดแยกว่า เป็นแอปเปิ้ลหรือส้มดังที่แสดงในรูปที่ 2-4



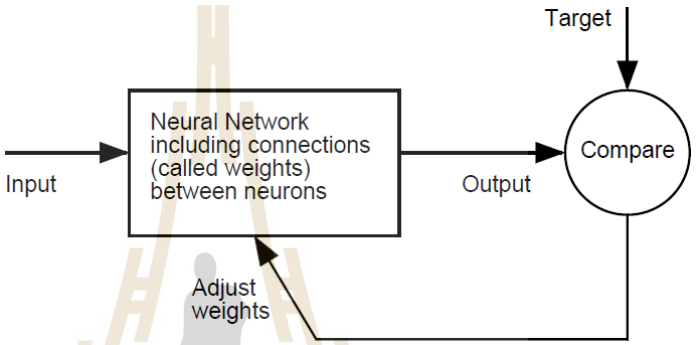
รูปที่ 2-4 ตัวอย่างปัญหาการใช้โครงข่ายประสาทเทียม (Hagan et al., 1996)

2.1.2 การเรียนรู้แบบมีผู้ฝึกสอน

ดังที่กล่าวไปแล้วว่าถ้าแบ่งโมเดลของโครงข่ายประสาทเทียมตามลักษณะของอัลกอริทึมสำหรับการเรียนรู้ สามารถแบ่งออกเป็น 3 แบบคือ แบบมีผู้ฝึกสอน แบบเสริมกำลัง และแบบไม่มีผู้ฝึกสอน โดยที่อัลกอริทึมสำหรับการเรียนรู้นั้นหมายถึง ขั้นตอน (Procedure) ที่ใช้สำหรับการปรับเปลี่ยน (Modify) ค่าน้ำหนัก (Weight) และค่าไบอัส (Bias) รวมถึงค่าพารามิเตอร์ (Parameters) ต่าง ๆ ที่ใช้ในแต่ละโครงข่าย ซึ่งขั้นตอนดังกล่าวอาจหมายถึง กฎการเรียนรู้ (Learning Rule) ที่ใช้สำหรับการเรียนรู้ (Learning) ในขั้นตอนการฝึกสอน (Training) ให้กับโครงข่ายก็ได้

สำหรับรูปที่ 2-5 เป็นแนวคิดของโมเดลที่เป็นการเรียนรู้แบบมีผู้ฝึกสอน นั่นคือกฎการเรียนรู้สำหรับโมเดลแบบมีผู้ฝึกสอนนั้นต้องใช้กับชุดข้อมูลตัวอย่าง (Set of Examples) ที่อยู่ในรูปของ $\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$, เมื่อ $\mathbf{p}_q$ เป็นเวกเตอร์ของอินพุตแต่ละตัวที่นำเข้าสู่โครงข่าย และ $\mathbf{t}_q$ คือค่าของเอาต์พุตเป้าหมาย (Target Output) ที่ต้องการของอินพุตตัวนั้น นั่นคือโมเดลที่

เป็นการเรียนรู้แบบมีผู้ฝึกสอนนั้น นอกเหนือจากที่ต้องป้อนเวกเตอร์อินพุต $(\mathbf{p}_q)$ แต่ละตัวให้กับโครงข่ายแล้ว ต้องป้อนค่าของเอาต์พุตเป้าหมาย $(\mathbf{t}_q)$ ที่สัมพันธ์กันกับอินพุตตัวนั้นด้วย เพื่อให้โครงข่ายสามารถนำค่าของเอาต์พุตเป้าหมายนั้นไปใช้สำหรับการเปรียบเทียบกับค่าของเอาต์พุตที่โครงข่ายหาออกมาว่ายังมีความผิดพลาด (Error) มากน้อยแค่ไหน เพื่อนำค่าความผิดพลาดดังกล่าวไปใช้ในขั้นตอนการปรับเปลี่ยน ค่าน้ำหนัก ค่าไบอัส รวมถึงค่าพารามิเตอร์ต่าง ๆ ตามกฎการเรียนรู้ที่กำหนดไว้ในแต่ละโครงข่ายต่อไป



รูปที่ 2-5 แนวคิดของการเรียนรู้แบบมีผู้ฝึกสอน (Demuth and Beale, 2004)

2.1.3 โครงข่ายเพอร์เซปตรอน

เพอร์เซปตรอนจัดเป็นโครงข่ายแบบชั้นเดียวที่มีจำนวนนิวรอนเพียงนิวรอนเดียวหรือมากกว่าก็ได้ สถาปัตยกรรมของเพอร์เซปตรอนที่เป็นโครงข่ายแบบนิวรอนเดียวได้กล่าวถึงไปก่อนหน้านี้แล้วดังรูปที่ 2-3 สำหรับสถาปัตยกรรมของเพอร์เซปตรอนแบบหลายนิวรอนแสดงดังรูปที่ 2-6 โดยที่รูปที่ 2-6(a) เป็นการเขียนโครงข่ายในรูปแบบเต็ม ที่แสดงการเชื่อมต่อกันทั้งหมดระหว่างแต่ละอินพุตกับทุก ๆ นิวรอน ส่วนรูปที่ 2-6(b) เป็นการเขียนโมเดลในรูปแบบย่อ เพื่อให้เข้าใจง่าย โดยมองภาพของข้อมูลทั้งหมดแต่ละส่วนในรูปแบบของเวกเตอร์และเมทริกซ์ เมื่อ R คือจำนวนองค์ประกอบของอินพุต และ S คือ จำนวนนิวรอน

จากสถาปัตยกรรมของเพอร์เซปตรอนที่แสดงในรูปที่ 2-6 จะเห็นว่าเอาต์พุต (a) ของโครงข่ายกำหนดโดยสมการที่ 2-1

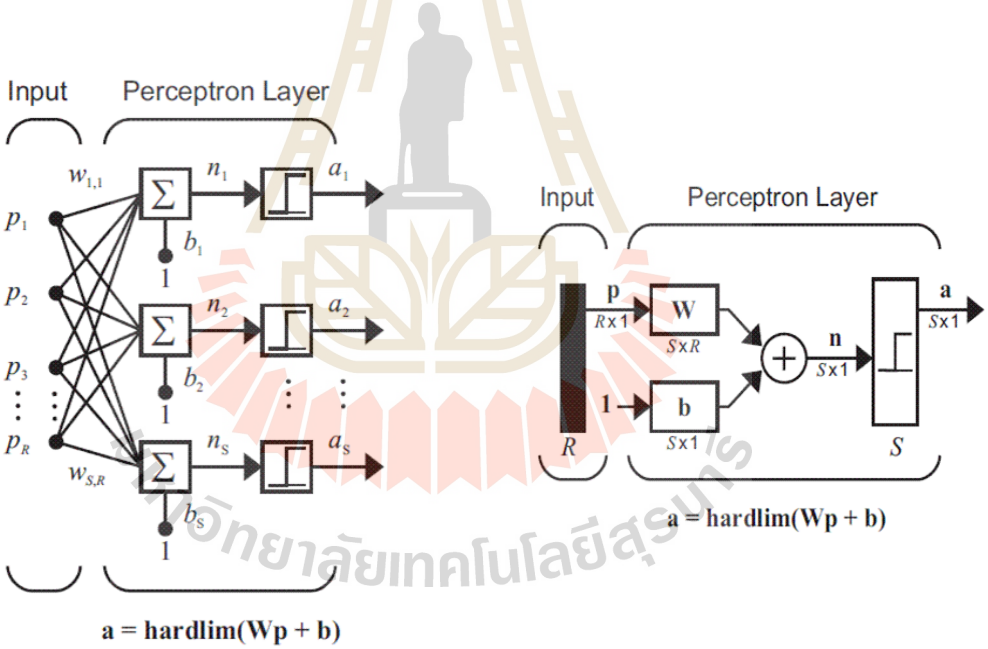
$$\mathbf{a} = hardlim(\mathbf{Wp} + \mathbf{b}) \quad = hardlim(\mathbf{n}) \tag{2-1}$$

เมื่อ **p** คือเวกเตอร์ของอินพุต **b** คือเวกเตอร์ของไบอัส **W** คือเมทริกซ์ของค่าน้ำหนัก และ **n** คือเวกเตอร์ที่เป็นผลลัพธ์จากการนำเมทริกซ์ของค่าน้ำหนักคูณกับเวกเตอร์ของอินพุตแล้วบวกด้วยเวกเตอร์ของไบอัส ซึ่งเมทริกซ์ของค่าน้ำหนัก **W** มีค่าในแต่ละองค์ประกอบคือ

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

โครงข่ายเพอร์เซปตรอนนั้นใช้ฟังก์ชันถ่ายโอนเป็นฟังก์ชันฮาร์ดลิมิต กำหนดโดยสมการที่ 2-2

$$\mathbf{a} = hardlim(\mathbf{n}) = \begin{cases} 1 & if\ \mathbf{n} \geq 0 \\ 0 & otherwise \end{cases} \tag{2-2}$$



(a) (b)

รูปที่ 2-6 โครงข่ายเพอร์เซปตรอนแบบหลายนิวรอน (Demuth and Beale, 2004)

(a) สถาปัตยกรรมของโครงข่าย

(b) การเขียนโมเดลในรูปแบบย่อ

กฎการเรียนรู้ของเพอร์เซปตรอน (Perceptron Learning Rule)

จากที่ได้กล่าวไปแล้วว่าเพอร์เซปตรอนเป็นโครงข่ายแบบมีผู้ฝึกสอน ดังนั้นเมื่อพิจารณาชุดข้อมูลตัวอย่าง ที่อยู่ในรูปของ $\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$, เมื่อ $\mathbf{p}_q$ เป็นเวกเตอร์ของอินพุตแต่ละตัวที่นำเข้าสู่โครงข่าย และ $\mathbf{t}_q$ คือค่าเวกเตอร์เป้าหมายของอินพุตตัวนั้น โดยกฎการเรียนรู้ของเพอร์เซปตรอนกำหนดดังสมการที่ 2-3, 2-4 และ 2-5

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{e}\mathbf{p}^T \tag{2-3}$$

$$\mathbf{b}^{new} = \mathbf{b}^{old} + \mathbf{e} \tag{2-4}$$

$$where, \qquad \mathbf{e} = \mathbf{t} - \mathbf{a} \tag{2-5}$$

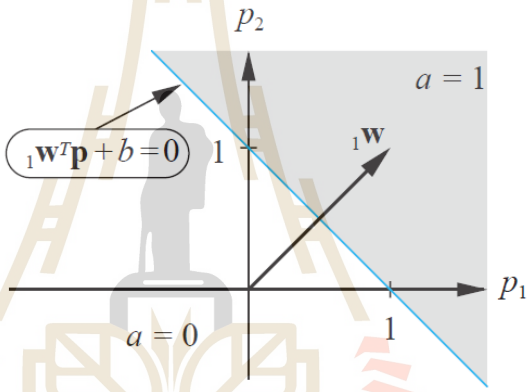
จากกฎการเรียนรู้จะเห็นว่าเป็นการนำค่าความผิดพลาด ($\mathbf{e}$) มาใช้สำหรับการปรับค่าของน้ำหนัก ($\mathbf{W}$) และไบอัส ($\mathbf{b}$) ซึ่งค่าความผิดพลาดมาจากการนำค่าเอาต์พุตเป้าหมาย ($\mathbf{t}$) ลบด้วยค่าเอาต์พุตที่ได้จากโครงข่าย ($\mathbf{a}$) ซึ่งจะสังเกตได้ว่าด้วยกฎการเรียนรู้ของเพอร์เซปตรอน ค่าของน้ำหนักจะไม่ถูกปรับ (มีค่าเท่าเดิม) เมื่อค่าความผิดพลาดเป็นศูนย์ นั่นคือเมื่อโครงข่ายสามารถเรียนรู้อินพุตตัวนั้นแล้วได้ค่าของเอาต์พุตจากโครงข่ายเหมือนกับค่าของเอาต์พุตเป้าหมายหรือเอาต์พุตที่เราต้องการจริง ๆ

ดังนั้นในขั้นตอนการฝึกสอนของโครงข่ายเพอร์เซปตรอนประกอบด้วยขั้นตอนย่อย ๆ ดังต่อไปนี้

1. รับข้อมูลจากชุดข้อมูลตัวอย่างเข้ามาครั้งละ 1 ตัว ในรูปของคู่ข้อมูลอินพุตและเอาต์พุตเป้าหมาย $\{\mathbf{p}_q, \mathbf{t}_q\}$
2. คำนวณค่าของเอาต์พุต ($\mathbf{a}$) ที่ได้จากโครงข่ายด้วยสมการที่ 2-1
3. คำนวณค่าความผิดพลาด ($\mathbf{e}$) ด้วยสมการที่ 2-5
4. ปรับค่าน้ำหนักและไบอัสด้วยสมการที่ 2-3 และ 2-4 ตามลำดับ
5. รับข้อมูลตัวต่อไป แล้วทำขั้นตอนที่ 2 ถึงขั้นตอนที่ 4 เช่นเดิม

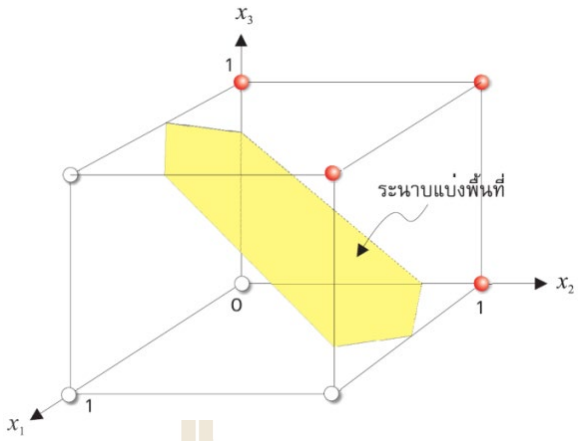
เมื่อรับข้อมูลมาครบทุกตัวแล้วถือว่าทำครบ 1 รอบใหญ่ (1 Epoch) จากนั้นจะดูค่าความผิดพลาดของข้อมูลทุกตัวว่าเป็นศูนย์ทั้งหมดแล้วหรือไม่ ถ้ายังก็ทำต่ออีกครั้งละ 1 Epoch ไปเรื่อย ๆ จนได้ค่าความผิดพลาดของข้อมูลทุกตัวเป็นศูนย์ทั้งหมดจึงหยุดการเรียนรู้

ค่าน้ำหนักทั้งหมดที่ได้หลังจากที่เพอร์เซปตรอนหยุดการเรียนรู้เมื่อนำไปแทนในสมการของค่า a ในสมการที่ 2-1 คือสมการเส้นตรงที่เมื่อแทนค่าข้อมูลใด ๆ บนเส้นตรงนั้นจะได้ค่าจากสมการเป็นศูนย์ ดังรูปที่ 2-7 จากรูปตัวอย่างเป็นกรณีที่อินพุต มี 2 องค์ประกอบ ($R = 2$) และในโครงข่ายใช้จำนวนนิวรอน 1 นิวรอน ($S = 1$) นั่นคือจะได้ตัวแบ่งแยกแบบเชิงเส้น (Linear Separable) ซึ่งตัวแบ่งแยกดังกล่าวสามารถแบ่งข้อมูลทั้งหมดออกเป็นสองส่วนได้ โดยที่เวกเตอร์ของค่าน้ำหนักจะตั้งฉากกับตัวแบ่งแยกเสมอ จากเส้นแบ่งแยกที่ได้ทำให้ทุกจุดที่อยู่ด้านบนเส้นแบ่งถือว่าอยู่ในกลุ่มที่ 1 ($a = 1$) เพราะทุกจุดนั้นเมื่อแทนในสมการของเส้นแบ่งแยกจะมีค่ามากกว่าศูนย์ทั้งหมด และทุกจุดที่อยู่ด้านล่างเส้นแบ่งแยกถือว่าอยู่ในกลุ่มที่ 2 ($a = 0$) เพราะเมื่อแทนในสมการของเส้นแบ่งแยกก็จะมีค่าน้อยกว่าหรือเท่ากับศูนย์ทั้งหมด

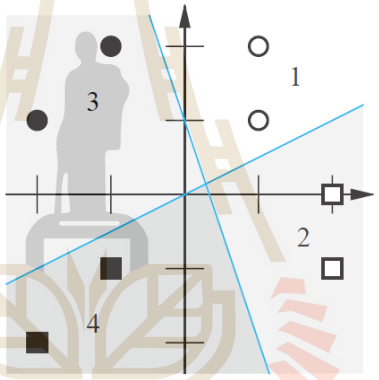


รูปที่ 2-7 เส้นแบ่งแยกที่ได้จากค่าน้ำหนักสุดท้ายของเพอร์เซปตรอน (Hagan et al., 1996)

รูปที่ 2-8 เป็นตัวอย่างการมองภาพของเพอร์เซปตรอนเมื่อแบ่งแยกข้อมูลที่มี $R = 3$ และใช้ $S = 1$ ให้สามารถแบ่งข้อมูลในสามมิติออกได้เป็นสองกลุ่ม ผลลัพธ์ที่ได้เป็นระนาบแบ่งแยกแบบเชิงเส้น (Linear) ที่แบ่งกล่องออกเป็นสองส่วน ส่วนรูปที่ 2-9 ยกตัวอย่างกรณีข้อมูลมี $R = 2$ และใช้ $S = 2$ เพื่อให้สามารถแบ่งแยกข้อมูลสองมิติที่ต้องการแบ่งออกเป็นสี่กลุ่ม นั่นคือได้ตัวแบ่งแยกเป็นเส้นตรงสองเส้น แต่ละเส้นแทนผลลัพธ์จากค่าน้ำหนักสุดท้ายของแต่ละนิวรอนหลังจบขั้นตอนการเรียนรู้



รูปที่ 2-8 ระนาบแบ่งแยกที่ได้กรณี *R=3, S=1* (อาทิตย์ ศรีแก้ว, 2558)



รูปที่ 2-9 เส้นแบ่งแยกที่ได้จากเส้นตรงสองเส้นประกอบกัน กรณี *R=2, S=2* (แบ่งข้อมูลสองมิติออกเป็น 4 กลุ่ม) (Hagan et al., 1996)

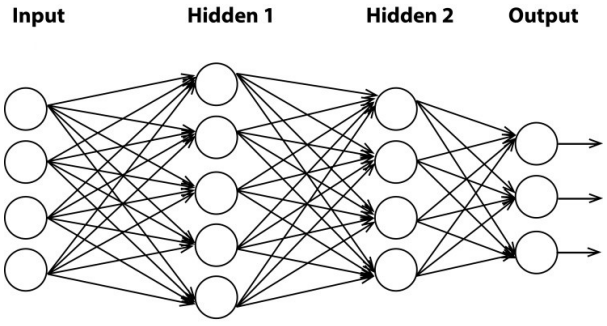
2.2 โครงข่ายเพอร์เซปตรอนแบบหลายชั้น

เนื่องจากคุณลักษณะของโมเดลเพอร์เซปตรอนนั้นเป็นโครงข่ายแบบชั้นเดียวที่มีข้อจำกัด คือ สามารถใช้จำแนกได้เฉพาะรูปแบบข้อมูลที่สามารถแบ่งแยกได้แบบเชิงเส้นเท่านั้น ในเวลาต่อมาถึงแม้ว่าจะมีการนำเสนอโครงข่าย ADALINE (ADAptiveLInearNEuron) โดย Windrow และ Hoff (1960) ซึ่งเป็นการนำเสนอโครงข่ายพร้อมกฎการเรียนรู้แบบกำลังสองเฉลี่ยน้อยที่สุด (Least Mean Square) หรือ LMS (Widrow, 1987) ซึ่ง ADALINE นั้นคล้ายคลึงกับเพอร์เซปตรอนแต่ฟังก์ชันถ่ายโอนที่ใช้ในโครงข่ายเป็นแบบเชิงเส้น (Linear) แทนที่จะเป็นแบบฮาร์ดลิมิต และการนำกฎการเรียนรู้แบบ LMS มาใช้นั้นทำให้โครงข่ายสามารถทนทานต่อสัญญาณรบกวนได้ดีกว่าเพอร์

เซปตรอน ซึ่ง ADALINE นั้นมีการนำไปประยุกต์ใช้งานจริงเกี่ยวกับงานทางด้านการประมวลผลสัญญาณดิจิทัลกันมาก อย่างไรก็ตามทั้งเพอร์เซปตรอนและ ADALINE ต่างก็มีข้อจำกัดเดียวกันตรงที่ใช้ได้กับปัญหาที่เป็นแบบแบ่งแยกได้แบบเชิงเส้นเท่านั้น จึงไม่สามารถนำไปประยุกต์ใช้ได้กับปัญหาส่วนใหญ่ได้มากนัก ในเวลาต่อมาแนวคิดเกี่ยวกับ โครงข่ายเพอร์เซปตรอนแบบหลายชั้น (Multilayer Perceptron, MLP) จึงได้ถูกนำเสนอขึ้น ซึ่งคุณลักษณะพื้นฐานของเพอร์เซปตรอนแบบหลายชั้นคือ (Haykin, 2009)

1. แต่ละนิวรอนในโครงข่ายจะมีการนำฟังก์ชันถ่ายโอนที่เป็นแบบ “Nonlinear” มาใช้
2. ชั้นซ่อนเร้นที่ประกอบด้วยนิวรอนที่ไม่ได้เป็นอินพุตและเอาต์พุต มีได้เป็นหนึ่งชั้นหรือมากกว่าก็ได้
3. โครงข่ายมีลักษณะการเชื่อมต่อกันสูง (High Degree of Connectivity) ด้วยค่าของค่าน้ำหนัก

จากคุณลักษณะพื้นฐานดังกล่าวของเพอร์เซปตรอนแบบหลายชั้นจึงสามารถนำไปใช้ประยุกต์ใช้ได้กับงานในหลาย ๆ ด้าน จึงเป็นที่รู้จักและเป็นที่นิยมในเวลาต่อมา โครงข่ายเพอร์เซปตรอนแบบหลายชั้นนั้นเป็นสถาปัตยกรรมแบบป้อนไปข้างหน้า (Feedforward) ที่ประกอบด้วย 3 ชั้นหลักคือ ชั้นอินพุต (Input layer) ชั้นซ่อนเร้น (Hidden layer) และชั้นเอาต์พุต (Output Layer) โดยที่ชั้นซ่อนเร้นสามารถมีได้มากกว่าหนึ่งชั้น ดังแสดงในรูปที่ 2-10 จากรูปเป็นตัวอย่างของโครงข่ายที่มีชั้นซ่อนเร้นสองชั้น โดยแต่ละชั้นซ่อนเร้นนั้นอาจมีจำนวนโหนดหรือจำนวนนิวรอนหนึ่งนิวรอนหรือมากกว่าได้ขึ้นอยู่การนำไปประยุกต์ใช้กับแต่ละปัญหา และในแต่ละชั้นก็ไม่จำเป็นต้องมีจำนวนนิวรอนเท่ากัน จากรูปที่ 2-10 เป็นตัวอย่างโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อนเร้นที่ 1 (Hidden 1) เป็น 5 นิวรอน มีจำนวนนิวรอนในชั้นซ่อนเร้นที่ 2 (Hidden 2) เป็น 4 นิวรอน และมีจำนวนนิวรอนในชั้นเอาต์พุตเป็น 3 นิวรอน โดยเพอร์เซปตรอนแบบหลายชั้นก็เป็นการเรียนรู้แบบมีผู้ฝึกสอนเช่นเดียวกันกับเพอร์เซปตรอนแบบชั้นเดียว



รูปที่ 2-10 สถาปัตยกรรมของโครงข่ายเพอร์เซปตรอนแบบหลายชั้น

2.2.1 เพอร์เซปตรอนแบบหลายชั้นที่ผนวกด้วยอัลกอริทึมแบบแพร่กลับ

โครงข่ายเพอร์เซปตรอนแบบหลายชั้นที่มีการทำงานร่วมกับเทคนิคการเรียนรู้หรืออัลกอริทึมแบบแพร่กลับ (Backpropagation Algorithm) ซึ่งนำเสนอโดย Rumelhart (1986a, 1986b) นั้นมีการนำมาประยุกต์ใช้อย่างกว้างขวางในหลายด้าน ไม่ว่าจะเป็นการรู้จำรูปแบบ (Pattern Recognition) การจำแนก (Classification) รวมถึงการประมาณค่า (Approximation) และการพยากรณ์ (Forecasting) การทำงานของอัลกอริทึมแบบแพร่กลับ ที่มีจำนวนนิวรอนในชั้นอินพุต ชั้นซ่อนเร้นและชั้นเอาต์พุตเป็น n, q และ p ตามลำดับ มีขั้นตอนดังแสดงในตารางที่ 2-1 (Kumar, 2005)

จากอัลกอริทึมแบบแพร่กลับในตารางที่ 2-1 จะเห็นว่าเป็นลักษณะการทำงานในสองขั้นตอนหลักคือ ขั้นตอนของการส่งค่าผ่านไปข้างหน้าและขั้นตอนแพร่กลับ โดยขั้นตอนของการส่งค่าผ่านไปข้างหน้าเป็นการคำนวณค่าต่าง ๆ ที่เกี่ยวข้องกันในแต่ละชั้นจากชั้นทางซ้ายไปสู่ชั้นทางขวาของโครงข่ายจนได้ออกมาเป็นค่าเอาต์พุตของโครงข่าย ส่วนขั้นตอนการแพร่กลับจะเป็นการคำนวณในลักษณะตรงกันข้ามคือ ส่งค่าจากชั้นทางขวาของโครงข่ายไปยังชั้นทางซ้าย ซึ่งเป็นขั้นตอนที่ใช้สำหรับการหาค่าความผิดพลาดต่าง ๆ (Delta/Error) และการหาค่าน้ำหนักที่เปลี่ยนแปลงไป แล้วนำค่าเหล่านั้นมาใช้สำหรับการปรับค่าของน้ำหนัก โดยขั้นตอนดังกล่าวนี้ต้องทำแบบย้อนกลับหรือแพร่กลับเพราะระหว่างสองชั้นใด ๆ ที่อยู่ติดกันค่า Delta/Error ของชั้นทางซ้ายจำเป็นต้องใช้ค่า Delta/Error จากชั้นทางขวาซึ่งต้องคำนวณมาก่อนด้วยสมการที่ใช้ในขั้นตอนแพร่กลับของอัลกอริทึม

ตารางที่ 2-1 เพอร์เซปตรอนแบบหลายชั้นที่ผนวกด้วยอัลกอริทึมแบบแพร่กลับ (Kumar, 2005)

กำหนดให้:

- *ข้อมูลแต่ละตัวที่ต้องการฝึกสอนคือเวกเตอร์* $X_k \in \mathbb{R}^n$
- *เอาต์พุตเป้าหมาย (Target Output) ของข้อมูลแต่ละตัวคือเวกเตอร์* $D_k \in \mathbb{R}^p$

กำหนดค่าเริ่มต้น:

- *กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้กับทุกค่าที่เชื่อมระหว่างชั้นอินพุตและชั้นซ่อนเร้น นั่นคือค่าของ* w_{ih}^1 *และกำหนดให้* $\Delta w_{ih}^0 = 0,\ i = 0, \dots, n;\ h = 1, \dots, q$
- *กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้กับทุกค่าที่เชื่อมระหว่างชั้นซ่อนเร้นและชั้นเอาต์พุต ซึ่งคือค่าของ* w_{hj}^1 *และกำหนดให้* $\Delta w_{hj}^0 = 0,\ h = 0, \dots, q;\ j = 1, \dots, p$
- *ให้* $k = 1$ *และกำหนดค่าโมเมนตัม* η *ค่าคงที่การเรียนรู้* α *และค่าความผิดพลาดที่ยอมรับได้* τ *ตามที่ต้องการ*

ตารางที่ 2-1 เพอร์เซปตรอนแบบหลายชั้นที่ผนวกด้วยอัลกอริทึมแบบแพร่กลับ (Kumar, 2005) (ต่อ)

วนลูปเพื่อทำซ้ำในแต่ละ k:

{

– *เลือกข้อมูลมาหนึ่งคู่ นั่นคือข้อมูลอินพุตที่ต้องการฝึกสอนและเอาต์พุตเป้าหมายที่ต้องการของข้อมูลตัวนั้น กำหนดให้เป็น* (X_k, D_k)

ขั้นตอนส่งค่าผ่านไปข้างหน้า (Forward) :

– *คำนวณค่าสัญญาณต่าง ๆ ตามลำดับของสมการต่อไปนี้*

$$\mathcal{S}(x_i^k) = x_i^k, \qquad i = 1, \dots, n$$

$$\mathcal{S}(x_0^k) = 1$$

$$\mathcal{Z}_h^k = \sum_{i=0}^{n} w_{ih}^k x_i^k, \qquad h = 1, \dots, q$$

$$\mathcal{S}(\mathcal{Z}_h^k) = \frac{1}{1 + \exp(-\mathcal{Z}_n^k)}, \qquad h = 1, \dots, q$$

$$\mathcal{S}(\mathcal{Z}_0^k) = 1$$

$$y_j^k = \sum_{h=0}^{q} w_{hj}^k \mathcal{S}(\mathcal{Z}_h^k), \qquad j = 1, \dots, p$$

$$\mathcal{S}(y_j^k) = \frac{1}{1 + \exp(-y_j^k)}, \qquad j = 1, \dots, p$$

ขั้นตอนแพร่กลับ (Backpropagate) :

– *คำนวณ Delta/Error ที่นิวรอนชั้นเอาต์พุตและหาค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นซ่อนเร้นและชั้นเอาต์พุตตามสมการคือ*

$$\delta_j^k = \left(d_j^k - \mathcal{S}(y_j^k)\right) \mathcal{S}'(y_j^k) \qquad j = 1, \dots, p$$

$$\Delta w_{hj}^k = \eta \delta_j^k \mathcal{S}(\mathcal{Z}_h^k) \qquad h = 0, \dots, q;\ j = 1, \dots, p$$

– *คำนวณ Delta/Error ที่นิวรอนชั้นซ่อนเร้น และหาค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นอินพุตและชั้นซ่อนเร้นตามสมการคือ*

$$\delta_h^k = \left(\sum_{j=1}^{p} \delta_j^k w_{hj}^k\right) \mathcal{S}'(\mathcal{Z}_h^k) \qquad h = 1, \dots, q$$

ตารางที่ 2-1 เพอร์เซปตรอนแบบหลายชั้นที่ผนวกด้วยอัลกอริทึมแบบแพร่กลับ (Kumar, 2005) (ต่อ)

$$\Delta w_{ih}^{k} = \eta \delta_{h}^{k} x_{i}^{k} \qquad i = 0, \ldots, n;\ h = 1, \ldots, q$$

– *ปรับค่าของน้ำหนักตามสมการคือ*

$$w_{hj}^{k+1} = w_{hj}^{k} + \Delta w_{hj}^{k} + \alpha \Delta w_{hj}^{k-1} \qquad h = 0, \ldots, q;\ j = 1, \ldots, p$$

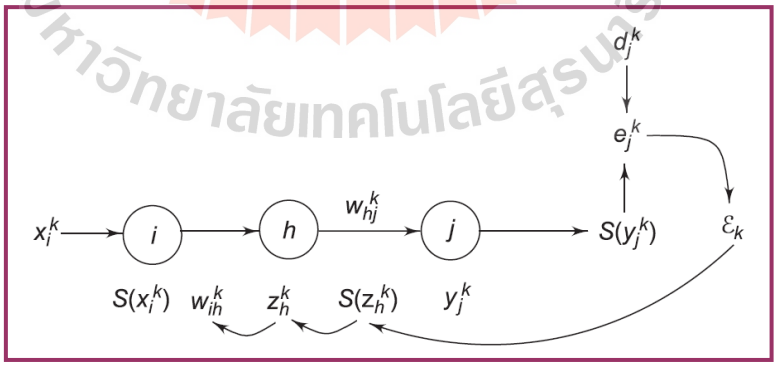
$$w_{ih}^{k+1} = w_{ih}^{k} + \Delta w_{ih}^{k} + \alpha \Delta w_{ih}^{k-1} \qquad i = 0, \ldots, n;\ h = 1, \ldots, q$$

– *คำนวณค่าความผิดพลาดให้เป็น* $\varepsilon_k = \frac{1}{2}\sum_{i=1}^{p}(d_j^k - s(y_j^k))^2$

} *ทำจนกระทั่ง* $(\varepsilon_{av} = \frac{1}{Q}\sum_{k=1}^{Q}\varepsilon_k < \tau)$ *เมื่อ* Q *คือจำนวนข้อมูล*

จากขั้นตอนของอัลกอริทึมดังกล่าว เพื่อให้เข้าใจการทำงานชัดเจนมากยิ่งขึ้น จึงยกตัวอย่างการคำนวณด้วยมือ (Hand Work Example) ที่แสดงรายละเอียดของค่าพารามิเตอร์ที่เกิดขึ้นในแต่ละขั้นตอน พารามิเตอร์ต่าง ๆ เหล่านั้นแสดงดังรูปที่ 2-11 เมื่อทิศทางของลูกศรจากซ้ายไปขวาหมายถึงพารามิเตอร์นั้นเกิดขึ้นในขั้นตอนส่งผ่านค่าไปข้างหน้า และเมื่อทิศทางของลูกศรจากขวาไปซ้ายจะเป็นพารามิเตอร์ที่เกิดขึ้นในขั้นตอนแพร่กลับของอัลกอริทึม

ตารางที่ 2-2 แสดงถึงสัญลักษณ์ต่าง ๆ ที่ใช้ในอัลกอริทึมแบบแพร่กลับที่สัมพันธ์กันกับแต่ละชั้นของโครงข่าย



รูปที่ 2-11 ค่าพารามิเตอร์ที่เกิดขึ้นในแต่ละขั้นตอนของส่วนการแพร่ไปข้างหน้าและส่วนการแพร่กลับของอัลกอริทึมแบบแพร่กลับ (Kumar, 2005)

ตารางที่ 2-2 สัญลักษณ์ที่ใช้ในอัลกอริทึมแบบแพร่กลับที่สัมพันธ์กันกับแต่ละชั้นของโครงข่าย (Kumar, 2005)

	Input	*Hidden*	*Output*
Number of neurons	$n+1$	$q+1$	p
Signal function	linear	sigmoidal	sigmoidal
Neuron index range	$i = 0, \ldots, n$	$h = 0, \ldots, q$	$j = 1, \ldots, p$
Activation	x_i	z_h	y_j
Signal	$S(x_i)$	$S(z_h)$	$S(y_j)$
Weights (including bias)	$\rightarrow w_{ih} \rightarrow$	$\rightarrow w_{hj} \rightarrow$	

สำหรับโครงข่ายที่ใช้ยกตัวอย่างนั้นมีสถาปัตยกรรมแบบ $n-p-q$ นั่นคือเป็นโครงข่ายแบบมีชั้นซ่อนเร้นเพียงชั้นเดียว ที่มีจำนวนนิวรอนในชั้นอินพุต 2 นิวรอน จำนวนนิวรอนในชั้นซ่อนเร้น 2 นิวรอน และจำนวนนิวรอนในชั้นเอาต์พุต 2 นิวรอนเช่นเดียวกัน (นั่นคือมี $n = 2,\ p = 2$ และ $q = 2$) สำหรับตารางที่ 2-3 แสดงค่าจำนวนของนิวรอน (Number of Neurons) ในชั้นอินพุตเป็น $n+1$ เพราะเป็นการมองค่าของไบอัสเป็นอีกนิวรอนหนึ่งด้วย เช่นเดียวกันกับในชั้นซ่อนเร้นที่เป็นมีจำนวนนิวรอนเป็น $q+1$ เพื่อความเป็นหนึ่งเดียวกันในการคำนวณของอัลกอริทึม

ตารางที่ 2-3 เป็นข้อมูลสองตัว (สอง Pattern Index) ที่ใช้ประกอบการยกตัวอย่าง โดยแต่ละตัวมีสององค์ประกอบ คือ (x_1, x_2) ส่วน (d_1, d_2) เป็นเอาต์พุตเป้าหมายที่สัมพันธ์กันของข้อมูลแต่ละตัว ซึ่งในงานประยุกต์จริงนั้นจำนวนข้อมูลมีมากกว่านี้มาก ในที่นี้เป็นการยกตัวอย่างข้อมูลที่กำหนดค่ามาเพื่อประกอบการคำนวณด้วยมือเท่านั้น ซึ่งสอดคล้องกับขั้นตอนในอัลกอริทึมคือ

กำหนดให้:

- *ข้อมูลแต่ละตัวที่ต้องการฝึกสอนคือเวกเตอร์* $X_k \in \mathbb{R}^n$
- *เอาต์พุตเป้าหมายของข้อมูลแต่ละตัวคือเวกเตอร์* $D_k \in \mathbb{R}^p$

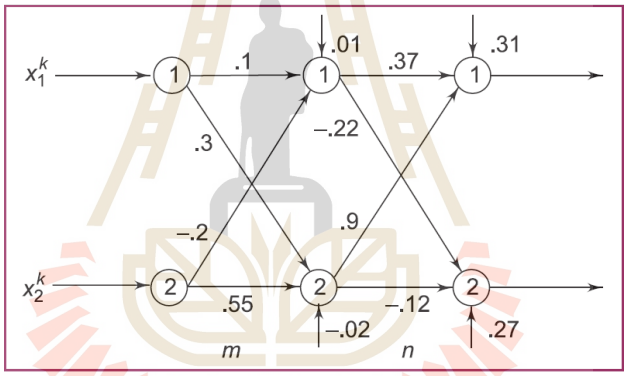
ตารางที่ 2-3 ข้อมูลที่ใช้ประกอบการยกตัวอย่าง (Kumar, 2005)

Pattern Index	x_1^k	x_2^k	d_1^k	d_1^k
1	0.5	-0.5	0.9	0.1
2	-0.5	0.5	0.1	0.9

รูปที่ 2-12 แสดงค่าน้ำหนักเริ่มต้นที่กำหนดให้กับโครงข่าย เมื่อ $k = 1$ นั่นคือ กำหนดค่าแบบสุ่ม (Random) ให้กับ w_{ih}^1 และ w_{hj}^1 ทุกตัว แต่ละค่าดังกล่าวแสดงด้วยค่าของตัวเลขบนแต่ละเส้นเชื่อมในรูปที่ 2-12 ซึ่งสอดคล้องกับส่วนของขั้นตอนในอัลกอริทึมคือ

กำหนดค่าเริ่มต้น:

- *กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มทุกตัวที่เชื่อมระหว่างชั้นอินพุตและชั้นซ่อนเร้น นั่นคือ* w_{ih}^1 *และให้* $\Delta w_{ih}^0 = 0,\ i = 0, \dots, n;\ h = 1, \dots, q$
- *กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มทุกตัวที่เชื่อมระหว่างชั้นซ่อนเร้นและชั้นเอาต์พุต ซึ่งคือ* w_{hj}^1 *และให้* $\Delta w_{hj}^0 = 0,\ h = 0, \dots, q;\ j = 1, \dots, p$
- *ให้* $k = 1$ *และกำหนดค่าคงที่การเรียนรู้* $\eta = 1.2$ *ค่าโมเมนตัม* $\alpha = 0.8$ *และค่าความผิดพลาดที่ยอมรับได้* τ *ตามที่ต้องการ*



รูปที่ 2-12 ค่าน้ำหนักเริ่มต้นแบบสุ่มที่กำหนดให้กับ w_{ih}^1 และ w_{hj}^1 (Kumar, 2005)

เมื่อ $k = 1$ จะพิจารณาข้อมูลใน Pattern Index ที่ 1 ของตารางที่ 2-3 แล้วคำนวณค่าสัญญาณต่าง ๆ ในขั้นตอนส่งค่าผ่านไปข้างหน้าของอัลกอริทึม นั่นคือ ณ $k = 1$ เป็นการพิจารณาข้อมูลอินพุตและเอาต์พุตเป้าหมายใน Pattern Index ที่ 1 ซึ่งกำหนดมาจากตารางที่ 2-3 มีค่าเมื่อแสดงในรูปแบบเวกเตอร์คือ

$$X_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \qquad D_1 = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$

นั่นคือเป็นการคำนวณตามลำดับของสมการจากส่วนของอัลกอริทึมต่อไปนี้

เมื่อ $\boldsymbol{k = 1}$:

- *เลือกข้อมูลมาหนึ่งคู่ ให้เป็นข้อมูลที่จะฝึกสอนและเอาต์พุตเป้าหมายของข้อมูลตัวนั้น กำหนดให้เป็น* (X_1, D_1) *นั่นคือ พิจารณา Pattern Index ที่ 1 ของตารางที่ 2-4*

<u>*ขั้นตอนส่งผ่านค่าไปข้างหน้า*</u> *:*

- *คำนวณค่าสัญญาณต่าง ๆ ตามลำดับของสมการต่อไปนี้*

$$\mathcal{S}(x_i^k) = x_i^k, \qquad i = 1, \dots, n$$
$$\mathcal{S}(x_0^k) = 1$$
$$Z_h^k = \sum_{i=0}^{n} w_{ih}^k x_i^k, \qquad h = 1, \dots, q$$
$$\mathcal{S}(Z_h^k) = \frac{1}{1 + \exp(-Z_n^k)}, \qquad h = 1, \dots, q$$
$$\mathcal{S}(Z_0^k) = 1$$
$$y_j^k = \sum_{h=0}^{q} w_{hj}^k \mathcal{S}(Z_h^k), \qquad j = 1, \dots, p$$
$$\mathcal{S}(y_j^k) = \frac{1}{1 + \exp(-y_j^k)}, \qquad j = 1, \dots, p$$

เมื่อนำข้อมูลอินพุตใน Pattern Index ที่ 1 ของตารางที่ 2-3 มาแทนในแต่ละสมการส่วนของอัลกอริทึมทั้งหมดในตารางด้านบน จะได้ผลลัพธ์ค่าสัญญาณต่าง ๆ ดังตารางที่ 2-4

ตารางที่ 2-4 ค่าสัญญาณต่าง ๆ ที่ได้จากการคำนวณในขั้นตอนการส่งผ่านค่าไปข้างหน้าเมื่อ $k = 1$ (Kumar, 2005)

k	x_1^k	x_2^k	$\mathcal{S}(x_1^k)$	$\mathcal{S}(x_2^k)$	z_1^k	z_2^k	$\mathcal{S}(z_1^k)$	$\mathcal{S}(z_2^k)$	y_1^k	y_2^k	$\mathcal{S}(y_1^k)$	$\mathcal{S}(y_2^k)$
1	.5	–.5	.5	–.5	.16	–.145	.5399	.4638	.9271	.0955	.7164	.5238

ขั้นตอนต่อมาเป็นขั้นตอนแพร่กลับ โดยเริ่มต้นจากการคำนวณค่า Delta/Error ที่นิวรอนชั้นเอาต์พุตและคำนวณ Delta/Error ที่นิวรอนชั้นซ่อนเร้นตามลำดับของสมการในตารางในหน้าถัดไปของขั้นตอนแพร่กลับ เพื่อนำค่า Delta/Error ที่ได้ไปใช้หาค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นซ่อนเร้นและชั้นเอาต์พุต รวมทั้งค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นอินพุตและชั้นซ่อนเร้น ตามลำดับ

ขั้นตอนแพร่กลับ :

- *คำนวณ Delta/Error ที่นิวรอนชั้นเอาต์พุต และหาค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นซ่อนเร้นและชั้นเอาต์พุตตามสมการคือ*

$$\delta_j^k = \left(d_j^k - \mathcal{S}\left(y_j^k\right)\right)\mathcal{S}'(y_j^k) \qquad j = 1, \dots, p$$

$$\Delta w_{hj}^k = \eta \delta_j^k \mathcal{S}(\mathcal{Z}_h^k) \qquad h = 0, \dots, q;\ j = 1, \dots, p$$

- *คำนวณ Delta/Error ที่นิวรอนชั้นซ่อนเร้น และหาค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นอินพุตและชั้นซ่อนเร้นตามสมการคือ*

$$\delta_h^k = \left(\sum_{j=1}^{p} \delta_j^k w_{hj}^k\right)\mathcal{S}'(\mathcal{Z}_h^k) \qquad h = 1, \dots, q$$

$$\Delta w_{ih}^k = \eta \delta_h^k x_i^k \qquad i = 0, \dots, n;\ h = 1, \dots, q$$

ค่าของ δ_1^1 ที่ได้เป็นค่า Delta/Error ของนิวรอนตัวที่ 1 ในชั้นเอาต์พุต และ δ_2^1 เป็นค่า Delta/Error ของนิวรอนตัวที่ 2 ในชั้นเอาต์พุตเช่นกัน

$$e_1^1 = d_1^1 - s_1^1 = 0.9 - 0.7164 = 0.1836$$
$$e_2^1 = d_2^1 - s_2^1 = 0.1 - 0.5238 = -0.4238$$
$$\delta_1^1 = 0.1836 \times 0.7164(1 - 0.7164) = 0.0373$$
$$\delta_2^1 = -0.4238 \times 0.5238(1 - 0.5238) = -0.1057$$

ค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นซ่อนเร้นและชั้นเอาต์พุตได้ผลลัพธ์ดังด้านล่าง ในที่นี้เพื่อไม่ให้สับสนกับพารามิเตอร์ต่าง ๆ ซึ่งอาจเหมือนกันหลังจากการแทนค่า จึงกำหนดให้ Δw_{hj}^k ในอัลกอริทึมแทนด้วย Δn_{hj}^k

$$\Delta n_{01}^1 = 1.2 \times 0.0373 \times 1.0 = 0.0447$$
$$\Delta n_{11}^1 = 1.2 \times 0.0373 \times 0.5399 = 0.0241$$
$$\Delta n_{21}^1 = 1.2 \times 0.0373 \times 0.4638 = 0.0207$$
$$\Delta n_{02}^1 = 1.2 \times -0.1057 \times 1.0 = -0.1268$$
$$\Delta n_{12}^1 = 1.2 \times -0.1057 \times 0.5399 = -0.0684$$
$$\Delta n_{22}^1 = 1.2 \times -0.1057 \times 0.4638 = -0.0588$$

ส่วนค่าของ δH_1^1 เป็นค่า Delta/Error ของนิวรอนตัวที่ 1 ในชั้นซ่อนเร้นและค่าของ δH_2^1 เป็น Delta/Error ของนิวรอนตัวที่ 2 ในชั้นซ่อนเร้นเช่นกัน สำหรับค่า Delta/Error ของนิวรอนในชั้นซ่อนเร้นนี้จริง ๆ แล้วในอัลกอริทึมคือค่าของ (δ_h^k) แต่เพื่อไม่ให้สับสนกับค่าของ Delta/Error ในชั้นเอาต์พุตที่ได้หาค่าไปก่อนหน้านี้แล้ว จึงใช้สัญลักษณ์ δH_h^k แทน สัญลักษณ์ δ_h^k

$$\delta H_1^1 = \left(0.0373 \times 0.37 + (-0.1057 \times -0.22)\right) \times 0.5399(1 - 0.5399) = 0.0092$$
$$\delta H_2^1 = \left(0.0373 \times 0.9 + (-0.1057 \times -0.12)\right) \times 0.4638(1 - 0.4638) = 0.0115$$

ค่าน้ำหนักที่เปลี่ยนแปลงไประหว่างชั้นอินพุตและชั้นซ่อนเร้นได้ผลลัพธ์ดังด้านล่าง ในที่นี้เพื่อไม่ให้สับสนกับพารามิเตอร์ต่าง ๆ ซึ่งอาจเหมือนกันหลังจากการแทนค่า จึงกำหนดให้ Δw_{ih}^k ในอัลกอริทึมแทนด้วย Δm_{ih}^k เช่นเดียวกัน

$$\Delta m_{01}^1 = 1.2 \times 0.0092 \times 1.0 = 0.011$$
$$\Delta m_{11}^1 = 1.2 \times 0.0092 \times 0.5 = 0.0055$$
$$\Delta m_{21}^1 = 1.2 \times 0.0092 \times -0.5 = -0.0055$$
$$\Delta m_{02}^1 = 1.2 \times 0.0115 \times 1.0 = 0.0138$$
$$\Delta m_{12}^1 = 1.2 \times 0.0115 \times 0.5 = 0.0069$$
$$\Delta m_{22}^1 = 1.2 \times 0.0115 \times -0.5 = -0.0069$$

ขั้นตอนถัดมา คือการปรับค่าน้ำหนักตามส่วนของอัลกอริทึมคือ

- *ปรับค่าของน้ำหนักตามสมการคือ*

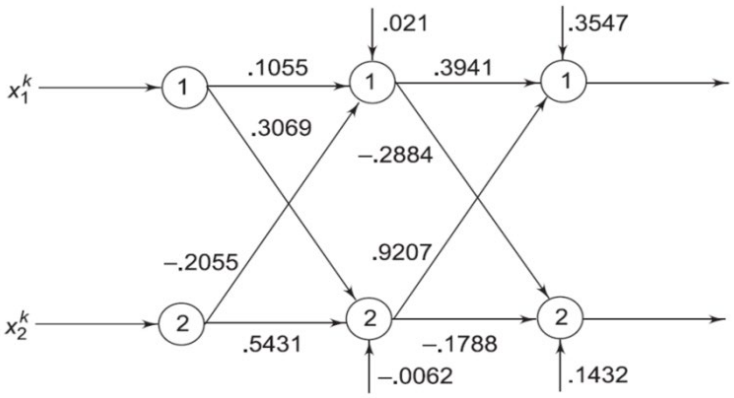
$$w_{hj}^{k+1} = w_{hj}^k + \Delta w_{hj}^k + \alpha \Delta w_{hj}^{k-1} \qquad h = 0, \ldots, q;\ j = 1, \ldots, p$$

$$w_{ih}^{k+1} = w_{ih}^k + \Delta w_{ih}^k + \alpha \Delta w_{ih}^{k-1} \qquad i = 0, \ldots, n;\ h = 1, \ldots, q$$

ค่าของน้ำหนักทั้งหมดที่ผ่านการปรับไปแล้วด้วยสมการในตารางด้านบนจะเป็นค่าน้ำหนักชุดใหม่ที่มีค่าดังแสดงด้านล่าง เมื่อ n_{hj}^2 คือค่าน้ำหนักใหม่ที่เชื่อมระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต (ซึ่งคือค่าของ w_{hj}^{k+1} ในอัลกอริทึม) ส่วน m_{ih}^2 คือค่าน้ำหนักใหม่ที่เชื่อมระหว่างอินพุตกับชั้นซ่อนเร้น (ซึ่งคือค่าของ w_{ih}^{k+1} ในอัลกอริทึม)

$$n_{01}^2 = 0.31 + 0.0447 = 0.3547 \qquad m_{01}^2 = 0.01 + 0.011 = 0.021$$
$$n_{11}^2 = 0.37 + 0.0241 = 0.3941 \qquad m_{11}^2 = 0.1 + 0.0055 = 0.1055$$
$$n_{21}^2 = 0.9 + 0.0207 = 0.9207 \qquad m_{21}^2 = -0.2 - 0.0055 = -0.2055$$
$$n_{02}^2 = 0.27 - 0.1268 = 0.1432 \qquad m_{02}^2 = -0.02 + 0.0138 = -0.0062$$
$$n_{12}^2 = -0.22 - 0.0684 = -0.2884 \qquad m_{12}^2 = 0.3 + 0.0069 = 0.3069$$
$$n_{22}^2 = -0.12 - 0.0588 = -0.1788 \qquad m_{22}^2 = 0.55 - 0.0069 = 0.5431$$

รูปที่ 2-13 แสดงค่าของน้ำหนักใหม่แต่ละค่าบนโครงข่ายหลังจบขั้นตอนของการรับข้อมูลตัวแรก นั่นคือเป็นการนำค่าของ m_{ih}^2 และ n_{hj}^2 มาใส่ในแต่ละเส้นเชื่อมที่สัมพันธ์กันกับค่าน้ำหนักของแต่ละนิวรอน



รูปที่ 2-13 ค่าน้ำหนักใหม่ทั้งหมดที่ได้ของโครงข่ายหลังรับข้อมูลตัวแรก (Kumar, 2005)

ขั้นตอนสุดท้ายของการทำงานเมื่อ $k = 1$ คือการหาค่าความผิดพลาดของ Pattern Index ที่ 1 ที่รับเข้ามาดังสมการที่ 2-6

$$\varepsilon_k = \frac{1}{2}\sum_{i=1}^{p}(d_j^k - s(y_j^k))^2 \qquad 2\text{-}6$$

จากนั้นโครงข่ายก็จะพิจารณา Pattern Index ถัดไป ซึ่งคือ ณ $k = 2$ โดยทำขั้นตอนต่าง ๆ ทั้งหมดตามลำดับเช่นเดียวกันกับที่ยกตัวอย่างสำหรับ Pattern Index ณ $k = 1$ ก่อนหน้านี้ จนจบของแต่ละ k นั่นคือ ถ้าข้อมูลมีทั้งหมด 100 ตัวก็จะทำจนจบ $k = 100$ นั่นคือใน 1 Epoch (1 รอบใหญ่) จะพิจารณาข้อมูลทุกตัวจนครบแล้วหาค่าความผิดพลาดรวม ตามสมการที่ 2-7

$$\varepsilon_{av} = \frac{1}{Q}\sum_{k=1}^{Q}\varepsilon_k < \tau \qquad 2\text{-}7$$

เมื่อ Q คือจำนวนข้อมูลทั้งหมด โดยที่ถ้าค่าความผิดพลาดรวมยังมากกว่าค่าที่ยอมรับได้ (τ) ที่ถูกกำหนดไว้ก็จะทำต่อไปครั้งละ 1 Epoch ไปเรื่อย ๆ จนกว่าค่าความผิดพลาดรวมอยู่ในย่านที่ยอมรับได้ ซึ่งค่า τ นั้นมักกำหนดเป็นค่าน้อย ๆ เช่น 0.01, 0.001 เป็นต้น

2.2.2 การวิเคราะห์เพื่อใช้งานอัลกอริทึมแบบแพร่กลับ

สิ่งสำคัญที่ต้องวิเคราะห์สำหรับการออกแบบใช้งานอัลกอริทึมแบบแพร่กลับ คือ การเลือกพารามิเตอร์ของโครงข่าย การลู่เข้า (Convergence) ของการฝึกสอน และการทำให้เป็นกรณีทั่วไป (Generalization) (อาทิตย์ ศรีแก้ว, 2558) พารามิเตอร์ของโครงข่ายถือเป็นสิ่งแรกที่ผู้ใช้โครงข่ายประสาทเทียมทุกรูปแบบต้องคำนึงถึงก่อนเป็นอันดับแรก การลู่เข้าของอัลกอริทึมการฝึกสอนเป็นส่วนที่จำเป็น เนื่องจากโครงข่ายจะไม่สามารถกล่าวได้ว่ามีการเรียนรู้ ถ้ากระบวนการฝึกสอนไม่มีแนวโน้มลู่เข้าสู่เงื่อนไขที่ต้องการ การทำให้เป็นกรณีทั่วไปถือเป็นคุณสมบัติสำคัญที่เราต้องการและคาดหวังจากโครงข่ายประสาทเทียม ด้วยการฝึกสอนด้วยชุดข้อมูลที่มีจำนวนจำกัด เราคาดหวังที่จะให้โครงข่ายเรียนรู้และสามารถทำงานกับชุดข้อมูลอื่น ๆ ที่ไม่เคยใช้ในการฝึกสอนมาก่อน ในหัวข้อต่อไปนี้จึงเป็นการนำเสนอรายละเอียดการวิเคราะห์ประสิทธิภาพของโครงข่ายที่ใช้อัลกอริทึมแบบแพร่กลับในมุมมองดังกล่าว

1) การเลือกพารามิเตอร์ของโครงข่าย

ในแต่ละงานประยุกต์ที่ต้องการนำโครงข่ายแบบหลายชั้นไปใช้งานนั้น การเลือกโครงสร้างของโครงข่ายมีผลโดยตรงต่อประสิทธิภาพที่ได้ นั่นคือต้องเลือกจำนวนนิวรอนและจำนวนชั้นซ่อนเร้นที่เหมาะสมกับแต่ละงานประยุกต์ ซึ่งโดยทั่วไปแล้วเราไม่สามารถบอกได้ว่าจำนวนดังกล่าวคืออะไร ตัวเลือกดังกล่าวย่อมขึ้นอยู่กับความซับซ้อนของงาน เช่นในงานการประมาณค่าฟังก์ชัน โครงข่ายแบบง่ายที่มีจำนวนชั้นซ่อนเร้นเพียงชั้นเดียวและมีจำนวนนิวรอนในชั้นดังกล่าวน้อยอาจจะสามารถประมาณค่าฟังก์ชันที่มีความซับซ้อนมากได้เพียงบางส่วนเท่านั้น เป็นต้น โดยเราสามารถเลือกที่จะปรับพารามิเตอร์ของโครงข่าย ซึ่งได้แก่จำนวนนิวรอนและจำนวนชั้นซ่อนเร้นเพื่อให้สามารถทำงานได้มีประสิทธิภาพเพิ่มขึ้นได้ อย่างไรก็ตามการเลือกจำนวนดังกล่าวที่มากเกินไปก็ไม่จำเป็นว่าจะให้ผลลัพธ์ที่ดีเสมอไป เนื่องมาจากโครงข่ายที่มีความซับซ้อนมากเกินไปจะทำให้การฝึกสอนยุ่งยากไปด้วย

2) การลู่เข้าของการฝึกสอน

ในโครงข่ายแบบหลายชั้นนั้น ถ้านำค่าความผิดพลาดมาแสดงเป็นพื้นผิวค่าความผิดพลาด (Error Surface) เปรียบเทียบกับค่าน้ำหนักหรือค่าไบอัส จะมีลักษณะของพื้นผิวค่าความผิดพลาดที่ซับซ้อนมาก ดังนั้นการฝึกสอนของโครงข่ายที่ต้องการให้ได้ค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าน้อยที่สุด อาจจะลู่เข้าสู่คำตอบของค่าเมทริกซ์น้ำหนักและไบอัสที่เป็นแบบวงแคบเฉพาะถิ่น (Local Maxima) ได้ ดังนั้นการฝึกสอนที่ลู่เข้าอาจจะไม่ได้หมายถึงคำตอบที่เหมาะสมที่สุดแบบวงกว้าง (Global Optima) ก็ได้ ซึ่งเราสามารถสังเกตได้จากลักษณะค่าเกรเดียน (Gradient) ของค่าความผิดพลาด นั่นคือในระหว่างการฝึกสอน ค่าเกรเดียนจะมีแนวโน้มที่ลดลง และการ

ฝึกสอนจะสิ้นสุดเมื่อค่าเกรเดียนมีค่าประมาณศูนย์ เอาต์พุตที่ได้อาจจะไม่ใช่เอาต์พุตที่ดีที่สุดก็ได้ เนื่องจากเกรเดียนอาจเป็นศูนย์ ณ จุดที่เป็นคำตอบแบบวงแคบเฉพาะถิ่น โดยทั่วไปแล้วอัลกอริทึมแบบแพร่กลับจะลู่เข้าสู่คำตอบได้เมื่อค่าคงที่การเรียนรู้มีค่าที่น้อยเพียงพอที่จะนำไปสู่คำตอบที่ต้องการได้ ค่าคงที่การเรียนรู้ที่มีค่าน้อยนั้นจะทำให้การปรับค่าน้ำหนักและไบอัสของโครงข่ายเป็นไปอย่างช้า ๆ (นั่นคือมีการเปลี่ยนค่าไปทีละน้อย) และแน่นอนว่าความเร็วในการเรียนรู้ก็จะช้าตามไปด้วย

3) การทำให้เป็นกรณีทั่วไป

โดยปกติในการฝึกสอนโครงข่ายจะใช้คู่อินพุตและเอาต์พุตเป้าหมายจากชุดข้อมูลจริงซึ่งมีจำนวนจำกัดจำนวนหนึ่ง จำนวนข้อมูลที่ใช้ฝึกสอนดังกล่าวมีความจำเป็นและแตกต่างกันออกไปตามความซับซ้อนของปัญหาหรืองานประยุกต์ ปัญหาที่มีความซับซ้อนมาก เช่น การจดจำใบหน้าคนด้วยภาพ อาจจะต้องใช้ข้อมูลภาพหน้าคนหลายหมื่นหลายแสนภาพ เนื่องมาจากภาพใบหน้าคนมีลักษณะความแตกต่างกันอย่างมากมายนับไม่ถ้วน ซึ่งในความเป็นจริงแล้ว คู่อินพุตและเอาต์พุตเป้าหมายนี้เป็นเพียงข้อมูลตัวแทนของข้อมูลที่เป็นไปได้ทั้งหมดที่ใช้ในขั้นตอนการฝึกสอนเท่านั้น ซึ่งข้อมูลที่เป็นไปได้ทั้งหมดจริง ๆ นั้นมีขนาดข้อมูลที่ใหญ่กว่าข้อมูลที่ใช้ฝึกสอนอีกมาก ดังนั้นสิ่งสำคัญอย่างหนึ่งในการใช้งานโครงข่ายด้วยอัลกอริทึมแบบแพร่กลับ คือ ความสามารถในการเป็นกรณีทั่วไปได้ นั่นคือจากการฝึกสอนคู่อินพุตและเอาต์พุตเป้าหมายด้วยจำนวนที่จำกัด โครงข่ายที่ดีต้องสามารถทำงานเป็นกรณีทั่วไปได้ จากการที่สามารถทำงานได้ครอบคลุมข้อมูลอินพุตจริงที่ไม่ใช่ตัวอย่างในชุดข้อมูลฝึกสอนได้อย่างถูกต้องได้ด้วย

2.2.3 การปรับแต่งอัลกอริทึมแบบแพร่กลับ

การประยุกต์ใช้งานอัลกอริทึมแบบแพร่กลับกับงานต่าง ๆ ในทางปฏิบัติอาจจะต้องใช้เวลาในการฝึกสอนนานมาก ได้มีงานวิจัยมากมายนำเสนอวิธีการปรับปรุงอัลกอริทึมแบบแพร่กลับให้ลู่เข้าสู่คำตอบได้เร็วขึ้น ซึ่งวิธีในการปรับปรุงดังกล่าวสามารถแบ่งได้เป็น 2 กลุ่มใหญ่ (อาทิตย์ ศรีแก้ว, 2558) คือ

1) กลยุทธแบบชาญฉลาด (Heuristic Strategy)

เป็นการปรับปรุงรายละเอียดของอัลกอริทึม เช่น วิธีปรับค่าคงที่การเรียนรู้ หรือค่าโมเมนตัม เป็นต้น หรือการใช้เทคนิคปัญญาเชิงคำนวณอย่างอื่นมาช่วยในการฝึกสอนโครงข่าย หรือใช้สำหรับการค้นหาค่าน้ำหนักและค่าไบอัสที่เหมาะสม เพื่อให้สามารถหาค่าที่เหมาะสมที่สุด (Optimal) ได้ โดยเทคนิคปัญญาเชิงคำนวณ (Computational Intelligence) ที่สามารถนำมาประยุกต์ใช้ฝึกสอนโครงข่าย เช่น จีนเนติกอัลกอริทึม (Genetic Algorithm) การค้นหาแบบตาบู (Tabu Search) อัลกอริทึมอบอ่อนจำลอง (Simulated Annealing Algorithm) เป็นต้น

2) เทคนิคเชิงตัวเลข (Numerical Method)

เป็นการปรับปรุงอัลกอริทึมในการคำนวณหาค่าที่เหมาะสมที่สุด เช่น อัลกอริทึมคอนจูเกตเกรเดียน (Conjugated Gradient Algorithm) หรืออัลกอริทึม Lavenberg-Marquardt เป็นต้น เทคนิควิธีดังกล่าวมีจุดประสงค์หลักเพื่อให้อัลกอริทึมทำการปรับค่าน้ำหนักและไบอัสได้อย่างรวดเร็วและได้ค่าน้ำหนักและไบอัสที่เหมาะสมที่สุด

การเรียนรู้แบบแพร่กลับถือเป็นอัลกอริทึมที่พัฒนาต่อยอดมาจากอัลกอริทึมแบบ LMS และสามารถใช้กับโครงข่ายแบบหลายชั้นได้อย่างมีประสิทธิภาพ ชื่อการเรียนรู้แบบแพร่กลับได้มาจากลักษณะการแพร่กลับของค่าความผิดพลาด จากชั้นเอาต์พุตของโครงข่าย กลับไปยังชั้นแรกสุดของโครงข่าย จนกระทั่งค่าความผิดพลาดมีค่าในย่านที่ต้องการ ทั้งการเรียนรู้แบบแพร่กลับและ LMS ใช้หลักการของอัลกอริทึมลงแบบชันสุด (Steepest Descent Algorithm) เพื่อลดค่าความผิดพลาดแบบกำลังสองเฉลี่ยของโครงข่าย จนกระทั่งลู่เข้าสู่คำตอบที่เหมาะที่สุด เทคนิคการปรับแต่งอัลกอริทึมแบบแพร่กลับได้ถูกนำเสนออย่างมากมาย โดยมีจุดประสงค์หลักเพื่อที่จะทำให้ความเร็วของการเรียนรู้เพิ่มขึ้น และสามารถลู่เข้าสู่คำตอบที่เหมาะที่สุดแบบวงกว้าง ทำให้โครงข่ายเพอร์เซปตรอนแบบหลายชั้นด้วยอัลกอริทึมแบบแพร่กลับกลายเป็นเครื่องมือสำคัญในการแก้ปัญหาต่าง ๆ ได้อย่างทรงประสิทธิภาพที่สุดอย่างหนึ่งในอดีต

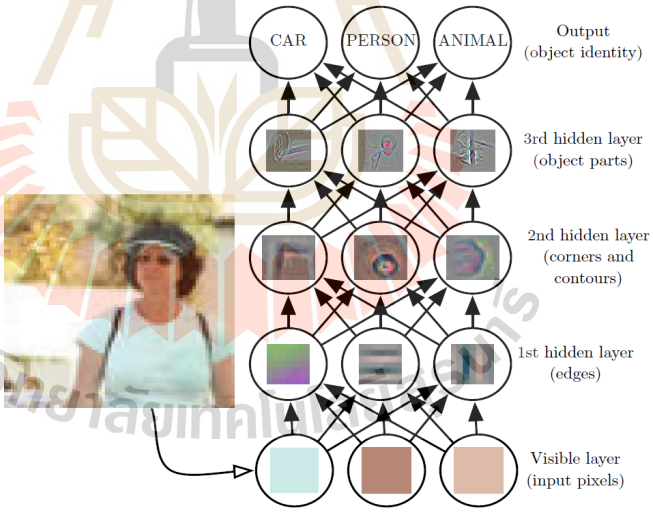
ถึงแม้ในตอนนี้จะเป็นที่ทราบกันว่า โครงข่ายเพอร์เซปตรอนแบบหลายชั้นมีข้อจำกัดในหลายด้านสำหรับการนำไปใช้ในงานประยุกต์ต่าง ๆ จากลักษณะของการเชื่อมต่อกันภายในโครงข่ายที่เป็นแบบเชื่อมถึงกันหมด (Fully Connected) แต่ในส่วนของอัลกอริทึมแบบแพร่กลับ ยังเป็นอัลกอริทึมที่ทรงพลัง และมีนำมาใช้งานต่อเนื่องอย่างกว้างขวางในขณะนี้ โดยเฉพาะในโมเดลต่าง ๆ ทางด้านการเรียนรู้เชิงลึก (Deep Learning) ซึ่งโมเดลโครงข่ายประสาทแบบคอนโวลูชันที่จะนำเสนอในหัวข้อถัดไปจากหัวข้อการเรียนรู้เชิงลึกก็เป็นโมเดลหนึ่งของการเรียนรู้เชิงลึกที่อัลกอริทึมสำหรับการเรียนรู้เป็นอัลกอริทึมแบบแพร่กลับเช่นเดียวกัน

2.3 การเรียนรู้เชิงลึก

การเรียนรู้เชิงลึก เป็นสาขาย่อยสาขาหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) ในการพยายามที่จะเรียนรู้รูปแบบทางนามธรรมในระดับสูง (High-Level Abstractions) จากข้อมูลโดยการใช้สถาปัตยกรรมแบบลำดับชั้น (Hierarchical Architectures) (Guo el al., 2016) ซึ่งพื้นฐานของการเรียนรู้เชิงลึกคือ ความพยายามที่จะสร้างแบบจำลองเพื่อสร้างตัวแทนข้อมูล (Data Representation) ในระดับสูงโดยการสร้างสถาปัตยกรรมข้อมูลขึ้นมาที่ประกอบไปด้วยโครงสร้างย่อย ๆ หลายชั้น และแต่ละชั้นนั้นได้มาจากการแปลงแบบไม่เป็นเชิงเส้น (Deng and Yu, 2014) โดย

ข้อมูลอินพุตในแต่ละชั้นได้มาจากการปฏิสัมพันธ์กับชั้นอื่น ๆ นั่นคือการเรียนรู้เชิงลึกพยายามหาความสัมพันธ์เชิงลึกจากข้อมูล โดยพยายามเรียนรู้เพื่อหารูปแบบการแทนข้อมูลอย่างมีประสิทธิภาพ เมื่อสถาปัตยกรรมแบบลำดับชั้นที่ใช้มีจำนวนชั้นและหน่วยประมวลผล (Processing Units) ที่อยู่ในแต่ละชั้นมากขึ้น คุณลักษณะ (Features) ที่ใช้แทนข้อมูลในชั้นสูง ๆ ก็จะยิ่งซับซ้อนหรือยิ่งเป็นนามธรรม (Abstracts) มากยิ่งขึ้น

ตัวอย่างของแนวคิดการเรียนรู้เชิงลึกแสดงดังรูปที่ 2-14 เมื่อรูปภาพภาพหนึ่งที่สามารถแทนได้ด้วยค่าของความสว่างของแต่ละจุดภาพ (Pixel) เมื่อใช้เป็นอินพุตเข้าสู่โมเดลการเรียนรู้เชิงลึก ชั้นแรกของโมเดลอาจมองส่วนที่เป็นเซ็ตของขอบ (Edges) ของวัตถุต่าง ๆ ในภาพ มองในระดับสูงขึ้นด้วยชั้นถัดไปด้วยเซ็ตของมุมและเส้นแสดงรูปร่าง (Corners and Contours) ตามด้วยการมองเป็นเซ็ตของส่วนต่าง ๆ ของวัตถุ (Object Parts) ในชั้นสุดท้าย แล้วโมเดลก็สามารถบอกได้ว่าภาพนั้นคือวัตถุอะไร ซึ่งการเรียนรู้เชิงลึกถือว่าเป็นวิธีการที่มีศักยภาพสูงในการจัดการกับคุณลักษณะที่ใช้แทนข้อมูล โดยสามารถนำไปใช้ได้กับการเรียนรู้ทั้งแบบมีผู้ฝึกสอน แบบไม่มีผู้สอน และแบบกึ่งมีผู้ฝึกสอน

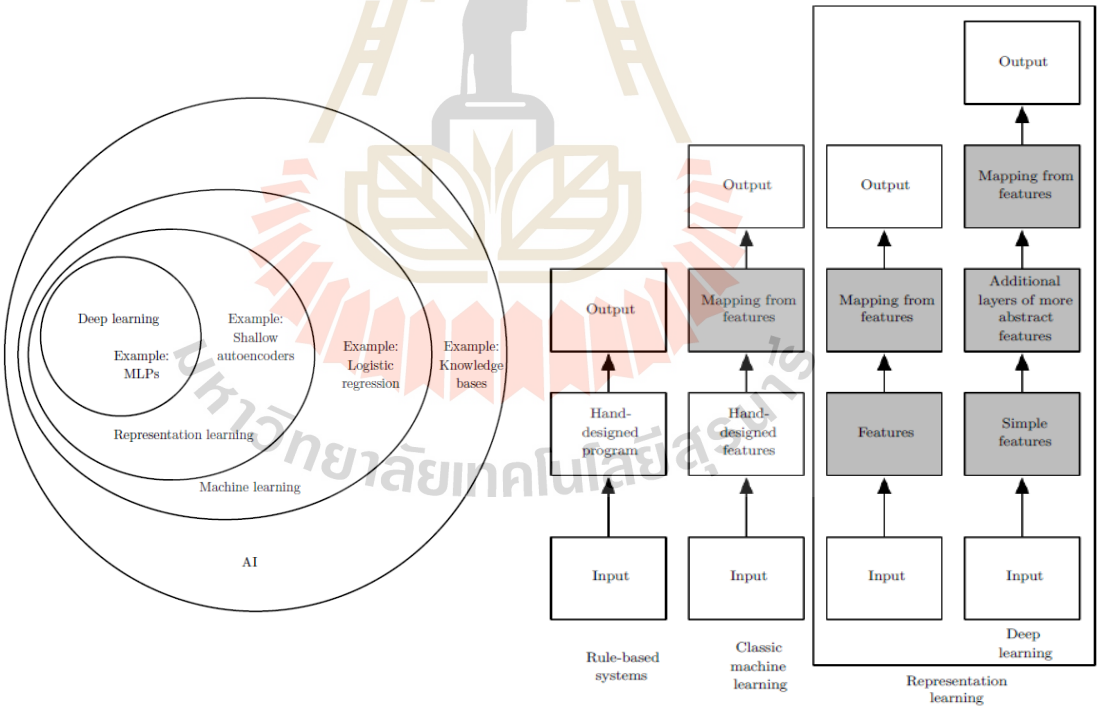


รูปที่ 2-14 แนวคิดของโมเดลการเรียนรู้เชิงลึก (Goodfellow et al., 2016)

งานวิจัยจำนวนมากได้มีการนำเสนอสถาปัตยกรรมการเรียนรู้หลายรูปแบบบนหลักการของการเรียนรู้เชิงลึกนี้ เช่น โครงข่ายประสาทเทียมเชิงลึก (Deep Artificial Neural Network) โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) โครงข่ายความเชื่อเชิงลึก (Deep Belief Network) และโครงข่ายประสาทแบบวนซ้ำ (Recurrent Neural Network) ซึ่งทุก

สถาปัตยกรรมมีการนำมาใช้งานอย่างแพร่หลายในทางการมองเห็นของเครื่อง (Computer Vision) การรู้จำเสียงพูด (Speech Recognition) การประมวลผลภาษาธรรมชาติ (Natural Language Processing) และชีวสารสนเทศศาสตร์ (Bioinformatics)

รูปที่ 2-15 เป็นแผนภาพแสดงความสัมพันธ์ระหว่างเทคนิคการเรียนรู้เชิงลึกกับเทคนิคอื่น ๆ ที่เกี่ยวข้องกัน ซึ่งทั้งหมดจัดว่าเป็นเทคนิคเกี่ยวกับความฉลาดเทียม (Artificial Intelligence) จากรูปที่ 2-15(a) จะเห็นว่าการเรียนรู้เชิงลึกจัดเป็นเทคนิคทางด้านการเรียนรู้ของเครื่องที่อยู่ในกลุ่มของการเรียนรู้เพื่อสร้างตัวแทน (Representation Learning) โดยแนวคิดของการเรียนรู้เพื่อสร้างตัวแทนด้วยการเรียนรู้เชิงลึกนั้นเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ นำเสนอดังรูปที่ 2-15(b) นั่นคือ การเรียนรู้เชิงลึกใช้ลักษณะการเรียนรู้เพื่อหาคุณลักษณะที่จะใช้แทนข้อมูลโดยอัตโนมัติจากข้อมูล โดยที่คุณลักษณะดังกล่าวไม่ต้องมีการกำหนดหรือออกแบบด้วยมือ (Handed-Designed) มาก่อนให้กับโมเดล แต่เทคนิคทางด้านการเรียนรู้ของเครื่องหรือความฉลาดเทียมแบบดั้งเดิมอื่น ๆ จำเป็นต้องมีการกำหนดคุณลักษณะที่ได้จากออกแบบด้วยมือดังกล่าว



(a) (b)

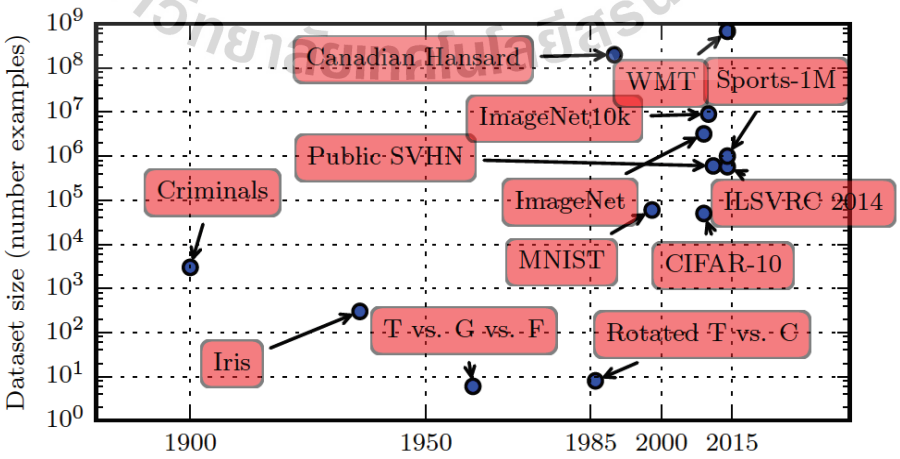
รูปที่ 2-15 แผนภาพแสดงความสัมพันธ์ระหว่างเทคนิคการเรียนรู้เชิงลึกกับเทคนิคอื่นที่เกี่ยวข้องกัน (Goodfellow et al., 2016)

การเรียนรู้เชิงลึกเป็นที่สนใจอย่างแพร่หลายในทุกวันนี้นับตั้งแต่ปี 2006 ที่ Hinton และคณะ (2006) ได้นำเสนอโครงข่ายที่เรียกว่า โครงข่ายความเชื่อเชิงลึก ที่สามารถเรียนรู้ได้อย่างมีประสิทธิภาพ ทั้งนี้เหตุผลที่การเรียนรู้เชิงลึกเป็นที่สนใจและมีการพัฒนาไปอย่างรวดเร็วนั้นเริ่มต้นมาจาก 3 ประเด็นหลักคือ (Ilango, 2017)

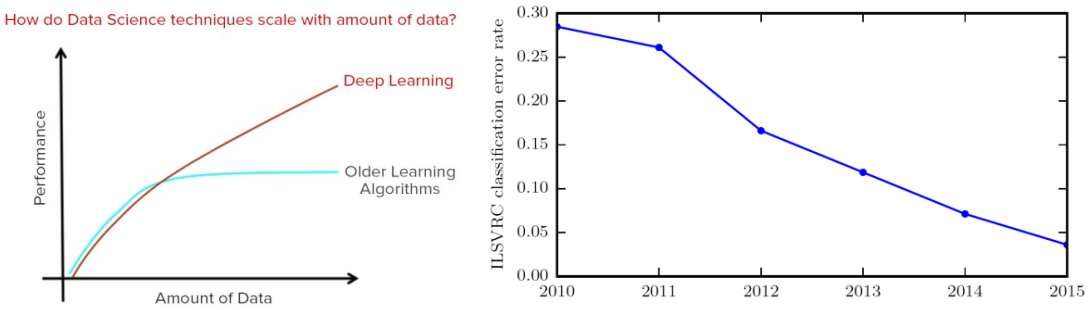
1. ปริมาณข้อมูลในแต่ละงานประยุกต์มีเป็นจำนวนที่สูงขึ้นมาก
2. ศักยภาพในการคำนวณของเครื่องสูงขึ้นมากด้วยความเร็วของ CPU และจากการใช้ GPUs/FPGAs
3. มีแนวคิดเดิมเกี่ยวกับโครงข่ายประสาทซึ่งจัดว่าเป็นเครื่องมือที่สำคัญที่สุด

รูปที่ 2-16 แสดงขนาดของแต่ละชุดข้อมูลในแต่ละช่วงเวลาตั้งแต่ปี 1990 ถึงปี 2015 ชุดข้อมูลเหล่านี้ที่เป็นที่รู้จักและมักจะนำมาใช้ในการทดสอบโมเดลทางด้านการเรียนรู้ของเครื่องรวมถึงการเรียนรู้เชิงลึก โดยส่วนใหญ่ขนาดของชุดข้อมูลดังกล่าวเพิ่มขึ้นตามลำดับเวลาที่เพิ่มขึ้น

จากปริมาณข้อมูลที่มีจำนวนมากขึ้นเป็นอย่างมากในแต่ละชุดข้อมูลของงานประยุกต์ต่าง ๆ เมื่อนำเทคนิคดั้งเดิมทางด้านวิทยาการข้อมูล (Data Science) ซึ่งหมายรวมถึงเทคนิคทางการเรียนรู้ของเครื่องแบบดั้งเดิมมาใช้กับชุดข้อมูลดังกล่าว แนวโน้มของค่าความถูกต้องที่ได้พบว่าไม่สามารถทำให้สูงขึ้นได้มากนักเมื่อเทียบกับโมเดลทางด้านการเรียนรู้เชิงลึกที่ได้ค่าความถูกต้องที่สูงขึ้นโดยตลอดอย่างชัดเจนดังแสดงในรูปที่ 2-17(a) ส่วนรูปที่ 2-17(b) แสดงอัตราความผิดพลาดที่ลดลงจากการนำเทคนิคการเรียนรู้เชิงลึกมาใช้ในงาน ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) ซึ่งจัดว่าเป็นการประสบความสำเร็จที่รู้กันอย่างแพร่หลายในแวดวงกลุ่มนักวิจัยทางด้านการเรียนรู้ของเครื่อง



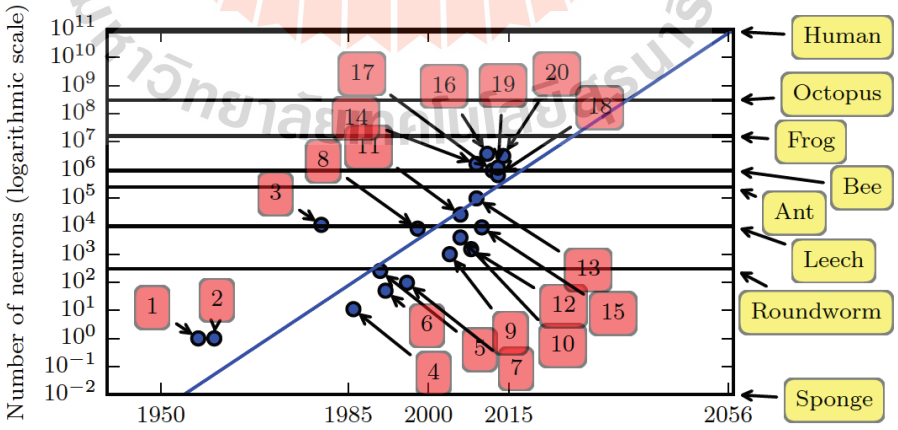
รูปที่ 2-16 ขนาดของชุดข้อมูลที่เพิ่มสูงขึ้นตลอดเวลา (Goodfellow et al., 2016)



(a) (Ilango, 2017) (b) (Goodfellow et al., 2016)

รูปที่ 2-17 ประสิทธิภาพที่ได้จากเทคนิคการเรียนรู้เชิงลึก

นอกจากนี้ผลสืบเนื่องจากที่ชุดข้อมูลมีขนาดใหญ่ขึ้นดังกล่าว พบว่าโมเดลที่นำมาใช้จะต้องมีความซับซ้อนตามไปด้วย เมื่อพิจารณาในแง่ของขนาดโมเดล (Model Sizes) ที่ต้องใช้ในงานประยุกต์ต่าง ๆ รูปที่ 2-18 แสดงวิวัฒนาการของจำนวนนิวรอนทั้งหมดที่ใช้ในชั้นซ่อนเร้นนับตั้งแต่เริ่มนำชั้นซ่อนเร้นมาใช้ในโครงข่ายประสาทเทียม ซึ่งพบว่าจำนวนนิวรอนที่ใช้จะเพิ่มขึ้นเป็นสองเท่าทุก ๆ 2.4 ปีโดยประมาณ (Goodfellow et al., 2016) จากกราฟในรูปแสดงเปรียบเทียบกับจำนวนนิวรอนในสิ่งมีชีวิตต่าง ๆ โดยถ้าแนวโน้มยังเป็นแบบเดิมต่อไปเช่นนี้ ประมาณปี 2056 จำนวนนิวรอนทั้งหมดที่ใช้ในชั้นซ่อนเร้นก็จะเท่ากันกับจำนวนนิวรอนของสมองมนุษย์ นอกจากว่าจะมีเทคโนโลยีใหม่ที่ทำให้อัตราการเพิ่มเร็วขึ้นกว่านี้



รูปที่ 2-18 วิวัฒนาการของจำนวนนิวรอนทั้งหมดที่ใช้ในชั้นซ่อนเร้นนับตั้งแต่เริ่มนำชั้นซ่อนเร้นมาใช้ในโครงข่ายประสาทเทียม (Goodfellow et al., 2016)

สำหรับทั้ง 20 งานประยุกต์ที่แสดงในรูปที่ 2-18 นั้นเป็นการนำโมเดลต่าง ๆ มาใช้โดยแสดงตามลำดับอ้างอิงดังค่าตัวเลขในภาพดังต่อไปนี้

1. Perceptron (Rosenblatt, 1958, 1962)
2. Adaptive linear element (Widrow and Hoff, 1960)
3. Neocognitron (Fukushima, 1980)
4. Early back-propagation network (Rumelhart et al., 1986b)
5. Recurrent neural network for speech recognition (Robinson and Fallside, 1991)
6. Multilayer perceptron for speech recognition (Bengio et al., 1991)
7. Mean field sigmoid belief network (Saul et al., 1996)
8. LeNet-5 (LeCun et al., 1998b)
9. Echo state network (Jaeger and Haas, 2004)
10. Deep belief network (Hinton et al., 2006)
11. GPU-accelerated convolutional network (Chellapilla et al., 2006)
12. Deep Boltzmann machine (Salakhutdinov and Hinton, 2009a)
13. GPU-accelerated deep belief network (Raina et al., 2009)
14. Unsupervised convolutional network (Jarrett et al., 2009)
15. GPU-accelerated multilayer perceptron (Ciresan et al., 2010)
16. MP-1 network (Coates and Ng, 2011)
17. Distributed autoencoder (Le et al., 2012)
18. Multi-GPU convolutional network (Krizhevsky et al., 2012)
19. COTS HPC unsupervised convolutional network (Coates et al., 2013)
20. GoogLeNet (Szegedy et al., 2014a)

โดยสรุปแล้ว นับตั้งแต่มีแนวคิดเกี่ยวกับการเรียนรู้เชิงลึก โมเดลดังต่อไปนี้จัดว่าเป็นโมเดลหรืออัลกอริทึมทางด้านการเรียนรู้เชิงลึกที่ประสบความสำเร็จในงานประยุกต์หลัก ๆ (Ilango, 2017) นั่นคือ

1. Convolutional Neural Networks (CNN) – ใช้สำหรับงานการรู้จำวัตถุ และการจำแนกรูปภาพ ซึ่งเหมาะที่จะใช้กับข้อมูลที่เป็นรูปภาพมากที่สุด
2. Recurrent Neural Networks (RNN) – ใช้สำหรับการจำแนกข้อความ (Text Classification) และปัญหาเกี่ยวกับการสร้างข้อความ (Text Generation problems) ซึ่งเหมาะที่จะใช้กับข้อมูลที่เป็นลำดับ (Sequences)

3. Long-Short Term Memory (LSTM) – ใช้สำหรับปัญหาที่ขึ้นต่อกันแบบระยะยาว (Long-Term Dependency Problems)
4. Generative Adversarial Networks (GAN) – เป็นการเรียนรู้แบบไม่มีผู้ฝึกสอน
5. Gated Recurrent Units (GRU) – เป็น RNNs ที่มีหน่วย Gating
6. Deep Belief Networks (DBN) – เป็นโครงข่ายประสาทเทียมแบบเชิงลึกที่มีจำนวนชั้นหลาย ๆ ชั้น
7. Stacked Auto-Encoders – ใช้ Sparse Autoencoders มาต่อด้วยกันหลาย ๆ ชั้น
8. Restricted Boltzmann Machine (RBM) – เป็นการเรียนรู้แบบไม่มีผู้ฝึกสอนที่เรียนรู้การแจกแจงความน่าจะเป็น (Probability Distribution) จากกลุ่มของอินพุต
9. Deep Reinforcement Learning (RL) – เป็นโมเดลที่กำลังเป็นที่สนใจมาก ณ เวลานี้
10. Online Learning – ข้อมูลเข้ามาเป็นลำดับเวลาเพื่อทำนายข้อมูล ณ เวลาถัดไป

2.4 การเรียนรู้เชิงลึกของโครงข่ายประสาทแบบคอนโวลูชัน

โมเดลการเรียนรู้เชิงลึกในชื่อ โครงข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Networks) หรือ ConvNets หรือ CNNs นั้นเป็นโครงข่ายประสาทแบบเฉพาะชนิดหนึ่งที่ใช้เพื่อการประมวลผลข้อมูลที่มีโครงสร้างแบบกริด (Grid-like Topology) เช่นถ้าข้อมูลเป็นอนุกรมเวลามองว่าเป็นแบบกริดหนึ่งมิติ (1D) ของข้อมูลแต่ละช่วงเวลา แต่ถ้าข้อมูลเป็นภาพก็มองเป็นกริดแบบสองมิติ (2D) ของข้อมูลแต่ละพิกเซล ที่มาของชื่อ “โครงข่ายประสาทแบบคอนโวลูชัน” เป็นการบ่งบอกว่าโครงข่ายใช้การดำเนินการทางคณิตศาสตร์แบบ “คอนโวลูชัน” นั่นคือ โครงข่ายแบบคอนโวลูชันนั้นเป็นโครงข่ายประสาทอย่างง่ายที่มีการใช้การคอนโวลูชันแทนที่การคูณเมทริกซ์ในโครงข่ายประสาทแบบดั้งเดิมอย่างน้อยหนึ่งชั้นของจำนวนชั้นทั้งหมดในโครงข่าย (Goodfellow et al., 2016) โครงข่ายประสาทแบบคอนโวลูชันได้รับความสนใจและเป็นที่รู้จักอย่างกว้างขวางมากในปัจจุบันนับตั้งแต่ประสบความสำเร็จจากการใช้ในงาน ILSVRC เมื่อปี 2012 (Krizhevsky et al., 2012) โดยเฉพาะในงานประยุกต์เกี่ยวกับ ข้อมูลภาพ เสียง ข้อความหรือวิดีโอ

เนื่องจากโครงข่ายประสาทแบบคอนโวลูชันจัดเป็นโมเดลหนึ่งในกลุ่มของโมเดลการเรียนรู้เชิงลึกที่โดยส่วนใหญ่นำมาประยุกต์ใช้กับข้อมูลรูปภาพ ในเบื้องต้นสำหรับหัวข้อนี้จึงเริ่มต้นจากการนำเสนอเกี่ยวกับพื้นฐานข้อมูลภาพและการคอนโวลูชัน เพราะมีการนำการคอนโวลูชันมาใช้ในส่วนหลักของโครงข่ายและเป็นที่มาของชื่อโครงข่ายแบบคอนโวลูชัน จากนั้นจึงเป็นการนำเสนอแนวคิดสำคัญและองค์ประกอบหลักของโครงข่ายประสาทแบบคอนโวลูชัน และรายละเอียดในแต่ละองค์ประกอบหลักของโครงข่าย ตามลำดับ

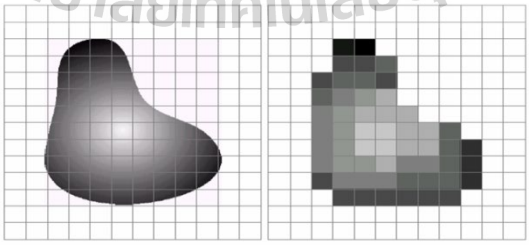
2.4.1 พื้นฐานข้อมูลรูปภาพและการคอนโวลูชัน

1) พื้นฐานข้อมูลรูปภาพ

การแทนภาพดิจิทัล (Digital Image Representation)

รูปภาพสามารถนิยามได้ในรูปของฟังก์ชันในสองมิติ $f(x,y)$ เมื่อ x และ y เป็นค่าระยะทางในแนวแกน x และแกน y และค่า Amplitude ของ f ณ แต่ละพิกัด (x,y) เรียกว่า Intensity ของภาพ ณ ตำแหน่งนั้น ๆ จากลักษณะของพิกัด (x,y) และค่าของ Amplitude แล้วรูปภาพถือว่ามีลักษณะที่ต่อเนื่อง พิจารณาการแปลงรูปภาพที่ต่อเนื่องไปเป็นแบบดิจิทัลต้องทำการดิจิไทซ์ (Digitized) ทั้งค่าของพิกัดจุด (Coordinates) และค่าของ Amplitude การดิจิไทซ์ค่าของพิกัดจุดนั้นเรียกว่าการ Sampling ส่วนการ ดิจิไทซ์ ค่าของ Amplitude เรียกว่าการ Quantization ดังนั้นเมื่อค่า x,y และค่า Amplitude ของ f มีขนาดที่จำกัด นั่นคือเป็นค่าแบบดิสครีต เราจึงเรียกรูปภาพที่ผ่านการดิจิไทซ์แล้วว่าภาพดิจิทัล (Digital Image) ดังตัวอย่างในรูปที่ 2-19(b) ซึ่งได้มาจากการ Sampling และการ Quantization ภาพที่ต่อเนื่องในรูปที่ 2-19(a)

ผลลัพธ์ที่ได้จากการ Sampling และ Quantization รูปภาพนั้นจะเป็นเมทริกซ์ของตัวเลขสมมุติรูปภาพ $f(x,y)$ ผ่านการ Sampling มาแล้วมีขนาด M แถวและ N คอลัมน์ นั่นคือเป็นรูปภาพที่มีขนาด $M \times N$ โดยค่าของพิกัดจุด (x,y) เป็นค่าแบบดิสครีต ในการประมวลผลภาพดิจิทัลโดยทั่วไปนั้น กำหนดให้พิกัดเริ่มต้นของรูปภาพอยู่ที่ตำแหน่ง $(x,y) = (0,0)$ ดังนั้นช่วงของค่าพิกัดในแกน x จึงอยู่ที่ 0 ถึง $M-1$ และ ช่วงของค่าพิกัดในแกน y อยู่ที่ 0 ถึง $N-1$ ดังรูปที่ 2-20(a) แต่รูปแบบของพิกัดจุดที่ใช้ในซอฟต์แวร์บางชนิดอาจแตกต่างกันออกไปคือ มักจะใช้การอ้างอิงตำแหน่งในเมทริกซ์ ในแนวแถว (r) และแนวคอลัมน์ (c) จึงอ้างอิงค่าในตำแหน่งเริ่มต้นเป็น 1 แทน ดังแสดงในรูปที่ 2-20(b) เมื่อค่าในแต่ละพิกัดจุดคือค่าในหนึ่งจุดภาพ (One Pixel)

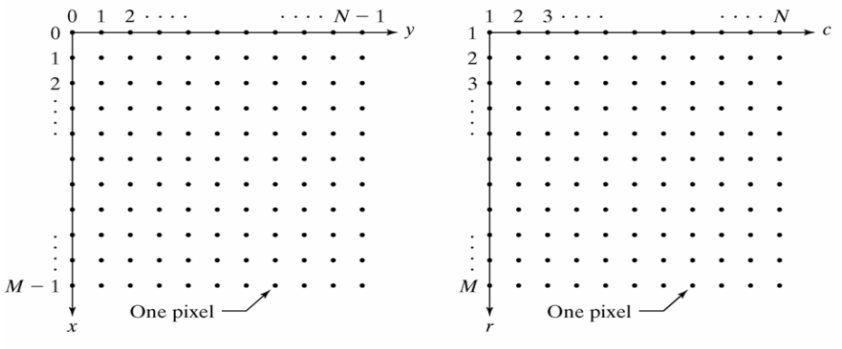


(a) (b)

รูปที่ 2-19 การแทนภาพดิจิทัล (Gonzalez et al., 2004)

(a) ภาพที่มีลักษณะต่อเนื่อง

(b) ภาพดิจิทัลที่ได้จากการ Sampling และการ Quantization



(a) (b)

รูปที่ 2-20 รูปแบบของพิกัดจุดในภาพ (Gonzalez et al., 2004)

(a) รูปแบบของพิกัดที่ใช้ในการประมวลผลภาพดิจิทัลโดยทั่วไป

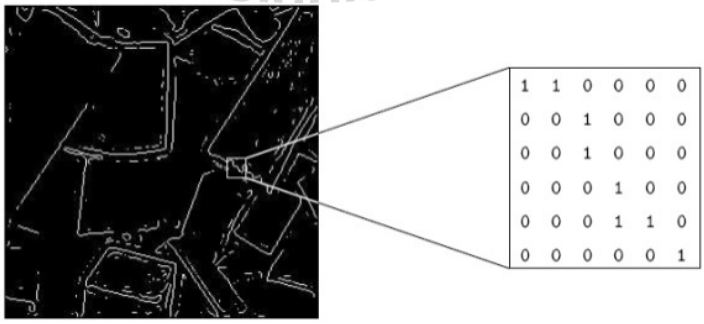
(b) รูปแบบของพิกัดที่ใช้ในบางซอฟต์แวร์

ชนิดของภาพดิจิทัล (Digital Image Types)

เมื่อพิจารณาจากสีของภาพ สามารถแบ่งชนิดของภาพดิจิทัลออกเป็น 4 ชนิด ด้วยกันคือ ภาพขาวดำ ภาพระดับเทา ภาพสีแบบ RGB และภาพสีแบบดัชนี (Indexed)

(1) ภาพขาวดำ

เป็นภาพที่ข้อมูลในภาพมีเพียงสองสีเท่านั้นคือ สีขาวและสีดำ ดังนั้นการจัดเก็บข้อมูลดังกล่าวในระบบดิจิทัลจึงใช้ค่าเพียงสองค่าคือ ค่าที่เป็น 0 และค่าที่เป็น 1 จากรูปที่ 2-21 ถ้าขยายเฉพาะบริเวณที่เป็นกรอบสี่เหลี่ยมเล็กในภาพ จะเห็นว่าค่าของข้อมูลในแต่ละจุดภาพจะมีค่าเป็น 0 และ 1 โดยจุดภาพสีขาวคือค่าที่เป็น 1 และจุดภาพสีดำคือค่าที่เป็น 0



รูปที่ 2-21 ตัวอย่างข้อมูลภาพขาวดำ (McAndrew, 2004)

(2) ภาพระดับเทา

ข้อมูลของภาพระดับเทาเกิดจากการไล่เฉดของสีเทา มองเสมือนการนำสีดำและสีขาวมาผสมกัน เกิดเป็นเฉดสีเทาที่แตกต่างกันทั้งหมด 256 เฉดสีหรือ 256 ค่า นั่นคือมีค่าตั้งแต่ 0–255 โดยค่าที่ใกล้ 0 จะเป็นเฉดสีแบบเทาเข้ม ส่วนค่าใกล้ 255 จะเป็นเฉดสีเทาอ่อน โดยค่า 0 จะเป็นสีดำ ส่วนค่า 255 เป็นสีขาว จากรูปที่ 2-22 ที่เป็นตัวอย่างข้อมูลภาพระดับเทา ถ้าขยายเฉพาะบริเวณที่เป็นกรอบสี่เหลี่ยมเล็กในภาพ จะเห็นว่าค่าของข้อมูลในแต่ละจุดภาพจะมีค่าตั้งแต่ 0–255 ซึ่งค่าที่อยู่ระหว่างกลางของช่วง เช่น 129 ก็จะเป็นเฉดสีเทาแบบกลาง ๆ

รูปที่ 2-22 ตัวอย่างข้อมูลภาพระดับเทา (McAndrew, 2004)

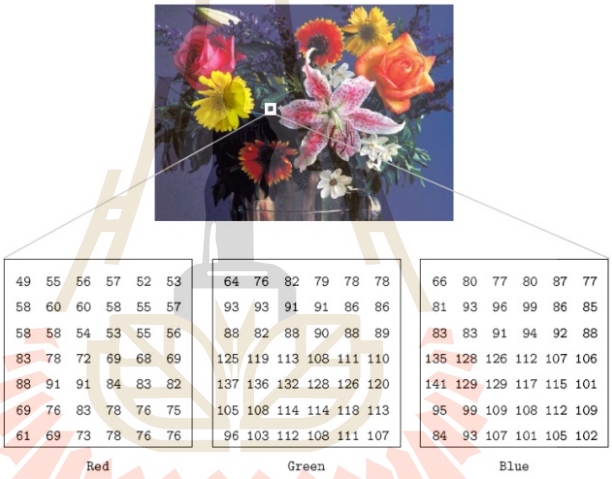
(3) ภาพสีแบบ RGB

การจัดเก็บข้อมูลภาพสีในระบบดิจิทัลนั้นมีความซับซ้อนกว่าภาพขาวดำและภาพระดับเทา ทำให้ใช้พื้นที่สำหรับจัดเก็บข้อมูลภาพมากกว่า การจัดเก็บข้อมูลภาพสีนั้นเป็นได้หลายรูปแบบขึ้นอยู่กับว่าจะอ้างอิงด้วยโมเดลสีแบบใด เช่น โมเดลสีแบบ RGB แบบ HSV หรือแบบ CMYK เป็นต้น ทั้งนี้จะมีสูตรที่สามารถแปลงจากโมเดลสีแบบหนึ่งไปยังโมเดลสีอีกแบบหนึ่งได้ ซึ่งโดยส่วนใหญ่มักจะใช้โมเดลแบบ RGB ที่ในหนึ่งจุดภาพจะประกอบด้วยค่าในสามช่องสี (Channel) หรือสามแกนสี คือแกนสีแดง (Red) แกนสีเขียว (Green) และแกนสีน้ำเงิน (Blue) ดังรูปที่ 2-23 ซึ่งถ้าขยายเฉพาะบริเวณกรอบสี่เหลี่ยมเล็ก ๆ ในภาพที่มีจำนวนจุดภาพในแนวกว้าง (แนวคอลัมน์) 6 จุดภาพ และมีจำนวนจุดภาพในแนวยาว (แนวแถว) 7 จุดภาพ นั่นคือบริเวณกรอบสี่เหลี่ยมนี้มีจำนวนจุดภาพทั้งหมดเป็น 6×7 จากรูปแสดงให้เห็นว่าแต่ละจุดภาพประกอบด้วยค่าในสามแกนสีแทนด้วยค่าตัวเลขในสามกรอบสี่เหลี่ยมด้านล่างของภาพ โดยจุดภาพแรกที่อยู่ตรงมุมด้านบนซ้ายสุดของกรอบสี่เหลี่ยมมีค่าในแกนสีแดงเป็น 49 มีค่าในแกนสีเขียวเป็น 64 และมีค่าในแกนสีน้ำเงินเป็น 66 ตามลำดับ ซึ่งค่าที่เป็นไปได้ในแต่ละแกนสีมีได้ตั้งแต่ 0–255 จากลักษณะของ

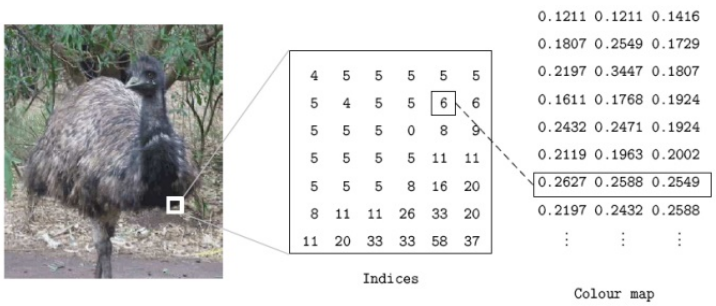
การจัดเก็บข้อมูลภาพสีเช่นนี้ประกอบเข้าด้วยกันกับกลไกของฮาร์ดแวร์ทำให้เราสามารถมองเห็นภาพที่เป็นสีทางจอภาพได้

(4) ภาพสีแบบดัชนี (Indexed)

การจัดเก็บข้อมูลภาพสีอีกรูปแบบหนึ่งที่นิยมใช้คือภาพสีแบบดัชนีที่จะไม่ใช้วิธีจัดเก็บค่าในสามแกนสีโดยตรงเช่นในภาพสีแบบ RGB แต่ใช้วิธีแบบจัดเก็บด้วยค่าของดัชนีแล้วใช้ค่าดัชนีนั้นไปอ้างอิงค่าในแต่ละแกนสีจากตารางที่จัดเก็บไว้ต่างหาก ดังแสดงในรูปที่ 2-24 ซึ่งส่วนของ Indices ในรูปคือค่าของดัชนี และส่วนของตารางที่ใช้ในการจัดเก็บค่าในแต่ละแกนสีของทุก ๆ ดัชนีเรียกว่าตารางแผนที่ค่าสี นั่นคือส่วนของ Colour Map ในรูป ซึ่งค่าในแต่ละแกนสีในตารางแผนที่ค่าสีนั้นมีค่าตั้งแต่ 0–1



รูปที่ 2-23 ตัวอย่างข้อมูลภาพสีแบบ RGB ที่แต่ละจุดภาพประกอบด้วยค่าในสามแกนสี (McAndrew, 2004)



รูปที่ 2-24 ตัวอย่างข้อมูลภาพสีแบบ Indexed (McAndrew, 2004)

2) การคอนโวลูชัน

การคอนโวลูชันจัดว่าเป็นการดำเนินการที่สำคัญมากอย่างหนึ่งในงานการประมวลผลสัญญาณ (Signal Processing) และการประมวลผลภาพ (Image Processing) ซึ่งสามารถดำเนินการได้ทั้งในแบบ 1 มิติ (เช่น การใช้กับข้อมูลเสียงพูด) 2 มิติ (เช่น การใช้กับข้อมูลภาพ) หรือ 3 มิติ (เช่น การใช้กับข้อมูลวีดีโอ) การคอนโวลูชันเป็นกลไกที่ใช้สำหรับการรวม (Combine) หรือการผสม (Blend) ฟังก์ชันสองฟังก์ชันเข้าด้วยกันแบบเกาะกันเป็นก้อน (Coherent) (Weisstein, 2018) โดยเมื่อพิจารณาในขอบเขต (Domain) แบบดิสครีตของการคอนโวลูชันภายใต้หนึ่งตัวแปร (การคอนโวลูชันใน 1 มิติ) กำหนดโดยสมการที่ 2-8 (Saxena, 2016)

$$y[n] = x[n] * h[n] = \sum_k x[k].h[n-k], \qquad k \in [-\infty, +\infty] \tag{2-8}$$

เมื่อ $x[n]$ เป็นฟังก์ชันของอินพุต $h[n]$ เป็นอีกฟังก์ชันหนึ่งที่ต้องการนำมาคอนโวลูชันกับอินพุต เช่นมองภาพว่าเป็นฟังก์ชันของตัวกรอง (Kernel Function) และ $y[n]$ เป็นเอาต์พุตที่ได้จากการคอนโวลูชัน เมื่อสัญลักษณ์ “$*$” หมายถึงการดำเนินการคอนโวลูชัน และสัญลักษณ์ “.” หมายถึงผลคูณต่อจุด (Pointwise Product)

สำหรับขอบเขตแบบดิสครีตของการคอนโวลูชันภายใต้สองตัวแปร (การคอนโวลูชันใน 2 มิติ) กำหนดโดยสมการที่ 2-9 นั่นคือเป็นการรวม (Summation) ผลคูณต่อจุดของค่าในฟังก์ชันตามลักษณะของการเข้าถึง (Traversal) ข้อมูลใน 2 มิติ (Saxena, 2016) ในแนวแกน x และแนวแกน y

$$(f * g)(x, y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u, v).g(x-u, y-v) \tag{2-9}$$

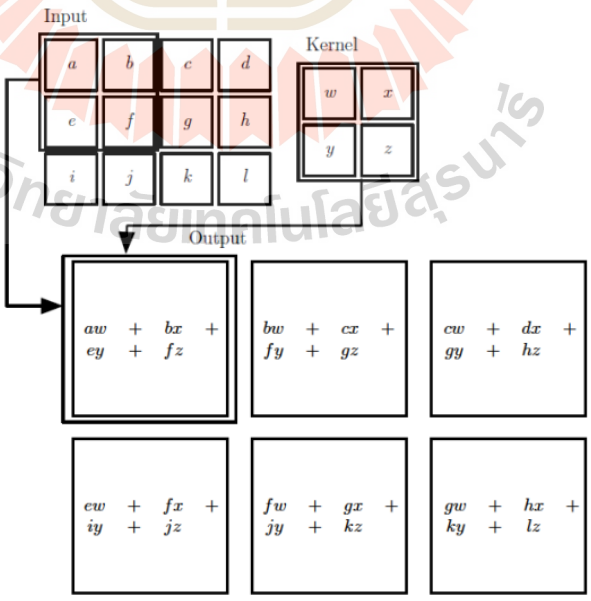
เมื่อ f เป็นฟังก์ชันของอินพุตและ g เป็นอีกฟังก์ชันหนึ่งที่ต้องการนำมาคอนโวลูชันกับอินพุต ที่อาจมองภาพว่าเป็นฟังก์ชันของตัวกรองเช่นเดียวกัน

สมการที่ 2-8 และสมการที่ 2-9 นั้นเป็นสมการที่เป็นนิยามของการดำเนินการคอนโวลูชันซึ่งใช้อ้างอิงในงานการประมวลผลสัญญาณดิจิทัลและการประมวลผลภาพดิจิทัล แต่การคอนโวลูชันที่นำมาใช้ในโครงข่ายประสาทแบบคอนโวลูชันที่เป็นการดำเนินการบนข้อมูลภาพนั้น

จริง ๆ แล้วไม่ได้เป็นการคอนโวลูชันโดยนิยามที่แท้จริง แต่เป็นการดำเนินการแบบสหสัมพันธ์ข้าม (Cross-correlation) (Weisstein, 2018) ที่นิยามโดยสมการที่ 2-10 (Saxena, 2016) สำหรับการคอนโวลูชันใน 2 มิติ ซึ่งเป็นลักษณะของการมองค่าฟังก์ชันตัวหนึ่งแบบกลับด้าน (Flip) เป็นตรงกันข้าม นั่นคือสมการที่ 2-10 เป็นการกลับด้านในค่าของฟังก์ชัน g ที่อาจมองภาพว่ากลับด้านของค่าในฟังก์ชันตัวกรอง จากการที่เปลี่ยนจากเครื่องหมายลบในสมการที่ 2-9 เป็นเครื่องหมายบวกในสมการที่ 2-10

$$(f * g)(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v).g(x+u,y+v) \qquad (2\text{-}10)$$

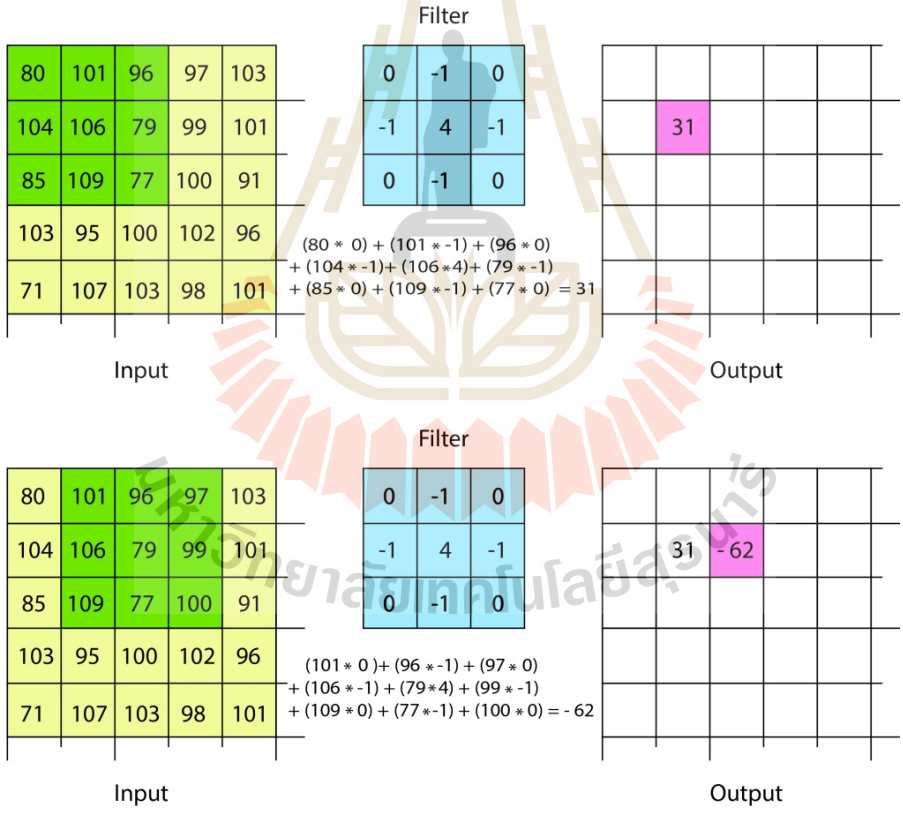
ทั้งนี้ในโครงข่ายประสาทแบบคอนโวลูชันนั้นมองลักษณะการดำเนินการทั้งแบบคอนโวลูชันและแบบสหสัมพันธ์ข้ามโดยนิยามที่กล่าวมานั้นว่าเป็นลักษณะเดียวกัน นั่นคือเป็น “การคอนโวลูชัน” ทั้งหมด และเป็นที่มาของชื่อโครงข่ายแบบคอนโวลูชัน ตัวอย่างการคอนโวลูชันดังกล่าวแสดงในรูปที่ 2-25 ซึ่งแสดงค่าที่เกิดขึ้นในแต่ละตำแหน่งจากการคอนโวลูชันข้อมูลอินพุต คือ $a, b, c, \ldots, l$ กับตัวกรอง (Kernel, Filter) ที่มีค่าเป็น w, x, y, z ซึ่งมองภาพเป็นลักษณะของการนำค่าของตัวกรองทั้งหมดไปครอบบนข้อมูลอินพุต



รูปที่ 2-25 ค่าในแต่ละตำแหน่งของการคอนโวลูชันใน 2 มิติ (Goodfellow et al., 2016)

การคอนโวลูชันเริ่มต้นจากตำแหน่งมุมบนสุดทางด้านซ้ายของข้อมูลใน 2 มิติ แล้วนำค่าที่ตรงกันในแต่ละตำแหน่งของข้อมูลกับตัวกรองนั้นมาคูณกันแล้วนำผลการคูณทั้งหมดทุกตำแหน่งนั้นมาบวกกันได้เป็นค่าเอาต์พุตในตำแหน่งแรก จากนั้นจึงเลื่อนค่าของตัวกรองเดิมทั้งหมดไปทางขวาหนึ่งตำแหน่งเพื่อครอบบนข้อมูลในตำแหน่งถัดไป แล้วทำเช่นเดิม นั่นคือเป็นลักษณะของการทำแบบวินโดว์ของการเลื่อน (Sliding Windows) จากซ้ายไปขวา และบนลงล่างตามลำดับ จากรูปที่ 2-25 เป็นตัวอย่างจากการใช้ขนาดของตัวกรองที่เป็นเมทริกซ์แบบ 2×2 โดยข้อมูลอินพุตที่ใช้มีขนาด 3×4 ดังนั้นเอาต์พุตจึงมีขนาดเป็น 2×3 ดังรูป

รูปที่ 2-26 เป็นตัวอย่างการคำนวณค่าจากการคอนโวลูชันในสองตำแหน่งแรก โดยรูปที่ 2-26(a) เป็นผลลัพธ์จากการการคอนโวลูชันในตำแหน่งที่หนึ่ง และรูปที่ 2-26(b) คือเอาต์พุตของการคอนโวลูชันในตำแหน่งที่สองตามลำดับ เมื่อตัวกรองที่ใช้มีขนาดเป็น 3×3

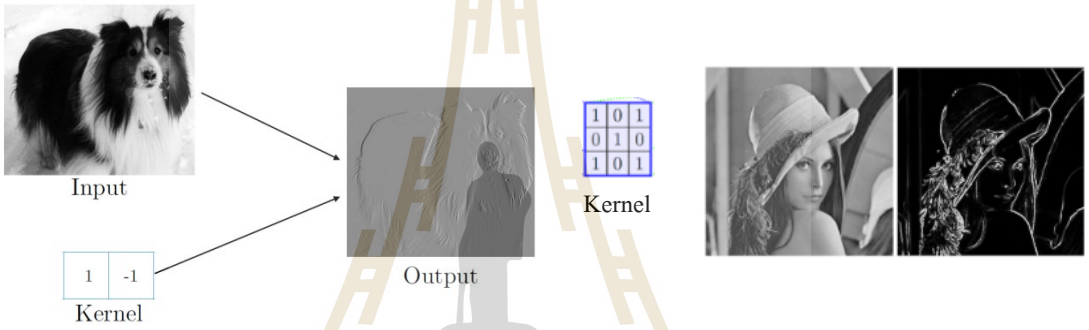


รูปที่ 2-26 ตัวอย่างการคำนวณค่าจากการคอนโวลูชัน

(a) ค่าที่เกิดขึ้นจากการคอนโวลูชันในตำแหน่งที่หนึ่ง

(b) ค่าที่เกิดขึ้นจากการคอนโวลูชันในตำแหน่งที่สอง

ในงานทางด้านการประมวลผลภาพนั้นมีการนำตัวกรองที่หลากหลายรูปแบบมาทำการคอนโวลูชันกับข้อมูลภาพตั้งต้นเพื่อต้องการเปลี่ยนข้อมูลภาพไปอยู่ในรูปแบบอื่นเพื่อเป็นประโยชน์สำหรับขั้นตอนอื่น ๆ ต่อไป เช่นการทำให้ภาพมีความคมชัดมากยิ่งขึ้น การทำให้ภาพพร่ามัว หรือการตรวจจับบริเวณที่เป็นขอบในภาพ เป็นต้น รูปที่ 2-27 เป็นตัวอย่างการคอนโวลูชันกับข้อมูลภาพทั้งภาพเมื่อมีการใช้ตัวกรองที่ต่างรูปแบบกัน โดยรูปที่ 2-27(a) ใช้ตัวกรองแบบ 1×2 ที่มีค่าเป็น 1 และ -1 จะทำให้เกิดภาพเอาต์พุตในลักษณะแบบมีผิวนูนขึ้น (Emboss) ส่วนรูปที่ 2-27(b) ใช้ตัวกรองแบบ 3×3 ที่มีค่าในแนวทแยงเป็น 1 ทั้งหมด ซึ่งเป็นลักษณะของตัวกรองที่ใช้สำหรับตรวจจับบริเวณที่เป็นขอบทั้งหมดในภาพ

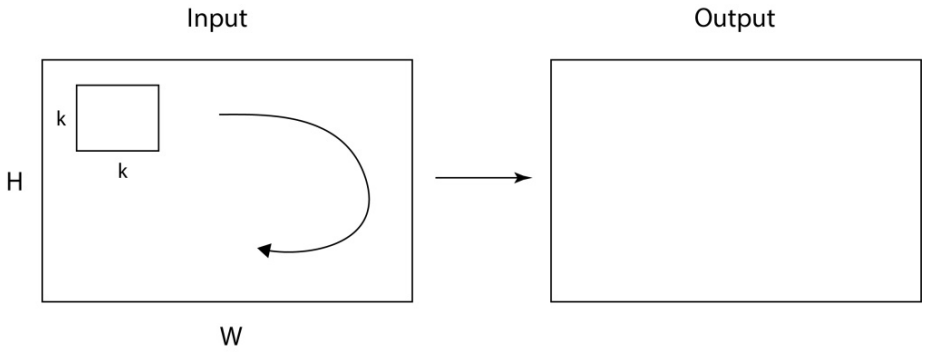


(a) (Goodfellow et al., 2016) (b) (Veličković, 2017)

รูปที่ 2-27 ตัวอย่างการคอนโวลูชันกับข้อมูลทั้งภาพเมื่อมีการใช้ตัวกรองที่ต่างรูปแบบกัน
(a) เมื่อใช้ตัวกรองแบบ 1×2
(b) เมื่อใช้ตัวกรองแบบ 3×3 ที่มีค่าในแนวทแยงเป็น 1 ทั้งหมด

การคอนโวลูชันใน 2 มิติ

การคอนโวลูชันใน 2 มิติที่ได้กล่าวไปแล้วนั้น สามารถเขียนเป็นแผนภาพเพื่อให้มองภาพและเข้าใจได้ชัดเจนยิ่งขึ้นได้ดังรูปที่ 2-28 เป็นการอธิบายว่า การคอนโวลูชันใน 2 มิตินั้นเป็นการนำเมทริกซ์ของตัวกรองขนาด $k \times k$ ไปคอนโวลูชันกับเมทริกซ์ของข้อมูลภาพขนาด W×H ที่เป็นลักษณะการทำแบบวินโดว์ของการเลื่อนไปในสองทิศทาง (อาจมองภาพเป็นทิศทางในแนวแกน x และแกน y) โดยทำไปทั่วทั้งเมทริกซ์ ซึ่งเมทริกซ์เอาต์พุตที่ได้ก็เป็น 2 มิติที่มีขนาดเป็น W×H



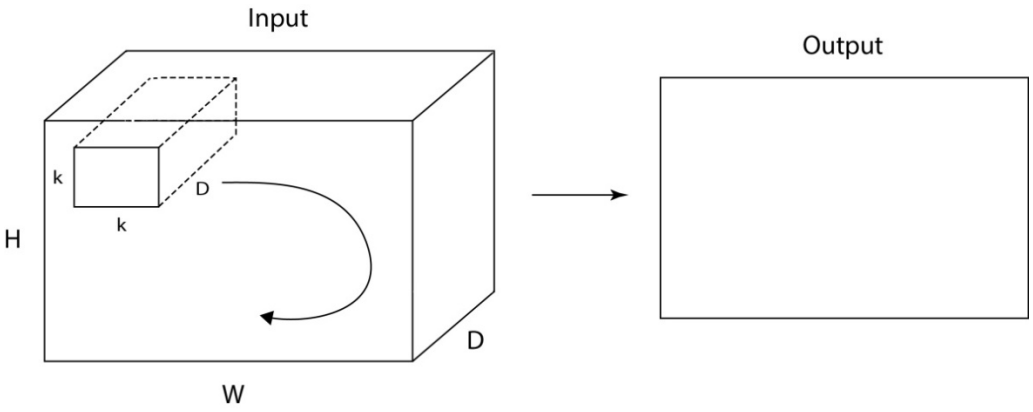
รูปที่ 2-28 การคอนโวลูชันใน 2 มิติ

โดยสรุปของการคอนโวลูชันใน 2 มิติคือ

- ทำสองทิศทางในแนว (x, y) เพื่อคำนวณการคอนโวลูชัน
- รูปร่างของเอาต์พุตที่ได้เป็นเมทริกซ์ใน 2 มิติ
- ขนาดของอินพุต = [W, H]
- ขนาดของตัวกรอง = $[k, k]$
- ขนาดของเอาต์พุต = [W, H]

การคอนโวลูชันใน 2 มิติกับอินพุต 3 มิติ

การคอนโวลูชันส่วนใหญ่ที่ใช้ในโครงข่ายประสาทแบบคอนโวลูชันเป็นลักษณะของการคอนโวลูชันใน 2 มิติกับข้อมูลอินพุต 3 มิติ ซึ่งแสดงดังรูปที่ 2-29 โดยเอาต์พุตที่ได้เป็นเมทริกซ์ใน 2 มิติที่มีขนาดเป็น W×H การคอนโวลูชันแบบนี้เกิดจากการนำตัวกรองในรูปแบบ $k \times k \times$D ไปคอนโวลูชันกับข้อมูลอินพุตใน 3 มิติที่มีความลึกขนาด D (เช่นถ้าข้อมูลภาพสีจากสามแกนสี R, G, B มองเป็นว่า D มีขนาดเป็น 3) ตัวอย่างเช่น ถ้าใช้ตัวกรองที่มีรูปแบบเป็น 5×5×3 เอาต์พุตจากการคอนโวลูชันในแต่ละตำแหน่งนั้นเกิดจากการนำค่าย่อย ๆ ทั้งหมด 75 ค่ามาบวกกัน โดย 25 ค่าแรกมาจากนำตัวกรองขนาด 5×5 ณ D = 1 ไปคอนโวลูชันกับอินพุตในแกนสี R + 25 ค่าถัดมาจากการใช้ตัวกรองขนาด 5×5 ณ D = 2 ไปคอนโวลูชันกับอินพุตแกนสี G ที่ตำแหน่งตรงกัน + 25 ค่าถัดมาจากการใช้ตัวกรองขนาด 5×5 ณ D = 3 ไปคอนโวลูชันกับอินพุตแกนสี B ที่ตำแหน่งตรงกัน ดังนั้นถึงแม้จะเป็นการคอนโวลูชันที่ทำกับข้อมูลอินพุตที่เป็น 3 มิติ เอาต์พุตที่ได้ก็ยังเป็นเมทริกซ์ใน 2 มิติที่มีขนาดเป็น W×H ดังแสดงในรูปที่ 2-29



รูปที่ 2-29 การคอนโวลูชันใน 2 มิติกับอินพุต 3 มิติ (กรณีใช้ตัวกรองรูปแบบเดียว)

การคอนโวลูชันใน 2 มิติกับข้อมูลอินพุต 3 มิติลักษณะเช่นนี้มักจะใช้ในโครงข่ายประสาทแบบคอนโวลูชันเมื่อมีการนำตัวกรองหลากหลายรูปแบบมาใช้ นั่นคือเมื่อตัวกรองที่ต้องการนำมาใช้มีหลาย ๆ ตัวกรอง ซึ่งอาจมีจำนวนเป็นหลายสิบ หรือหลายร้อยตัวกรองก็ได้ขึ้นอยู่กับแต่ละงานประยุกต์ โดยเอาต์พุตที่ได้จากคอนโวลูชันกับแต่ละตัวกรองเป็นเมทริกซ์ใน 2 มิติที่มีขนาดเป็น W×H ถ้าตัวกรองมีจำนวน N ตัวกรอง เอาต์พุตทั้งหมดก็จะเป็น N เมทริกซ์ใน 2 มิติที่มีขนาดเป็น W×H นั่นคือ มองภาพรูปร่างของเอาต์พุตทั้งหมดเป็นกองซ้อน (Stack) ของ N เมทริกซ์ใน 2 มิติที่มีขนาดเป็น W×H ดังนั้นโดยสรุปของการคอนโวลูชันใน 2 มิติกับข้อมูลอินพุต 3 มิติในกรณีใช้ตัวกรองรูปแบบเดียวและในกรณีที่ใช้ตัวกรอง N รูปแบบ คือ

กรณีใช้ตัวกรองรูปแบบเดียว:

- ทำสองทิศทางในแนว (x, y) เพื่อคำนวณการคอนโวลูชัน
- รูปร่างของเอาต์พุตที่ได้เป็นเมทริกซ์ใน 2 มิติ
- ขนาดของอินพุต = [W, H, D]
- ขนาดของตัวกรอง = $[k, k, \mathrm{D}]$
- ขนาดของเอาต์พุต = [W, H]

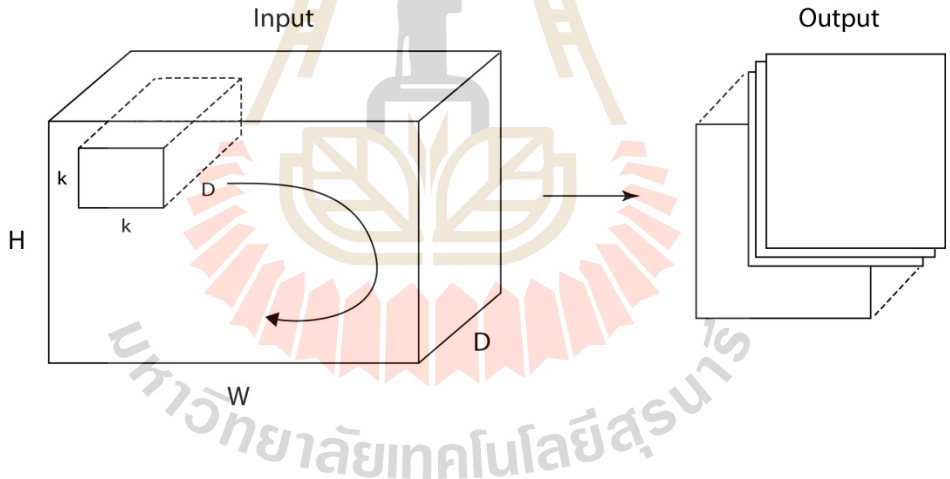
กรณีใช้ตัวกรอง N รูปแบบ:

- ทำสองทิศทางในแนว (x, y) เพื่อคำนวณการคอนโวลูชันกับแต่ละตัวกรอง
- รูปร่างของเอาต์พุตที่ได้เป็นกองซ้อนของเมทริกซ์ใน 2 มิติ
- ขนาดของอินพุต = [W, H, D]
- ขนาดของตัวกรอง = $[k, k, \mathrm{D}]$ จำนวน N รูปแบบ (N ตัวกรอง)

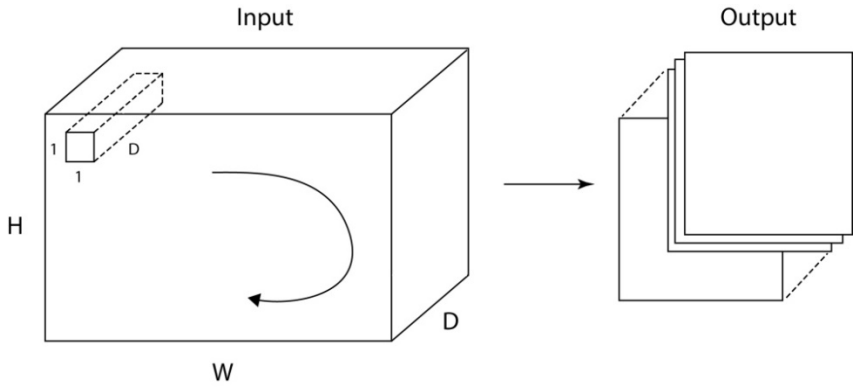
– ขนาดของเอาต์พุต = กองซ้อนของเอาต์พุตจากแต่ละตัวกรอง นั่นคือรูปร่างของเอาต์พุตเป็น 3 มิติที่เกิดจากนำเมทริกซ์ใน 2 มิติ N เมทริกซ์มาวางซ้อนกัน ดังแสดงในรูปที่ 2-30

การคอนโวลูชันแบบ 1×1

เป็นการทำคอนโวลูชันแบบการกรองทางลึก (Depth-Wise Filtering) ซึ่งเริ่มต้นนำมาใช้ในโครงข่ายประสาทแบบคอนโวลูชันของ GoogleNet (Szegedy et al., 2014) ที่เป็นลักษณะของการคอนโวลูชันแบบเดียวกันกับการคอนโวลูชันใน 2 มิติกับข้อมูลอินพุต 3 มิติที่กล่าวไปก่อนหน้านี้ แต่ในการคอนโวลูชันแบบ 1×1 นั้นจะใช้ตัวกรองที่มีรูปแบบเป็น 1×1×D เท่านั้นแทนที่จะเป็นรูปแบบ $k\times k\times$D ที่ได้กล่าวไปก่อนหน้านี้ การคอนโวลูชันแบบ 1×1 แสดงดังรูปที่ 2-31 เมื่อเป็นการนำตัวกรองแบบ 1×1×D จำนวน N รูปแบบมาคอนโวลูชันกับภาพอินพุตที่เป็น 3 มิติ ซึ่งขนาดของเอาต์พุตก็จะเป็นกองซ้อนของเอาต์พุตจากแต่ละตัวกรอง นั่นคือรูปร่างของเอาต์พุตจะเป็นลักษณะ 3 มิติที่เกิดจากนำเมทริกซ์ใน 2 มิติ N เมทริกซ์มาวางซ้อนกัน



รูปที่ 2-30 การคอนโวลูชันใน 2 มิติกับอินพุต 3 มิติ (กรณีใช้ตัวกรอง N รูปแบบ)



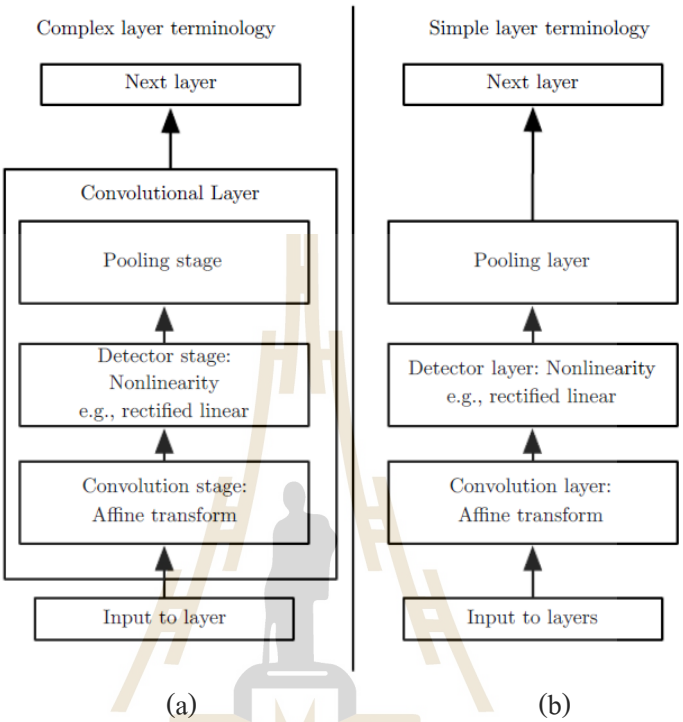
รูปที่ 2-31 การคอนโวลูชันแบบ 1×1 (กรณีใช้ตัวกรอง *N* รูปแบบ)

2.4.2 แนวคิดสำคัญและองค์ประกอบหลักของโครงข่ายประสาทแบบคอนโวลูชัน

จากพื้นฐานเกี่ยวกับการดำเนินการแบบคอนโวลูชันร่วมกันกับคุณลักษณะของโครงข่ายประสาทนำมาซึ่งสามแนวคิดหลักที่เป็นที่มาของการเกิดเป็นโมเดลการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทแบบคอนโวลูชัน นั่นคือแนวคิดเกี่ยวกับ การมีปฏิสัมพันธ์แบบหยาบ (Sparse Interaction) การใช้พารามิเตอร์ร่วมกัน (Parameter Sharing) และการสร้างตัวแทนแบบต่างร่วม (Equivariant Representations) นอกจากนั้นการดำเนินการแบบคอนโวลูชันยังสามารถทำให้เกิดเอาต์พุตที่หลากหลายกันไปเมื่อนำไปใช้กับอินพุตที่มีขนาดแตกต่างกัน จากแนวคิดหลักดังกล่าวเป็นที่มาของโครงข่ายประสาทแบบคอนโวลูชันที่ประกอบด้วยองค์ประกอบหลักต่าง ๆ ในโครงข่ายแสดงดังรูปที่ 2-32 โดยสามารถมองภาพขององค์ประกอบต่าง ๆ ได้เป็นสองรูปแบบขึ้นอยู่กับว่าแต่ละสถาปัตยกรรมที่ได้นำเสนอไว้นั้นมองภาพเป็นรูปแบบใด

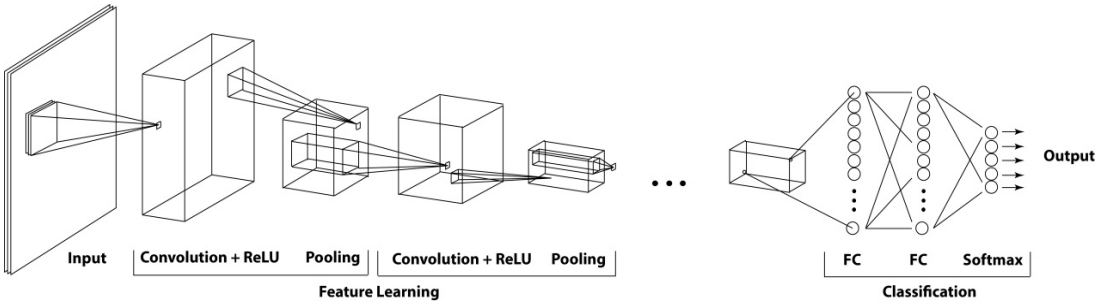
จากรูปที่ 2-32(a) เป็นการมองภาพว่าแต่ละชั้นในโครงข่ายมีรูปแบบที่ซับซ้อน (Complex Layer) จากการที่มองว่าในแต่ละชั้นของการคอนโวลูชัน (Convolution Layer) มีส่วนของขั้นตอนย่อยอื่นประกอบอยู่ภายใน นั่นคือ ขั้นตอนการคอนโวลูชัน (Convolution Stage) ขั้นตอนการตรวจจับ (Detector Stage) ซึ่งเป็นขั้นตอนของการแปลงค่าที่ได้จากการคอนโวลูชันด้วยฟังก์ชันแบบไม่เป็นเชิงเส้น เช่นฟังก์ชัน ReLU (Rectified Linear Unit) ตามด้วยขั้นตอนการทำพูลลิ่ง (Pooling Stage) ตามลำดับ โดยที่ชั้นของการคอนโวลูชันก่อนหน้าก็จะเชื่อมต่อกับชั้นของการคอนโวลูชันถัดไปเรื่อย ๆ ดังรูป ส่วนรูปที่ 2-32(b) เป็นการมองภาพว่าแต่ละชั้นในโครงข่ายมีรูปแบบที่เป็นชั้นอย่างง่าย (Simple Layer) นั่นคือเป็นการมองแต่ละขั้นตอน (Stage) ย่อย ๆ จากรูปที่ 2-32(a) เป็นหนึ่งชั้นของโครงข่ายในรูปที่ 2-32(b) ดังนั้นชั้นของการคอนโวลูชันในรูปแบบชั้น

อย่างง่ายนั้นจะทำเพียงขั้นตอนของการคอนโวลูชันเท่านั้น ไม่ได้ประกอบด้วยสามขั้นตอนย่อย เหมือนในรูปที่ 2-32(a)



รูปที่ 2-32 องค์ประกอบหลักของโครงข่ายประสาทแบบคอนโวลูชัน (Goodfellow et al., 2016)
(a) มองภาพว่าแต่ละชั้นในโครงข่ายมีรูปแบบที่ซับซ้อน
(b) มองภาพว่าแต่ละชั้นในโครงข่ายมีรูปแบบที่เป็นชั้นอย่างง่าย

รูปที่ 2-33 แสดงแนวคิดในภาพรวมของโมเดลโครงข่ายประสาทแบบคอนโวลูชัน จากการนำองค์ประกอบย่อย ๆ ที่กล่าวไปแล้วในรูปที่ 2-32 มาใช้ในโมเดลในส่วนของการเรียนรู้คุณลักษณะ (Feature Learning) โดยผลลัพธ์ที่ได้จากการเรียนรู้ดังกล่าวนำไปใช้ในขั้นตอนของการจำแนก (Classification) ด้วยโครงข่ายลักษณะเดียวกันกับเพอร์เซปตรอนแบบหลายชั้นแบบดั้งเดิมที่เป็นลักษณะของการเชื่อมถึงกันหมด (Fully Connected, FC)



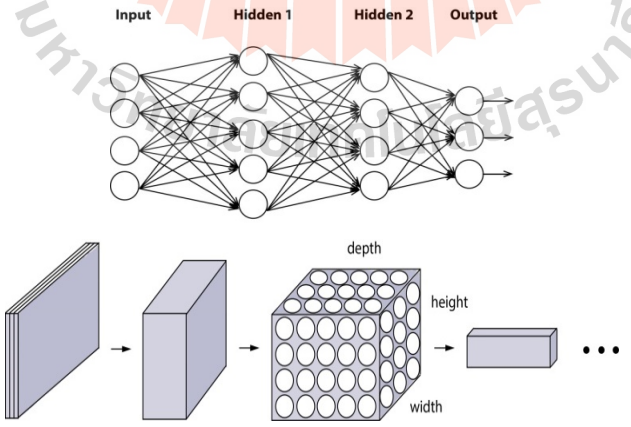
รูปที่ 2-33 แนวคิดในภาพรวมของโมเดลโครงข่ายประสาทแบบคอนโวลูชัน

การทำงานภายในของโครงข่ายนั้นเป็นการนำตัวกรองขนาดเดียวกันหลากหลายรูปแบบอย่างสุ่ม มาทำการคอนโวลูชันกับข้อมูลภาพในแต่ละชั้นของการคอนโวลูชัน เช่นชั้นคอนโวลูชันแรกใช้ตัวกรองขนาด 9×9 จำนวน 20 ตัวกรอง ชั้นต่อมาใช้ตัวกรองขนาด 5×5 จำนวน 50 ตัวกรอง เป็นต้น โดยเป็นลักษณะของการคอนโวลูชันแบบ 2 มิติกับข้อมูลใน 3 มิติ แบบเดียวกันกับที่ได้กล่าวไปแล้วก่อนหน้านี้ ดังนั้นเอาต์พุตที่ได้จากการคอนโวลูชันจึงเป็นข้อมูลแบบ 3 มิติจากการนำเอาต์พุตที่ได้จากการการคอนโวลูชันกับแต่ละตัวกรองมาวางเป็นกองซ้อนกัน โดยผลลัพธ์ที่ได้จากการทำคอนโวลูชันในแต่ละตำแหน่งจะผ่านการแปลงค่าด้วยฟังก์ชันไม่เป็นเชิงเส้น เช่น ฟังก์ชัน Rectified Linear Unit (ReLU) ดังนั้นเอาต์พุตที่ได้จากการจากคอนโวลูชันกับแต่ละตัวกรองมองว่าเป็นเอาต์พุตของแผนที่คุณลักษณะ (Feature Map Output) ซึ่งตัวกรองหนึ่งแบบก็จะได้เป็นหนึ่งแบบของแผนที่คุณลักษณะ เอาต์พุตของแผนที่คุณลักษณะทั้งหมดจากชั้นก่อนหน้าก็จะเป็นอินพุตของการทำพูลลิ่ง (Pooling) ซึ่งการทำพูลลิ่งนั้นเป็นแนวคิดของการทำเพื่อสุ่มต่ำ (Subsampling) กับแต่ละแผนที่คุณลักษณะ ดังนั้นแต่ละแผนที่คุณลักษณะเมื่อผ่านการทำพูลลิ่งแล้วขนาดก็จะเล็กลง ดังรูปที่ 2-33 ที่ลักษณะของกล่องมีขนาดในแนวกว้างและแนวยาวที่ลดลง ในชั้นท้าย ๆ ของโครงข่ายก่อนชั้นเอาต์พุตจะเป็นการนำค่าจากแผนที่คุณลักษณะในชั้นก่อนหน้านั้นมาเรียงเป็นข้อมูลแบบเวกเตอร์ (ชั้น FC แรกในรูป) และใช้โครงข่ายแบบเชื่อมถึงกันหมดแบบเดียวกันกับที่ใช้ในโครงข่ายเพอร์เซปตรอนหลายชั้นแบบดั้งเดิม โดยเอาต์พุตจากชั้นสุดท้ายของการเชื่อมถึงกันหมดจะผ่านฟังก์ชันซอฟต์แมกซ์ (Softmax Function) เพื่อให้ได้เป็นเอาต์พุตของโครงข่ายที่อยู่ในรูปของค่าความน่าจะเป็น

ถ้ามองภาพลักษณะการเรียงตัวของนิวรอนในโครงข่ายประสาทแบบคอนโวลูชันเมื่อเปรียบเทียบกับโครงข่ายเพอร์เซปตรอนหลายชั้นแบบดั้งเดิมสามารถมองได้ดังรูปที่ 2-34 เมื่อแต่ละนิวรอนคือวงกลมแต่ละวงในรูป โดยรูปที่ 2-34(a) เป็นการเรียงตัวของนิวรอนในแต่ละชั้นของโครงข่ายเพอร์เซปตรอนหลายชั้นแบบดั้งเดิม ส่วนรูปที่ 2-34(b) เป็นการมองภาพการเรียงตัว

ของนิวรอนในโครงข่ายประสาทแบบคอนโวลูชัน ซึ่งการเรียงตัวของนิวรอนอยู่ในรูปแบบที่เรียกว่าเทนเซอร์ (Tensor) ที่เป็นปริมาตรใน 3 มิติ เมื่อแกนหนึ่งของเทนเซอร์คือความกว้าง (Width) ของภาพ อีกแกนหนึ่งคือความสูง (Height) ของภาพ และอีกแกนหนึ่งคือความลึก (Depth) โดยความลึกเกิดจากการนำเอาแต่ละแผนที่คุณลักษณะมาวางเป็นกองซ้อนกัน ดังนั้นข้อมูลอินพุตที่เป็นภาพสีแบบ RGB จึงเป็นเทนเซอร์ที่มีความลึกเป็น 3 (มาจากการนำค่าของจุดภาพในแต่ละแกนสีมาวางเป็นกองซ้อนกัน โดยแต่ละจุดภาพในเทนเซอร์นี้คือแต่ละนิวรอนอินพุต) ดังนั้นวงกลมแต่ละวงที่อัดกันอยู่ในเทนเซอร์ก็คือแต่ละนิวรอนของโครงข่ายแบบคอนโวลูชัน

แนวคิดของโครงข่ายประสาทแบบคอนโวลูชันนั้นมองว่าแต่ละตัวกรองที่นำมาใช้ในขั้นตอนการคอนโวลูชันคือแต่ละรูปแบบของค่าน้ำหนักที่เป็นลักษณะของการทำงานแบบการใช้ค่าน้ำหนักร่วมกัน (Weight Sharing) จากลักษณะของการคอนโวลูชันแบบ 2 มิติกับข้อมูลใน 3 มิติ โดยมองว่ารูปแบบของค่าน้ำหนักหรือตัวกรองที่ได้จากการเรียนรู้ของโครงข่ายในแต่ละชั้นซ่อนเร้นนั้น ยิ่งจำนวนชั้นซ่อนเร้นยิ่งลึก ลักษณะรูปแบบของค่าน้ำหนักหรือตัวกรองก็ยิ่งซับซ้อน (Complex) ขึ้น ดังนั้นโครงข่ายก็จะยิ่งสามารถเรียนรู้จากข้อมูลเพื่อสร้างแผนที่คุณลักษณะที่ซับซ้อนและเหมาะสมขึ้นเรื่อย ๆ ได้ แผนที่คุณลักษณะสุดท้ายที่ได้จะนำไปสู่ขั้นตอนการคัดแยกด้วยชั้นของการเชื่อมถึงกันหมดเพื่อให้ได้ออกมาเป็นเอาต์พุตสุดท้ายของโครงข่าย *โดยสรุปคือโครงข่ายประสาทแบบคอนโวลูชันเป็นการเรียนรู้เพื่อหาค่าน้ำหนักที่เหมาะสมที่สุด (ซึ่งคือค่าที่เหมาะสมที่สุดในแต่ละรูปแบบของตัวกรอง) ในแต่ละชั้นของโครงข่ายเพื่อให้สามารถสร้างเป็นรูปแบบการแทนข้อมูลที่ดีที่สุดจากการแทนข้อมูลที่อยู่ในรูปของแผนที่คุณลักษณะ*



(a)

(b)

รูปที่ 2-34 ลักษณะของแต่ละนิวรอนในโครงข่าย

(a) นิวรอนของโครงข่ายเพอร์เซปตรอนหลายชั้นแบบดั้งเดิม

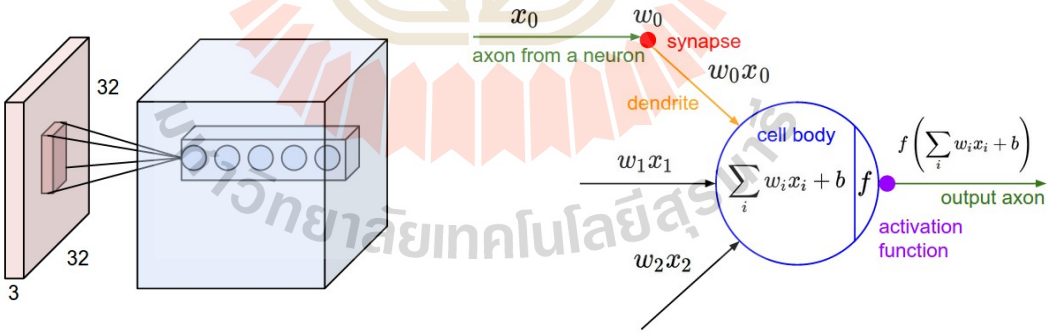
(b) นิวรอนของโครงข่ายประสาทแบบคอนโวลูชัน

2.4.3 รายละเอียดในแต่ละองค์ประกอบหลักของโครงข่ายประสาทแบบคอนโวลูชัน

เนื่องจากองค์ประกอบหลักของโครงข่ายประสาทแบบคอนโวลูชันคือ ขั้นตอนการคอนโวลูชัน ขั้นตอนการแปลงค่าด้วยฟังก์ชัน ReLU ขั้นตอนการทำพูลลิ่งและขั้นตอนของการเชื่อมถึงกันหมด ดังที่แสดงในรูปที่ 2-33 ในที่นี้จึงนำเสนอรายละเอียดของแต่ละขั้นตอนดังกล่าวตามลำดับ

1) ขั้นตอนการคอนโวลูชัน

ภาพประกอบในรูปที่ 2-35 ยกตัวอย่างลักษณะข้อมูลในเทนเซอร์ของโครงข่ายแบบคอนโวลูชัน ในที่นี้สมมุติเป็นการคอนโวลูชันโดยใช้ตัวกรองทั้งหมด 5 ตัวกรองกับข้อมูลภาพอินพุตที่เป็นภาพสีแบบ RGB ขนาด 32×32 จากรูปที่ 2-35(a) จะเห็นว่า ข้อมูลภาพอินพุตเป็นเทนเซอร์ที่มีค่าในแนวลึกเป็น 3 นั้นมาจากการนำค่าในสามแกนสีของภาพแบบ RGB มาวางเป็นลักษณะกองซ้อนกัน ด้านขวาของรูปที่ 2-35(a) เป็นเทนเซอร์เก็บเอาต์พุตที่เกิดจากการนำตัวกรองทั้งหมด 5 ตัวกรองมาคอนโวลูชันกับภาพอินพุต ในลักษณะของการคอนโวลูชันแบบ 2 มิติ กับข้อมูล 3 มิติ ในรูปแสดงตำแหน่งของ 5 นิวรอนในแนวลึกของเทนเซอร์ว่าคือนิวรอนเอาต์พุตที่เกิดจากการคอนโวลูชันในขอบเขตรับสัญญาณ (Receptive Field) เดียวกันในข้อมูลอินพุตแต่ใช้ตัวกรองที่แตกต่างกัน 5 ตัวกรอง (แต่ละตัวกรองก็เป็นเทนเซอร์ขนาดเล็ก ๆ เช่น 7×7×3, 5×5×3 โดยที่ค่าในแนวลึกของเทนเซอร์ตัวกรองต้องเท่ากับค่าในแนวลึกของเทนเซอร์อินพุตเสมอ)



(a) (b)

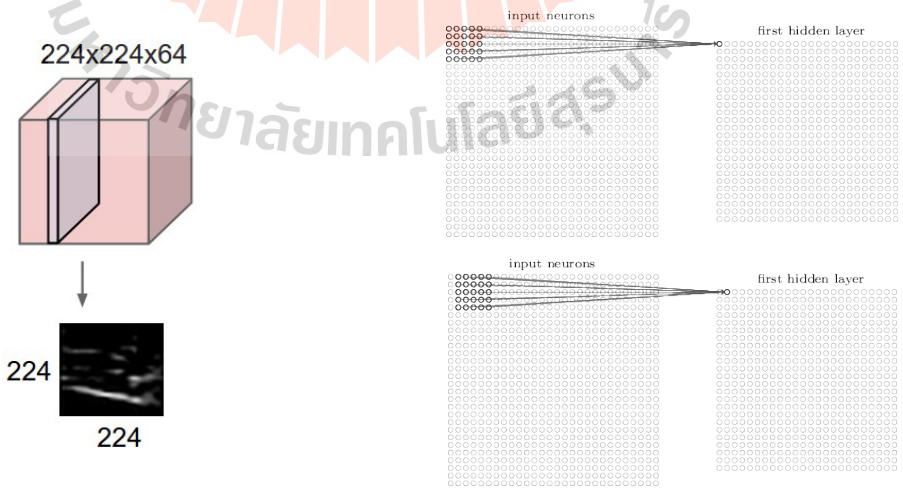
รูปที่ 2-35 ลักษณะข้อมูลในเทนเซอร์และการคำนวณค่าของนิวรอน (Li et al., 2017)

(a) ลักษณะข้อมูลในเทนเซอร์ที่มีขอบเขตรับสัญญาณเดียวกัน

(b) การคำนวณค่าในแต่ละนิวรอน

รูปที่ 2-35(b) แสดงการคำนวณค่าของแต่ละนิวรอนเช่นเดียวกันกับในเพอร์เซปตรอนแบบชั้นเดียวหรือเพอร์เซปตรอนแบบหลายชั้น ที่เกิดจากการรวมกันของผลคูณระหว่างค่าน้ำหนัก (ค่า w_i ในรูป) กับอินพุต (ค่า x_i ในรูป) แล้วบวกด้วยค่าของไบอัส (ค่า b ในรูป) แต่ค่าน้ำหนักที่ใช้ในโครงข่ายแบบคอนโวลูชันคือค่าในตัวกรองแต่ละตัวกรอง โดยมักเริ่มต้นจากการกำหนดค่าเป็นแบบสุ่ม (Random) เช่นเดียวกันกับการสุ่มค่าของน้ำหนักในเพอร์เซปตรอน ส่วนอินพุตนั้นใช้เฉพาะในบริเวณขอบเขตรับสัญญาณเท่านั้นตามลักษณะของการคอนโวลูชันแบบ 2 มิติ กับข้อมูล 3 มิติ

เมื่อมองเข้าไปในข้อมูลแต่ละแผ่นในแนวลึก (Depth Slice) ของเทนเซอร์ เมื่อเทนเซอร์ในรูปที่ 2-36(a) เป็นเอาต์พุตที่เกิดจากการนำตัวกรอง 64 ตัวกรองมาคอนโวลูชันกับอินพุต เกิดเป็นเทนเซอร์แบบ 3 มิติที่มีขนาดในแนวกว้าง แนวยาว และ แนวลึก เป็น 224, 224 และ 64 ตามลำดับ ถ้ามองลึกลงไปในข้อมูลแต่ละแผ่น (Slice) จะพบว่าเป็นลักษณะของข้อมูลใน 2 มิติที่มีขนาดเป็น 224×224 นั่นคือข้อมูลแต่ละ Slice ดังกล่าวคือแต่ละรูปแบบของแผนที่คุณลักษณะจากทั้งหมด 64 รูปแบบ โดยข้อมูลในแต่ละตำแหน่ง (อาจมองว่าคือแต่ละจุดภาพ) ของแผนที่คุณลักษณะนั้นก็คือค่าของนิวรอนแต่ละนิวรอนในชั้นซ่อนเร้นแรกของโครงข่าย ดังแสดงในรูปที่ 2-36(b) นั่นคือค่าของนิวรอนตัวแรกที่แสดงด้วยวงกลมสีเข้มด้านบนของรูป เกิดจากการคอนโวลูชันในขอบเขตรับสัญญาณแรก (จากภาพใช้ขนาดของตัวกรองเป็น 5×5) และค่าของนิวรอนตัวที่สองที่แสดงด้านล่างของรูปเกิดจากการคอนโวลูชันในขอบเขตรับสัญญาณถัดมาจากการเลื่อนไปในตำแหน่งถัดไปของขอบเขตรับสัญญาณ ตามลำดับ



(a) (Li et al., 2017) (b) (Nielsen, 2017)

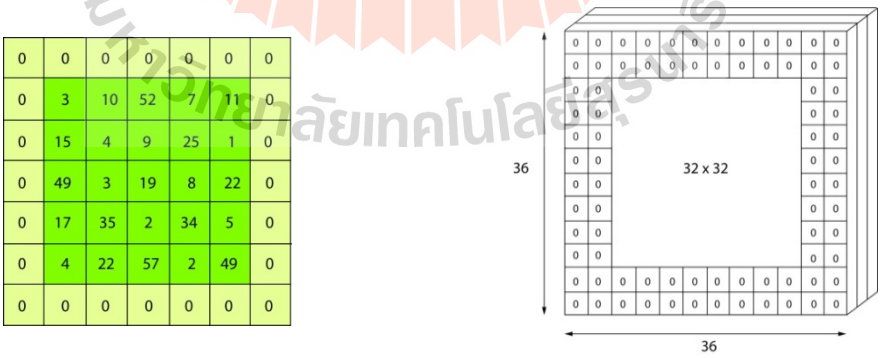
รูปที่ 2-36 ข้อมูลแต่ละแผ่นในแนวลึกของเทนเซอร์

การทำแพดดิ้ง (Padding)

การทำแพดดิ้งคือ การปรับขนาดของอินพุตก่อนขั้นตอนของการคอนโวลูชัน เพื่อให้ได้ขนาดในแนวกว้าง และแนวยาวของเอาต์พุตตามที่ต้องการ มักใช้ในกรณีที่ต้องการให้ขนาดของแต่ละแผนที่คุณลักษณะมีขนาดเท่ากันกับอินพุตหลังจากการทำคอนโวลูชัน โดยส่วนใหญ่มักใช้การทำแบบแพดดิ้งด้วยค่าศูนย์ (Zero Padding) โดยเป็นการเติมค่าที่เป็นศูนย์บริเวณรอบ ๆ ภาพอินพุต รูปที่ 2-37(a) เป็นตัวอย่างการทำแพดดิ้งด้วยค่าศูนย์ด้วยขนาดเท่ากับ 1 ที่ทำให้ขนาดของภาพจากเดิม 5×5 ขยายเป็น 7×7 ส่วนรูปที่ 2-37(b) เป็นตัวอย่างเมื่อทำแพดดิ้งด้วยค่าศูนย์ที่มีขนาดเท่ากับ 2 กับข้อมูลในเทนเซอร์เดิมที่เป็นแบบ $32\times32\times3$ มีผลทำให้ได้เทนเซอร์ใหม่ที่มีขนาดในแนวกว้าง ยาว และลึกเป็น $36\times36\times3$ ดังรูป

การสไทรด์ (Stride)

ค่าที่กำหนดให้กับการสไทรด์ คือจำนวนตำแหน่งของการให้ตัวกรองเลื่อนไปในภาพทั้งในแนวกว้างและแนวยาวในขั้นตอนของการคอนโวลูชัน ซึ่งจากลักษณะของการคอนโวลูชันปกติจะเป็นการเลื่อนตัวกรองไปครั้งละ 1 ตำแหน่ง แต่ในโครงข่ายแบบคอนโวลูชันนั้นสามารถกำหนดค่าการสไทรด์ เป็น 2 หรือ 3 หรือค่าอื่น ๆ ก็ได้ รูปที่ 2-38 ยกตัวอย่างการสไทรด์ ด้วยค่าที่แตกต่างกันในขั้นตอนของการคอนโวลูชันด้วยตัวกรองแบบ 3×3 กับอินพุตขนาด 7×7 เมื่อรูปที่ 2-38(a) ใช้ขนาดของการสไทรด์ เป็น 1 ส่วนรูปที่ 2-38(b) ใช้ขนาดของการสไทรด์ เป็น 2 จะเห็นว่าด้วยขนาดของการสไทรด์ที่แตกต่างกัน ทำให้เกิดขนาดของเอาต์พุตที่แตกต่างกันด้วย นั่นคือถ้าขนาดของการสไทรด์ยิ่งมาก ขนาดของเอาต์พุตก็ยิ่งลดลง

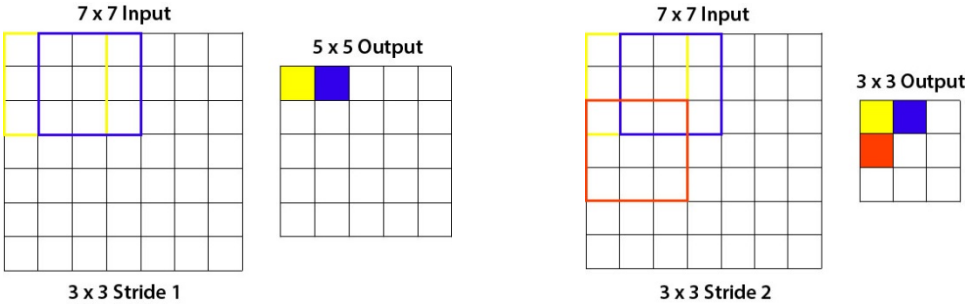


(a) (b)

รูปที่ 2-37 การทำแพดดิ้งด้วยค่าศูนย์

(a) ทำแพดดิ้งด้วยขนาดเท่ากับ 1 กับภาพอินพุต

(b) ทำแพดดิ้งด้วยขนาดเท่ากับ 2 กับแผนที่คุณลักษณะ

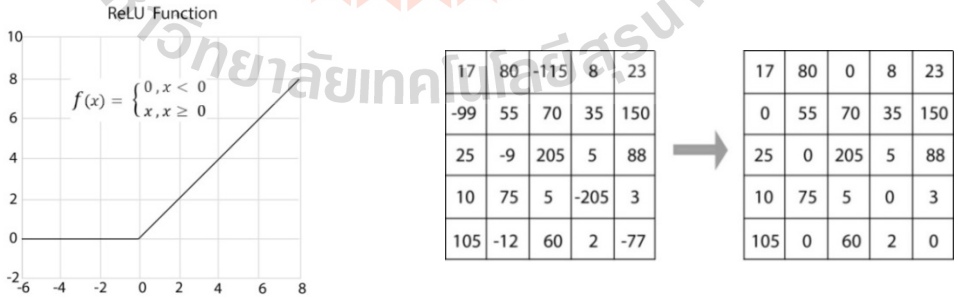


(a) ขนาดของการสไทรด์เป็น 1 (b) ขนาดของการสไทรด์เป็น 2

รูปที่ 2-38 การสไทรด์ด้วยค่าที่แตกต่างกัน

2) ขั้นตอนการแปลงค่าด้วยฟังก์ชัน ReLU

ในโครงข่ายแบบคอนโวลูชันนั้น หลังจากผ่านขั้นตอนของการคอนโวลูชันแล้ว ค่าที่ได้ในแต่ละตำแหน่งของการทำคอนโวลูชันจะนำมาผ่านการแปลงค่าด้วยฟังก์ชันแบบไม่เป็นเชิงเส้น โดยส่วนใหญ่มักจะใช้ฟังก์ชัน ReLU ซึ่งเป็นขั้นตอนของการแปลงข้อมูลแบบเชิงเส้นที่เกิดจากการคอนโวลูชันไปเป็นข้อมูลแบบไม่เป็นเชิงเส้นด้วยค่าของฟังก์ชันการแปลงแสดงดังรูปที่ 2-39(a) นั่นคือเป็นการปัดค่าที่เป็นลบจากการคอนโวลูชันไปเป็นศูนย์ ในขณะที่ค่าอื่นยังคงไว้เหมือนเดิม ตัวอย่างค่าของข้อมูลที่เกิดขึ้นจากการใช้ฟังก์ชัน ReLU แสดงดังภาพประกอบที่ 2-39(b) เมื่อด้านซ้ายคือข้อมูลก่อนผ่านฟังก์ชัน ส่วนด้านขวาคือผลลัพธ์ที่ได้หลังจากผ่านฟังก์ชันดังกล่าว



(a) (b)

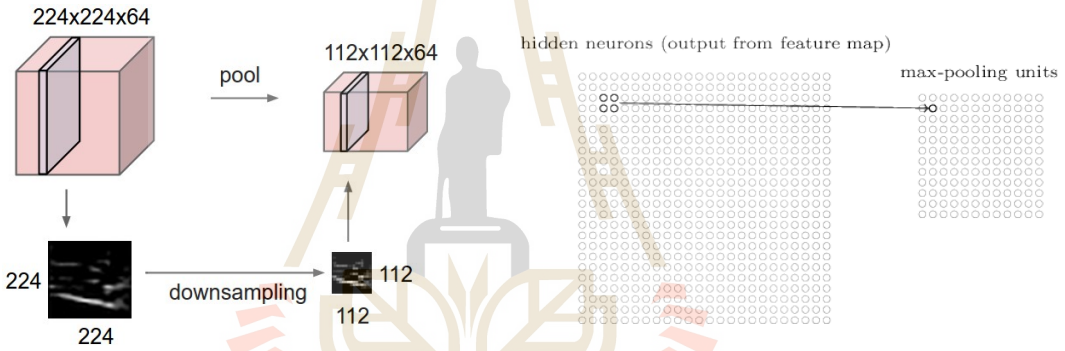
รูปที่ 2-39 การแปลงค่าด้วยฟังก์ชัน ReLU

(a) ค่าของฟังก์ชัน ReLU

(b) ตัวอย่างค่าของข้อมูลที่เกิดขึ้นจากการใช้ฟังก์ชัน ReLU

3) ขั้นตอนการทำพูลลิ่ง (Pooling)

โดยส่วนใหญ่ของโครงข่ายแบบคอนโวลูชันนั้น หลังจากผ่านขั้นตอนของการคอนโวลูชันและขั้นตอนการแปลงค่าด้วยฟังก์ชัน ReLU แล้วมักจะตามมาด้วยขั้นตอนของการทำพูลลิ่ง ที่มีเป้าหมายเพื่อต้องการทำการสุ่มต่ำจากข้อมูลในแต่ละแผนที่คุณลักษณะจากขั้นตอนก่อนหน้า ซึ่งการทำพูลลิ่งนั้นจะกำหนดด้วยขอบเขตของการพูลลิ่งและขนาดของการสไทรด์ รูปที่ 2-40(a) ยกตัวอย่างการทำพูลลิ่งในขอบเขต 2×2 กับข้อมูลในเทนเซอร์แบบ 224×224×64 ด้วยขนาดของการสไทรด์ เป็น 2 ทำให้ขนาดของเทนเซอร์เอาต์พุตหลังการทำพูลลิ่งลดลงเป็น 112×112×64 นั่นคือการทำพูลลิ่งในขอบเขต 2×2 ด้วยขนาดของการสไทรด์เท่ากับ 2 นั้นเป็นการสุ่มต่ำจากข้อมูลในแต่ละแผนที่คุณลักษณะจากแต่ะ 2×2 นิวรอนเหลือเป็น 1×1 นิวรอน (นั่นคือ 1 นิวรอน) ดังแสดงในรูปที่ 2-40(b)



(a) (Li et al., 2017) (b) (Nielsen, 2017)

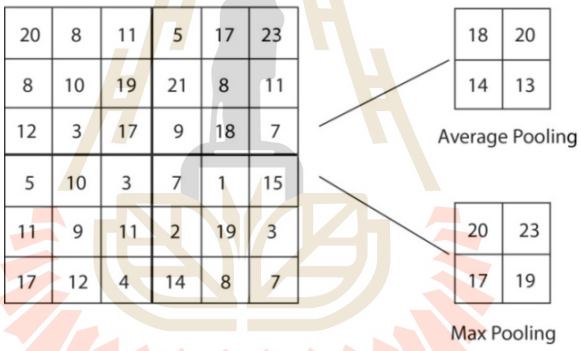
รูปที่ 2-40 การทำพูลลิ่งกับข้อมูลในเทนเซอร์
(a) เอาต์พุตจากการสุ่มต่ำในขั้นตอนการทำพูลลิ่ง
(b) การพูลลิ่งในขอบเขต 2×2

การทำพูลลิ่งนั้น นอกเหนือจากเป็นการสุ่มต่ำแล้วยังสามารถกำหนดได้ว่าจะต้องการค่าจากการสุ่มต่ำนั้นเป็นรูปแบบใด เช่น รูปที่ 2-41 เป็นตัวอย่างค่าที่ได้จากการทำพูลลิ่งแบบเฉลี่ย (Average-Pooling) และแบบแม็กซ์ (Max-Pooling) กับข้อมูลอินพุตชุดเดียวกัน ที่เป็นการพิจารณาข้อมูลอินพุตในขอบเขต 3×3 เพื่อหาค่าที่สูงที่สุดจากเก้าค่าที่อยู่ในขอบเขตนั้นแล้วเลือกค่าที่สูงที่สุดเป็นเอาต์พุตในกรณีที่เป็นพูลลิ่งแบบแม็กซ์ ส่วนพูลลิ่งแบบเฉลี่ยก็จะเป็นการนำอินพุตทั้งเก้าค่านั้นมาเฉลี่ยกัน โดยกำหนดค่าขนาดของการสไทรด์ เป็น 3 ซึ่งนอกจากพูลลิ่งแบบแม็กซ์และแบบเฉลี่ยแล้ว อาจจะใช้พูลลิ่งแบบ L2 (L2-Pooling) หรือแบบอื่น ๆ มาใช้ในขั้นตอน

การทำพูลลิ่งก็ได้ โดยขอบเขตของการพูลลิ่งก็สามารถกำหนดเป็น 3×3, 4×4, 10×10 หรือเท่าไหร่ก็ได้ และขนาดการสไทรด์ ก็กำหนดให้มากกว่า 2 ก็ได้ นอกจากนี้การทำแพดดิ้งก็สามารถนำมาใช้ในขั้นตอนของการทำพูลลิ่งได้ด้วยเช่นกัน

4) ชั้นการเชื่อมถึงกันหมด

จากแนวคิดในภาพรวมของโมเดลโครงข่ายประสาทแบบคอนโวลูชันที่นำเสนอไปแล้วในรูปที่ 2-33 จะพบว่าโครงข่ายประกอบด้วยสองส่วนหลักคือ ส่วนการเรียนรู้เพื่อหาคุณลักษณะและส่วนของการจำแนก โดยข้อมูลในเทนเซอร์ที่ได้จากชั้นสุดท้ายของส่วนของการเรียนรู้เพื่อหาคุณลักษณะจะถูกนำมาเรียงเป็นข้อมูลแบบเวกเตอร์เพื่อเป็นอินพุตเข้าสู่ส่วนของการจำแนก แล้วใช้โครงข่ายแบบเชื่อมถึงกันหมดที่มีลักษณะเช่นเดียวกันกับโครงข่ายเพอร์เซปตรอนหลายชั้นแบบดั้งเดิม โดยผลลัพธ์ที่ได้จากชั้นสุดท้ายของการเชื่อมถึงกันหมดจะผ่านฟังก์ชันซอฟต์แมกซ์ก่อนที่จะได้เป็นเอาต์พุตสุดท้ายของโครงข่าย



รูปที่ 2-41 ตัวอย่างค่าจากการทำพูลลิ่งแบบเฉลี่ยและแบบแม็กซ์

ลักษณะของฟังก์ชันซอฟต์แมกซ์ แสดงดังสมการที่ 2-11

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{j} e^{y_i}} \tag{2-11}$$

เมื่อ $S(y_i)$ คือซอฟต์แมกซ์ของเอาต์พุต y จากแต่ละเอาท์พุตนิวรอน i เป็นค่าเอกซ์โปเนนเชียลของ y_i หารด้วยผลรวมของเอกซ์โปเนนเชียลของเวกเตอร์ y ทั้งหมด เมื่อ j คือแต่ละองค์ประกอบของเวกเตอร์ y

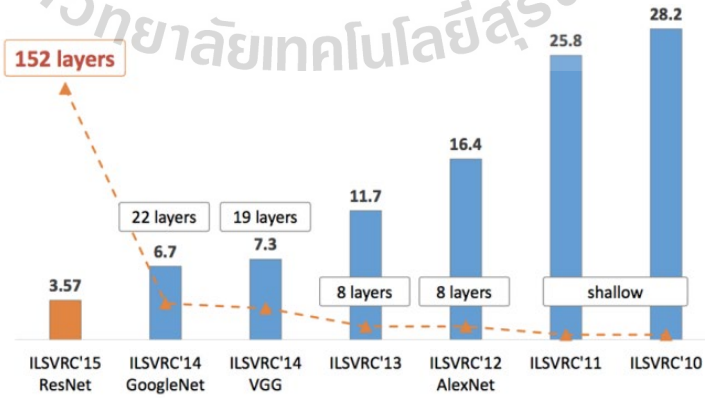
ซึ่งค่าที่เกิดขึ้นจากฟังก์ชันซอฟต์แมกซ์ในสมการที่ 2-11 แสดงดังตัวอย่างคือ

$$Z = \text{softmax}\left[\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}\right] = \begin{pmatrix} 0.1050 & 0.1125 & 0.1050 \\ 0.1125 & 0.1300 & 0.1125 \\ 0.1050 & 0.1125 & 0.1050 \end{pmatrix}$$

ในส่วนของอัลกอริทึมที่ใช้สำหรับการปรับค่าน้ำหนักในขั้นตอนการเรียนรู้ของโครงข่ายประสาทแบบคอนโวลูชันนั้นก็เป็นการใช้อัลกอริทึมแบบแพร่กลับเช่นเดียวกันกับในโครงข่ายเพอร์เซปตรอนแบบหลายชั้น

2.5 สถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชัน

ถึงแม้ว่าโครงข่ายประสาทแบบคอนโวลูชันนั้นจะเริ่มต้นเป็นที่รู้จักอย่างแพร่หลายหลังจากงาน ILSVRC เมื่อปี 2012 หลังจากที่ Krizhevsky และคณะ (2012) ใช้สถาปัตยกรรมในชื่อ AlexNet สำหรับการแข่งขันในงานเพื่อจำแนกข้อมูลภาพจากชุดข้อมูล ImageNet แต่จริง ๆ แล้ว LeCun และคณะ ได้เคยนำเสนอสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันมาก่อนแล้ว ตั้งแต่ปี 1998 ภายใต้สถาปัตยกรรมชื่อ LeNet (LeCun et al., 1998) แต่ยังไม่ได้เป็นที่รู้จักมากนักในยุคนั้น ในส่วนนี้จึงเริ่มต้นจากการนำเสนอสถาปัตยกรรมต่าง ๆ ที่เกิดขึ้นเรียงตามลำดับเวลา เริ่มจากการพัฒนา LeNet ตามด้วยแต่สถาปัตยกรรมที่ชนะการแข่งขันและที่โดดเด่นในงาน ILSVRC นับตั้งแต่ปี 2012 จนถึงปี 2015 ซึ่งรูปที่ 2-42 แสดงวิวัฒนาการของความลึกที่ใช้ในโครงข่ายประสาทแบบคอนโวลูชัน



รูปที่ 2-42 วิวัฒนาการของความลึกที่ใช้ในแต่ละสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชัน (He, 2017)

จากกราฟแท่งที่นำเสนอในรูปที่ 2-42 แสดงวิวัฒนาการของความลึกที่ใช้ในแต่ละสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันที่แข่งขันแล้วชนะและที่โดดเด่นจากงาน ILSVRC ImageNet Challenge ในแต่ละปี โดยค่าตัวเลขที่แสดงบนแท่งของกราฟเป็นค่าอัตราความผิดพลาดที่ได้จากการจำแนกภาพแบบ Top-5 Error Rate (วัดจากจำนวนภาพในชุดข้อมูลทดสอบที่เอาท์พุตจากการทำนายใน 5 ลำดับแรกของระบบหรือโครงข่ายไม่มีคำตอบที่ถูกต้อง นั่นคือไม่ตรงกับเอาต์พุตเป้าหมาย) จากข้อมูลภาพในชุดข้อมูล ImageNet โดยรายละเอียดของแต่ละสถาปัตยกรรมจะนำเสนอตามลำดับถัดจากสถาปัตยกรรมของ LeNet

สำหรับภาพในชุดข้อมูล ImageNet นั้นประกอบด้วยภาพความละเอียดสูงที่มีเอาต์พุตเป้าหมายบอกกำกับไว้ (Labeled High-Resolution Images) ว่าเป็นภาพที่จัดอยู่ในประเภทใด ซึ่งทั้งหมดในชุดข้อมูลมีประมาณ 15 ล้านภาพ แบ่งออกเป็นประเภท (Categorites) ของวัตถุต่าง ๆ ประมาณ 22,000 ประเภท แต่ในการแข่งขันในงาน ILSVRC แต่ละปีนั้นเลือกมาใช้เพียง 1,000 ประเภทหรือกลุ่ม (Class) และในแต่ละประเภทมีประมาณ 1,000 ภาพ โดยใช้ภาพในชุดฝึกสอนทั้งหมดประมาณ 1.2 ล้านภาพ ชุดตรวจสอบความสมเหตุสมผล (Validation Set) 50,000 ภาพ และชุดทดสอบ 150,000 ภาพ ผลการแข่งขัน รายงานผลเป็นอัตราความผิดพลาดจากการที่ให้โมเดลทำนายเอาต์พุตออกมาเป็นจำนวน 5 กลุ่มที่คาดว่าใกล้เคียงกับเอาต์พุตเป้าหมายมากที่สุดของแต่ละภาพในชุดข้อมูลทดสอบ และวัดออกมาเป็น Top-1 Error และ Top-5 Error เมื่อ Top-5 Error หรือ Rank-5 Error นั้นคำนวณมาจาก Rank- N Error Metric ที่มีค่า $N = 5$ โดย Rank- N Error คืออัตราส่วนของจำนวนข้อมูลทดสอบ x_i ซึ่งเอาต์พุตเป้าหมาย y_i ไม่ปรากฏอยู่ใน N เอาต์พุตแรกจากการทำนาย (Top- N Predicted Result) ของโมเดลเมื่อเอาต์พุตเรียงลำดับจากมากไปน้อยของค่าความเชื่อมั่น (Confidence) หรือ $P(y_i|x_i)$ ซึ่ง Error Metric (e) (Image-net, 2012) แสดงดังสมการที่ 2-12

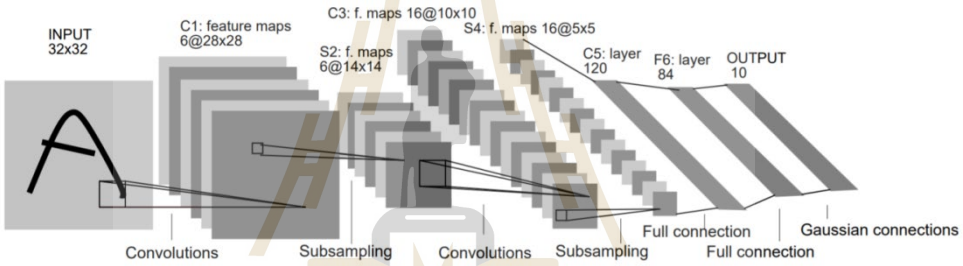
$$e = \frac{1}{n} \cdot \sum_{k} \min_{i} d(c_i, C_k) \qquad 2\text{-}12$$

$$\text{เมื่อ} \quad d(a,b) = \begin{cases} 0 & if\ a = b \\ 1 & otherwise \end{cases}$$

เมื่อ c_i คือ Class ที่ได้จากการทำนายและ C_k คือ Class จริง (Ground Truth Label) n คือจำนวนภาพในชุดข้อมูลทดสอบทั้งหมด และ $k = 1, \ldots, n$

2.5.1 สถาปัตยกรรมของ LeNet

เป็นสถาปัตยกรรมที่จัดว่าเป็นงานริเริ่ม (Pioneer Work) ของโครงข่ายประสาทแบบคอนโวลูชัน จากการนำเสนอโดย LeCun และคณะ (1998) ภายใต้ชื่อ LeNet-5 ในปี 1998 แต่ในเวลาต่อมามักเป็นที่รู้จักกันในชื่อ LeNet ที่มาของสถาปัตยกรรมของ LeNet นั้นมาจากการนำเสนอว่า ข้อมูลภาพนั้นไม่เหมาะกับการใช้อินพุตที่เป็นแต่ละจุดภาพแยกกันแบบที่ต้องใช้ในโครงข่ายประสาทแบบหลายชั้นดั้งเดิม เพราะข้อมูลภาพมีสหสัมพันธ์กันเชิงพื้นที่สูง (Highly Spatially Correlated) การใช้อินพุตที่เป็นแต่ละจุดภาพแยกกันจะไม่สามารถใช้ประโยชน์จากการมีสหสัมพันธ์กันดังกล่าว สถาปัตยกรรมของ LeNet จึงนำเสนอโดยใช้แนวคิดหลักจากการใช้การดำเนินการแบบคอนโวลูชัน แสดงดังรูปที่ 2-43 ซึ่งสถาปัตยกรรมนี้ใช้สำหรับการรู้จำข้อมูลตัวเลขที่เขียนด้วยมือ (Handwritten Digit Recognition) ในชุดข้อมูล MNIST



รูปที่ 2-43 สถาปัตยกรรมของ LeNet (LeCun et al., (1998)

รายละเอียดในแต่ละชั้นของ LeNet แสดงสรุปดังตารางที่ 2-5 ซึ่งประกอบด้วยชั้นซ่อนเร้น (ชั้นที่ไม่ได้เป็นชั้นอินพุตและเอาต์พุต) ทั้งหมด 6 ชั้น

ชั้นซ่อนเร้นที่ 1 เป็นชั้นการคอนโวลูชัน (C1) ที่ใช้ตัวกรองขนาด 5×5 ทั้งหมด 6 ตัวกรอง ไปคอนโวลูชันกับภาพอินพุตขนาด 32×32 (จริง ๆ แล้วภาพในชุดข้อมูล MNIST มีขนาด 28×28 แต่ใน LeNet ใช้การทำแพดดิ้งด้วยค่าศูนย์ให้กับภาพอินพุตก่อนนำเข้าสู่โครงข่าย) ด้วยขนาดการสไทรด์เท่ากับ 1 แล้วค่าจากการคอนโวลูชันนำไปผ่านฟังก์ชัน tanh เกิดเป็นเอาต์พุตของแผนที่คุณลักษณะขนาด 28×28 ทั้งหมด 6 แผนที่คุณลักษณะจากการใช้ 6 ตัวกรอง

ชั้นซ่อนเร้นที่ 2 เป็นชั้นการสุ่มต่ำกับข้อมูลด้วยการทำพูลลิ่ง (S2) แบบเฉลี่ยที่ใช้ขอบเขตแบบ 2×2 ด้วยขนาดการสไทรด์ เท่ากับ 2 แล้วค่าจากการพูลลิ่งนำไปผ่านฟังก์ชัน tanh เกิดเป็นเอาต์พุตของแผนที่คุณลักษณะขนาด 14×14 ทั้งหมด 6 แผนที่คุณลักษณะ (ทุก ๆ แผนที่คุณลักษณะจากชั้นก่อนหน้าจะถูกสุ่มต่ำกับข้อมูลด้วยการทำพูลลิ่ง)

ตารางที่ 2-5 รายละเอียดในแต่ละชั้นของ LeNet (LeCun et al., (1998)

Layer	Type	Maps	Size	Kernel Size	Stride	Activation
Out	Fully Connected	-	10	-	-	RBF
F6	Fully Connected	-	84	-	-	tanh
C5	Convolution	120	1×1	5×5	1	tanh
S4	Avg Pooling	16	5×5	2×2	2	tanh
C3	Convolution	16	10×10	5×5	1	tanh
S2	Avg Pooling	6	14×14	2×2	2	tanh
C1	Convolution	6	28×28	5×5	1	tanh
In	Input	1	32×32	-	-	-

ชั้นซ่อนเร้นที่ 3 เป็นชั้นการคอนโวลูชัน (C3) แบบเดียวกันกับชั้น C1 แต่ใช้ตัวกรองจำนวน 16 ตัวกรอง เช่นเดียวกันกับที่ ชั้นซ่อนเร้นที่ 4 เป็นชั้นการสุ่มต่ำกับข้อมูลด้วยการทำพูลลิ่ง (S4) ในลักษณะเดียวกันกับชั้น S2 ส่วนชั้นซ่อนเร้นที่ 5 เป็นชั้นสุดท้ายของการคอนโวลูชัน (C5) ที่สามารถมองภาพว่าเอาต์พุตของแผนที่คุณลักษณะทั้งหมดที่ได้ อยู่ในรูปของเวกเตอร์ที่มีจำนวน 120 องค์ประกอบ จากนั้นข้อมูลจากเวกเตอร์ดังกล่าวไปเชื่อมกับนิวรอนจำนวน 84 นิวรอนของชั้นซ่อนเร้นที่ 6 ในลักษณะแบบเชื่อมถึงกันหมด (F6) และชั้น F6 ก็เชื่อมกับชั้นเอาต์พุตแบบเชื่อมถึงกันหมดเช่นเดียวกัน

ชั้นเอาต์พุตใช้นิวรอนจำนวน 10 นิวรอนเพื่อจำแนกตัวเลขออกเป็น 10 กลุ่ม (คือตัวเลข 0-9) เมื่อฟังก์ชันถ่ายโอนที่ใช้เป็นแบบ Radial Basis Function (RBF) นั่นคือในชั้นเอาต์พุตจะไม่ใช้การคูณกันจุดต่อจุดระหว่างค่าเวกเตอร์ของอินพุตกับเวกเตอร์ค่าน้ำหนักแล้วนำมาผ่านฟังก์ชันถ่ายโอนแบบที่ใช้โดยทั่วไป แต่เอาต์พุตของแต่ละนิวรอนจะคำนวณมาจากการยกกำลังสองของค่าระยะทางแบบยูคลิเดียน (Euclidian Distance) ระหว่างเวกเตอร์ของอินพุตกับเวกเตอร์ค่าน้ำหนักแทน

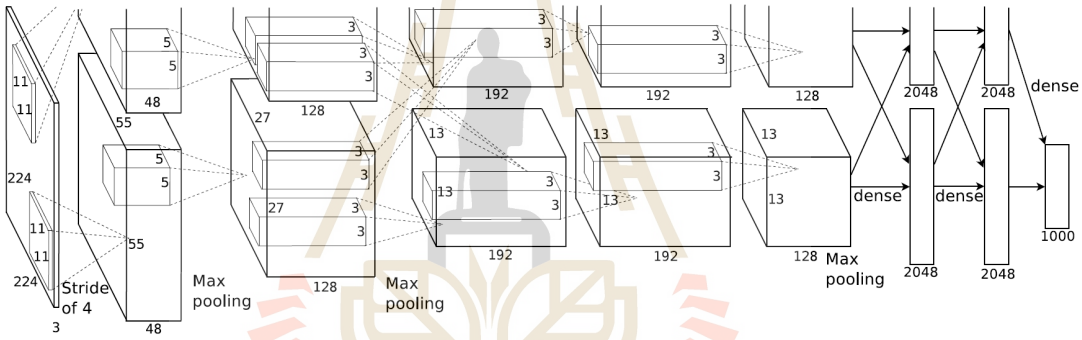
LeNet นั้นนำเสนอขึ้นมาในยุคที่ความเร็วของ CPU ยังไม่สูงและยังไม่มีการนำ GPU มาใช้ แต่ก็มีการนำไปประยุกต์ใช้งานจริงสำหรับการรู้จำรหัสไปรษณีย์ (Zip Code) และ

ตัวเลข และเป็นสถาปัตยกรรมที่จัดว่าเป็นจุดตั้งต้นและเป็นแรงจูงใจสำหรับให้เกิดการพัฒนาสถาปัตยกรรมอื่นๆ ในเวลาต่อมา

2.5.2 สถาปัตยกรรมของ AlexNet

เป็นสถาปัตยกรรมที่ทำให้โครงข่ายประสาทแบบคอนโวลูชันเริ่มต้นเป็นที่รู้จักอย่างแพร่หลายโดยเฉพาะในงานทางด้านการมองเห็นของเครื่อง โดย Krizhevsky และคณะ (2012) ได้ใช้สถาปัตยกรรมในชื่อ AlexNet ดังแสดงในรูปที่ 2-44 สำหรับการแข่งขันในงาน ILSVRC 2012 เพื่อจำแนกข้อมูลภาพจากชุดข้อมูล ImageNet

รายละเอียดในแต่ละชั้นของสถาปัตยกรรม AlexNet แสดงดังตารางที่ 2-6 ซึ่งจะคล้ายคลึงกับ LeNet แต่เป็นโครงข่ายที่ใหญ่กว่าและมีความลึกมากกว่า ซึ่งคุณลักษณะสำคัญของ AlexNet คือ



รูปที่ 2-44 สถาปัตยกรรมของ AlexNet (Krizhevsky et al., 2012)

- นำฟังก์ชันถ่ายโอนแบบ ReLU มาใช้เป็นครั้งแรกในชั้น CONV1, 2, 3, 4, 5 และ FC6, 7
- ใช้ฟังก์ชัน Softmax ในชั้น FC8
- มีชั้น Normalization ที่เป็นการนอร์มอลไลซ์ค่าจากแผนที่คุณลักษณะที่อยู่ใกล้ ๆ กัน
- ใช้การทำพูลลิ่งแบบ Overlapping Max Pooling โดยกำหนดให้ขนาดการสไทรด์ในแนวนอนน้อยกว่าในแนวตั้ง
- นำหลักการ Data Augmentation มาใช้เพื่อเพิ่มความเป็นกรณีทั่วไป
- นำเทคนิค Dropout มาใช้ในชั้น FC ด้วยค่า 0.5 (ลดจำนวนนิวรอนลงครึ่งหนึ่งแบบสุ่ม)

- ใช้ Batch Size ขนาด 128
- ใช้ค่า SGD Momentum เป็น 0.9
- ใช้ค่าคงที่การเรียนรู้เริ่มต้นเป็น 0.01 และลดค่าลงครั้งละ 10 เท่าด้วยมือ (Manually)
- ใช้หลักการ Regularization แบบ L2 Weight Decay
- ฝึกสอนโดยใช้ 2 GPUs (GTX 580 GPU) และใช้หน่วยความจำ 3GB โดยแบ่งนิวรอน (แผนที่คุณลักษณะ) เป็นครึ่งหนึ่งในแต่ละ GPU นั่นคือ
  - CONV1, CONV2, CONV4, CONV5 : มีการเชื่อมกันเฉพาะแผนที่คุณลักษณะที่อยู่บน GPU เดียวกัน (ติดต่อสื่อสารบน GPU เดียวกัน)
  - CONV3, FC6, FC7, FC8 : มีการเชื่อมกันกับทุกแผนที่คุณลักษณะในชั้นก่อนหน้า (ติดต่อสื่อสารข้าม GPU)

ตารางที่ 2-6 รายละเอียดในแต่ละชั้นของ AlexNet (Li et al., 2017)

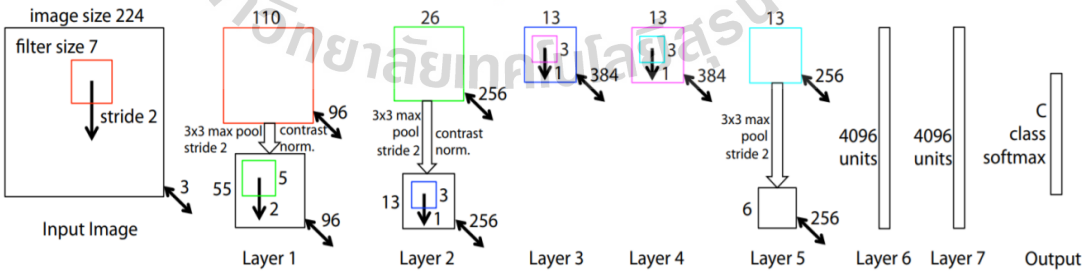
Feature Maps	**Layer**	**Details**
[227×227×3]	INPUT	
[55×55×96]	CONV1	96 11×11 filters at stride 4, pad 0
[27×27×96]	MAX POOL1	3×3 filters at stride 2
[27×27×96]	NORM1	Normalization layer
[27×27×256]	CONV2	256 5×5 filters at stride 1, pad 2
[13×13×256]	MAX POOL2	3×3 filters at stride 2
[13×13×256]	NORM2	Normalization layer
[13×13×384]	CONV3	384 3×3 filters at stride 1, pad 1
[13×13×384]	CONV4	384 3×3 filters at stride 1, pad 1
[13×13×256]	CONV5	256 3×3 filters at stride 1, pad 1
[6×6×256]	MAX POOL3	3x3 filters at stride 2
[4096]	FC6	4096 neurons
[4096]	FC7	4096 neurons
[1000]	FC8	1000 neurons (class scores)

สถาปัตยกรรม AlexNet นั้นชนะการแข่งขันในงาน ILSVRC 2012 เพื่อจำแนกข้อมูลภาพจากชุดข้อมูล ImageNet ด้วย Top-5 Error Rate ที่ 16.4 ในขณะที่ผู้ชนะอันดับถัดมาได้ Top-5 Error Rate ที่ 26.2 นั่นคือต่างกันถึงประมาณ 10% ทำให้โครงข่ายประสาทแบบคอนโวลูชันเป็นดาวรุ่งของโครงข่ายการเรียนรู้เชิงลึกตั้งแต่นั้นมา

2.5.3 สถาปัตยกรรมของ ZF Net

เป็นสถาปัตยกรรมที่นำเสนอโดย Zeiler และ Fergus (2013) ดังแสดงในรูปที่ 2-45 และชนะการแข่งขันในงาน ILSVRC 2013 ด้วย Top-5 Error Rate ที่ 11.7% ซึ่งเป็นโครงข่ายที่ปรับปรุงมาจาก AlexNet ด้วยการปรับเปลี่ยนไฮเปอร์พารามิเตอร์ในสถาปัตยกรรม (Tweaking Architecture Hyperparameters) เมื่อไฮเปอร์พารามิเตอร์ หมายถึง จำนวนตัวกรองที่ใช้ในแต่ละชั้น ขนาดในแนวกว้างและแนวยาวของตัวกรอง ขนาดของการสไทรด์ หรือ ชนิดของการแพดดิ้ง เป็นต้น ซึ่งการปรับเปลี่ยนหลักคือ ในชั้นแรก ๆ ของชั้นการคอนโวลูชันจะใช้ขนาดตัวกรองและขนาดการสไทรด์ ที่เล็กลง และในชั้นกลาง ๆ ของชั้นการคอนโวลูชันจะใช้จำนวนตัวกรองเพิ่มขึ้น นั่นคือ ZF Net นั้นคือ AlexNet ที่มีการปรับเปลี่ยนคือ

- CONV1 (Layer1) : เปลี่ยนจากตัวกรองขนาด 11×11 ด้วยขนาดการสไทรด์เท่ากับ 4 ไปเป็นตัวกรองขนาด 7×7 ด้วยขนาดการสไทรด์ เท่ากับ 2
- CONV2 (Layer2) : จากจำนวนตัวกรองคือ 256 เปลี่ยนไปใช้ด้วยจำนวนตัวกรอง 512
- CONV5 (Layer5) : จากจำนวนตัวกรองคือ 256 เปลี่ยนไปใช้ด้วยจำนวนตัวกรอง 512

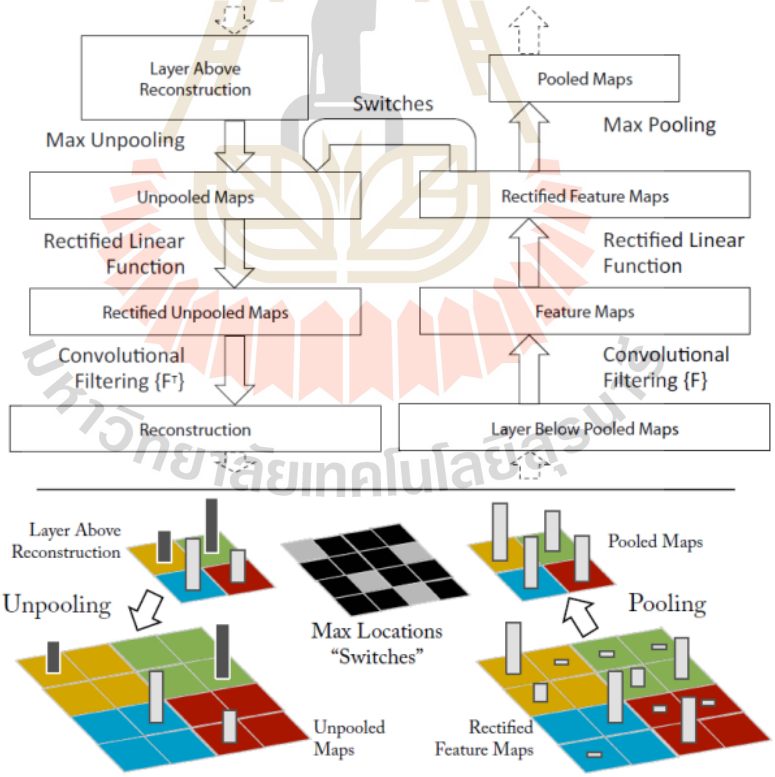


รูปที่ 2-45 สถาปัตยกรรมของ ZF Net (Zeiler and Fergus, 2013)

นอกเหนือจากที่ ZF Net ประสบความสำเร็จในงาน ILSVRC 2013 จากการปรับเปลี่ยนสถาปัตยกรรมจาก AlexNet ดังกล่าวแล้ว ในงานของ Zeiler และ Fergus ยังได้นำเสนอ

สิ่งที่น่าสนใจมาก ในส่วนของการแสดงผลแผนที่คุณลักษณะ (Visualizing Feature Maps) นั่นคือ ได้พัฒนาเทคนิคการแสดงผล (Visualization Technique) ภายใต้ชื่อ “Deconvolutional Network” เพื่อช่วยในการตรวจสอบความแตกต่างของค่าที่เกิดขึ้นในแผนที่คุณลักษณะและความสัมพันธ์ระหว่างแผนที่คุณลักษณะกับข้อมูลอินพุต ที่มาของชื่อ “Deconvnet” เพราะเป็นการแปลง (Map) จากคุณลักษณะไปยังจุดภาพ (Features to Pixels) ซึ่งตรงกันข้ามกันกับการทำคอนโวลูชัน (Deshpande, 2016) โดยเทคนิคการแสดงผลภายใต้ชื่อ Deconvolutional Network ดังกล่าวแสดงดังรูปที่ 2-46

เทคนิคการแสดงผลภายใต้ชื่อ Deconvolutional Network ในรูปที่ 2-46 นั้น ด้านซ้ายของรูปเป็นชั้น Deconvnet ที่นำมาแนบเพิ่มไปกับชั้น Convnet ที่อยู่ด้านขวา โดยชั้น Deconvnet จะทำการกู้คืน (Reconstruct) ค่าโดยประมาณจากคุณลักษณะของชั้น Convnet ที่อยู่ข้างใต้ (Beneath) สำหรับด้านล่างของรูปที่ 2-46 แสดงการดำเนินการแบบ Unpooling ของ Deconvnet โดยการใช้ Switches สำหรับบันทึกตำแหน่งของ Local Max ในแต่ละขอบเขตของการ พูลลิ่ง



รูปที่ 2-46 Deconvolutional Network ที่ใช้สำหรับการแสดงผลแผนที่คุณลักษณะ (Zeiler and Fergus, 2013)

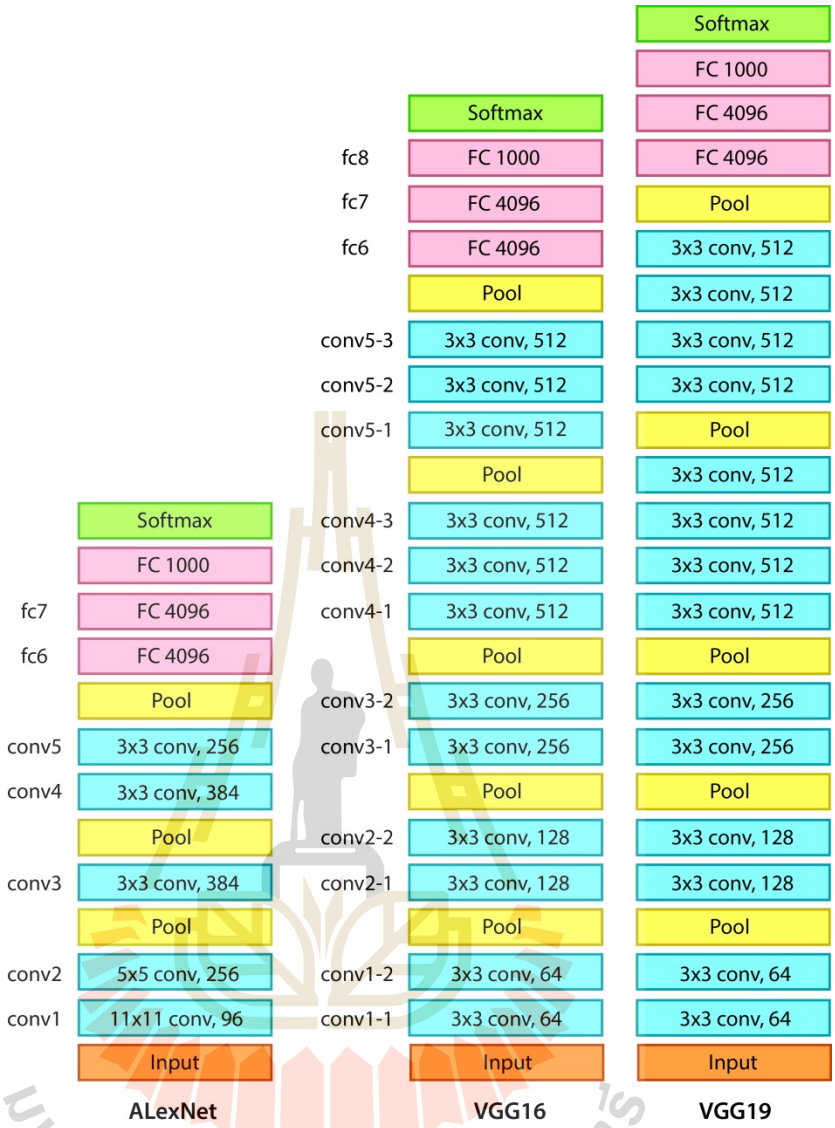
เทคนิคการแสดงผลที่นำเสนอนี้มีประโยชน์มากสำหรับแสดงภาพการทำงานของโครงข่ายแบบคอนโวลูชัน ในการช่วยสำหรับการวิเคราะห์เพื่อเพิ่มประสิทธิภาพให้กับโครงข่ายได้ นั่นคือเทคนิคการแสดงผลนี้ไม่เพียงแค่ช่วยอธิบายการทำงานภายในของโครงข่ายเท่านั้น แต่ยังช่วยให้สามารถมองลึกเข้าไปข้างในเพื่อปรับปรุงสถาปัตยกรรมของโครงข่ายได้ (Deshpande, 2016)

2.5.4 สถาปัตยกรรมของ VGG Net

สถาปัตยกรรมของ VGG Net นำเสนอโดย Simonyan และ Zisserman (2014) ที่มาจากแนวคิดหลักคือการใช้ตัวกรองขนาดเล็กและโครงข่ายที่ลึกขึ้น ดังแสดงในรูปที่ 2-47 ที่เป็นสถาปัตยกรรมของ VGG Net แบบ VGG16 (16 Weight Layers) และ VGG19 (19 Weight Layers) เมื่อเปรียบเทียบกับ AlexNet

คุณลักษณะสำคัญของ VGG Net คือ

- แต่ละชั้นของการคอนโวลูชันใช้เฉพาะตัวกรองขนาด 3×3 ด้วยขนาดการสไทรด์ เท่ากับ 1 และทำแพดดิ้งด้วยค่าศูนย์ขนาดเท่ากับ 1
- แต่ละชั้นของการทำพูลลิ่งใช้เฉพาะพูลลิ่งแบบแมกซ์ด้วยขอบเขตแบบ 2×2 และขนาดการสไทรด์ เท่ากับ 2
- จำนวนตัวกรองในชั้นคอนโวลูชันจะเพิ่มขึ้นเป็นสองเท่าหลังจากชั้นการทำพูลลิ่ง นั่นคือเป็นการสนับสนุนแนวคิดการใช้องค์ประกอบเชิงพื้นที่น้อยลง (Shrinking Spatial Dimensions) แต่เพิ่มในทางลึกมากขึ้น (Growing Depth)
- สามารถใช้ได้ดีกับทั้งการจำแนกภาพและงานการระบุตำแหน่งของวัตถุ (Object Localization Tasks) นั่นคือ VGG19 ชนะเลิศสำหรับงานการระบุตำแหน่งของวัตถุในการแข่งขัน ILSVRC 2014 และชนะเป็นลำดับที่สอง (7.3% Top-5 Error Rate) สำหรับงานการจำแนกภาพในปีเดียวกัน ซึ่งผู้ชนะเลิศในงานการจำแนกภาพของปี 2014 นั้นคือสถาปัตยกรรมของ Google Net (6.7% Top-5 Error Rate) ที่จะนำเสนอในลำดับถัดไป
- ใช้เทคนิค Scale Jittering เป็นเทคนิคหนึ่งในการทำ Data Augmentation
- ใช้ฟังก์ชันถ่ายโอนแบบ ReLU หลังจากแต่ละชั้นของการคอนโวลูชัน
- ขั้นตอนในการฝึกสอนใช้ลักษณะเดียวกันกับ AlexNet
- ไม่มีการทำ Local Response Normalisation (LRN)
- ฝึกสอนโดยใช้ Nvidia Titan Black GPU จำนวน 4 GPUs ใช้เวลาประมาณ 2-3 สัปดาห์

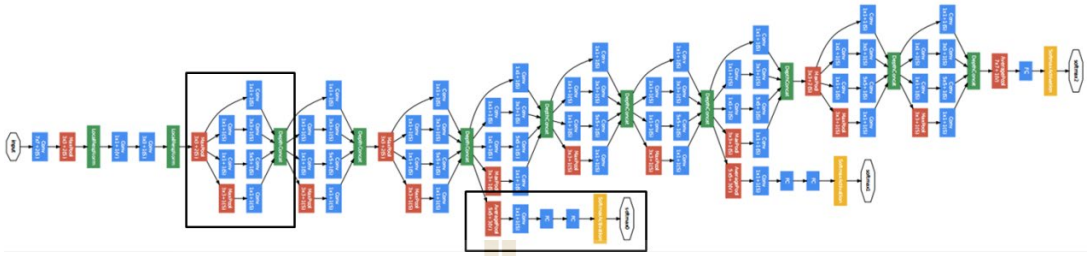


รูปที่ 2-47 สถาปัตยกรรมของ VGG Net เมื่อเปรียบเทียบกับ AlexNet (Li et al., 2017)

2.5.5 สถาปัตยกรรมของ GoogleNet

เป็นสถาปัตยกรรมที่พัฒนาโดย Szegedy และคณะ (2014) จากทีมของ Google Research ที่ชนะการแข่งขันในงานการจำแนกภาพของ ILSVRC 2014 จากการใช้โครงข่ายที่มีความลึกมากกว่าโครงข่ายอื่น ๆ ที่ผ่านมา ดังแสดงในรูปที่ 2-48 ในโครงข่ายหลักจะประกอบด้วยโครงข่ายย่อยที่เรียกว่า Inception module (แสดงด้วยกรอบสี่เหลี่ยมแนวตั้งในรูปที่ 2-48) ทั้งหมด 9 Module โดยที่ชั้นต่าง ๆ ในแต่ละ Inception Module ทำงานกันแบบ Parallel และประกอบด้วยอีกสองโครงข่ายย่อยที่เรียกว่า Auxiliary Module (แสดงด้วยกรอบสี่เหลี่ยมแนวนอนในรูปที่ 2-48) ซึ่ง

ทั้งหมดของโครงข่ายนับว่ามีความลึก 22 ชั้น เมื่อนับเฉพาะชั้นที่มีค่าของ Weights และ Biases ดังแสดงรายละเอียดในแต่ละชั้นดังกล่าวของสถาปัตยกรรม GoogleNet ในตารางที่ 2-7



รูปที่ 2-48 สถาปัตยกรรมของ GoogleNet (Szegedy et al., 2014)

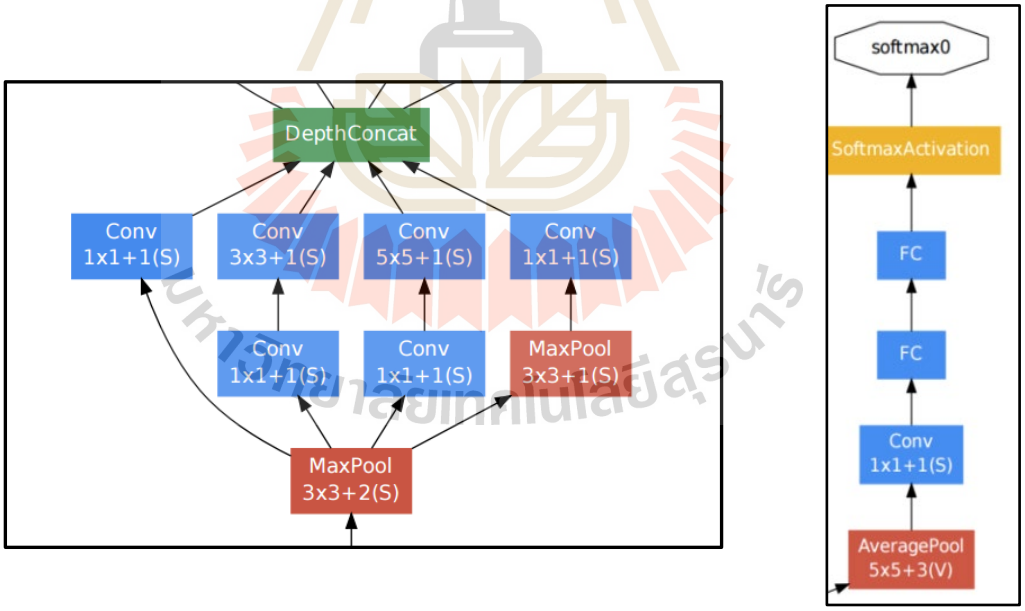
แนวคิดหลักของ GoogleNet คือต้องการปรับปรุงการใช้ทรัพยากรในการคำนวณ (Computing Resources) ภายในโครงข่ายให้มีประสิทธิภาพมากยิ่งขึ้นด้วยการเพิ่มขนาดของโครงข่ายทั้งในแนวกว้างและแนวลึก ซึ่งการคำนวณที่มีประสิทธิภาพนั้นเป็นผลมาจากการใช้ Inception module ในโครงข่าย โดยจำนวนพารามิเตอร์ทั้งหมดที่ใช้ (จำนวนของค่าน้ำนำหนักและไบอัสทั้งหมด) มีประมาณ 5 ล้านพารามิเตอร์ ซึ่งน้อยกว่าของ AlexNet ประมาณ 12 เท่า (Li et al., 2017) ทำให้ GoogleNet นี้ชนะเลิศในงานการจำแนกภาพของ ILSVRC 2014 ที่ 6.7% Top-5 Error

จากรายละเอียดภายในของสถาปัตยกรรมที่แสดงในตารางที่ 2-7 จะเห็นว่าโครงข่ายมีการนำ Inception Module ทั้งหมด 9 Modules มาใช้ในชั้นหลักที่ 3, 4 และ 5 โดยมองว่าแต่ละ Inception Module มีความลึกเป็น 2 ดังนั้นความลึกทั้งหมด 22 ชั้นของโครงข่ายมาจากการนับเฉพาะจำนวนชั้นของการคอนโวลูชัน (ค่าในคอลัมน์ Depth ในตาราง) และชั้นที่เป็น Linear ก่อนชั้นการแปลงค่าด้วยฟังก์ชัน Softmax ซึ่งเป็นการนับเฉพาะชั้นที่มีการใช้พารามิเตอร์ (ชั้นที่มีค่าน้ำหนักและไบอัส) แต่ถ้านับจำนวนชั้นย่อย ๆ ทั้งหมด (จำนวน Block ย่อย ๆ ทั้งหมดในรูปที่ 2-48) แล้วจะมีทั้งหมดประมาณ 100 ชั้น

สำหรับส่วนของ Inception Module และ Auxiliary Module ที่แสดงด้วยกรอบสี่เหลี่ยมในรูปที่ 2-48 นั้นมีรายละเอียดเมื่อขยายออกมาแสดงดังรูปที่ 2-49 ในส่วนรายละเอียดของ Inception Module ที่แสดงในรูปที่ 2-49(a) นั้น Block ด้านล่างสุดที่เป็นการทำพูลลิ่งแบบแมกซ์ของ $3\times3+2(S)$ หมายถึงการทำพูลลิ่งแบบแมกซ์ในขอบเขตแบบ 3×3 ด้วยขนาดการสไทรด์เท่ากับ 2 และทำแพดดิ้งแบบ SAME (คือการทำแพดดิ้งด้วยค่าศูนย์ให้เอาต์พุตมีขนาดเท่ากันกับอินพุต)

ตารางที่ 2-7 รายละเอียดภายในของสถาปัตยกรรม GoogleNet (Szegedy et al., 2014)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

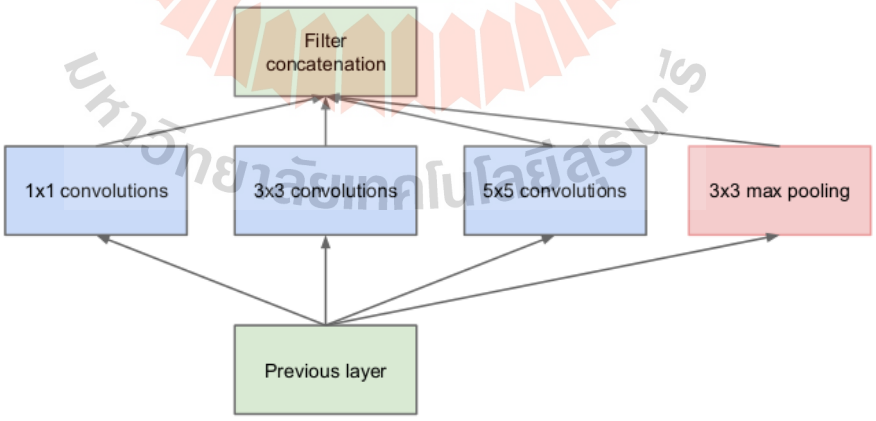


(a) Inception Module (b) Auxiliary Module

รูปที่ 2-49 ภาพขยายแสดงรายละเอียดของสองโครงข่ายย่อยภายใน GoogleNet (Li et al., 2017)

เนื่องจากขนาดของตัวกรองจัดว่าเป็นไฮเปอร์พารามิเตอร์หลักอย่างหนึ่งที่ต้องเลือกใช้ในโครงข่ายแบบคอนโวลูชันโดยทั่วไปว่าในแต่ละชั้นการคอนโวลูชันที่ต่อเนื่องกันควรใช้ขนาดเป็นเท่าไร แต่สำหรับการออกแบบในส่วน Inception Module ของ GoogleNet นั้นใช้แนวคิดว่า ทดลองทำในทุก ๆ ขนาด (โดยสังเกตว่าจะใช้ขนาดที่เล็ก เช่น $1\times1, 3\times3$ หรือ 5×5 ที่เป็นผลสืบเนื่องมาจากการใช้ในสถาปัตยกรรมที่นำเสนอก่อนหน้านี้แล้วว่าดีกว่าขนาดที่ใหญ่) แล้วให้โครงข่ายเลือกเองว่าอะไรดีที่สุด (Mulc, 2016) ซึ่งเป็นการให้โมเดลตรวจจับได้ทั้ง Local Features จากการใช้ขนาดตัวกรองที่เล็กและ High Abstracted Features จากขนาดตัวกรองที่ใหญ่กว่า ดังนั้นจึงเกิดเป็น Inception Module ขึ้นในโมเดล ที่ในแต่ละขั้นตอนย่อยของ Inception Module สามารถทำงานได้แบบ Parallel

จาก Inception Module ในรูปที่ 2-49(a) จะสังเกตได้ว่าก่อนขั้นตอนการคอนโวลูชันแบบ $3\times3, 5\times5$ และหลังขั้นตอนการทำพูลลิ่งแบบแมกซ์จะเป็นขั้นตอนการคอนโวลูชันแบบ 1×1 (โดยขั้นตอนการทำการคอนโวลูชันแบบ 1×1 ในลักษณะเช่นนี้ใน GoogleNet เรียกว่า 1×1 Conv “Bottleneck” Layer) ซึ่งถือว่าเป็นกุญแจสำคัญของการออกแบบในส่วน Inception Module ของ GoogleNet สำหรับการช่วยลดมิติของข้อมูล (Dimensional Reduction) ในแผนที่คุณลักษณะที่เป็นอินพุต และช่วยลดการคำนวณลงได้มาก นั่นคือถ้าไม่มีการใช้ส่วนของการคอนโวลูชันแบบ 1×1 ดังกล่าว จะเป็นลักษณะของ Naive Inception Module ที่แสดงในรูปที่ 2-50 ที่จำนวน Operation ทั้งหมดที่จะต้องคำนวณจะสูงกว่ากันมาก



รูปที่ 2-50 Naive Inception Module (Szegedy et al., 2014)

นอกเหนือจากที่ช่วยลดมิติของข้อมูลแล้ว ประเด็นหลักของการใช้การคอนโวลูชันแบบ 1×1 คือสามารถช่วยลดการคำนวณลงได้มาก ยกตัวอย่างดังรูปที่ 2-51 ที่เป็นการยกตัวอย่างส่วนหนึ่งของการคอนโวลูชันในชั้น 3(a) จากตารางที่ 2-8 ของ GoogleNet (ชั้นที่มีกรอบสี่เหลี่ยมกำกับไว้) เมื่อทำการคอนโวลูชันแบบ 5×5 ในชั้นดังกล่าวในกรณีที่เป็นแบบ Naive Inception Module (แสดงดังรูปที่ 2-51(a)) และในกรณีที่เป็นแบบ Inception Module ที่ใช้ใน GoogleNet ที่ทำการคอนโวลูชันแบบ 1×1 ก่อนแล้วจึงตามด้วยแบบ 5×5 (แสดงดังรูปที่ 2-51(b))

จากการคอนโวลูชันที่แสดงในรูปที่ 2-51 ถ้าคำนวณออกมาเป็นจำนวน Operation ทั้งหมดที่ต้องใช้ของแต่ละแบบ จะพบว่า

– แบบ Naive Inception Module ที่แสดงในรูปที่ 2-51(a)

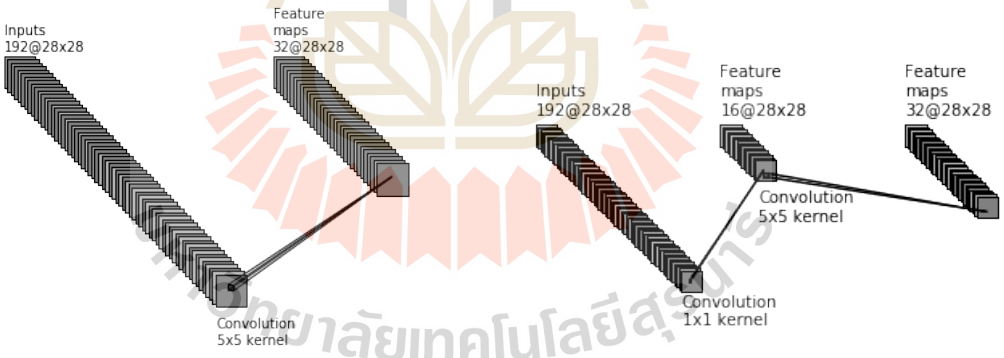
จำนวน Operations $= 5^2*(28)^2*(192)*(32)$ $= 120,422,400$ *Operations*

– แบบ Inception Module ของ GoogleNet ที่แสดงในรูปที่ 2-51(b)

จำนวน Operations $= [1^2*(28)^2*(192)*(16)] + [5^2*(28)^2*(16)*(32)]$

$= 2,408,448 + 10,035,200$

$= 12,443,648$ *Operations*



(a) (b)

รูปที่ 2-51 ส่วนหนึ่งของการคอนโวลูชันในชั้น 3(a) ของ GoogleNet (Mulc, 2016)

(a) การคอนโวลูชันแบบ 5×5 ใน Naive Inception Module

(b) การคอนโวลูชันแบบ 1×1 ตามด้วย 5×5 ใน Inception Module ของ GoogleNet

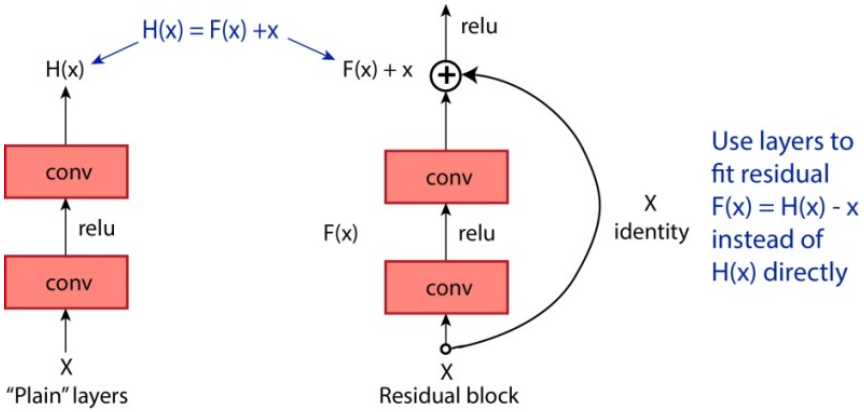
จะพบว่าจำนวน Operation ที่ใช้ใน Inception Module ของ GoogleNet นั้นน้อยกว่าที่ใช้ในแบบ Naive Inception Module ประมาณสิบเท่า อันเป็นผลมาจากการใช้การคอนโวลูชันแบบ 1x1 มาช่วย นอกจากนี้จากตารางที่ 2-8 จะเห็นว่าในโครงข่ายใช้การพูลลิ่งแบบเฉลี่ยหลังจาก Inception Module สุดท้ายของโครงข่าย (หลังจากชั้น Inception 5(b)) แทนที่จะเป็นชั้นแบบ FC ที่ใช้โดยทั่วไปนั้นก็สามารถลดจำนวนพารามิเตอร์ลงได้อีกมากเช่นกัน ซึ่งหลังจากที่ GoogleNet นำเสนอโมเดลแบบ 22 ชั้นนี้ในงาน ILSVRC 2014 แล้วได้มีการพัฒนาต่ออีกหลาย Version ตามมา โดย Version ล่าสุดคือ Inception-v4

2.5.6 สถาปัตยกรรมของ ResNet

เป็นสถาปัตยกรรมที่ชนะการแข่งขันใน ILSVRC 2015 ในชื่อ Residual Network (หรือ ResNet) (He et al., 2015) ที่พัฒนาขึ้นโดย He และคณะ ที่สามารถทำให้ Top-5 Error Rate ลดลงมาอยู่ที่ 3.57% จากการใช้โครงข่ายที่มีความลึกมากถึง 152 ชั้น ซึ่งจริง ๆ แล้วจากการทดลองโดย He และคณะพบว่า ถ้าจะสร้างโครงข่ายในลักษณะเชิงลึกแบบ “Plain” จากการนำชั้นต่าง ๆ มาซ้อนต่อกันไปเรื่อย ๆ นั้น ณ ความลึกที่ 56 ชั้นพบว่าค่าความผิดพลาดที่เกิดขึ้นจะแย่ลงทั้งผิดพลาดจากการฝึกสอนและการทดสอบ ซึ่งไม่ได้มีสาเหตุมาจากการเกิด Overfitting แต่เป็นเพราะว่าจากปัญหาการ Optimization เพราะถ้าโครงข่ายยิ่งลึกจะยิ่งยากในการ Optimize จึงมีสมมุติฐานสำหรับออกแบบโครงข่ายในลักษณะที่ว่า โมเดลที่ลึกกว่านั้นอย่างน้อยที่สุดแล้วควรจะสามารถทำงานได้ดีพอ ๆ กับโมเดลที่ตื้นกว่า จึงเป็นที่มาของการสร้างโครงข่ายจากการทำซ้ำ (Copying) ชั้นการเรียนรู้จากโมเดลที่อยู่ตื้นกว่าและมีชั้นที่สร้างเพิ่มขึ้นมาใหม่กำหนดให้เป็น Identity Mapping ซึ่งแนวคิดดังกล่าวแสดงดังรูปที่ 2-52 ที่เป็นการนำ Residual Block มาใช้ในโครงข่ายแทนที่จะนำแต่ละชั้นมาต่อ ๆ กันในลักษณะ Plain Layer

รูปที่ 2-52 เป็นการอธิบายว่าในการฝึกสอนโครงข่ายโดยทั่วไปนั้นมีเป้าหมายคือให้โครงข่ายโมเดล Target Function H(x) แต่ถ้าเราบวกอินพุต x ให้กับเอาต์พุตของโครงข่าย (นั่นคือเป็นการเพิ่ม Skip Connection เข้าไป) จะทำให้โครงข่ายถูกบังคับให้โมเดล F(x) = H(x) – x แทนที่จะให้โมเดล H(x) โดยตรง ลักษณะการโมเดลแบบนี้เรียกกว่า Residual Learning

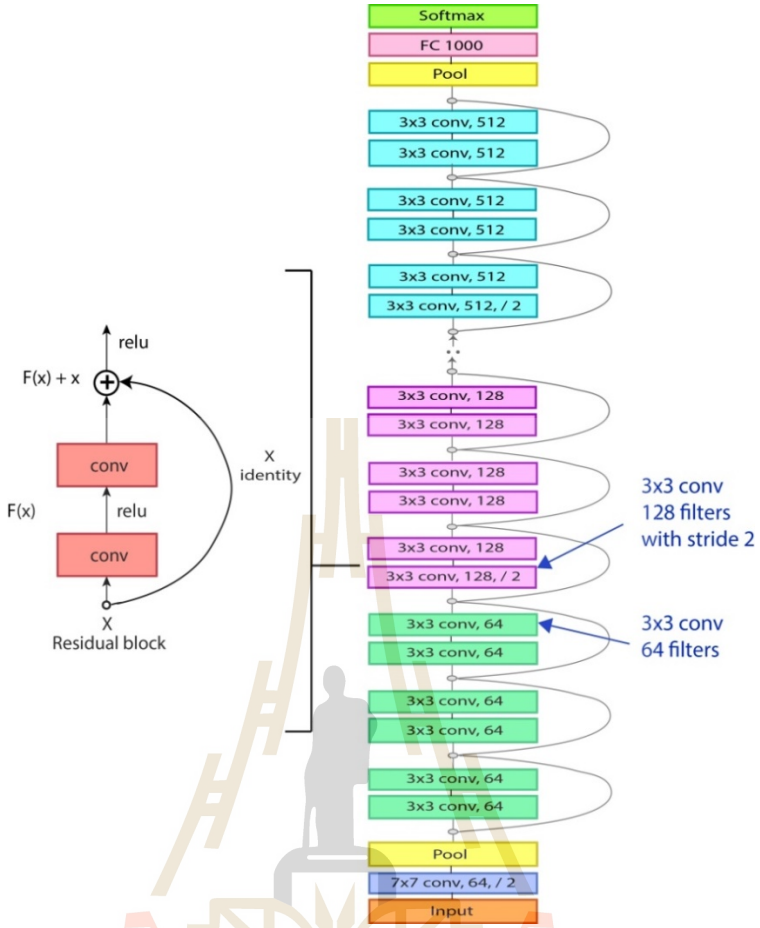
จากลักษณะการโมเดลของแนวคิด Residual Learning โดยการใช้ Residual Block ดังกล่าว สถาปัตยกรรมของ ResNet จึงนำเสนอขึ้นมาจากการนำหลาย ๆ Residual Blocks มาซ้อนกัน ดังรูปที่ 2-53 ด้านขวา ที่มีคุณลักษณะหลักคือ



รูปที่ 2-52 แนวคิดของการใช้ Residual Block ใน ResNet (Li et al., 2017)

- ทุก Residual Block มีชั้นการคอนโวลูชันสองชั้นที่ใช้ตัวกรองแบบ 3×3 ทั้งสองชั้น
- มีการเพิ่มจำนวนตัวกรองเป็นสองเท่าและทำการสุ่มต่ำด้วยขนาดการสไทรด์เท่ากับสองเป็นช่วง ๆ
- มีชั้นการคอนโวลูชันหนึ่งชั้นตอนเริ่มต้น
- ไม่มีการใช้ชั้น FC หลังชั้นคอนโวลูชันสุดท้าย

ความลึกที่ใช้สำหรับการทดลองในชุดข้อมูล ImageNet คือกำหนดเป็น 18, 34, 50, 101 และ 152 ชั้นดังแสดงรายละเอียดในตารางที่ 2-8 โดยที่ความลึก 152 ชั้นคือโมเดลที่ได้ Top-5 Error Rate เป็น 3.57% จากตารางที่ 2-8 จะเห็นว่าโมเดลที่ใช้ระดับความลึกเป็น 18 และ 34 นั้นในแต่ละResidual Block ประกอบด้วย 2 ชั้นย่อยที่เป็นการคอนโวลูชันแบบ 3×3 ทั้งหมด แต่สำหรับโมเดลที่ใช้ระดับความลึกเป็น 50, 101 และ 152 นั้น แต่ละResidual Block จะประกอบด้วย 3 ชั้นย่อยที่เป็นการคอนโวลูชันแบบ 1×1 แบบ 3×3 และแบบ 1×1 ตามลำดับ จะเห็นได้ว่าเมื่อใช้ระดับความลึก 50 ชั้นขึ้นไป มีการนำแนวคิดของ “Botteneck” Layer จากการนำการคอนโวลูชันแบบ 1×1 มาใช้ในส่วนของ Residual Blocks เพื่อเพิ่มประสิทธิภาพให้กับโครงข่ายเช่นเดียวกันกับใน GoogleNet

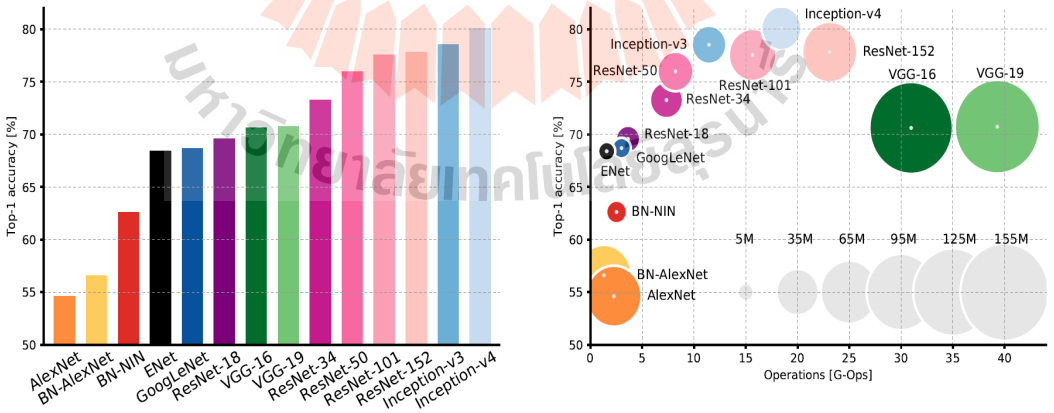


รูปที่ 2-53 สถาปัตยกรรมของ ResNet (Li et al., 2017)

ตารางที่ 2-8 รายละเอียดของโครงข่ายที่ใช้ความลึกที่แตกต่างกันสำหรับทดลองกับชุดข้อมูล ImageNet (He et al., 2015)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ที่กล่าวมาทั้งหมดในหัวข้อนี้เป็นรายละเอียดของแต่ละโครงข่ายประสาทแบบคอนโวลูชันที่เด่น ๆ และประสบความสำเร็จจากการนำไปใช้งาน ILSVRC ซึ่งเมื่อไม่นานมานี้ Canziani และคณะ (2017) ได้ทำการศึกษาเปรียบเทียบและวิเคราะห์โมเดลต่าง ๆ ของโครงข่ายประสาทเชิงลึกสำหรับการประยุกต์ใช้ในทางปฏิบัติที่มีโครงข่ายดังที่ได้กล่าวมาแล้วรวมอยู่ด้วย รวมถึงบางโครงข่ายที่มีการพัฒนาต่อยอดเพิ่มขึ้นมา ผลการวิเคราะห์ดังกล่าวแสดงด้วยกราฟเปรียบเทียบกันดังรูปที่ 2-54 ซึ่งพบว่า โมเดล Inception-v4 ที่พัฒนาต่อยอดมาจาก GoogleNet และ ResNet มีประสิทธิภาพดีที่สุด จากการวัดค่าแบบ Top-1 Accuracy (จากที่แสดงด้วยกราฟแท่งทางซ้าย) โดย Inception-v4 นั้นเป็นการผสมผสานระหว่างแนวคิด Residual Learning จาก ResNet ร่วมกับแนวคิด Inception Module ของ GoogleNet ส่วนทางด้านขวาเป็นการแสดงแผนภาพวงกลมที่บอกถึงขนาดของจำนวน Operation ที่ต้องใช้ในการคำนวณสำหรับการทำงานหนึ่งรอบของขั้นตอนส่งผ่านค่าไปข้างหน้าของแต่ละโมเดล เช่นเป็นการบอกว่า VGG Net นั้นต้องใช้หน่วยความจำมากที่สุดจากการที่ต้องมีจำนวน Operation มากที่สุด GoogleNet จัดว่ามีประสิทธิภาพมากที่สุดจากการได้ค่าความถูกต้องที่สูงที่สุด ในส่วนของ AlexNet นั้นถึงจะมีการคำนวณน้อยกว่า VGG Net แต่ก็ยังต้องใช้หน่วยความจำสูงและมีความถูกต้องน้อย ส่วน ResNet นั้นมีประสิทธิภาพในระดับกลาง ๆ ขึ้นอยู่กับแต่ละระดับความลึกของโมเดล แต่จัดว่ามีความถูกต้องสูง เป็นต้น



รูปที่ 2-54 ผลการวิเคราะห์โมเดลต่าง ๆ ของโครงข่ายประสาทเชิงลึกสำหรับการประยุกต์ใช้ในทางปฏิบัติ (Canziani et al., 2017)

2.6 การประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในทางปฏิบัติ

ในการนำโครงข่ายประสาทแบบคอนโวลูชันไปประยุกต์ใช้สำหรับงานใด ๆ นั้นสามารถประยุกต์ใช้ได้ใน 2 รูปแบบหลักคือ รูปแบบของการนำสถาปัตยกรรมของโครงข่ายที่ผ่านการฝึกสอนมาก่อน (Pre-Trained Architecture) จากชุดข้อมูลอื่นมาใช้ในแนวคิดการเรียนรู้แบบถ่ายโอน (Transfer Learning) และในรูปแบบที่เป็นการสร้างสถาปัตยกรรมขึ้นมาใหม่ (New Architecture Generating) จากการสร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายขึ้นมาเองเพื่อการฝึกสอนโครงข่ายด้วยชุดข้อมูลที่ต้องการศึกษา (Training from Scratch) โดยตรง ในที่นี้จึงเป็นการนำเสนอแนวคิดสำหรับการประยุกต์ใช้งานโครงข่ายประสาทแบบคอนโวลูชันใน 2 รูปแบบดังกล่าว

(1) การนำสถาปัตยกรรมของโครงข่ายที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้

เป็นการนำสถาปัตยกรรมของโครงข่ายที่ผ่านการฝึกสอนมาก่อนจากชุดข้อมูลอื่นที่มีการเผยแพร่มาประยุกต์ใช้ในลักษณะของแนวคิดการเรียนรู้แบบถ่ายโอน โดยโมเดลที่ผ่านการฝึกสอนมาก่อนที่มีการเผยแพร่ส่วนใหญ่จะผ่านการฝึกสอนมาจากชุดข้อมูลภาพของ ImageNet ที่มาจากการแข่งขันในงาน ILSVRC ดังที่ได้กล่าวถึงไปในหัวข้อที่ 2.5 เช่นสถาปัตยกรรมของ AlexNet หรือ สถาปัตยกรรมของ GoogleNet ในการนำสถาปัตยกรรมดังกล่าวมาใช้ สามารถประยุกต์ใช้ในอีก 2 รูปแบบ คือในรูปแบบของการยึดคุณลักษณะจากตัวสกัด (Fixed Feature Extractor) และในรูปแบบของการปรับแต่งการเรียนรู้ (Fine-Tune Learning) (Karpathy, 2018)

การประยุกต์ใช้งานแบบยึดคุณลักษณะจากตัวสกัด เป็นรูปแบบของการถ่ายโอนการเรียนรู้ที่ไม่ต้องมีการฝึกสอนใหม่ด้วยชุดข้อมูลที่ต้องการศึกษาให้กับโมเดลที่เคยผ่านการฝึกสอนมาก่อนแล้วนั้น นั่นคือเราสามารถนำข้อมูล เป็นการนำค่าของ Feature Maps ที่ได้จากชั้นสุดท้ายก่อนชั้นเอาต์พุตของโครงข่ายแบบที่ผ่านการฝึกสอนมาก่อนมาใช้ในขั้นตอนการจำแนกกับข้อมูลที่ต้องการศึกษาได้เลยโดยที่ไม่ต้องผ่านการฝึกสอนใด ๆ นั่นคือเป็นการนำข้อมูลที่ต้องการศึกษาไปแปลงให้อยู่ในรูปแบบของ Feature Maps จากชั้นสุดท้ายก่อนชั้นเอาต์พุตของโครงข่ายแล้วนำค่าเหล่านั้นจำแนกด้วยตัวจำแนก (Classifier) ใดก็ได้ที่ต้องการใช้

อีกรูปแบบหนึ่งของการนำสถาปัตยกรรมที่ผ่านการฝึกสอนมาก่อนมาใช้มาใช้ในทางปฏิบัติคือการประยุกต์ใช้แบบการปรับแต่งการเรียนรู้ ที่เป็นการนำสถาปัตยกรรมของโครงข่ายแบบที่ผ่านการฝึกสอนมาก่อนมาปรับแต่งเพิ่มเติมด้วยการฝึกสอนใหม่กับชุดข้อมูลที่ต้องการศึกษา นั่นคือเป็นการปรับแต่งโครงข่ายที่นำมาใช้ให้เข้ากับชุดข้อมูลที่ศึกษาด้วยการฝึกสอนใหม่ แต่เป็นการฝึกสอนที่ยึดตามโครงสร้างของสถาปัตยกรรมนั้น และใช้ค่าพารามิเตอร์ต่าง ๆ ในตอนเริ่มต้นฝึกสอนจากโมเดลที่เคยผ่านการฝึกสอนมาก่อนแล้วนั้น ซึ่งในขั้นตอนการปรับแต่งนั้นสามารถ

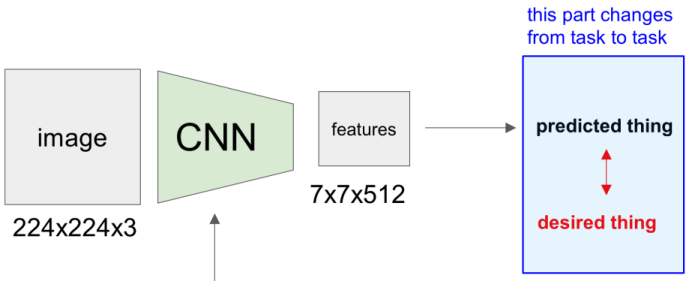
เลือกกำหนดได้ว่า จะปรับแต่งกับเฉพาะบางช่วงชั้นของโครงข่าย เช่น 10 ชั้นสุดท้าย หรือ 3 ชั้นสุดท้ายเป็นต้น หลังจากผ่านขั้นตอนการเรียนรู้การ Fine Tune ก็จะได้คุณลักษณะจากการปรับแต่ง (Fine Tune Feature) สำหรับการนำไปใช้ในขั้นตอนของการจำแนกต่อไป

(2) การสร้างสถาปัตยกรรมขึ้นมาใหม่

ในทางปฏิบัติแล้ว เราสามารถสร้างสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันสำหรับใช้ในการฝึกสอนชุดข้อมูลที่เราต้องการศึกษาขึ้นมาได้เอง โดยต้องมีการสร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายขึ้นมาก่อน แล้วนำสถาปัตยกรรมที่สร้างขึ้นมาไปฝึกสอนด้วยข้อมูลชุดฝึกสอนที่ศึกษา เพราะรูปแบบของสถาปัตยกรรมมีผลโดยตรงกับคุณลักษณะที่ได้จากขั้นตอนการเรียนรู้เพื่อหาคุณลักษณะของโครงข่าย ตัวอย่างของการพิจารณารูปแบบของสถาปัตยกรรมที่เหมาะสม เช่น การที่จะพิจารณาว่าจะมีจำนวนชั้นการคอนโวลูชันกี่ชั้นในโครงข่าย ขนาดของตัวกรองและจำนวนตัวกรองที่ใช้ในแต่ละชั้นเป็นเท่าไร ใช้วิธีการทำพูลลิ่งแบบใด ลำดับการเรียงของชั้นต่าง ๆ เป็นอย่างไร เป็นต้น ซึ่งในสถาปัตยกรรมแบบที่สร้างขึ้นมาเองนั้นจำเป็นต้องมีการพิจารณาเพื่อหาสถาปัตยกรรมที่เหมาะสมดังกล่าว การใช้สถาปัตยกรรมที่สร้างขึ้นมาอย่างเหมาะสมกับรูปแบบของข้อมูลจะช่วยให้โครงข่ายสามารถเรียนรู้เพื่อสร้างคุณลักษณะได้อย่างเหมาะสมด้วย ส่งผลให้โครงข่ายสามารถจำแนกข้อมูลที่ต้องการศึกษาได้อย่างมีประสิทธิภาพ

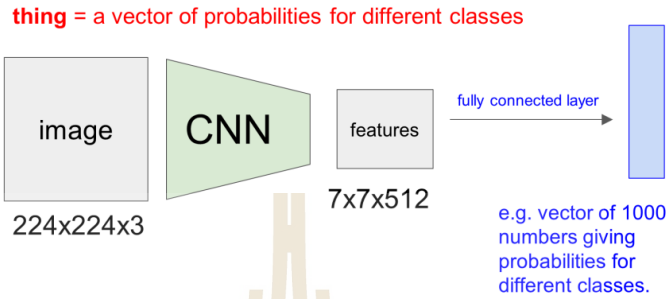
2.7 งานประยุกต์ที่ใช้โครงข่ายประสาทแบบคอนโวลูชัน

โครงข่ายประสาทแบบคอนโวลูชันนั้นเป็นการเรียนรู้เพื่อสร้างตัวแทนข้อมูล ดังนั้นตัวแทนข้อมูลที่อยู่ในรูปของแผนที่คุณลักษณะที่ได้จากเรียนรู้ของโครงข่ายจึงสามารถนำไปใช้งานต่อได้ในหลากหลายรูปแบบ ขึ้นอยู่กับความต้องการในแต่ละงานประยุกต์ แผนภาพในรูปที่ 2-55 แสดงถึงรูปแบบของงานประยุกต์ที่ใช้โครงข่ายประสาทแบบคอนโวลูชัน เมื่อส่วนด้านขวาของแผนภาพสามารถปรับเปลี่ยนได้ตามแต่ละงานประยุกต์ที่ต้องการ



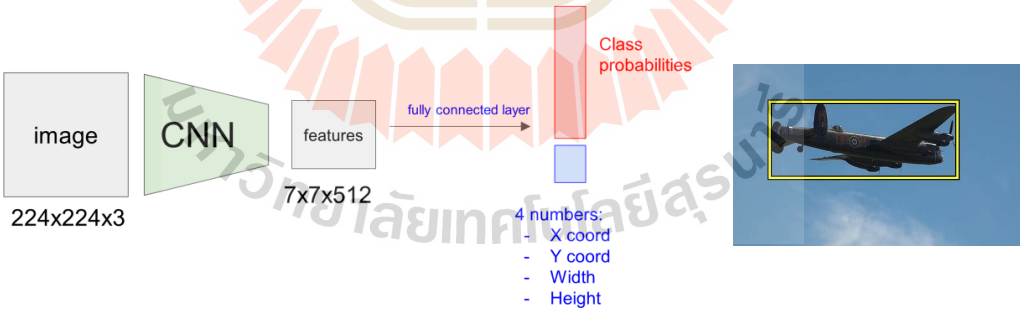
รูปที่ 2-55 รูปแบบของงานประยุกต์ที่ใช้โครงข่ายประสาทแบบคอนโวลูชัน (Karpathy, 2016)

ตัวอย่างที่เห็นได้ชัดเจนคือการนำไปใช้ในงานประยุกต์ทางด้านการจำแนกข้อมูล ดังแสดงในรูปที่ 2-56 เช่นในงานสำหรับการจำแนกภาพในชุดข้อมูล ImageNet ที่ให้เอาต์พุตจากโครงข่ายสามารถบอกได้ว่าภาพนั้นอยู่ใน Class ใด จากทั้งหมด 1,000 Classes



รูปที่ 2-56 งานประยุกต์ทางด้านการจำแนกข้อมูล (Karpathy, 2016)

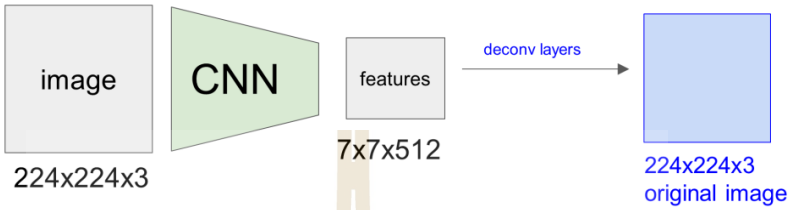
รูปที่ 2-57(a) เป็นตัวอย่างการใช้ในงานการระบุตำแหน่งของวัตถุในภาพ ที่นอกจากเอาต์พุตจากโครงข่ายจะต้องสามารถบอกได้ว่าภาพนั้นอยู่ใน Class ใดแล้วยังต้องบอกออกมาเป็นค่าตัวเลขอีกสี่ค่าที่ให้สามารถระบุตำแหน่งของวัตถุนั้นได้ด้วย ส่วนรูปที่ 2-57(b) เป็นตัวอย่างผลลัพธ์จากการระบุตำแหน่งของวัตถุที่ได้จากการใช้โครงข่ายในรูปที่ 2-57(a)



(a) (b)

รูปที่ 2-57 การนำไปใช้ในงานประยุกต์เกี่ยวกับการระบุตำแหน่งของวัตถุ (Object Localization) (Karpathy, 2016)

รูปที่ 2-58 แสดงแผนภาพการนำใช้ในงานประยุกต์เกี่ยวกับการเข้ารหัสข้อมูลแบบอัตโนมัติ โดยการนำแผนที่คุณลักษณะที่ได้จากการเรียนรู้ในโครงข่ายไปผ่านขั้นตอนการ Deconvolution เพื่อให้ได้เอาต์พุตเป็นภาพต้นฉบับ นั่นคือแผนที่คุณลักษณะที่ได้จากการเรียนรู้ในโครงข่ายเป็นขั้นตอนการเข้ารหัส ส่วนขั้นตอนการ Deconvolution นั้นเป็นการถอดรหัสออกมา



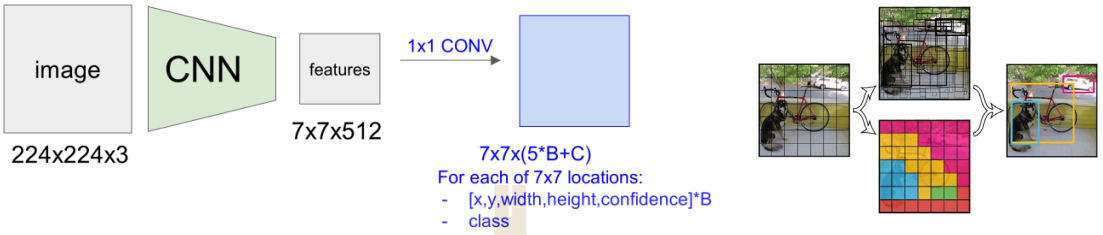
รูปที่ 2-58 การนำใช้ในงานประยุกต์การเข้ารหัสข้อมูลแบบอัตโนมัติ (Autoencoder) (Karpathy, 2016)

สำหรับงานการแยกส่วนภาพ (Image Segmentation) นั้นสามารถนำไปใช้ได้ในลักษณะคล้ายกันกับงานการเข้ารหัสข้อมูลแบบอัตโนมัติ แต่ภาพเอาต์พุตของงานการแยกส่วนภาพเป็นการระบุว่าแต่ละจุดภาพเป็นของวัตถุใด จากการที่กำหนดค่าในแนวลึกของเทนเซอร์เอาต์พุตให้เป็นจำนวนวัตถุทั้งหมดที่มีในภาพเพื่อต้องการให้แยกส่วนวัตถุเหล่านั้นออกจากกัน ดังแสดงตัวอย่างในรูปที่ 2-59 ที่ค่าในแนวลึกของเทนเซอร์เอาต์พุตเป็น 20 หมายความว่าให้โครงข่ายแยกส่วนของวัตถุ 20 ชนิดออกจากกัน

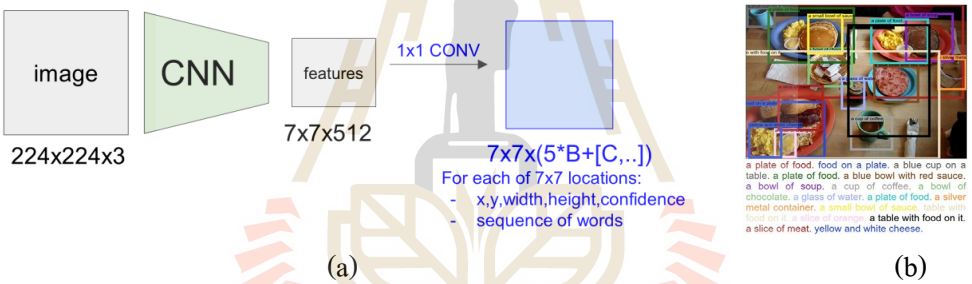
รูปที่ 2-59 การนำใช้ในงานประยุกต์การแยกส่วนภาพ (Karpathy, 2016)

รูปที่ 2-60 แสดงแผนภาพการนำไปใช้ในงานประยุกต์การตรวจจับกลุ่มของวัตถุในภาพ (Image Class Detection) จากภาพอินพุตที่มีวัตถุหลายชนิด แล้วให้โครงข่ายสามารถระบุได้ว่าในภาพมีวัตถุอะไรบ้างพร้อมทั้งสามารถระบุตำแหน่งของแต่ละวัตถุเหล่านั้นได้

การใช้ในงานประยุกต์การบรรยายภาพแบบหนาแน่น (Dense Image Captioning) แสดงดังภาพที่ 2-61 ที่นอกจากโครงข่ายจะต้องสามารถระบุได้ว่าแต่ละวัตถุในภาพอยู่ในบริเวณใดแล้ว ยังต้องบรรยายแต่ละส่วนของวัตถุเหล่านั้นออกมาเป็นภาษาธรรมชาติ (Natural Language) ได้ด้วย

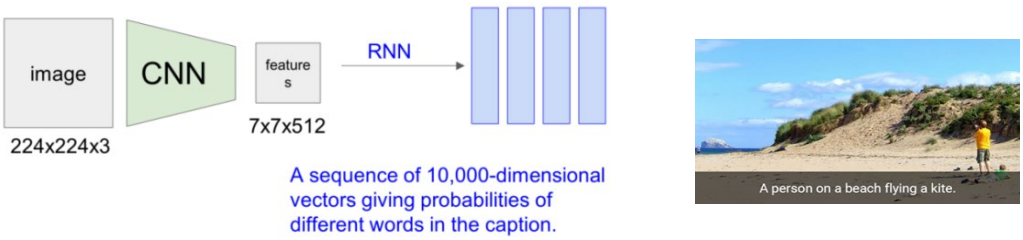


รูปที่ 2-60 การนำไปใช้ในงานประยุกต์การตรวจจับกลุ่มของวัตถุในภาพ (Karpathy, 2016)



รูปที่ 2-61 การนำไปใช้ในงานประยุกต์การบรรยายภาพแบบหนาแน่น (Karpathy, 2016)

นอกจากการนำไปใช้ในงานการบรรยายภาพแบบหนาแน่นแล้ว โครงข่ายแบบคอนโวลูชันสามารถนำไปใช้ร่วมกับโครงข่ายการเรียนรู้เชิงลึกแบบอื่น เช่น นำไปใช้ร่วมกับโครงข่าย RNN ที่แสดงดังรูปที่ 2-62(a) สำหรับงานการบรรยายภาพ (Image Captioning) โดยเอาต์พุตที่ต้องการของงานประยุกต์นี้แสดงดังรูปที่ 2-62(b) ในการสร้างคำบรรยายให้กับภาพหนึ่งภาพ ซึ่งงานการบรรยายภาพนี้เป็นการนำโมเดลการเรียนรู้เชิงลึกไปใช้ร่วมกันกับการประมวลผลภาษาธรรมชาติ



(a) (b)

รูปที่ 2-62 การนำไปประยุกต์ใช้ในงานการบรรยายภาพ (Karpathy, 2016)

2.8 Frameworks ของโครงข่ายประสาทแบบคอนโวลูชัน

หลังจากที่แนวคิดเกี่ยวกับการเรียนรู้เชิงลึกนำเสนอขึ้นมาแล้วพบว่ามีประสิทธิภาพในการนำไปใช้งาน โมเดลต่าง ๆ เกี่ยวกับการเรียนรู้เชิงลึกจึงมีการพัฒนาและมีการนำไปประยุกต์ใช้ในงานต่าง ๆ อย่างรวดเร็วและกว้างขวาง ส่วนหนึ่งของการพัฒนาดังกล่าวเกิดจากการมีซอฟต์แวร์หรือ Frameworks ที่มีผู้พัฒนาขึ้นมารองรับสำหรับการนำไปประยุกต์ใช้ ในส่วนนี้จึงเป็นการนำเสนอ Frameworks ที่เด่น ๆ ของการเรียนรู้เชิงลึก ซึ่ง Frameworks นั้นมีส่วนของโมเดลโครงข่ายประสาทแบบคอนโวลูชันรวมอยู่ด้วย

สำหรับ Frameworks ที่ต้องการนำเสนอเป็นหลักในส่วนของรายละเอียดในส่วนนี้คือ Theano, Tensorflow, Torch, Caffe และ Keras

- Theano: เป็น Open Source ที่พัฒนาโดย Machine Learning Group ของ Université de Montréal เมื่อปี 2009
- Tensorflow: เป็น Library ที่พัฒนาโดย Google Brain Team แบบ Open Source เมื่อปี 2015
- Torch: พัฒนาขึ้นจาก Facebook AI Research, Twitter, Google DeepMind
- Caffe: พัฒนาโดย Berkeley Vision and Learning Center (BVLC) แห่งมหาวิทยาลัย University of California เมื่อปี 2013
- Keras: เป็น Python API ที่พัฒนาโดยขึ้น François Chollet เมื่อปี 2015 (ปัจจุบันเป็น software engineer ของ Google) ซึ่งมีแรงจูงใจในการพัฒนามาจาก Torch โดยสามารถใช้ส่วนของ backends เป็น Theano, TensorFlow and Deeplearning4j Keras เป็น library ที่พัฒนาขึ้นในรูปแบบ API จึงจัดว่าเป็น library ที่ใช้งานง่าย

รูปที่ 2-63 เป็นแผนภาพแสดงการเปรียบเทียบคุณลักษณะของ frameworks ต่าง ๆ ของการเรียนรู้เชิงลึกที่มี Theano, Tensorflow, Torch, Caffe และ Keras รวมอยู่ในแผนภาพนี้ด้วย แผนภาพใช้จำนวนของเครื่องหมายบวกสำหรับบอกถึงระดับของคุณลักษณะต่าง ๆ นั่นคือคุณลักษณะใดมีจำนวนเครื่องหมายบวกมาก หมายความว่า Framework นั้นเด่นในคุณลักษณะด้านนั้น จากแผนภาพเมื่อพิจารณาคุณลักษณะต่าง ๆ พบว่า

- Frameworks ส่วนใหญ่พัฒนาด้วยภาษา Python และ C++
- ในแง่ของ Material สำหรับการฝึกสอนและคำแนะนำเกี่ยวกับวิธีการใช้งานนั้น Tensorflow จะสามารถหาข้อมูลได้ง่ายที่สุด รองลงมาคือ Theano
- เกี่ยวกับความสามารถของโมเดลโครงข่ายประสาทแบบคอนโวลูชัน (CNN Modeling Capability) พบว่า Theano และ Tensorflow มีความสามารถใกล้เคียงกัน
- เกี่ยวกับความสามารถของโมเดลแบบ RNN (RNN Modeling Capability) พบว่า Theano Tensorflow และ Torch มีความสามารถในระดับเดียวกัน
- ในประเด็นทางด้านความง่ายในการใช้งานพบว่า Tensorflow ใช้งานได้ง่ายที่สุด
- ในแง่ของความเร็ว Torch เร็วที่สุด
- ความสามารถในการรองรับการทำงานแบบ multiple GPU พบว่า Tensorflow และ Torch มีความสามารถไล่เลี่ยกัน
- เนื่องจาก Keras เป็น API ที่พัฒนาโดยสามารถใช้ส่วนของ backends เป็น Theano และTensorFlow ดังนั้น Keras จึงรองรับกับการใช้งานในสอง Frameworks ดังกล่าว

นอกเหนือจาก 5 Frameworks ที่นำเสนอนี้แล้ว ยังมี Frameworks อื่น ๆ เช่น MXNet, Neon, CNTK ที่มีคุณลักษณะต่าง ๆ แสดงในรูปที่ 2-63 ด้วยเช่นกัน รวมทั้ง Framework อื่นที่ไม่ได้มีแสดงในที่นี้เช่น Deeplearning4j, Lasagne, BigDL, MatConvNet, Blocks, Apache Singa และอื่น ๆ อีกมากมายซึ่งสามารถอ้างอิงได้จาก http://deeplearning.net/software_links/

	Languages	Tutorials and training materials	CNN modeling capability	RNN modeling capability	Architecture: easy-to-use and modular front end	Speed	Multiple GPU support	Keras compatible
Theano	Python, C++	++	++	++	+	++	+	+
Tensor-Flow	Python	+++	+++	++	+++	++	++	+
Torch	Lua, Python (new)	+	+++	++	++	+++	++	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
Neon	Python	+	++	+	+	++	+	
CNTK	C++	+	+	+++	+	++	+	

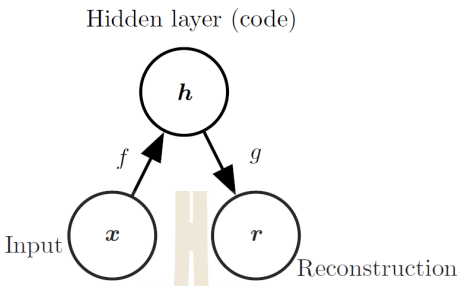
รูปที่ 2-63 แผนภาพเปรียบเทียบคุณลักษณะของ Frameworks ต่าง ๆ ของการเรียนรู้เชิงลึก (Rubashkin, 2017)

2.9 โครงข่ายเครื่องเข้ารหัสอัตโนมัติ (Autoencoder Network)

เครื่องเข้ารหัสอัตโนมัติ (Autoencoder) เป็นโครงข่ายของนิวรอนที่ทำการฝึกสอนเพื่อให้เอาต์พุตของโครงข่ายคือการพยายามทำสำเนาของอินพุต (Goodfellow et al., 2016) นั่นคือเป็นการกำหนดให้เอ๊าพุตเป้าหมาย (Target Output) ที่ป้อนให้กับโครงข่ายมีค่าเหมือนกับอินพุต โครงสร้างภายในของโครงข่ายนั้นจะมีชั้นซ่อนเร้น (Hidden Layer; h) จำนวนหนึ่งชั้นสำหรับใช้แทน (Represent) ลักษณะของรหัส (Code) ที่จะถูกใช้แทนอินพุต สามารถมองภาพลักษณะของโครงข่ายได้ว่าประกอบด้วยสองส่วนคือ ส่วนของฟังก์ชันการเข้ารหัส (Encoder Function) $h = f(x)$ และส่วนของการถอดรหัส (Decoder) ที่ทำการกู้คืน (Reconstruction) ค่าจากการเข้ารหัส $r = g(h)$ นั่นคือสถาปัตยกรรมของเครื่องเข้ารหัสอัตโนมัติสามารถแสดงได้ดังรูปที่ 2.64

จากลักษณะสถาปัตยกรรมของโครงข่ายในรูปที่ 2. 64 นั้นถ้าเป็นการให้เครื่องเข้ารหัสอัตโนมัติทำการเรียนรู้เพื่อให้ได้เอาต์พุตคือ $g(h) = g(f(x)) = x$ อย่างสมบูรณ์จะถือว่าไม่ได้ประโยชน์ใด ๆ จากการเรียนรู้ของโครงข่าย เพราะเป็นการเรียนรู้เพื่อให้ได้ฟังก์ชันเอกลักษณ์ (Identity Function) ดังนั้นเครื่องเข้ารหัสโดยอัตโนมัติจึงถูกออกแบบให้ไม่สามารถเรียนรู้ให้ทำสำเนาได้อย่างสมบูรณ์ โดยทั่วไปจะเป็นการไปจำกัดการเรียนรู้เพื่อให้เป็นการทำสำเนาโดยประมาณและทำสำเนาเฉพาะส่วนที่คล้ายคลึงกับข้อมูลที่ฝึกสอน การที่โครงข่ายถูกกำหนด

เงื่อนไขบังคับไว้ก่อน (Prior Constraint) ว่ารูปลักษณ์ (Aspects) แบบใดของข้อมูลถึงควรจะถูกทำสำเนา จะเป็นการทำให้เครื่องเข้ารหัสโดยอัตโนมัติสามารถเรียนรู้คุณสมบัติต่าง ๆ ที่มีประโยชน์จากข้อมูลได้



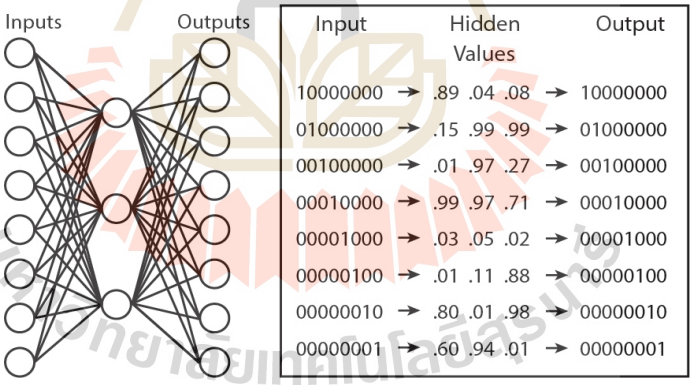
รูปที่ 2.64 สถาปัตยกรรมของโครงข่ายเข้ารหัสอัตโนมัติ (Goodfellow et al., 2016)

ถึงแม้เครื่องเข้ารหัสอัตโนมัตินั้นมีโครงสร้างเช่นเดียวกันกับโครงข่ายนิวรอนโดยทั่วไปที่มีชั้นซ่อนเร้นหนึ่งชั้น แต่สิ่งที่สนใจจากการฝึกสอนเครื่องเข้ารหัสอัตโนมัตินั้นไม่ได้เป็นส่วนเอาต์พุตจากโครงสร้างของโครงข่ายเหมือนโครงข่ายนิวรอนโดยทั่วไป แต่เป็นส่วนของการเข้ารหัสที่ได้จากการเรียนรู้ในชั้นซ่อนเร้น (นั่นคือส่วนของ h ในรูปที่ 2. 64) โดยการเข้ารหัสดังกล่าวเก็บไว้ในส่วนเมทริกซ์ของค่าน้ำหนัก (Weight Matrix) ที่เชื่อมอยู่กับชั้นซ่อนเร้นนั้น ลักษณะการเรียนรู้ของเครื่องเข้ารหัสอัตโนมัติจึงจัดว่าเป็นการเรียนรู้แบบไม่มีผู้ฝึกสอน (Usupervised Learning) ที่เป็นการนำการเรียนรู้แบบแพร่กลับ (Backpropagation Learning) มาประยุกต์ใช้ โดยกำหนดให้ค่าเอาต์พุตเป้าหมายเหมือนกันกับค่าของอินพุต และจัดว่าเป็นลักษณะของการเรียนรู้เพื่อสร้างตัวแทน (Representative Learning) ให้กับข้อมูลที่สนใจอีกรูปแบบหนึ่งด้วยเช่นกัน

รูปที่ 2. 65 แสดงตัวอย่างจากการเรียนรู้ของเครื่องเข้ารหัสอัตโนมัติสำหรับการเข้ารหัสข้อมูลเลขฐานสอง (Binary Number) โดยใช้โครงข่ายแบบ 8 นิวรอนในชั้นอินพุต 3 นิวรอนในชั้นซ่อนเร้นและ 8 นิวรอนในชั้นเอาต์พุต ลักษณะโครงสร้างของโครงข่ายแสดงในรูปที่ 2.65(a) จะเห็นได้ว่าในการฝึกสอนนั้นกำหนดให้เอาต์พุตเป้าหมายที่ป้อนให้กับโครงข่ายมีค่าเหมือนกับอินพุต ดังแสดงในรูปที่ 2.65(b) หลังจากผ่านขั้นตอนการฝึกสอนให้กับโครงข่าย ค่าของแต่ละนิวรอนในชั้นซ่อนเร้นเมื่อนำอินพุตที่ใช้ฝึกสอนแต่ตะตัวส่งผ่านเข้าไปในโครงข่าย (นั่นคือการนำอินพุตมาแปลงเป็นคุณลักษณะด้วยค่า weights และ biases ที่ได้มาจากการฝึกสอน) จะได้ผลลัพธ์ในส่วนของนิวรอนชั้นซ่อนเร้น (ส่วนของ Hidden Values) ดังแสดงในรูปที่ 1.65(b) สังเกตได้ว่าจากค่าที่แสดง

ดังกล่าว ถ้าทำการปัดค่า (Round) ไปเป็นค่า 0 หรือ 1 (เช่น .89 .04 .08 ปัดค่าเป็น 100) ผลลัพธ์ที่ได้จะเหมือนกันกับการใช้หลักการเข้ารหัสสำหรับเลข 8 บิตแบบวิธีมาตรฐาน (Mitchell, 1997) จากตัวอย่างนี้เป็นการแสดงให้เห็นถึงสิ่งที่ได้จากการเรียนรู้ของเครื่องเข้ารหัสโดยอัตโนมัติว่าผลลัพธ์ที่สนใจจากการเรียนรู้คือค่าต่าง ๆ ที่เกิดขึ้นในชั้นซ่อนเร้น นั่นคือเป็นลักษณะการเรียนรู้เพื่อสร้างตัวแทน (Representative Learning) ให้กับข้อมูลที่สนใจ และจัดว่าเป็นการเรียนรู้แบบไม่มีผู้ฝึกสอน (Unsupervised Learning) เพราะไม่ได้มีการนำเอาต์พุตเป้าหมายจริง ๆ มาใช้ในขั้นตอนการฝึกสอน

สามารถแบ่งชนิดของเครื่องเข้ารหัสอัตโนมัติได้ตามลักษณะของเงื่อนไข (Constraints) ที่นำมาบังคับให้โครงข่ายเรียนรู้ ซึ่งแบ่งออกเป็นเครื่องเข้ารหัสอัตโนมัติแบบต่ำกว่าสมบูรณ์ (Undercomplete Autoencoder) ที่จัดว่าเป็นเครื่องเข้ารหัสอัตโนมัติแบบแบบดั้งเดิม (Traditional Autoencoder) นั่นคือแบบที่ได้กล่าวไปก่อนหน้านี้ และเครื่องเข้ารหัสอัตโนมัติแบบ Regularized (Regularized Autoencoder) โดยเครื่องเข้ารหัสอัตโนมัติแบบ Regularized สามารถแบ่งออกเป็นเครื่องเข้ารหัสอัตโนมัติแบบลดสัญญาณรบกวน (Denoise Autoencoder) เครื่องเข้ารหัสอัตโนมัติแบบเบาบาง (Sparse Autoencoder) เครื่องเข้ารหัสอัตโนมัติแบบหดตัว (Contractive Autoencoder) และเครื่องเข้ารหัสอัตโนมัติแบบแปรผัน (Variational Autoencoder)



รูปที่ 2.65 ตัวอยางการเรียนรู้ที่ได้จากชั้นซ่อนเร้น (ค่าของ Hidden Values) ของเครื่องเข้ารหัสอัตโนมัติเพื่อเข้ารหัสข้อมูลจาก 8 บิตเป็น 3 บิต (Mitchell, 1997)

2.9.1 เครื่องเข้ารหัสอัตโนมัติแบบต่ำกว่าสมบูรณ์ (Undercomplete Autoencoders)

ในการฝึกสอนเครื่องเข้ารหัสอัตโนมัติเพื่อให้สามารถทำสำเนาของอินพุตได้นั้นจะทำให้ผลลัพธ์ที่ได้จากเรียนรู้ในชั้นซ่อนเร้น h มีคุณสมบัติที่มีประโยชน์สำหรับการนำไปใช้วิธีการหนึ่งสำหรับการได้มาซึ่งคุณลักษณะที่มีประโยชน์ (Useful Features) จาก h คือการจำกัดให้

ชั้นซ่อนเร้น h มีมิติ (Dimension) ที่น้อยกว่าจำนวนมิติของอินพุต จึงเป็นที่มาของชื่อ ต่ำกว่าสมบูรณ์ (Undercomplete) ดังกล่าว จากรูป 2.65 ที่กล่าวไปก่อนหน้านี้ซึ่งเป็นเครื่องเข้ารหัสอัตโนมัติแบบดั้งเดิมก็เป็นการใช้หลักการแบบต่ำกว่าสมบูรณ์เช่นเดียวกัน นั่นคือมิติของ h น้อยกว่ามิติของอินพุต จากการที่กำหนดให้จำนวนนิวรอนในชั้นซ่อนเร้นเป็นสามในขณะที่จำนวนนิวรอนในชั้นอินพุตแป็นแปด (นั่นคือมิติของ h เป็นสามในขณะที่มิติของข้อมูลเข้าเป็นแปด ซึ่งสาม < แปด)

การเรียนรู้เพื่อให้ได้รูปแบบการแทนข้อมูลแบบที่ต่ำกว่าสมบูรณ์ (Undercomplete Representation) นั้นเป็นการบังคับให้เครื่องเข้ารหัสอัตโนมัติสามารถตรวจจับคุณลักษณะเด่น (Salient Features) หรือโครงสร้างที่น่าสนใจ (Interesting Structures) จากข้อมูลที่ใช้ฝึกสอนได้ กระบวนการเรียนรู้เกิดจากการพยายามหาค่าน้อยที่สุดของฟังก์ชันการสูญเสีย (Minimizing a Loss Function) ซึ่งคือ

$$L(x, g(f(x)))$$

เมื่อ L เป็นฟังก์ชันการสูญเสียที่เกิดจากการทำโทษ (Penalizing) ให้ $g(f(x))$ มีความแตกต่างไปจาก x เช่นการใช้ L ด้วยค่าความผิดพลาดเฉลี่ยยกกำลังสอง (Mean Square Error) เป็นต้น

เมื่อใช้ฟังก์ชันการเข้ารหัส f และฟังก์ชันการถอดรหัส g เป็นแบบเชิงเส้น (นั้นคือเมื่อ Activation Function เป็นเชิงเส้น) และใช้ L เป็นค่าความผิดพลาดเฉลี่ยยกกำลังสอง เครื่องเข้ารหัสอัตโนมัติแบบต่ำกว่าสมบูรณ์จะทำการเรียนรู้เพื่อให้ได้สับสเปซ (Subspace) เช่นเดียวกันการใช้หลักการของ PCA (Principal Component Analysis) นั่นคือการที่ฝึกสอนให้เครื่องเข้ารหัสอัตโนมัติทำการทำสำเนาข้อมูลนั้นทำให้ได้ผลข้างเคียง (Side-Effect) จากการฝึกสอนเป็นสับสเปซหลัก (Principal Subspace) ของข้อมูลที่ใช้ฝึกสอน นอกจากนี้เครื่องเข้ารหัสอัตโนมัติที่ใช้ฟังก์ชันการเข้ารหัส f และฟังก์ชันการถอดรหัส g เป็นแบบไม่เชิงเส้นจะมีความสามารถที่เหนือกว่า PCA ในการเรียนรู้ความเป็นทั่วไปแบบไม่เชิงเส้น (Nonlinear Generalization) (Goodfellow et al., 2016)

2.9.2 เครื่องเข้ารหัสอัตโนมัติแบบ Regularization (Regularization Autoencoders)

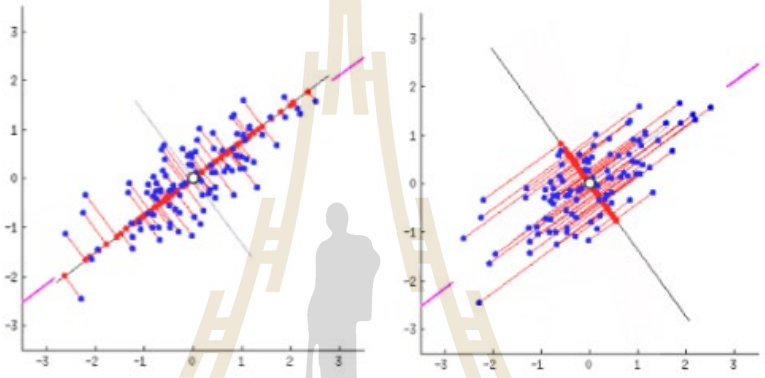
Regularization คือการปรับเปลี่ยนใด ๆ ที่เราทำกับอัลกอริทึมสำหรับการเรียนรู้ (Learning Algorithm) เพื่อต้องการที่จะลดความผิดพลาดแบบทั่วไป (Generalization Error) แต่ไม่ใช่ความผิดพลาดจากการฝึกสอน (Training Error) (Goodfellow et al., 2016) ดังนั้นแนวคิดของเครื่องเข้ารหัสอัตโนมัติแบบ Regularized จึงสอดคล้องตามความหมายดังกล่าว ที่นำมาใช้เพื่อปรับเปลี่ยนกระบวนการเรียนรู้ของเครื่องเข้ารหัสอัตโนมัติ โดยนำมาใช้ในกรณีที่เครื่องเข้ารหัสอัตโนมัติเป็นแบบเกินกว่าสมบูรณ์ (Overcomplete) นั่นคือเมื่อจำนวนนิวรอนในชั้นซ่อนเร้นมีจำนวนมิติสูงกว่าจำนวนอินพุต

เครื่องเข้ารหัสอัตโนมัติแบบเกินกว่าสมบูรณ์ที่ไม่ได้มีการสร้างข้อจำกัดใด ๆ ให้กับโครงข่ายนั้นฟังก์ชันการเข้ารหัสและการถอดรหัสแบบเชิงเส้นสามารถที่จะเรียนรู้เพื่อให้สามารถทำสำเนาอินพุตไปเป็นเอาต์พุตได้โดยไม่ได้เกิดการเรียนรู้อะไรที่เป็นประโยชน์จากลักษณะการกระจายของข้อมูล (Data Distribution) เลย แต่ในทางอุดมคติแล้ว เราสามารถที่จะฝึกสอนสถาปัตยกรรมใด ๆ ของเครื่องรหัสอัตโนมัติได้จากการเลือกมิติของรหัส (Code Dimension) และจากความสามารถ (Capacity) ของตัวเข้ารหัสและตัวถอดรหัสโดยใช้ความซับซ้อนของการกระจาย (Complexity of Distribution) ที่เราต้องการจะสร้างโมเดล ดังนั้นแทนที่จะจำกัดความสามารถของโมเดลด้วยการให้มิติของรหัสมีจำนวนน้อย ๆ เครื่องเข้ารหัสโดยอัตโนมัติแบบ Regularized นั้นจะสร้างข้อจำกัดโดยใช้ฟังก์ชันความสูญเสีย เพื่อบังคับให้แบบจำลองมีคุณสมบัติอื่น ๆ นอกเหนือจากความสามารถในการทำสำเนาอินพุตไปเป็นเอาต์พุตแทน คุณสมบัติอื่น ๆ ดังกล่าว เช่น การแทนด้วยความเบาบาง (Sparsity of the Representation) การแทนด้วยค่าที่น้อยที่สุดของผลต่าง (Smallness of the Derivative of the Representation) และความทนทานต่อสัญญาณรบกวนหรือข้อมูลเข้าที่สูญหาย (Robustness to Noise or to Missing Inputs) เป็นต้น ดังนั้นเครื่องเข้ารหัสอัตโนมัติแบบ Regularized จึงแบ่งออกเป็นชนิดต่าง ๆ ตามคุณสมบัติที่แตกต่างกันในการสร้างข้อจำกัดให้กับฟังก์ชันความสูญเสียดังที่กล่าวมา เช่น เครื่องเข้ารหัสโดยอัตโนมัติแบบลดสัญญาณรบกวน (Denoise Autoencoder) เครื่องเข้ารหัสโดยอัตโนมัติแบบเบาบาง (Sparse Autoencoder) เครื่องเข้ารหัสโดยอัตโนมัติแบบหดตัว (Contractive Autoencoder) และเครื่องเข้ารหัสโดยอัตโนมัติแบบแปรผัน (Variational Autoencoder) ซึ่งแต่ละชนิดนั้นมีการนำไปประยุกต์ใช้ในรูปแบบที่แตกต่างกันออกไป

2.10 การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis, PCA)

PCA เป็นเทคนิคที่ใช้สำหรับการหาองค์ประกอบหลัก (Principal Components) จากชุดข้อมูลตั้งต้น ด้วยการแปลงข้อมูลในระบบพิกัด (Coordinate System) เดิมไปอยู่ในระบบพิกัดใหม่หรือสับสเปซ (Subspace) ใหม่ที่มีจำนวนมิติ (หรือจำนวน Variables) น้อยกว่าจำนวนมิติของข้อมูลตั้งต้น โดยแต่ละแกนในระบบพิกัดใหม่นั้นแทนแต่ละองค์ประกอบหลักที่ได้ ซึ่งแต่ละองค์ประกอบหลักที่ได้นั้นจะไม่มีความสัมพันธ์กันในแบบเชิงเส้น (Linearly Uncorrelated) เมื่อองค์ประกอบหลักลำดับที่หนึ่ง (แกนที่หนึ่ง) คือองค์ประกอบที่มีค่าความแปรปรวน (Variance) สูงที่สุดจากการ Projected ข้อมูลตั้งต้นลงบนแกนดังกล่าว และองค์ประกอบถัด ๆ ไปก็จะมีความแปรปรวนลดหลั่นกันไปตามลำดับ รูปที่ 2.66 ยกตัวอย่างการ Projected ข้อมูลตั้งต้นเดียวกัน (เมื่อข้อมูลตั้งต้นเป็น 2 มิติ) ลงบนแกนหลักสองแกน โดยแกนหลักลำดับที่หนึ่งที่ได้คือรูปที่ 2.66(a) ซึ่ง

เป็นแกนหลักที่เมื่อนำข้อมูลทั้งหมดมา Projected ลงบนแกนดังกล่าวแล้วเกิดค่าความแปรปรวนจากการ Projected สูงที่สุด (ค่าความแปรปรวนพิจารณาจากความกว้างของช่วงของจุดที่ถูก Projected ลงบนแกนหลัก) ส่วนรูปที่ 2.66(b) เป็นแกนหลักที่มีความกว้างของช่วงของจุดที่ถูก Projected ลงบนแกนหลักนั้นแคบที่สุดจากข้อมูลตั้งต้นเดียวกัน แสดงว่ามีค่าความแปรปรวนน้อยที่สุด ดังนั้นในกรณีที่ข้อมูลตั้งต้นเป็น 2 มิติ ถ้าต้องการลดมิติของข้อมูลเหลือเพียงหนึ่งมิติ ก็สามารถทำได้ด้วยการเลือกใช้เพียงองค์ประกอบหลักที่หนึ่งในรูปที่ 2.66(a) สำหรับการแปลงข้อมูลตั้งต้นด้วยเวกเตอร์ขององค์ประกอบหลักนั้นไปเป็นข้อมูลในสับสเปซใหม่

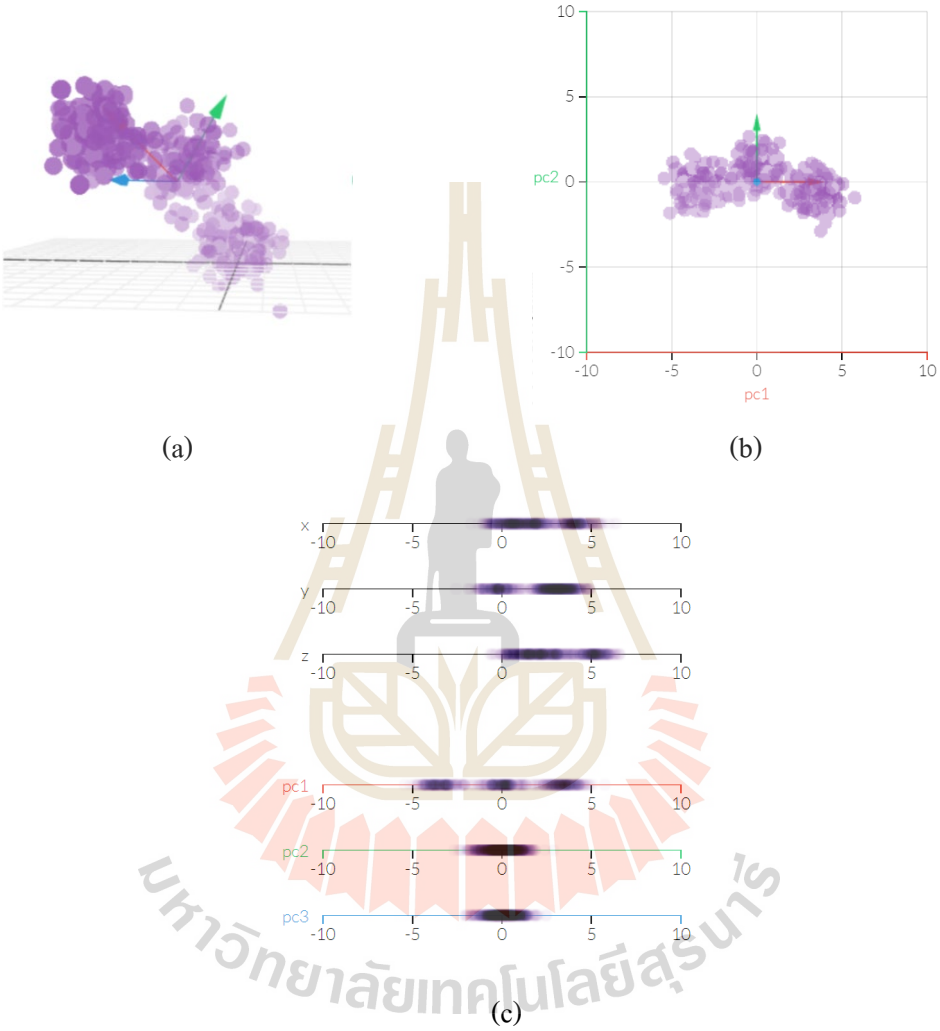


(a) (b)

รูปที่ 2.66 ตัวอย่างการ Projected ข้อมูลตั้งต้นลงบนแกนหลักและการพิจารณาค่าความแปรปรวนที่เกิดขึ้นบนแกนหลัก (Boonmarueng, 2019)
(a) แกนหลักที่มีค่าความแปรปรวนมากที่สุด
(b) แกนหลักที่มีค่าความแปรปรวนน้อยที่สุด

รูปที่ 2.67 ยกตัวอย่างการนำ PCA ไปใช้ในกรณีที่ข้อมูลตั้งต้นเป็น 3 มิติ โดยข้อมูลตั้งต้นที่ใช้แสดงดังรูปที่ 2.67(a) สำหรับรูปที่ 2.67(b) แสดงผลลัพธ์ที่ได้หลังจากการเลือกใช้แกนหลักสองแกนแรกมาแปลงข้อมูลตั้งต้นไปเป็นข้อมูลในสับสเปซใหม่ เมื่อ pc1 ในรูปแทนแกนขององค์ประกอบหลักที่หนึ่งและ pc2 แทนแกนขององค์ประกอบหลักที่สองตามลำดับ ส่วนรูปที่ 2.67(c) ส่วนบนแสดงให้เห็นความแปรปรวนที่เกิดขึ้นจากแต่ละมิติ (แต่ละ variable) ของข้อมูลตั้งต้น และส่วนล่างเป็นความแปรปรวนที่เกิดขึ้นจากแต่ละองค์ประกอบหลักด้วยวิธี PCA จะเห็นได้ว่าช่วงของข้อมูลในแกน pc1 มีความกว้างมากที่สุด ดังนั้นแกน pc1 จึงมีค่าความแปรปรวนมากที่สุด แกน pc2 มีความกว้างมากรองลงมา จึงมีค่าความแปรปรวนมากเป็นอันดับที่สอง ส่วนแกน

pc3 มีความกว้างน้อยที่สุดจะมีค่าความแปรปรวนมากเป็นอันดับที่สาม ดังนั้นหลังจากนำเฉพาะเวกเตอร์ที่แทน pc1 และ pc2 มาใช้ในการแปลงข้อมูลตั้งต้นจึงได้ผลลัพธ์ดังรูปที่ 2.67(b) นั่นคือข้อมูลจาก 3 มิติเดิมจึงถูกแปลงไปเป็นข้อมูลใน 2 มิติของสับสเปซใหม่



รูปที่ 2.67 การนำ PCA ไปใช้ในกรณีที่ข้อมูลตั้งต้นเป็น 3 มิติ (Power, 2019)

ด้วยวิธีการของ PCA จึงเป็นการหาจำนวนแกนหลักที่เหมาะสมสำหรับใช้แทนข้อมูลเดิม ดังนั้น PCA จึงจัดว่าเป็นวิธีการหนึ่งสำหรับการแทนข้อมูล (Data Representation) ที่เป็นวิธีการเพื่อลดคุณลักษณะ (Feature Reducing) ของข้อมูลตั้งต้นโดยใช้หลักการพิจารณาองค์ประกอบหลักเพื่อสร้างคุณลักษณะใหม่ขึ้นมา ดังนั้นวิธีการภายในของ PCA จึงเป็นลำดับขั้นตอนทางคณิตศาสตร์เพื่อใช้สำหรับการลดมิติของข้อมูลนั่นเอง

ขั้นตอนของ PCA แบบมาตรฐานอธิบายได้ดังต่อไปนี้

1) จัดเตรียมข้อมูล:
   - Center the Data: การนำค่า Mean ของข้อมูลในแต่ละมิติ (แต่ละ Variable) มาลบออกจากค่าเดิมของ Variable นั้น ๆ นั่นคือเป็นการทำให้ข้อมูลมีค่า Mean เป็นศูนย์
   - Scale the Data: สเกลข้อมูลจากขั้นตอนก่อนหน้าให้เป็น Unit Variance ด้วยการนำค่า SD (Standard Deviation) ของแต่ละ Variable มาหารกับค่าของ Variable เดิม
2) คำนวณ Covariance Matrix
3) คำนวณ Eigenvectors และ Eigenvalues จาก Covariance Matrix ที่ได้
4) เลือกองค์ประกอบหลัก: องค์ประกอบหลักจะถูกเลือกจากลำดับของ Eigenvectors ที่ได้มาจากการนำ Eigenvalues มาเรียงลำดับจากมากไปน้อย โดยเลือกนำ Eigenvectors d ตัวแรกไปใช้ (จำนวนของ d Eigenvectors ที่เลือกจะเป็นตัวบอกจำนวนมิติของข้อมูลในสับสเปซใหม่)
5) คำนวณค่าของข้อมูลแต่ละตัวในสับสเปซใหม่:
   - ทำการ Transpose เมทริกซ์ของ d Eigenvectors: ให้ rows แทนแต่ละ Eigenvectors
   - ทำการ Transpose เมทริกซ์ของชุดข้อมูลตั้งต้น: ให้ rows เป็นแต่ละ Variable และ column ข้อมูลแต่ละตัว

ค่าของข้อมูลในสับสเปซใหม่ได้มาจากการนำเมทริกซ์ข้างต้นสองเมทริกซ์นั้นมาคูณกัน

2.11 เครื่องเวกเตอร์เกื้อหนุน (Support Vector Machine, SVM)

เครื่องเวกเตอร์เกื้อหนุน หรือ SVM เป็นโมเดลแบบเชิงเส้นที่สามารถนำมาใช้สำหรับการจำแนกข้อมูลและการวิเคราะห์การถดถอย (Regression Analysis) ถูกพัฒนาขึ้นโดย Vapnik (1995) ซึ่งเป็นโมเดลที่สามารถประยุกต์ใช้ได้ทั้งกับปัญหาที่เป็นแบบเชิงเส้นและไม่เป็นเชิงเส้น แนวคิดหลักของโมเดล SVM คือการสร้างไฮเปอร์เพลนการแบ่งแยกที่เหมาะสมที่สุด (Optimal Separating Hyperplane) เพื่อจำแนกข้อมูลออกเป็น 2 กลุ่ม (Binary Classification) นั่นคือถ้าข้อมูลเป็น 2 มิติ ผลลัพธ์ที่ได้จาก SVM จะเป็นเส้นตรงที่เหมาะสมที่สุด แต่ถ้าข้อมูล 3 มิติขึ้นไปผลลัพธ์ที่ได้จะเป็นระบาบการแบ่งแยก (Separatig Hyperplane) ที่เหมาะสมที่สุดที่สามารถแบ่งข้อมูลออกเป็น 2 กลุ่มได้ดีที่สุด

(1) SVM สำหรับการจำแนกข้อมูลที่เป็นแบบเชิงเส้น

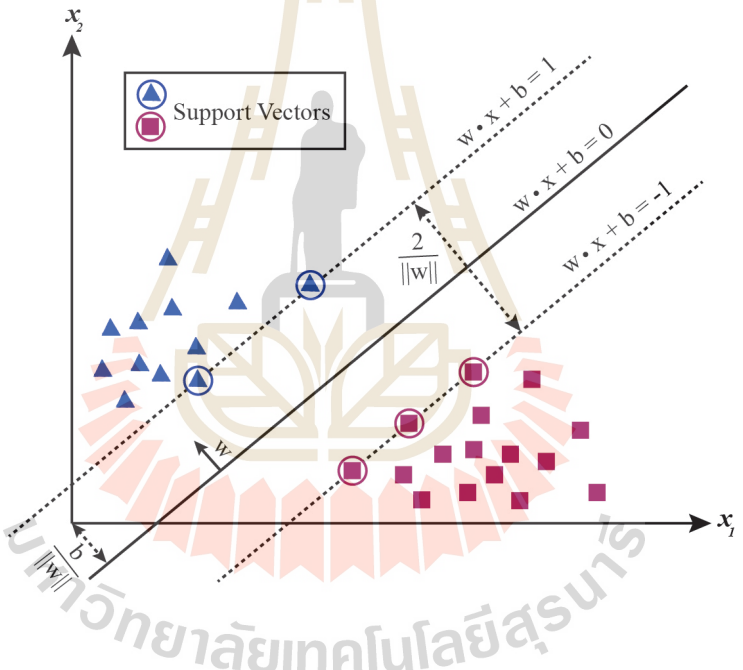
ในกรณีที่ปัญหาเป็นแบบเชิงเส้นและข้อมูลเป็น 2 มิตินั้น ด้วยขั้นตอนวิธีของ SVM จะทำการค้นหา Support Vectors ของแต่ละกลุ่มข้อมูล เมื่อ Support Vectors คือข้อมูลที่อยู่บริเวณขอบ ๆ ของแต่ละกลุ่ม แล้วนำตำแหน่งของ Support Vectors ระหว่าง 2 กลุ่มมาใช้สำหรับการหาระยะแบ่ง (Margin) ที่จะแบ่งข้อมูลออกเป็น 2 กลุ่ม โดยเป้าหมายหลักคือการจะพยายามหาระยะแบ่งที่กว้างที่สุด (Maximun Margin) เพราะมองว่าระยะแบ่งที่กว้างที่สุดนั้นจะสามารถแบ่งข้อมูลออกเป็น 2 กลุ่มได้ดีที่สุด ดังนั้นระยะแบ่งที่ได้จึงเป็นระยะการตัดสินใจ (Decision Boundary) ที่ได้จากขั้นตอนวิธีของ SVM รูปที่ 2.68 แสดงตัวอย่างแนวคิดของ SVM สำหรับการจำแนกข้อมูลแบบเชิงเส้น เมื่อข้อมูลเป็น 2 มิติ จากลักษณะของข้อมูลตัวอย่างในสองกลุ่ม นั่นคือกลุ่มที่เป็นรูปสามเหลี่ยมและกลุ่มที่เป็นรูปสี่เหลี่ยม จะได้ว่า Support Vectors ของแต่ละกลุ่ม (ข้อมูลตัวที่ถูกล้อมรอบด้วยวงกลม) คือข้อมูลตัวที่อยู่บนระยะตัดสินใจ (เส้นตรงที่เป็นเส้นประในรูป) ของแต่ละฝั่ง ดังนั้นจากรูปจะเห็นว่าระยะแบ่งที่กว้างที่สุดจะมีค่าเป็น $\frac{2}{\|\mathbf{w}\|}$ เมื่อ $\mathbf{w}$ คือเวกเตอร์ของค่า weight ที่ได้จากขั้นตอนวิธีของ SVM โดยเวกเตอร์ $\mathbf{w}$ นั้นจะตั้งฉากกับระนาบการแบ่งแยกที่ได้เสมอ เมื่อระนาบการแบ่งแยกที่เหมาะสมที่สุดจากตัวอย่างในรูปที่ 2.68 คือระนาบ $\mathbf{w} \cdot \mathbf{x} + b = 0$ (ในที่นี้เป็นเส้นตรงเพราะข้อมูลเป็น 2 มิติ) เมื่อ $\mathbf{x}$ คือเวกเตอร์ของข้อมูล และ b คือค่าไบอัส โดยระนาบการแบ่งแยกที่ได้จะอยู่ห่างจากจุดกำเหนิดด้วยระยะเท่ากับ $\frac{b}{\|\mathbf{w}\|}$ เมื่อเส้นตรง $\mathbf{w} \cdot \mathbf{x} + b = 1$ ที่เป็นเส้นประในรูปคือระยะตัดสินใจสำหรับข้อมูลที่อยู่ในกลุ่มบวก (Class บวก) และเส้นตรง $\mathbf{w} \cdot \mathbf{x} + b = -1$ คือระยะตัดสินใจสำหรับข้อมูลที่อยู่ในกลุ่มลบ (Class ลบ) นั่นคือด้วยวิธีการของ SVM นั้นแต่ละโมเดลของ SVM จะใช้สำหรับการแบ่งข้อมูลออกเป็นสองกลุ่มเท่านั้น โดยจะมองข้อมูลสองกลุ่มนั้นว่าเป็นกลุ่มที่เป็นบวกและกลุ่มที่เป็นลบ

SVM จัดว่าเป็นวิธีการแบบมีผู้ฝึกสอน นั่นคือต้องมีการนำค่าของเอาต์พุตเป้าหมายมาใช้ร่วมกับข้อมูลในขั้นตอนของการสร้างโมเดลเพื่อหาไฮเปอร์เพลนการแบ่งแยกที่เหมาะสมที่สุดแล้วนำไฮเปอร์เพลนที่ได้นั้นไปใช้สำหรับการจำแนกข้อมูลตัวใหม่ ๆ ต่อไปเพื่อทำการตัดสินใจว่าข้อมูลตัวใหม่นั้นจะอยู่ในกลุ่มบวกหรือกลุ่มลบจากการแทนค่าของข้อมูลลงไปในสมการของไฮเปอร์เพลน

(2) SVM สำหรับการจำแนกข้อมูลที่ไม่เป็นแบบเชิงเส้น

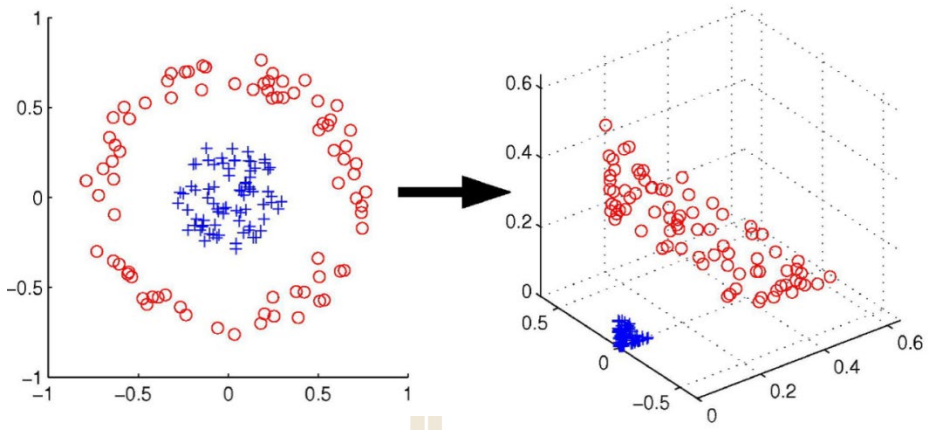
ในกรณีที่ข้อมูลตั้งต้นไม่เป็นเชิงเส้น วิธีการของ SVM จะใช้แนวคิดของการแปลงข้อมูลหรือการแม็ป (Mapping) นั่นคือเป็นการแมปข้อมูลเดิมให้ไปอยู่ในปริภูมิลักษณะ (Feature Space) ใหม่ด้วยการใช้ฟังก์ชันแก่น (Kernel Function) เพื่อให้ข้อมูลที่อยู่ในปริภูมิลักษณะใหม่นั้นเรียงตัวอยู่ในรูปแบบที่สามารถแบ่งออกเป็นสองกลุ่มได้ง่ายขึ้นโดยใช้ตัวแบ่งแยกแบบเชิงเส้น ตัวอย่าง

แนวคิดในการแม็ปแสดงดังรูปที่ 2.69 จะเห็นว่าข้อมูลเดิมที่แสดงด้านซ้ายของรูปซึ่งเป็นข้อมูลใน 2 มิตินั้นไม่สามารถใช้ตัวแบ่งแยกแบบเชิงเส้นมาแบ่งกลุ่มข้อมูลออกจากกันได้ (นั่นคือเป็นข้อมูลแบบไม่เป็นเชิงเส้น) ด้วยหลักการของ SVM จะแม็ปข้อมูลดังกล่าวให้ไปอยู่ในปริภูมิลักษณะใหม่ จากตัวอย่างจะเห็นว่าข้อมูลในปริภูมิลักษณะใหม่มีมิติที่สูงขึ้น โดยการฟังก์ชันการแม็ปคือ $(x_1, x_2) \rightarrow (z_1, z_2, z_3) = \left(x_1^2, \sqrt{2x_1x_2}, x_2^2\right)$ นั่นคือข้อมูลในปริภูมิลักษณะใหม่เป็น 3 มิติ และจากลักษณะของข้อมูลใหม่สามารถใช้ตัวแบ่งแยกแบบเชิงเส้นมาแบ่งข้อมูลออกเป็นสูงกลุ่มได้อย่างชัดเจน ซึ่งด้วยหลักการของ SVM ในกรณีที่ข้อมูลตั้งต้นไม่เป็นเชิงเส้น หลังจากการแม็ปข้อมูลเดิมไปอยู่ในปริภูมิลักษณะใหม่แล้ว การแบ่งกลุ่มข้อมูลในปริภูมิลักษณะใหม่ก็สามารถทำได้เช่นเดียวกันกับกรณีของการจำแนกข้อมูลที่เป็นแบบเชิงเส้นที่ได้กล่าวไปแล้วก่อนหน้านี้



รูปที่ 2.68 แนวคิดในการสร้างตัวแบ่งแยกข้อมูลด้วย SVM สำหรับการจำแนกข้อมูลใน 2 มิติที่เป็นแบบเชิงเส้นออกเป็นสองกลุ่ม

ดังนั้นในกรณีที่ข้อมูลตั้งต้นไม่เป็นเชิงเส้นจึงจำเป็นต้องเลือกฟังก์ชันแก่นที่เหมาะสมสำหรับการแม็ปข้อมูลเดิมไปอยู่ในปริภูมิลักษณะใหม่ โดยฟังก์ชันแก่นที่นิยมใช้เช่นฟังก์ชันพหุนาม (Polynomial Function) ฟังก์ชันฐานรัศมีแบบเกาส์เซียน (Gaussion Radial Basis Function) หรือฟังก์ชันซิกมอยด์ (Sigmoid Function) เป็นต้น



รูปที่ 2.69 แนวคิดในการแม็ปข้อมูลตั้งต้นไปอยู่ในปริภูมิลักษณะใหม่ด้วยฟังก์ชันแก่น (Jordan, 2004)

(3) SVM สำหรับการจำแนกแบบหลายกลุ่ม (Multi-Class Classification)

เนื่องจากแนวคิดหลักของ SVM นั้นเป็นการสร้างโมเดลเพื่อหาไฮเปอร์เพลนการแบ่งแยกที่เหมาะสมที่สุดสำหรับการแบ่งข้อมูลออกเป็น 2 กลุ่มนั่นคือกลุ่มที่เป็นบวกและกลุ่มที่เป็นลบ ในการนำแนวคิดของ SVM ไปใช้สำหรับการจำแนกข้อมูลแบบหลายกลุ่มจึงต้องแบ่งการแก้ปัญหาแบบหลายกลุ่ม (Multi-Class Problem) ให้อยู่ในรูปแบบของชุดของปัญหาย่อยแบบ 2 กลุ่ม (Series of Binary Class Problem) นั่นคือเป็นการสร้างตัวจำแนก (Classifier) สำหรับการแบ่งข้อมูลครั้งละ 2 กลุ่มขึ้นมาหลาย ๆ ตัวจำแนก แล้วนำตัวจำแนกทั้งหมดเหล่านั้นมาพิจารณาเพื่อตัดสินใจจำแนกข้อมูลแบบหลายกลุ่มต่อไป วิธีการในการพิจารณาสร้างตัวจำแนกทั้งหมดมี 2 วิธีคือวิธีการแบบ One Versus One และวิธีการแบบ One Versus All

ถ้าข้อมูลที่ต้องการจำแนกมี N กลุ่ม ด้วยวิธีการแบบ One Versus One จะสร้างตัวจำแนกขึ้นมาทั้งหมดจำนวน $N(N-1)/2$ ตัว เช่น ถ้า $N=4$ จะต้องสร้างตัวจำแนกทั้งหมดขึ้นมา 6 ตัวที่ประกอบด้วย

- Classifier#1: สำหรับจำแนกกลุ่มที่ 1 กับ กลุ่มที่ 2
- Classifier#2: สำหรับจำแนกกลุ่มที่ 1 กับ กลุ่มที่ 3
- Classifier#3: สำหรับจำแนกกลุ่มที่ 1 กับ กลุ่มที่ 4
- Classifier#4: สำหรับจำแนกกลุ่มที่ 2 กับ กลุ่มที่ 3
- Classifier#5: สำหรับจำแนกกลุ่มที่ 2 กับ กลุ่มที่ 4
- Classifier#6: สำหรับจำแนกกลุ่มที่ 3 กับ กลุ่มที่ 4

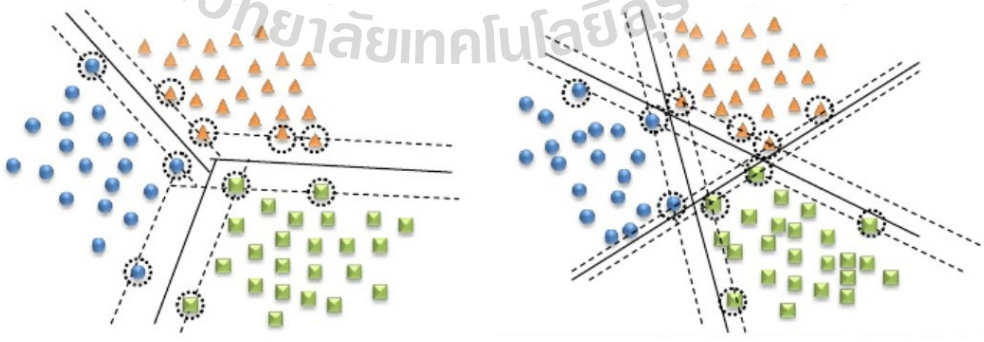
ดังนั้นหลังจากสร้างทั้ง 6 ตัวจำแนกขึ้นมาแล้วจากชุดข้อมูลสำหรับการฝึกสอน ในขั้นตอนการทดสอบ ข้อมูลแต่ละตัวที่ต้องการทดสอบจะต้องนำไปทดสอบกับทั้ง 6 ตัวจำแนกว่าแต่ละตัวจำแนกให้ผลลัพธ์ออกมาว่าอยู่ในกลุ่มใด จากนั้นสามารถใช้วิธีการโหวต (Voting) เพื่อตัดสินว่าข้อมูลตัวที่ทดสอบนั้นอยู่กลุ่มใด

สำหรับวิธีการแบบ One Versus All เมื่อต้องการจำแนกข้อมูล N กลุ่ม จะสร้างตัวจำแนกขึ้นมาทั้งหมดจำนวน N ตัว เช่น ถ้า $N = 4$ จะต้องสร้างตัวจำแนกทั้งหมดขึ้นมา 4 ตัวที่ประกอบด้วย

- Classifier#1: สำหรับจำแนกกลุ่มที่ 1 กับ กลุ่มที่ไม่ใช่ 1 (อื่น ๆ ทั้งหมด)
- Classifier#2: สำหรับจำแนกกลุ่มที่ 2 กับ กลุ่มที่ไม่ใช่ 2 (อื่น ๆ ทั้งหมด)
- Classifier#3: สำหรับจำแนกกลุ่มที่ 3 กับ กลุ่มที่ไม่ใช่ 3 (อื่น ๆ ทั้งหมด)
- Classifier#4: สำหรับจำแนกกลุ่มที่ 4 กับ กลุ่มที่ไม่ใช่ 4 (อื่น ๆ ทั้งหมด)

หลังจากสร้างทั้ง 4 ตัวจำแนกขึ้นมาแล้วจากชุดข้อมูลสำหรับการฝึกสอน ในขั้นตอนการทดสอบ ข้อมูลแต่ละตัวที่ต้องการทดสอบจะต้องนำไปทดสอบกับทั้ง 4 ตัวจำแนกว่าแต่ละตัวจำแนกให้ผลลัพธ์ออกมาว่าอยู่ในกลุ่มใด จากนั้นสามารถใช้วิธีการให้คะแนน (Scoring) เพื่อตัดสินว่าข้อมูลตัวที่ทดสอบนั้นอยู่กลุ่มใด

รูปที่ 2.70 แสดงตัวอย่างของไฮเปอร์เพลนการแบ่งแยกที่เหมาะสมที่สุดที่ได้จากวิธีการแบบ One Versus One และวิธีการแบบ One Versus All ที่ SVM ใช้สำหรับการจำแนกข้อมูลที่มีทั้งหมด 3 กลุ่ม เมื่อรูปที่ 2.70(a) เป็นผลลัพธ์จากวิธีการแบบ One Versus One และรูปที่ 2.70(b) เป็นผลลัพธ์จากวิธีการแบบ One Versus All



(a) One Versus One (b) One Versus All

รูปที่ 2.70 ตัวอย่างของไฮเปอร์เพลนการแบ่งแยกที่เหมาะสมที่สุดที่ได้จากวิธีการที่ SVM ใช้สำหรับการจำแนกข้อมูลที่มีทั้งหมด 3 กลุ่ม (Herrero, 2010)

2.12 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้างในอดีตที่ผ่านมาไม่ได้มีการนำเสนอไว้มากนักเมื่อเทียบกับการจำแนกภาพในงานประยุกต์อื่น แต่ปัจจุบันการจำแนกภาพวัสดุในงานก่อสร้างกำลังเป็นที่สนใจและมีการนำเสนอแนวคิดที่หลากหลายมากขึ้นจากการที่มีเทคโนโลยีที่ทันสมัยมากยิ่งขึ้นมารองรับ รวมถึงมีความพยายามที่จะหาวิธีการแบบอัตโนมัติหรือกึ่งอัตโนมัติมาช่วยในอุตสาหกรรมการก่อสร้างที่เกี่ยวกับการบริหารจัดการงานก่อสร้างในส่วนงานต่าง ๆ โดยเฉพาะอย่างยิ่งสำหรับงานการตรวจติดตามความคืบหน้างานก่อสร้างแบบอัตโนมัตินั้นมีความจำเป็นอย่างยิ่งที่ต้องมีส่วนของงานย่อยหลักเกี่ยวการจำแนกภาพวัสดุในงานก่อสร้าง เพื่อจะได้ประเมินงานส่วนที่แล้วเสร็จและส่วนที่ยังไม่เสร็จได้อย่างถูกต้องแม่นยำ โดยงานวิจัยเกี่ยวกับการคัดแยกประเภทของวัสดุในงานก่อสร้างจากภาพนั้นอาจมองได้ว่าคล้ายคลึงกับงานวิจัยทางด้านการคัดแยกวัสดุ (Material Classification) โดยทั่วไป (Varma and Zisserman, 2009) หรืองานการคัดแยกพื้นผิว (Texture Classification) (Sifre and Mallat, 2013; Cimpoi et al., 2015) ซึ่งเป็นทิศทางของงานวิจัยที่มีการนำเสนอไว้ค่อนข้างหลากหลายและมีการพัฒนาไปมากในแวดวงของการประมวลผลภาพ การมองเห็นของเครื่อง รวมถึงการเรียนรู้ของเครื่อง แต่จากการศึกษาโดย Dimitrov และ Golparvar-Fard (2014) พบว่าเมื่อนำวิธีการที่เคยนำเสนอไว้สำหรับการคัดแยกวัสดุโดยทั่วไปและการคัดแยกพื้นผิวมาใช้กับภาพวัสดุในงานก่อสร้างที่ถ่ายมาจากสถานที่ก่อสร้างจริง โดยส่วนใหญ่ความถูกต้องจะลดลงจากประมาณ 95% เหลือประมาณ 70% หรือต่ำกว่า อันเนื่องมาจากคุณลักษณะเฉพาะของภาพวัสดุในงานก่อสร้างมีความแตกต่างในรายละเอียดออกไปจากภาพของวัสดุโดยทั่วไปและภาพของพื้นผิว

ดังนั้นความพยายามในการหาวิธีการที่เหมาะสมสำหรับการคัดแยกภาพวัสดุในงานก่อสร้างจึงมักเริ่มต้นจากการรวบรวมชุดข้อมูล (Dataset) ที่เป็นภาพวัสดุในงานก่อสร้างโดยเฉพาะแล้วหาวิธีการที่เหมาะสมเพื่อจำแนกภาพจากชุดข้อมูลเหล่านั้น วิธีการโดยส่วนใหญ่ที่มีการนำเสนอไว้ในงานวิจัยที่ผ่านมาจะเป็นวิธีการที่อยู่บนพื้นฐานของการหาคุณลักษณะ (Feature Based Methods) ที่นำเทคนิคทางการประมวลผลภาพ เทคนิคการมองเห็นของเครื่องหรือเทคนิคอื่น ๆ มาช่วยสำหรับการสกัดคุณลักษณะสำคัญบางอย่างออกมาจากภาพ แล้วนำคุณลักษณะเหล่านั้นมาใช้สำหรับการจำแนกความแตกต่างระหว่างแต่ละวัสดุ โดยอาจใช้เทคนิคการเปรียบเทียบความแตกต่างหรือเทคนิคการเรียนรู้ของเครื่องมาใช้ในขั้นตอนการจำแนก โดยสรุปจากการศึกษางานวิจัยที่เกี่ยวข้องกับการคัดแยกภาพวัสดุในงานก่อสร้างมีดังต่อไปนี้

Brilakis และคณะ (2006) ได้นำเสนอวิธีการแบบอัตโนมัติสำหรับการค้นคืนภาพ (Image Retrieval) ภายใต้พื้นฐานของการใช้เนื้อหา (Content Based) ในการระบุส่วนที่เป็นวัสดุและวัตถุ

ต่าง ๆ ในเนื้อหาของภาพ (Image Content) แล้วทำการแบ่งกลุ่มของวัสดุ (Material Clusters) ด้วยการเปรียบเทียบกับตัวอย่างที่รู้ชนิดของวัสดุจากฐานข้อมูลความรู้ (Knowledge Database) วิธีการที่นำเสนอนี้ เริ่มต้นจากการนำภาพมาหาคุณลักษณะพื้นฐานคือ คุณลักษณะทางด้านสี (Color) พื้นผิว (Texture) และโครงสร้าง (Structure) ด้วยการใช้ชุดของตัวกรอง (Filter Bank) จากนั้นจึงแบ่งภาพออกเป็นบริเวณต่าง ๆ โดยใช้การแบ่งกลุ่ม คำนวณ Cluster Signature ของแต่ละกลุ่ม แล้วทำการเปรียบเทียบแต่ละ Cluster Signature กับฐานข้อมูลความรู้ เพื่อระบุชนิดของวัสดุในภาพต้นฉบับแล้วกำหนดชนิดของวัสดุที่ได้ให้เป็น Attribute หนึ่งของภาพสำหรับการนำไปใช้ค้นคืนภาพในฐานข้อมูล งานวิจัยนี้ใช้ค่าขีดแบ่ง (Threshold) สำหรับการกำหนดช่วงของแต่ละ Feature Signature ในฐานข้อมูลความรู้ และใช้ระยะทางแบบยูคลิเดียน (Euclidean Distance) สำหรับการเปรียบเทียบความแตกต่างของ Feature Signature ในการทดลองใช้ภาพทั้งหมด 1025 ภาพที่แบ่งกลุ่มของวัสดุต่าง ๆ ออกเป็น 20 กลุ่ม โดยใช้ข้อมูลจาก 3 กลุ่มคือ Earth Concrete และ Paint สำหรับการทดสอบแล้ววัดประสิทธิภาพของวิธีการที่นำเสนอด้วยด้วยค่า Precision และ Recall โดยแสดงผลเปรียบเทียบด้วยกราฟระหว่างเปอร์เซ็นต์ของ Precision และเปอร์เซ็นต์ของ Recall ซึ่งพบว่า ค่า Precision จะแปรผกผันกับค่า Recall และ ณ ค่า Recall ต่ำ ๆ ค่า Precision จะสูงเกือบ 100%

Zhu และ Brilakis (2010) นำเสนอวิธีการใหม่สำหรับการระบุ (Identification) บริเวณที่เป็นคอนกรีตในภาพโดยใช้เทคนิคการเรียนรู้ของเครื่อง โดยเริ่มต้นจากขั้นตอนการประมวลผลก่อน (Pre-Process) เพื่อกำจัดสัญญาณรบกวน (Noise) ในภาพจากการใช้ Median Filter จากนั้นแบ่งภาพออกเป็นแต่ละบริเวณด้วยวิธีการแยกส่วนภาพ (Image Segmentation) ทำการคำนวณ Visual Features ของแต่ละบริเวณแล้วจำแนก Visual Features เหล่านั้นด้วยตัวจำแนกที่ผ่านการเรียนรู้มาก่อน (Pre-Trained Classifier) ด้วยการใช้เทคนิคการเรียนรู้ของเครื่องคือ SVM และ ANN โดยผลลัพธ์ที่ได้จากการจำแนกด้วยวิธีการที่นำเสนอทำการเปรียบเทียบกับผลลัพธ์จากการจำแนกด้วยมือ (Manual Classification) สำหรับส่วนของ Visual Features ที่ใช้ในงานวิจัยนี้ประกอบด้วยส่วนของ Color และ Texture โดยคุณลักษณะเกี่ยวกับ Color นั้นใช้อัตราส่วนของสีแดงต่อสีเขียว (R2G) และอัตราส่วนของน้ำเงินต่อสีเขียว (B2G) และคุณลักษณะเกี่ยวกับ Texture ได้มาจากการใช้ชุดตัวกรองแบบ RFS (Root Filter Set) ในขั้นตอนการจำแนก ทำการฝึกสอนโดยใช้ 114 ภาพตัวอย่าง ที่เป็นกลุ่มของคอนกรีตซึ่งจัดเป็นกลุ่มบวก (Positive Concrete) จำนวน 63 ภาพ และกลุ่มที่ไม่ใช่คอนกรีตซึ่งจัดเป็นกลุ่มลบ (Negative Concrete) จำนวน 51 ภาพ สำหรับการทดสอบนั้น ใช้ภาพตัวอย่าง 167 ภาพ ที่เป็นกลุ่มบวก 53 ภาพ และกลุ่มลบ 114 ภาพ ผลลัพธ์จากการจำแนกด้วย SVM และ ANN พบว่า ANN มีประสิทธิภาพมากกว่า SVM จากการที่ได้ค่าเฉลี่ยของ Precision และ Recall ประมาณ 80%

Rashidi และคณะ (2016) ได้ทำการศึกษาเปรียบเทียบเพื่อประเมินประสิทธิภาพของเทคนิคต่าง ๆ ทางด้านการเรียนรู้ของเครื่องสำหรับการระบุประเภทวัสดุที่เป็นวัสดุของอาคาร (Building Materials) ใน 3 ประเภทคือ Concrete, Red Brick, และ OSB (Oriented Strand Boards) โดยตัวจำแนกจากเทคนิคการเรียนรู้ของเครื่องที่ศึกษาคือ MLP, RBF และ SVM จากผลการศึกษาพบว่า SVM ดีกว่าตัวจำแนกอื่น ๆ จากการวัดผลด้วยค่า Accuracy งานวิจัยนี้นำเสนอการใช้คุณลักษณะหลักจากภาพใน 3 รูปแบบคือ RGB Histogram, HSV Histogram และ Histogram of Dominant Edge โดยที่ Histogram of Dominant Edge ได้มาจากการการแปลงภาพ RGB เป็นภาพระดับเทา นำภาพระดับเทานั้นมาผ่านการทำ Gabor Wavelet เพื่อสกัด Dominant Edge ออกมาแล้วนับจุดภาพทั้งหมดที่ได้ออกมาเป็นค่าของ Histogram คุณลักษณะจากภาพใน 3 รูปแบบที่หาออกมาได้ดังกล่าวถูกนำไปจำแนกกับแต่ละตัวจำแนกที่ต้องการศึกษาเปรียบเทียบ โดยทดลองกับภาพทั้งหมด 750 ภาพ ที่เป็นลักษณะการทดลองกับปัญหาแบบ 2 กลุ่มคือกลุ่มที่เป็นวัสดุเป้าหมาย (Target Material) และกลุ่มที่ไม่เป็นวัสดุเป้าหมาย เช่นกลุ่มที่เป็น Concrete กับกลุ่มที่ไม่เป็น Concrete จากการทดลองพบว่า SVM ด้วยการใช้ RBF Kernel สามารถจำแนกได้ด้วยค่า Accuracy ดีที่สุด โดยค่า Accuracy จากการระบุ Concrete และ OSB Board อยู่ที่ 75-95% ค่า Precision และ Recall สำหรับการระบุอิฐแดงอยู่ที่ 94% และ 96% ตามลำดับ

Son และคณะ (2014) ได้ศึกษาประสิทธิภาพของตัวจำแนกเดี่ยว ๆ จำนวน 6 ตัวเปรียบเทียบกับตัวจำแนกแบบ Ensemble ที่นำเสนอขึ้นสำหรับการนำมาใช้จำแนกข้อมูลภาพวัสดุในงานก่อสร้าที่ประกอบด้วย 3 กลุ่มคือ Concrete, Steel และ Wood ในลักษณะการทดลองกับปัญหาแบบ 2 กลุ่ม คือกลุ่มเป้าหมายและไม่ใช่กลุ่มเป้าหมาย โดยสร้างตัวจำแนกแบบ Voting-Based Ensemble จาก 6 ตัวจำแนกเดี่ยวคือ SVM, ANN, C4.5 (Commercial Version 4.5), Naïve Bayes (NB), Logistic Regression (LR) และ KNN (k-Nearest Neighbors) และเปรียบเทียบผลลัพธ์ที่ได้ด้วยค่า Accuracy, Precision, Sensitivity, Specificity และ S (Average Score) คุณลักษณะที่นำมาใช้สำหรับงานวิจัยนี้มีเพียง 3 คุณลักษณะจากค่าในแต่ละแกนสีของโมเดลสี HSV คือค่าในแกนสี H แกนสี S และแกนสี V ผลการทดลองพบว่าตัวจำแนกแบบ Ensemble ที่นำเสนอได้ค่าความถูกต้องในการระบุชนิดของวัสดุโดยรวมที่ดีกว่าแต่ละตัวจำแนกเดี่ยว ๆ

Dimitrov และ Golparvar-Fard (2014)ได้นำเสนอวิธีการแบบ Vision-Based ในการจำแนกวัสดุในงานก่อสร้างจากภาพที่ไม่ทราบมุมมอง (Viewpoint) และไม่ทราบสภาวะของแสง (Illumination Condition) โดยนำเสนออัลกอริทึมสำหรับการโมเดล Material Appearance ด้วย Joint Probability Distribution ของคุณลักษณะต่าง ๆ ที่ได้มาจากการใช้ชุดของตัวกรอง และจากค่าในแต่ละแกนสีของโมเดลสี HSV แล้วนำมาจำแนกด้วย SVM การนำเสนอหลักของงานวิจัยนี้คือ

การนำ Bag of Words (BoW) Pileline มาสร้าง Statistical Distribution ของค่าที่ได้จากการใช้ชุดของตัวกรอง LM (Leung and Malik) และค่าในแต่ละแกนสีของโมเดลสี HSV และใช้การแบ่งกลุ่มด้วย k-Means Clustering สำหรับสร้างเป็น Codebooks ของ Material Appearance โดยกำหนดจุดกึ่งกลางของแต่ละกลุ่มเป็น *Textons* ดังนั้นในขั้นตอนการฝึกสอนจึงเป็นการเรียนรู้เพื่อโมเดล Texton Frequencies แล้วใช้ Histogram ของ Texton Frequencies สร้างเป็น Codebooks งานวิจัยนี้ได้สร้างฐานข้อมูลภาพสำหรับการทดลองที่ประกอบด้วย 20 ชนิดของวัสดุในงานก่อสร้าง ในแต่ละชนิดมีมากกว่า 150 ภาพ จากผลลัพธ์ที่ได้พบว่าได้ค่าความถูกต้องจากการจำแนกโดยเฉลี่ย 97.1% สำหรับการทดลองกับภาพที่มีขนาดเป็น 200×200 และเมื่อขนาดของภาพเล็กลงเป็น 30×30 ได้ค่าความถูกต้องจากการจำแนกโดยเฉลี่ยที่ 90.8%

DeGol และคณะ (2016) ได้ทำการศึกษาเกี่ยวกับการนำคุณลักษณะทาง 3D Geometry คือ Surface Normal, Camera Intrinsic และ Extrinsic Parameters มาใช้ร่วมกันกับคุณลักษณะทาง 2D คือ Texture และ Color ในการเพิ่มประสิทธิภาพการจำแนกวัสดุในงานก่อสร้าง คุณลักษณะทาง 2D ที่นำมาใช้ทดลองคือ คุณลักษณะจาก RFS Filter Bank และ MR8 Filter Bank, Fisher Vector, HSV Color และ CNN จากโครงข่าย Pre-Trained VGG-M โดยคุณลักษณะจาก Filter Bank และจาก HSV Color ใช้แนวคิดของการทำ Clustering มาใช้ร่วมด้วย และนำ SVM มาใช้ในขั้นตอนการจำแนก งานวิจัยนี้ทำการทดลองด้วยชุดข้อมูลที่ประกอบด้วยวัสดุต่าง ๆ ในงานก่อสร้างจำนวน 19 ประเภท ซึ่งเป็นชุดข้อมูลที่ผู้วิจัยได้รวบรวมขึ้นมาเองสำหรับใช้ในการทดลอง ที่ประกอบด้วยข้อมูล 2 ชุดคือ ชุดที่เป็นแบบ Focus Scale และแบบ Scene Scale โดยชุดที่เป็น Focus Scale ใช้สำหรับการฝึกสอนและการทดสอบด้วย ส่วนแบบ Scene Scale ใช้เฉพาะการทดสอบ จากการทดลองในหลากหลายรูปแบบของการนำคุณลักษณะต่าง ๆ มาใช้ร่วมกันพบว่า ในกรณีที่ไม่นำคุณลักษณะทางด้าน 3D มาใช้นั้น Fisher Vector กับ CNN เมื่อนำมาใช้ร่วมกัน จะได้ผลลัพธ์จากการจำแนกโดยเฉลี่ยดีที่สุดคือ 68.92% และเมื่อนำคุณลักษณะทางด้าน 3D มาใช้ร่วมด้วยพบว่า เมื่อนำ Surface Normal มาใช้ร่วมกันกับทั้ง Fisher Vector และ CNN จะได้ผลลัพธ์จากการจำแนกโดยเฉลี่ยดีที่สุดคืออยู่ 73.80%

จากวิธีการที่เคยนำเสนอไว้ในงานวิจัยส่วนใหญ่พบว่า แทบทั้งหมดใช้ข้อมูลภาพจากฐานข้อมูลที่สร้างขึ้นเองสำหรับงานวิจัยนั้น ๆ และไม่มีการเผยแพร่ข้อมูลภาพดังกล่าว ถึงแม้ว่าในบางงานวิจัย เช่นงานวิจัยของ DeGol และคณะ (2016) จะมีการสร้างฐานข้อมูลที่ค่อนข้างหลากหลายเกี่ยวกับภาพวัสดุในงานก่อสร้างมาใช้และมีการนำฐานข้อมูลดังกล่าวมาเผยแพร่ แต่ผลลัพธ์ที่ได้จากการจำแนกในงานวิจัยดังกล่าวก็ยังถือว่าไม่สูง นอกจากนี้เทคนิคใหม่ ๆ ทางด้านการเรียนรู้ของเครื่องที่กำลังเป็นที่สนใจอย่างกว้างขวาง เช่น เทคนิคการเรียนรู้เชิงลึก ก็ยังไม่ได้มี

นำเสนอเพื่อศึกษาอย่างครอบคลุมสำหรับการประยุกต์ใช้เพื่อจำแนกภาพวัสดุในงานก่อสร้าง ดังนั้น งานวิจัยนี้จึงต้องการนำเสนอการศึกษาเพื่อนำเทคนิคการเรียนรู้เชิงลึกที่เหมาะสมกับข้อมูลรูปภาพมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ นั่นคือคือโครงข่ายประสาทแบบคอนโวลูชัน ซึ่งเป็นสถาปัตยกรรมของโครงข่ายที่เป็นความก้าวหน้าใหม่ทางด้านการเรียนรู้ของเครื่อง ที่กำลังเป็นที่สนใจในการนำไปประยุกต์ใช้ในงานต่าง ๆ อย่างกว้างขวาง โดยเฉพาะงานที่เกี่ยวข้องกับข้อมูลภาพ แต่จากการสำรวจงานวิจัยที่เกี่ยวข้องพบว่า โครงข่ายประสาทแบบคอนโวลูชันยังไม่มีการนำมาประยุกต์ใช้โดยตรงสำหรับการจำแนกข้อมูลภาพเกี่ยวกับวัสดุในงานก่อสร้าง นั่นคือวิธีการหรือแนวคิดในการนำโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้เพื่อการจำแนกข้อมูลภาพวัสดุในงานก่อสร้างจึงยังไม่เคยมีการนำเสนอไว้ ดังนั้นงานวิจัยนี้จึงมุ่งศึกษาในส่วนของวิธีการและแนวคิดต่าง ๆ ในการประยุกต์ใช้ดังกล่าว เพื่อต้องการนำเสนอวิธีการหรือแนวคิดที่มีประสิทธิภาพสำหรับการนำเทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ

จากงานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้างที่กล่าวมาข้างต้นรวมทั้งงานที่ต้องการนำเสนอเพื่อศึกษาในวิทยานิพนธ์ฉบับนี้ เมื่อนำมาสรุปเปรียบเทียบกันในแต่ละกระบวนการที่เกี่ยวข้อง แสดงดังตารางที่ 2-9

ตารางที่ 2-9 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้าง

กระบวนการที่เกี่ยวข้อง	งานวิจัยที่เกี่ยวข้อง						
	ก	ข	ค	ง	จ	ฉ	ช*
คุณลักษณะที่ใช้							
Intensity	✓						
RGB Descriptors	✓	✓					
HSV Descriptors				✓	✓		
Textures (Used Filter Bank)	✓	✓			✓		
RGB Histogram			✓				
HSV Histogram			✓				
Histogram of Dominant Edge			✓				
Fisher Vector						✓	
Surface Normal						✓	
CNN (Fixed Feature from Pre-Trained Model)						✓	✓
CNN (Fine Tune of Pre-Trained Model)							✓
CNN (Self-Train on Studied Data Set)							✓

ตารางที่ 2-9 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้าง (ต่อ)

กระบวนการที่เกี่ยวข้อง	งานวิจัยที่เกี่ยวข้อง						
	ก	ข	ค	ง	จ	ฉ	ช*
คุณลักษณะที่ใช้							
Included Knowledge Database	✓						
Included Clustering Based	✓				✓		
Proposed Features							✓
เทคนิคที่ใช้ในการจำแนก							
Thresholds	✓						
Euclidean Distance	✓						
Artificial Neural Network (ANN)		✓	✓				
Support Vector Machine (SVM)		✓	✓		✓	✓	✓
Radial Basis Function (RBF)			✓				
Ensemble (SVM, ANN, C4.5, NB, LR, KNN)				✓			
Convolution Neural Network (CNN)							✓
มาตรวัดที่ใช้วัดประสิทธิภาพของโมเดล							
Accuracy		✓		✓	✓	✓	✓
Precision	✓	✓	✓	✓	✓		✓
Recall	✓	✓	✓		✓		✓
Generality		✓					
Sensitivity				✓			
Specificity				✓			
F-Measure							✓
AUC (Quantitative describing the ROC curve)				✓			
S (Average Performance Score)				✓			
ชุดข้อมูลที่ใช้							
Non-Public	✓	✓	✓	✓	✓		
Public						✓	✓
จำนวนกลุ่มของข้อมูลที่พิจารณา							
2 Classes (Focus on 1 Category)		✓					
2 Classes (Focus on Multiple Categories)			✓	✓			
Multiple Classes	✓				✓	✓	✓

ตารางที่ 2-9 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกภาพวัสดุในงานก่อสร้าง (ต่อ)

กระบวนการที่เกี่ยวข้อง	งานวิจัยที่เกี่ยวข้อง						
	ก	ข	ค	ง	จ	ฉ	ช*
วัตถุประสงค์ของการวิจัย							
Construction Materials Images Retrieval	✓						
Construction Materials Detection		✓	✓	✓			
Construction Materials Classification	✓	✓	✓	✓	✓	✓	✓

งานวิจัยที่เกี่ยวข้องประกอบด้วย

ก แทนงานวิจัยของ Brilakis และคณะ (2006)

ข แทนงานวิจัยของ Zhu และ Brilakis (2010)

ค แทนงานวิจัยของ Rashidi และคณะ (2016)

ง แทนงานวิจัยของ Son และคณะ (2014)

จ แทนงานวิจัยของ Dimitrov และ Golparvar-Fard (2014)

ฉ แทนงานวิจัยของ DeGol และคณะ (2016)

ช* แทนงานของวิทยานิพนธ์ฉบับนี้

บทที่ 3
วิธีดำเนินงานวิจัย

วัตถุประสงค์ของงานวิจัยนี้เป็นการนำเทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยต้องการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนคือโมเดล ResNet101 มาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ เพื่อนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสแบบอัตโนมัติ และเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มสำหรับการเพิ่มประสิทธิภาพในการจำแนกภาพวัสดุในงานก่อสร้าง เพื่อให้การดำเนินงานวิจัยสอดคล้องตามวัตถุประสงค์ดังกล่าว ในบทนี้จึงเป็นการนำเสนอในส่วนของ ชุดข้อมูลที่ศึกษา กรอบแนวคิดงานวิจัย งานวิจัยที่นำเสนอวิธีการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในรูปแบบอื่นเพื่อการศึกษาเปรียบเทียบกับวิธีการที่นำเสนอ และการเปรียบเทียบประสิทธิภาพของการจำแนก ตามลำดับ

3.1 ชุดข้อมูลที่ศึกษา

ชุดข้อมูลที่ศึกษาสำหรับงานวิจัยนี้คือข้อมูลภาพวัสดุในงานก่อสร้างที่ประกอบด้วยข้อมูลสี่ชนิด (4 Classes) ของวัสดุในงานก่อสร้าง โดยสามชนิดของภาพวัสดุในงานก่อสร้างมาจากข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะ และอีกหนึ่งชนิดของวัสดุเป็นข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าที่สร้างขึ้นเองสำหรับงานวิจัยนี้ ซึ่งได้มาจากการถ่ายภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า โดยข้อมูลทั้งสี่ชนิดของวัสดุในงานก่อสร้างใช้ในการทดลองสำหรับวิธีการที่นำเสนอขึ้นในงานวิจัยนี้ ส่วนในขั้นตอนของการศึกษาเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการในรูปแบบอื่นๆ นั้นทำการทดลองกับเฉพาะชุดข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะที่ประกอบด้วยสามชนิดของภาพวัสดุในงานก่อสร้าง โดยรายละเอียดของแต่ละชุดข้อมูลภาพอธิบายดังข้อ (1) และข้อ (2)

(1) ข้อมูลภาพวัสดุในงานก่อสร้างจากฐานข้อมูลที่เผยแพร่

เป็นข้อมูลภาพที่นำมาจากส่วนหนึ่งของข้อมูลภาพที่เผยแพร่ในงานวิจัยของ Degol และคณะ ที่ประกอบด้วยภาพวัสดุที่ใช้กันโดยทั่วไปในงานก่อสร้างในสามวัสดุหลักคือ อิฐ (Brick)

คอนกรีต (Concrete) และไม้ (Wood) โดยเป็นภาพย่อยๆ (Patches) ที่ตัดมาจากภาพใหญ่แบบความละเอียดสูง (high-resolution) จากการสุ่มตัวอย่างในภาพความละเอียดสูงแต่ละภาพที่ความละเอียด 100×100, 200×200, 400×400, และ 800×800 แล้วปรับขนาด (Resize) ทุกภาพย่อยเป็น 100×100 ดังรายละเอียดการสร้างชุดข้อมูลภาพนี้ใน (Degol el al., 2016) ซึ่งแต่ละกลุ่ม (Class) ของวัสดุจะแบ่งเป็นภาพที่ใช้สำหรับการฝึกสอนทั้งหมด 400 ภาพ และเป็นภาพที่ใช้สำหรับการทดสอบ 200 ภาพ ดังนั้นสำหรับชุดข้อมูลนี้มีภาพที่ใช้ในการฝึกสอนทั้งหมด 1200 ภาพ และภาพที่ใช้ในการทดสอบทั้งหมด 600 ภาพ ตัวอย่างภาพที่ใช้สำหรับการทดสอบในแต่ละกลุ่มของวัสดุแสดงดังรูปที่ 3-1

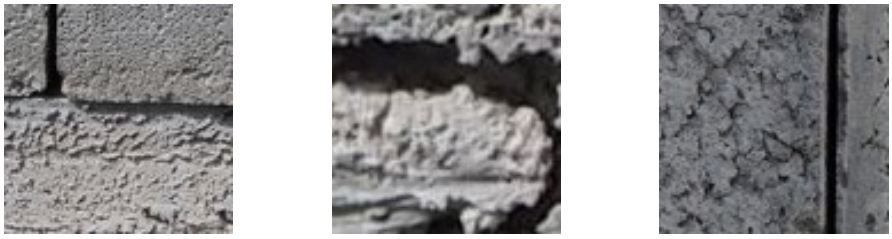
(a) Brick (b) Concrete (a) Wood

รูปที่ 3-1 ตัวอย่างข้อมูลภาพจากชุดทดสอบที่เผยแพร่ในงานวิจัยของ Degol และคณะ

(2) ข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า

เป็นข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าที่สร้างขึ้นเองสำหรับงานวิจัยนี้ ซึ่งได้มาจากการถ่ายภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าที่พัฒนาโดย ผศ. ดร. ธีรวัฒน์ สินศิริ สาขาวิชาวิศวกรรมโยธา มหาวิทยาลัยเทคโนโลยีสุรนารี ข้อมูลภาพชุดนี้สร้างมาจากการตัดภาพย่อย ๆ มาจากภาพใหญ่แบบความละเอียดสูง (high-resolution) จากการสุ่มตัวอย่างในภาพความละเอียดสูงแต่ละภาพที่ความละเอียด 100×100, 200×200, 400×400, และ 800×800 แล้วปรับขนาดทุกภาพย่อยเป็น 100×100 ดังรายละเอียดการสร้างเช่นเดียวกันกับชุดข้อมูลที่มาจากงานวิจัยของ Degol และคณะ ซึ่งชุดข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่านี้แบ่งเป็นภาพที่ใช้สำหรับการฝึกสอน

ทั้งหมด 400 ภาพ และเป็นภาพที่ใช้สำหรับการทดสอบ 200 ภาพ ตัวอย่างภาพที่ใช้สำหรับการทดสอบแสดงดังรูปที่ 3-2



รูปที่ 3-2 ตัวอย่างข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่าจากชุดทดสอบ

ดังนั้นข้อมูลที่ใช้สำหรับงานวิจัยนี้ประกอบด้วยข้อมูล 2 ชุดคือ

ชุดข้อมูลที่ 1: เป็นข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะซึ่งประกอบด้วย 3 กลุ่มคือ อิฐ ไม้ และคอนกรีต

ชุดข้อมูลที่ 2: เป็นข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะรวมข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า ดังนั้นจึงประกอบด้วย 4 กลุ่มคือกลุ่มคือ อิฐ ไม้ คอนกรีตและอิฐมวลเบาคอนกรีตเซลลูล่า

3.2 กรอบแนวคิดงานวิจัย

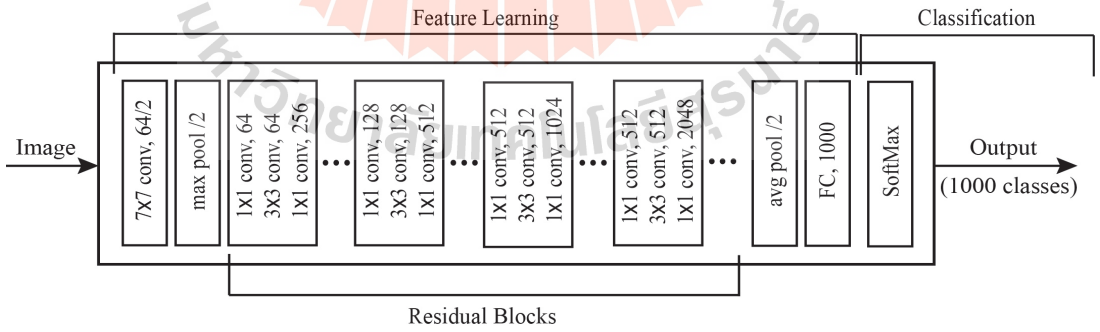
วัตถุประสงค์ของการพัฒนางานวิจัยนี้คือ การนำเสนอการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยต้องการนำเสนอแนวคิดการประยุกต์ใช้ในรูปแบบของการเรียนรู้แบบถ่ายโอน ที่เป็นการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนคือโมเดล ResNet101 มาประยุกต์ใช้ เพื่อนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสแบบอัตโนมัติ เพื่อสร้างรูปแบบการแทนข้อมูลที่เหมาะสม และนำเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มมาใช้สำหรับการเพิ่มประสิทธิภาพในการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ ซึ่งนอกเหนือจากวิธีการในรูปแบบที่นำเสนอดังกล่าวแล้ว งานวิจัยนี้ได้ทำการศึกษาเปรียบเทียบวิธีการในรูปแบบอื่น ๆ สำหรับการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติด้วย

งานวิจัยนี้นำโมเดลแบบที่ผ่านการฝึกสอนมาก่อนของโมเดล ResNet101 มาใช้เพราะโมเดล ResNet101 นั้นเป็นโมเดลที่มีความลึก 101 ชั้นที่มีการนำหลักการของ Residual Learning จากการนำ Residual Blocks มาใช้ในโครงข่ายในลักษณะเดียวกันกับโมเดล ResNet152 ที่มีความลึก 152 ชั้นที่ชนะการแข่งขันในงาน ILSVRC 2015 แต่เนื่องจากโมเดล ResNet152 นั้นไม่มีการเผยแพร่โมเดลแบบฝึกสอนมาก่อนสำหรับการใช้งานในโปรแกรม Matlab ซึ่งงานวิจัยนี้เป็นการพัฒนาโดยใช้โปรแกรม Matlab ดังนั้นโมเดล ResNet101 ที่มีระดับความลึกรองลงมาจึงถูกเลือกเพื่อนำมาใช้สำหรับการพัฒนางานวิจัยนี้เพราะเป็นโมเดลมีการเผยแพร่สถาปัตยกรรมแบบฝึกสอนมาก่อนสำหรับการใช้งานในโปรแกรม Matlab

เพื่อให้การเนินงานวิจัยสอดคล้องตามวัตถุประสงค์ที่กล่าวข้างต้น และเพื่อให้มองเห็นถึงมุมมองโดยรวมในส่วนของวิธีการที่นำเสนอ รายละเอียดต่าง ๆ ที่เกี่ยวข้องกับวิธีการที่นำเสนอจึงอธิบายดังต่อไปนี้

3.2.1 การประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันด้วยการเรียนรู้แบบถ่ายโอน

รูปที่ 3-3 แสดงรายละเอียดในส่วนต่าง ๆ ของโมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนด้วยชุดข้อมูลภาพของ ImageNet (มองว่าเป็นข้อมูลภาพจาก Source Task) ซึ่งเป็นข้อมูลที่มี 1000 Classes โดยตั้งแต่ชั้นแรกของโครงข่ายจนถึงชั้น FC, 1000 สามารถมองว่าคือส่วนของการทำการเรียนรู้เพื่อหาคุณลักษณะ (Feature Learning) และส่วนท้ายของโครงข่ายที่เป็นชั้น SoftMax ซึ่งเป็นการนำคุณลักษณะที่เรียนรู้มาได้จากชั้นก่อนหน้ามาผ่านฟังก์ชัน SoftMax เพื่อสร้างเป็น Output ของโครงข่ายนั้นมองว่าเป็นส่วนของการจำแนก (Classification)



รูปที่ 3-3 ส่วนต่าง ๆ ของโมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนด้วยชุดข้อมูลภาพของ ImageNet (Source Task) ซึ่งเป็นข้อมูลที่มี 1000 Classes

ในการนำโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้ในรูปแบบของการเรียนรู้แบบถ่ายโอนนั้นสามารถประยุกต์ใช้ได้ใน 2 รูปแบบคือ แบบยึดคุณลักษณะจากตัวสกัด (Fixed Feature Extractor) และแบบปรับแต่งการเรียนรู้ (Fine Tune Learning) ซึ่งรายละเอียดของแนวคิดสำหรับการประยุกต์ใช้ใน 2 รูปแบบดังกล่าวอธิบายในข้อ (1) และข้อ (2) ตามลำดับต่อไปนี้

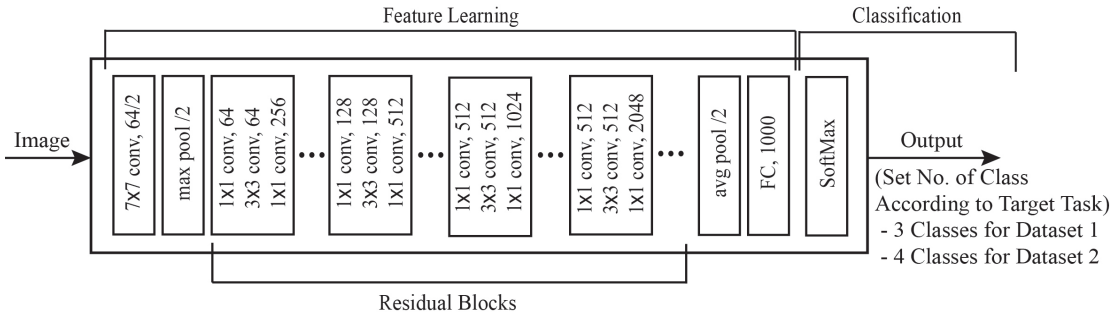
(1) แนวคิดของการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัด

ในการนำโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้ในรูปแบบของการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัดด้วยการใช้โมเดล ResNet101 นั้นเริ่มต้นจากเปลี่ยนจำนวนกลุ่ม (Class) ของ Output ในโครงข่ายจากเดิม 1000 Classes ของรูปที่ 3-3 ให้มีจำนวนตามจำนวนกลุ่มของข้อมูลที่ต้องการจำแนกในชุดข้อมูลที่ศึกษา (มองว่าชุดข้อมูลที่ศึกษา คือ Target Task สำหรับการเรียนรู้แบบถ่ายโอน) ดังแสดงในส่วนของ Output ในรูปที่ 3-4 นั่นคือสำหรับงานวิจัยนี้ซึ่งทำการศึกษาใน 2 ชุดข้อมูล ดังนั้นเมื่อศึกษาชุดข้อมูลที่ 1 (Dataset 1) จึงกำหนดให้ส่วนของ Output เป็น 3 Classes และเมื่อศึกษาชุดข้อมูลที่ 2 (Dataset 2) ก็จะกำหนดให้ส่วนของ Output เป็น 4 Classes ตามรายละเอียดของแต่ละชุดข้อมูลที่กล่าวไปแล้วในหัวข้อ 3.1

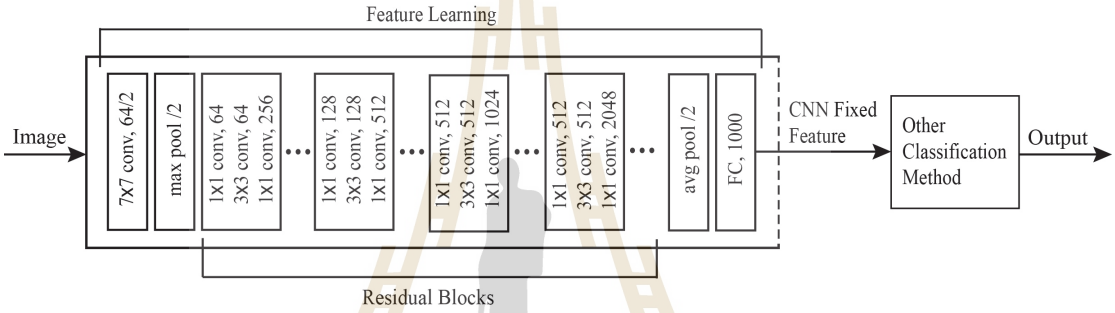
การเปลี่ยนจำนวนกลุ่มของ Output ในโครงข่ายให้สอดคล้องตามจำนวนกลุ่มของชุดข้อมูลที่ศึกษานั้นเป็นขั้นตอนแรกสุดที่ต้องจัดการเมื่อต้องการนำการเรียนรู้แบบถ่ายโอนมาใช้ทั้งในแบบยึดคุณลักษณะจากตัวสกัดและแบบปรับแต่งการเรียนรู้ โดยในการประยุกต์ใช้ในแบบยึดคุณลักษณะจากตัวสกัดนั้นสามารถแบ่งรูปแบบของแนวคิดในการประยุกต์ใช้ออกเป็นอีก 3 รูปแบบย่อยดังแสดงในรูปที่ 3-4 รูปที่ 3-5 และรูปที่ 3-6 ตามลำดับ

รูปที่ 3-4 เป็นการประยุกต์ใช้ในแบบยึดคุณลักษณะจากตัวสกัดที่ยังคงยึดตามรูปแบบเดิมของโครงข่ายที่ผ่านการฝึกสอนมาก่อนมากที่สุด นั่นคือเป็นการนำข้อมูลภาพในชุดข้อมูลที่ศึกษามาแปลงเป็นข้อมูลของคุณลักษณะจากการใช้ค่าของ Weights และ Biases ที่ผ่านการฝึกสอนมาก่อนแล้วและใช้ฟังก์ชัน SoftMax สำหรับขั้นตอนการจำแนกตามลักษณะเดิมของโครงข่าย

รูปที่ 3-5 เป็นการประยุกต์ใช้ในแบบยึดคุณลักษณะจากตัวสกัดที่นำวิธีการจำแนกแบบอื่นมาใช้แทนฟังก์ชัน SoftMax จากรูปจะเห็นได้ว่าข้อมูลภาพจะถูกนำมาแปลงเป็นข้อมูลของคุณลักษณะจากการใช้ค่าของ Weights และ Biases ที่ผ่านการฝึกสอนมาก่อนจนถึงชั้น FC, 1000 แล้วนำคุณลักษณะดังกล่าวที่ได้ (เรียกว่า CNN Fixed Feature ในรูปที่ 3-5) ไปผ่านขั้นตอนการจำแนกโดยใช้วิธีการจำแนกแบบอื่นที่ไม่ใช่ฟังก์ชัน SoftMax

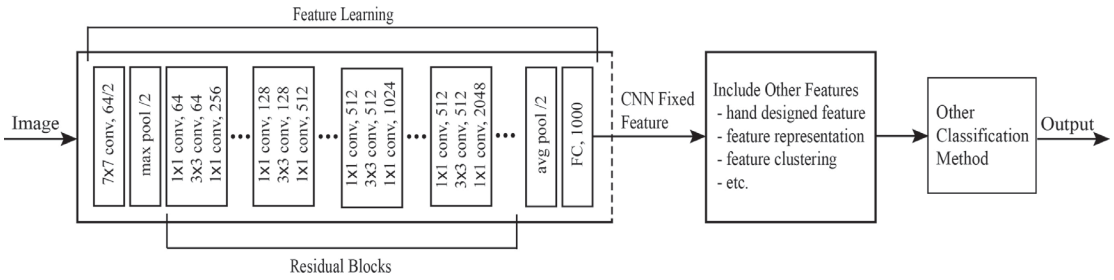


รูปที่ 3-4 การกำหนดจำนวน Class ของ Output ตาม Target Task ในแบบยึดคุณลักษณะจากตัวสกัด



รูปที่ 3-5 การนำวิธีการจำแนกแบบอื่นมาใช้แทนฟังก์ชัน SoftMax ในแบบยึดคุณลักษณะจากตัวสกัด

รูปที่ 3-6 เป็นการประยุกต์ใช้ในแบบยึดคุณลักษณะจากตัวสกัดที่เป็นการนำคุณลักษณะที่สกัดออกมาได้จากโครงข่าย CNN (ซึ่งคือส่วนของ CNN Fixed Feature ในรูปที่ 3-6) ที่ได้มาด้วยวิธีเดียวกันกับในรูปที่ 3-5 โดยที่ข้อมูลภาพจะถูกนำมาแปลงเป็นข้อมูลของคุณลักษณะจากการใช้ค่าของ Weights และ Biases ที่ผ่านการฝึกสอนมาก่อนจนถึงชั้น FC, 1000 แล้วนำ CNN Fixed Feature ที่ได้มานั้นไปใช้ร่วมกันกับวิธีการสำหรับการหาคุณลักษณะแบบอื่น (ซางคือขั้นตอน Include Other Features ในรูปที่ 3-6) แล้วไปผ่านขั้นตอนการจำแนกโดยใช้วิธีการจำแนกแบบอื่นที่ไม่ใช่ฟังก์ชัน SoftMax ซึ่งวิธีการที่นำเสนอขึ้นสำหรับงานวิจัยนี้เพื่อการจำแนกภาพวัสดุในงานก่อสร้างนั้นเป็นการประยุกต์โดยใช้ในรูปแบบดังแสดงในรูปที่ 3-6 ดังกล่าว



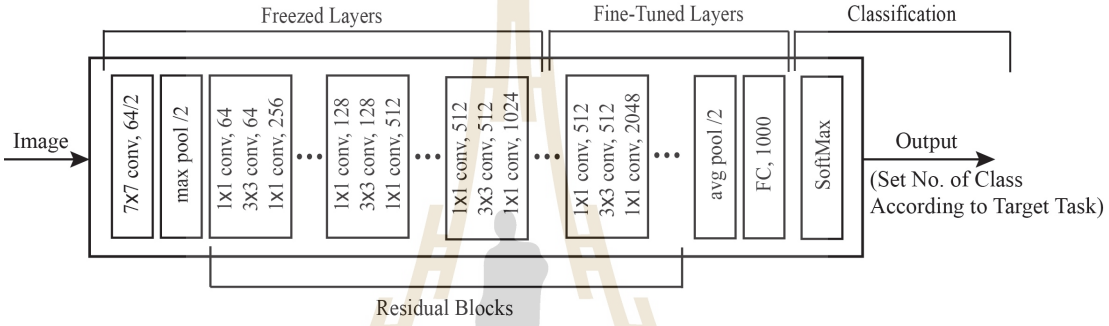
รูปที่ 3-6 การนำวิธีการหาคุณลักษณะแบบอื่นมาใช้ร่วมกับ CNN Fixed Feature และนำวิธีการจำแนกแบบอื่นมาใช้ในแบบยึดคุณลักษณะจากตัวสกัด

(2) แนวคิดของการเรียนรู้แบบถ่ายโอนในแบบปรับแต่งการเรียนรู้

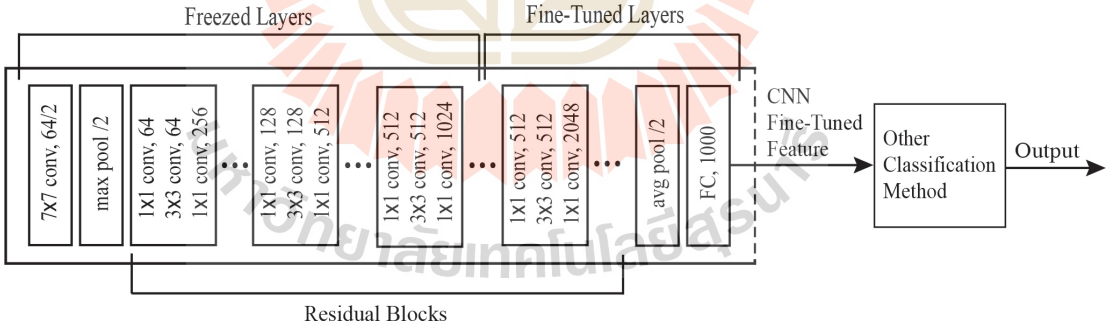
การเรียนรู้แบบถ่ายโอนในแบบปรับแต่งการเรียนรู้นั้นเป็นการนำบางส่วนของโครงข่ายที่ผ่านการฝึกสอนมาก่อนแล้วนั้นมาผ่านการเรียนรู้เพิ่มเติมด้วยชุดข้อมูลที่ต้องการศึกษา โดยมักจะฝึกสอนเพิ่มเติมในส่วนของชั้นท้าย ๆ ของส่วน Feature Learning แต่ชั้นในส่วนต้น ๆ จะคงไว้เหมือนเดิม นั่นคือเป็นการมองว่า ค่าของ Weights และ Biases ของชั้นที่ไม่ได้ฝึกสอนเพิ่มเติมขะถูกแช่แข็งไว้ (Freezed) หมายความว่าค่า Weights และ Biases ของชั้นเหล่านั้นไม่มีการเปลี่ยนแปลงยังคงใช้ตามค่าที่ได้รับการถ่ายโอนมาจากโมเดลที่ผ่านการฝึกสอนมาก่อน ในขณะที่ค่า Weights และ Biases ของชั้นที่ได้รับการฝึกสอนใหม่จะถูกปรับแต่ง (Fine Tune) เพื่อให้เหมาะสมกับชุดข้อมูลที่ศึกษามากยิ่งขึ้น จากแนวคิดที่กล่าวมา รายละเอียดต่าง ๆ ของการเรียนรู้แบบถ่ายโอนในแบบปรับแต่งการเรียนรู้เมื่อนำ ResNet101 มาใช้จึงแสดงได้ดังรูปที่ 3-7 เมื่อ Freezed Layers ในรูปคือช่วงของชั้นที่ไม่ต้องการฝึกสอนเพิ่มเติม ซึ่งในทางปฏิบัติสามารถกำหนดเป็นจำนวนกี่ช่วงชั้นก็ได้ตามความเหมาะสมของแต่ละ Target Task ในส่วนของ Fine-Tuned Layers คือช่วงของชั้นที่ต้องการให้มีการปรับแต่งหรือฝึกสอนเพิ่มเติม

ในการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในแบบปรับแต่งการเรียนรู้นั้นสามารถแบ่งรูปแบบของแนวคิดในการประยุกต์ใช้ออกเป็นอีก 3 รูปแบบย่อยเช่นเดียวกันกับที่กล่าวมาแล้วในแบบยึดคุณลักษณะจากตัวสกัด ดังแสดงในรูปที่ 3-7 รูปที่ 3-8 และรูปที่ 3-9 ตามลำดับ ซึ่งความแตกต่างหลักระหว่างแบบยึดคุณลักษณะจากตัวสกัดและแบบปรับแต่งการเรียนรู้คือ แบบยึดคุณลักษณะจากตัวสกัดไม่ต้องมีการฝึกสอนใหม่ในขณะที่แบบปรับแต่งการเรียนรู้ต้องมีการฝึกสอนใหม่สำหรับบางช่วงชั้น

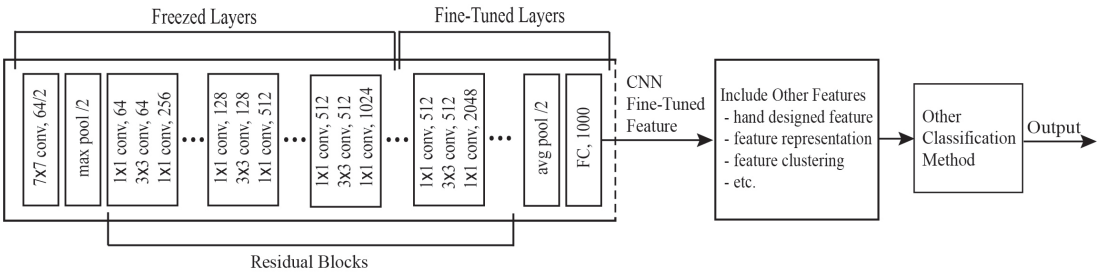
ดังนั้นคุณลักษณะที่ได้จากขั้นตอน Feature Learning จากแบบปรับแต่งการเรียนรู้จึงเป็นคุณลักษณะที่ผ่านการปรับแต่งจากการฝึกสอนใหม่ คุณลักษณะที่ได้ดังกล่าว (นั่นคือ CNN Fine-Tuned Feature) จึงนำไปใช้ต่อได้ใน 3 รูปแบบย่อยคือ แบบที่ใช้ฟังก์ชัน SoftMax สำหรับขั้นตอนการจำแนกตามลักษณะเดิมของโครงข่าย ดังแสดงในรูปที่ 3-7 แบบที่นำวิธีการจำแนกแบบอื่นมาใช้แทนฟังก์ชัน SoftMax ดังแสดงในรูปที่ 3-8 และแบบที่นำ CNN Fine-Tuned Feature ที่ได้มานั้นไปใช้ร่วมกันกับวิธีการสำหรับการหาคุณลักษณะแบบอื่น แล้วไปผ่านขั้นตอนการจำแนกโดยใช้วิธีการจำแนกแบบอื่นที่ไม่ใช่ฟังก์ชั้น SoftMax ดังแสดงในรูปที่ 3-9



รูปที่ 3-7 การกำหนดจำนวน Class ของ Output ตาม Target Task ในแบบปรับแต่งการเรียนรู้



รูปที่ 3-8 การนำวิธีการจำแนกแบบอื่นมาใช้แทนฟังก์ชัน SoftMaxในแบบปรับแต่งการเรียนรู้



รูปที่ 3-9 การนำวิธีการหาคุณลักษณะแบบอื่นมาใช้ร่วมกับ CNN Fine-Tuned Feature และนำวิธีการจำแนกแบบอื่นมาใช้ในแบบปรับแต่งการเรียนรู้

3.2.2 ขั้นตอนหลักของวิธีการที่นำเสนอ

วิธีการที่นำเสนอสำหรับงานวิจัยนี้เป็นการนำโครงข่ายประสาทแบบคอนโวลูชันจากโมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้ในแบบยึดคุณลักษณะตามรูปแบบที่แสดงในรูปที่ 3-6 โดยแบ่งวิธีการที่นำเสนอออกเป็นขั้นตอนหลัก 4 ขั้นตอนดังแผนภาพในรูปที่ 3-10 ในแต่ละขั้นตอนดังกล่าวมีรายละเอียดดังต่อไปนี้

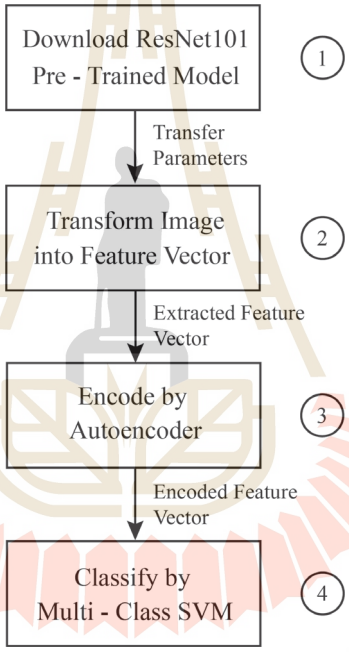
(1) การ download โมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อน (Download ResNet101 Pre-Trained Model)

เป็นการ download เพื่อนำโมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วและมีการนำมาเผยแพร่ให้สามารถนำไปใช้ประโยชน์ต่อได้ ซึ่งโมเดล ResNet101 ที่นำมาใช้นั้นเป็นโมเดลรูปแบบเดียวกันกับโมเดล ResNet152 ที่ชนะการแข่งขันในงาน ILSVRC 2015 แต่จำนวนชั้นต่างกัน นั่นคือโมเดลที่นำมาใช้สำหรับงานวิจัยนี้มีความลึก 101 ชั้น แต่โมเดลที่ชนะการแข่งขันในงาน ILSVRC 2015 มีความลึก 152 ชั้น ซึ่งโมเดลของ ResNet101 ที่นำมาใช้นี้ผ่านการฝึกสอนมาก่อนจากชุดข้อมูลภาพของ ImageNet 1.2 ล้านภาพ (อาจเรียกว่าภาพจาก Source Task) ดังนั้นขั้นตอนการ download เพื่อนำโมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วสำหรับงานวิจัยนี้คือขั้นตอนการ import โมเดล ResNet101 เข้ามาใน Matlab R2018a เพื่อนำมาประยุกต์ใช้ต่อในรูปแบบของการถ่ายโอนการเรียนรู้

(2) การแปลงข้อมูลภาพไปเป็นเวกเตอร์คุณลักษณะ (Transform Image into Feature Vector)

ในการนำโมเดลที่ผ่านการฝึกสอนมาก่อนมาใช้ต่อในรูปแบบของการถ่ายโอนการเรียนรู้นั้น ข้อมูลที่ถ่ายโอนมาจากโมเดลที่ผ่านการฝึกสอนมาก่อนคือรายละเอียดเกี่ยวกับสถาปัตยกรรมของโครงข่ายและค่าพารามิเตอร์ต่าง ๆ ของโครงข่ายที่ผ่านการฝึกสอนมาแล้ว ซึ่งคือค่าของ

Weights และ Biases ทั้งหมดที่ใช้ในสถาปัตยกรรมของโครงข่าย (ซึ่งคือ Transfer Parameters ในรูปที่ 3-10) ดังนั้นเมื่อต้องการนำโมเดลดังกล่าวมาประยุกต์ใช้แบบการถ่ายโอนการเรียนรู้ด้วยวิธีการยึดคุณลักษณะจากตัวสกัดนั้น ข้อมูลของภาพที่เราต้องการศึกษาแต่ละภาพ (อาจเรียกว่าภาพจาก Target Task) จะถูกส่งผ่านเข้าไปในโครงข่าย 1 รอบ (1 pass) เพื่อนำค่าของ Weights และ Biases ที่ผ่านการฝึกสอนมาก่อนแล้วนั้นสำหรับการแปลงข้อมูลภาพไปเป็นข้อมูลของเวกเตอร์คุณลักษณะตามรายละเอียดในแต่ละชั้นของโครงข่ายจนถึงชั้นสุดท้ายที่ต้องการนำผลลัพธ์ของเวกเตอร์คุณลักษณะที่สกัดออกมาได้ (Extracted Feature Vector) นั้นไปใช้งานต่อไป ซึ่งขั้นตอนนี้คือขั้นตอนหมายเลข 2 ในรูปที่ 3-10



รูปที่ 3-10 สรุป 4 ขั้นตอนหลักของวิธีการที่นำเสนอ

(3) การเข้ารหัสด้วยเครื่องเข้ารหัสอัตโนมัติ (Encode by Autoencoder)

เป็นการนำเวกเตอร์คุณลักษณะที่สกัดออกมาได้จากขั้นตอนที่ 2 มาผ่านการหารูปแบบของการแทนข้อมูล (Data Representation) หรือการเข้ารหัสข้อมูลด้วยการใช้โครงข่ายการเข้ารหัสอัตโนมัติ เพื่อช่วยสำหรับการลดมิติของข้อมูลและเป็นการหารูปแบบการแทนข้อมูลที่เหมาะสมมากยิ่งขึ้นก่อนที่จะนำไปใช้สำหรับการจำแนกต่อไป ในงานวิจัยนี้นำเสนอการใช้เทคนิคของเครื่องเข้ารหัสอัตโนมัติสำหรับการหารูปแบบของการแทนข้อมูลดังกล่าว โดยผลลัพธ์จากขั้นตอนนี้คือเวกเตอร์คุณลักษณะที่ถูกเข้ารหัส (Encoded Feature Vector) ดังแสดงในรูปที่ 3-10

(4) การจำแนกด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม (Classify by Multi-Class SVM)

เป็นขั้นตอนที่ใช้สำหรับการจำแนกกลุ่มของข้อมูลเพื่อให้ได้ผลลัพธ์ที่ต้องการว่าภาพวัสดุในงานก่อสร้างแต่ละภาพที่นำมาจำแนกนั้นคือวัสดุอะไร ข้อมูลที่นำเข้าสู่ขั้นตอนนี้คือเวกเตอร์คุณลักษณะที่ถูกเข้ารหัสจากขั้นตอนที่ 3 ก่อนหน้านี้ เนื่องจากงานวิจัยนี้เป็นการศึกษาเพื่อจำแนกข้อมูลแบบหลายกลุ่ม โมเดลของเครื่องเวกเตอร์เกื้อหนุนจึงเป็นโมเดลสำหรับการจำแนกแบบหลายกลุ่มในกลยุทธ์แบบหนึ่งต่อหนึ่ง (One versus One) นั่นคือเป็นการการสร้างตัวจำแนกของเครื่องเวกเตอร์เกื้อหนุนแบบจำแนกเป็นสองกลุ่ม (Binary Classification) หลาย ๆ ตัวมาใช้ประกอบร่วมกันแบบหนึ่งต่อหนึ่งสำหรับการจำแนกข้อมูลแบบมีมากกว่าสองกลุ่ม ขั้นตอนนี้คือขั้นตอนที่ 4 ของรูปที่ 3-10

3.2.3 เครื่องมือที่ใช้สำหรับการวิจัย

เครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ ประกอบด้วย

(1) เครื่องคอมพิวเตอร์สำหรับพัฒนา มีรายละเอียดดังนี้

หน่วยประมวลผลกลาง : Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz

หน่วยประมวลผลกราฟิก : NVIDIA GeForce GTX 1060 3GB

หน่วยความจำหลัก : 16 GB

หน่วยความจำสำรอง : 1 TB

(2) ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับพัฒนา ประกอบด้วย

ระบบปฏิบัติการ : Windows 7 Ultimate 64 bits

เครื่องมือที่ใช้ในการพัฒนา : Matlab R2018a

3.3 งานวิจัยที่นำเสนอ (Proposed Work)

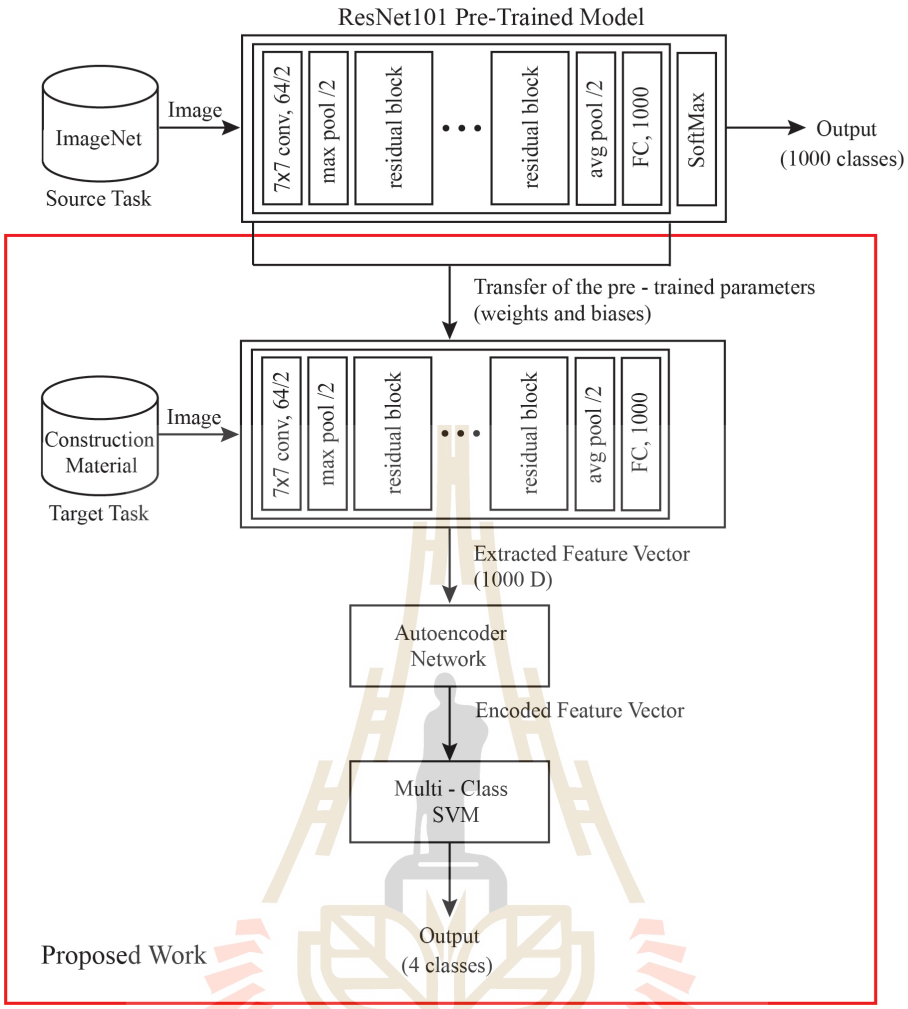
จากกรอบแนวคิดงานวิจัยที่อธิบายด้วยขั้นตอนหลักของวิธีการที่นำเสนอที่ได้กล่าวไปในหัวข้อ 3.2 ก่อนหน้านี้ เพื่อให้เห็นภาพชัดเจนยิ่งขึ้นในส่วนรายละเอียดของวิธีการที่นำเสนอในงานวิจัยนี้ รูปที่ 3-11 จึงแสดงรายละเอียดเพิ่มเติมเกี่ยวกับวิธีการที่นำเสนอดังกล่าว นั่นคือในการนำโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้ในการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัตินั้นเริ่มต้นจากการนำเข้า (Import) โมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนเข้ามาในโปรแกรม Matlab เวอร์ชัน R2018a เพื่อต้องการนำรูปแบบของสถาปัตยกรรมและค่าพารามิเตอร์ต่าง ๆ จากโมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วนั้นมาใช้ในลักษณะของการถ่ายโอนการเรียนรู้ ซึ่งงานวิจัยนี้ใช้การถ่ายโอนการเรียนรู้ในแบบยึดคุณลักษณะจากตัวสกัด วิธีการในการ

การถ่ายโอนการเรียนรู้ดังกล่าวเริ่มต้นจากการนำภาพแต่ละภาพจากชุดข้อมูลที่ศึกษามาส่งผ่านเข้าไปในโมเดล 1 รอบเพื่อนำค่าพารามิเตอร์ (ค่า Weights และBiases) ของโมเดลที่ผ่านการฝึกสอนมาก่อนแล้วนั้นสำหรับการแปลงข้อมูลภาพที่ส่งเข้าไปเป็นข้อมูลของเวกเตอร์คุณลักษณะ นั่นคือข้อมูลภาพจะถูกส่งผ่านไปในแต่ละชั้นของโมเดลและค่อย ๆ แปลงข้อมูลภาพดั้งต้นนั้นไปเป็นแผนที่คุณลักษณะ (Feature Map) ตามลักษณะสถาปัตยกรรมของโครงข่ายในแต่ละชั้นจนถึงชั้นสุดท้ายก่อนขั้นตอนการจำแนก นั่นคือชั้น FC, 1000 (ชั้นการเชื่อมถึงกันหมดที่มีจำนวน 1000 นิวรอน) ดังนั้นผลลัพธ์จากขั้นตอนนี้คือเวกเตอร์คุณลักษณะที่สกัดออกมาได้จากโมเดล เป็นเวกเตอร์ที่มีจำนวนมิติเป็น 1000 ตามจำนวนของนิวรอนที่ใช้ในชั้น FC, 1000 ของโมเดล ResNet101

จากรูปที่ 3-11 จะเห็นว่าชั้นของ SoftMax ที่เป็นชั้นสุดท้ายของโครงข่าย ResNet101 จะไม่ได้ถูกนำมาใช้ในขั้นตอนนี้ เพราะเป้าหมายสำหรับการทำขั้นตอนนี้เพื่อต้องการนำเวกเตอร์คุณลักษณะที่สกัดออกมาได้ไปใช้งานในขั้นตอนอื่นต่อไป ยังไม่ได้เป็นการจำแนกกลุ่มของข้อมูล นั่นคือเป็นการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในรูปแบบตามรายละเอียดในรูปที่ 3-6 ที่ได้นำเสนอไปก่อนหน้านี้ ดังนั้นเวกเตอร์คุณลักษณะที่สกัดออกมาได้จากวิธีการถ่ายโอนการเรียนรู้ในแบบยึดคุณลักษณะจากตัวสกัดนี้จึงถูกนำไปผ่านขั้นตอนที่ 2 ต่อไป นั่นคือคือเป็นอินพุตเข้าสู่โครงข่ายการเข้ารหัสอัตโนมัติ (Autoencoder Network ในรูปที่ 3-11) เพื่อสร้างรูปแบบการแทนข้อมูลที่เหมาะสมเป็นเวกเตอร์คุณลักษณะที่ถูกเข้ารหัส แล้วนำเวกเตอร์ดังกล่าวไปจำแนกด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มต่อไป

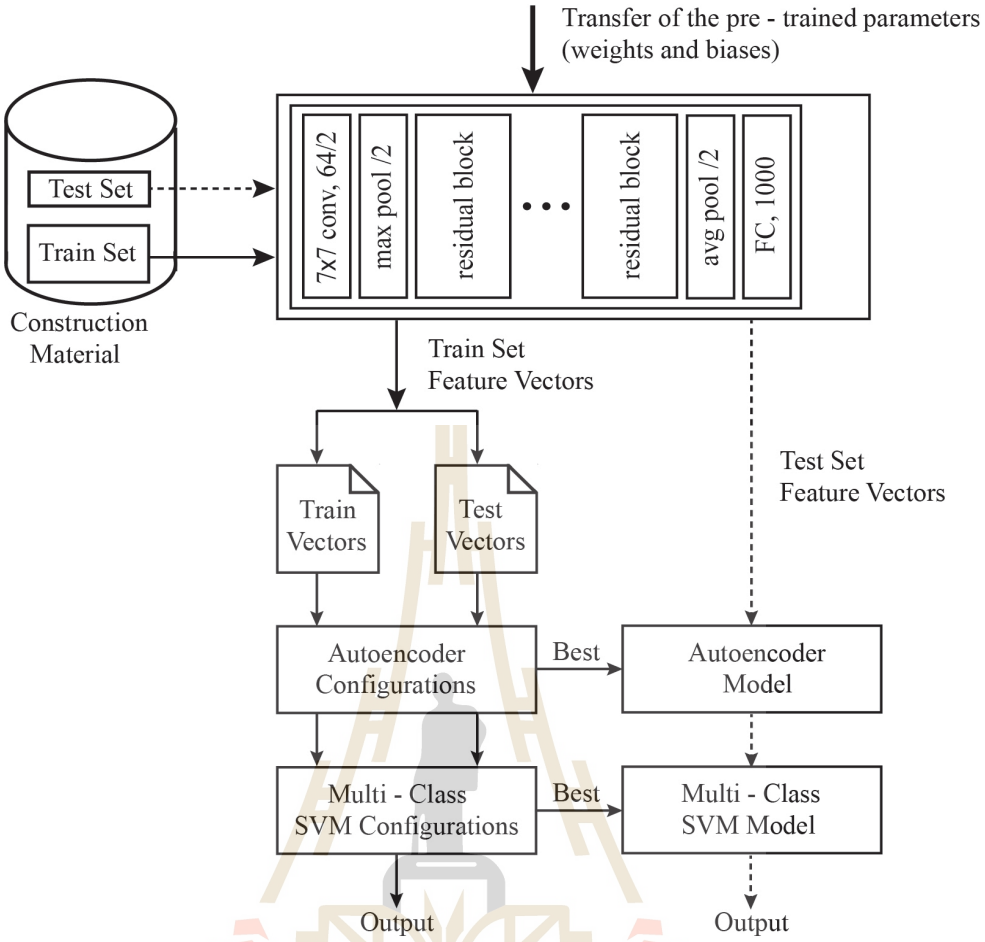
3.3.1 รายละเอียดการฝึกสอนและการทดสอบในวิธีการที่นำเสนอ

เนื่องจากวิธีการที่นำเสนอสำหรับงานวิจัยนี้ นอกเหนือจากการประยุกต์ใช้งานโครงข่ายประสาทแบบคอนวูชันแล้ว เพื่อให้ผลลัพธ์ที่ได้จากการจำแนกที่มีประสิทธิภาพมากยิ่งขึ้น จึงนำโครงข่ายของเครื่องเข้ารหัสอัตโนมัติมาใช้สำหรับการสร้างตัวแทนข้อมูลก่อนเข้าสู่ขั้นตอนการจำแนก ซึ่งเครื่องเข้ารหัสอัตโนมัติจัดได้ว่าเป็นโครงข่ายประสาทเทียมชนิดหนึ่ง ดังนั้นเพื่อให้ได้โมเดลที่เหมาะสมของโครงข่ายสำหรับชุดข้อมูลที่ศึกษา จำเป็นต้องมีขั้นตอนของการการฝึกสอนให้กับโครงข่ายด้วยชุดข้อมูลที่ศึกษาและขั้นตอนของการทดสอบสำหรับการหาโมเดลที่เหมาะสมที่สุด เพื่อให้เห็นภาพชัดเจนมากยิ่งขึ้นสำหรับขั้นตอนดังกล่าว รูปที่ 3-12 จึงแสดงรายละเอียดที่ขยายความเพิ่มเติมจากรูปที่ 3-11 ในส่วนที่เกี่ยวข้องกับขั้นตอนการฝึกสอนและการทดสอบในวิธีการที่นำเสนอทั้งหมด



รูปที่ 3-11 รายละเอียดของวิธีการในงานวิจัยที่นำเสนอ

จากรูปที่ 3-12 แสดงรายละเอียดให้เห็นว่าวิธีการที่นำเสนอสำหรับงานวิจัยนี้แบ่งชุดข้อมูลภาพวัสดุในงานก่อสร้างที่ศึกษาออกเป็นสองชุดคือชุดฝึกสอน (Train Set) และชุดทดสอบ (Test Set) ซึ่งภาพวัสดุแต่ละชนิดจะอยู่ในชุดฝึกสอนจำนวน 400 ภาพ และอยู่ในชุดทดสอบจำนวน 200 ภาพ ขั้นตอนที่เกี่ยวข้องกับการฝึกสอนในวิธีการที่นำเสนอแสดงด้วยลูกศรที่เป็นเส้นทึบในรูปที่ 3-12 ส่วนขั้นตอนที่เกี่ยวข้องกับการทดสอบแสดงด้วยลูกศรที่เป็นเส้นประ



รูปที่ 3-12 รายละเอียดการฝึกสอน (Train) และการทดสอบ (Test) ในวิธีการที่นำเสนอ

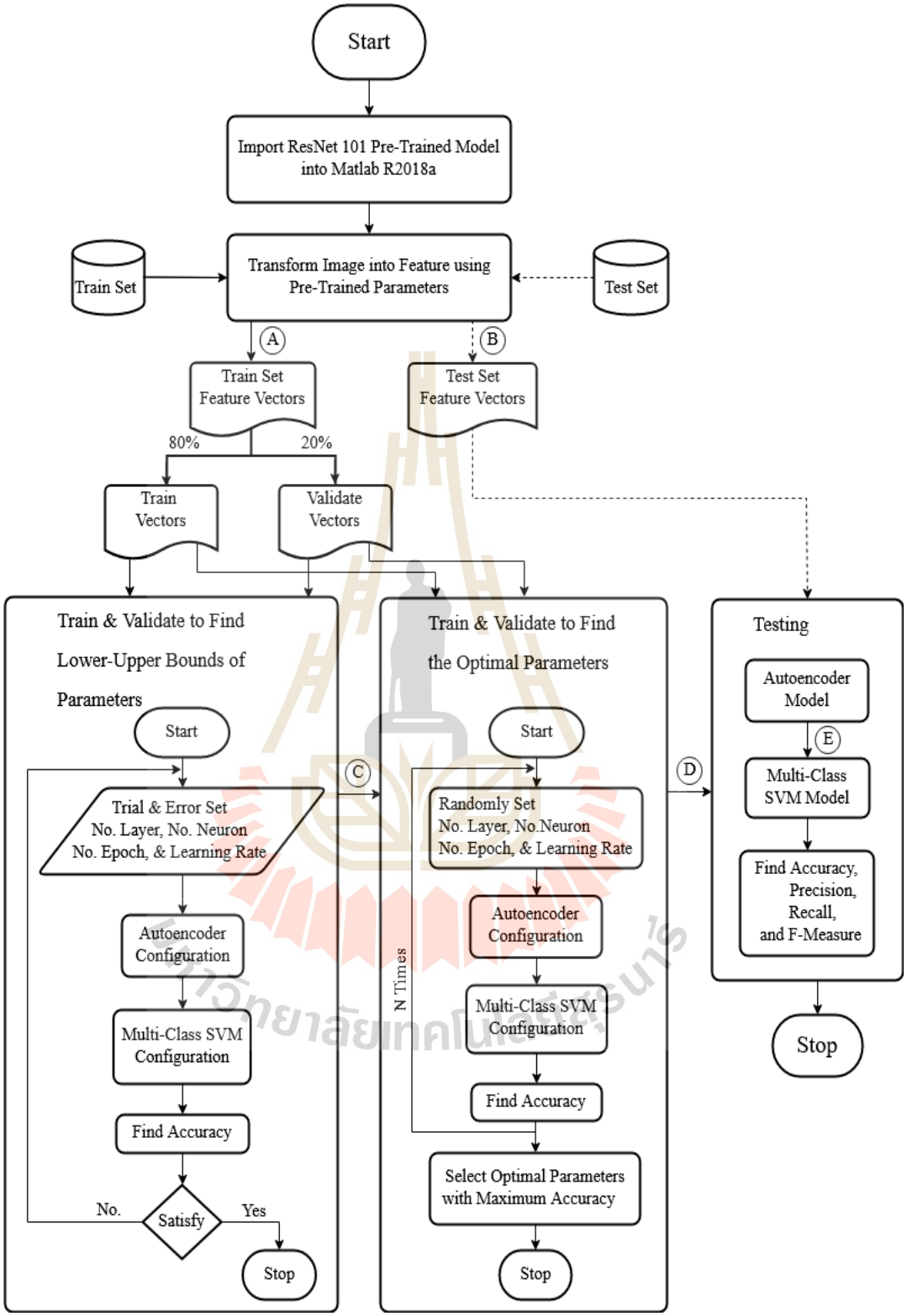
ขั้นตอนการฝึกสอนเริ่มต้นจากการนำภาพในชุดฝึกสอนแต่ละภาพมาส่งเข้าสู่โมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วและนำมาประยุกต์ใช้ในรูปแบบการถ่ายโอนการเรียนรู้แบบยึดคุณลักษณะจากตัวสกัดเพื่อแปลงข้อมูลภาพแต่ละภาพไปเป็นข้อมูลเวกเตอร์คุณลักษณะด้วยการใช้ Weights และ Biases ที่ถูกถ่ายโอนมา ภาพแต่ละภาพจึงถูกแปลงเป็นเวกเตอร์คุณลักษณะที่มีจำนวนมิติเป็น 1000 (1000 D) ตามคุณลักษณะของ output จากชั้น FC, 1000 ของโมเดล ResNet101 ดังนั้นเมื่อทุกภาพในข้อมูลชุดฝึกสอนถูกส่งผ่านเข้าไปในโมเดลแล้วจะได้ผลลัพธ์ทั้งหมดเป็นเวกเตอร์คุณลักษณะของชุดฝึกสอน (Train Set Feature Vectors) จากนั้นจึงนำเวกเตอร์คุณลักษณะเหล่านั้นมาแบ่งเป็นสองชุดอีกเพื่อใช้สำหรับการฝึกสอนและการทดสอบให้กับโครงข่ายเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุน ดังนั้นจากเวกเตอร์คุณลักษณะทั้งหมดจากข้อมูลชุดฝึกสอนที่เป็นผลลัพธ์จากโมเดล ResNet101 จึงถูกนำมาแบ่งเป็นชุดของเวกเตอร์สำหรับการฝึกสอน (Train Vectors) และชุดของเวกเตอร์สำหรับการทดสอบ (Test Vectors) ดังแสดงในรูป

ที่ 3-12 นั่นคือชุดของเวกเตอร์สำหรับการฝึกสอนจะนำมาใช้สำหรับการฝึกสอนให้กับโครงข่ายเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุนเพื่อต้องการหารูปแบบของโมเดลที่เหมาะสมสำหรับการสร้างรูปแบบการแทนข้อมูลและการจำแนกข้อมูล ดังนั้นในขั้นตอนนี้โครงแบบต่าง ๆ ของเครื่องเข้ารหัสอัตโนมัติ (Autoencoder Configurations) และโครงแบบต่าง ๆ ของเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม (Multi-Class SVM Configurations) จึงถูกสร้างขึ้นและฝึกสอนจากการใช้เวกเตอร์สำหรับการฝึกสอน โดยโครงแบบที่ดีที่สุด (The Best Configuration) จะได้มาจากการประเมินด้วยเวกเตอร์สำหรับการทดสอบ ดังนั้นโครงแบบที่ดีที่สุดของเครื่องเข้ารหัสอัตโนมัติจึงถูกเลือกไปใช้เป็นโมเดลของเครื่องเข้ารหัสอัตโนมัติ (Autoencoder Model) และโครงแบบที่ดีที่สุดของเครื่องเวกเตอร์เกื้อหนุนก็จะถูกเลือกไปใช้เป็นโมเดลของเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม (Multi-Class SVM Model) สำหรับการทดสอบกับข้อมูลชุดทดสอบต่อไป ตามรายละเอียดของขั้นตอนในรูปที่ 3-12

ขั้นตอนการทดสอบนั้นเป็นขั้นตอนที่ใช้เพื่อวัดประสิทธิภาพของการจำแนกจากวิธีการที่นำเสนอหลังจากผ่านขั้นตอนการฝึกสอนแล้ว โดยใช้ชุดข้อมูลภาพในชุดทดสอบ ซึ่งเริ่มต้นจากการนำภาพแต่ละภาพจากชุดทดสอบส่งเข้าสู่โมเดล ResNet101 เพื่อแปลงข้อมูลภาพเป็นข้อมูลเวกเตอร์คุณลักษณะเช่นเดียวกันกับที่ทำในขั้นตอนการฝึกสอน และเรียกเวกเตอร์คุณลักษณะทั้งหมดเหล่านั้นว่า เวกเตอร์คุณลักษณะของข้อมูลชุดทดสอบ (Test Set Feature Vectors) ดังแสดงในรูปที่ 3-12 ในส่วนที่แสดงด้วยลูกศรเส้นประ จากนั้นจึงนำเวกเตอร์คุณลักษณะทั้งหมดนั้นไปหารูปแบบการแทนข้อมูลด้วยโมเดลของเครื่องเข้ารหัสอัตโนมัติ แล้วนำรูปแบบการแทนข้อมูลที่ได้ไปจำแนกด้วยโมเดลของเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม แล้ววัดประสิทธิภาพของการจำแนกจากวิธีการที่นำเสนอโดยใช้มาตรวัดคือ Accuracy, Precision, Recall และ F-Measure.

3.3.2 ผังงาน (Flowchart) แสดงขั้นตอนการทำงานของวิธีการที่นำเสนอ

ลำดับขั้นตอนการทำงานในรายละเอียดทั้งหมดของวิธีการที่นำเสนอแสดงดังผังงานในรูปที่ 3-13 โดยเริ่มต้นจากการนำโมเดล ResNet101 ที่ผ่านการฝึกสอนมาก่อนเข้ามาในโปรแกรม Matlab Version R2018a จากการนำเข้า (Import) ด้วยเมนู Add-Ons ในโปรแกรม นั่นคือเป็นการ Download และ Install โมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนให้สามารถใช้งานร่วมกันกับส่วนอื่น ๆ ของโปรแกรม Matlab ได้ หลังจากที่นำเข้ามาได้แล้ว คำสั่งต่าง ๆ ที่มีในโปรแกรม Matlab ก็จะสามารถใช้ในการจัดการเพื่อนำรูปแบบของสถาปัตยกรรมและค่าพารามิเตอร์ต่าง ๆ จากโมเดล ResNet101 มาใช้งานต่อในรูปแบบของการเรียนรู้แบบถ่ายโอนได้



รูปที่ 3-13 ผังงาน (Flowchart) แสดงขั้นตอนการทำงานของวิธีการที่นำเสนอ

ดังนั้นด้วยวิธีการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัด ขั้นตอนต่อมาจึงเป็นการแปลงข้อมูลภาพให้อยู่ในรูปแบบของคุณลักษณะด้วยการใช้ค่าพารามิเตอร์ต่าง ๆ ที่ถ่ายโอนมาโดยแปลงจนถึงชั้น FC, 1000 ของโมเดล ResNet101 ตามรายละเอียดในรูปที่ 3-12 เพื่อนำคุณลักษณะที่ได้จากชั้นดังกล่าวไปผ่านขั้นตอนอื่น ๆ ต่อไป

ตัวอย่างข้อมูลที่ได้หลังจากการแปลงข้อมูลภาพให้อยู่ในรูปแบบของคุณลักษณะแสดงดังตารางที่ 3-1 และตารางที่ 3-2 เมื่อตารางที่ 3-1 คือข้อมูลในส่วนของ A ในรูปที่ 3-13 ซึ่งข้อมูลดังกล่าวถูกนำมาเก็บเป็น Train Set Feature Vectors ในรูป นั่นคือ Train Set Feature Vectors ที่แสดงในตารางที่ 3-1 เป็นข้อมูลของคุณลักษณะที่ได้มาจากการแปลงภาพในชุดฝึกสอน (Train Set) ของชุดข้อมูลที่ 2 ที่ใช้ในงานวิจัยนี้ (นั่นคือเป็นชุดข้อมูลที่มี 4 กลุ่มของวัสดุ ซึ่งชุดฝึกสอนมี 400 ภาพต่อหนึ่งชนิดของวัสดุ ภาพทั้งหมดในชุดฝึกสอนจึงมี 1600 ภาพ) จากตารางจะเห็นได้ว่าจำนวนข้อมูลทั้งหมดมีจำนวนมิติเป็น 1600×1000 อันเนื่องมาจากมีจำนวนภาพทั้งหมด 1600 ภาพ และจำนวนนิวรอนในชั้น FC, 1000 นั้นมีจำนวน 1000 นิวรอน นั้นคือภาพหนึ่งภาพจะถูกแปลงไปเป็นข้อมูลที่มีจำนวน 1000 องค์ประกอบ ดังค่าของข้อมูลตัวอย่างในตารางที่ 3-1 ซึ่งแสดงตัวอย่างเพียงบางส่วนของ 20 แถวแรก และ 10 คอลัมภ์แรก

ตารางที่ 3-2 คือข้อมูลในส่วนของ B ของรูปที่ 3-13 ซึ่งนำมาเก็บเป็น Test Set Feature Vectors เป็นข้อมูลของคุณลักษณะที่ได้มาจากการแปลงภาพในชุดทดสอบ (Test Set) ของชุดข้อมูลที่ 2 ที่ใช้ในงานวิจัยนี้ (นั่นคือเป็นชุดข้อมูลที่มี 4 กลุ่มของวัสดุ ที่ชุดทดสอบมี 200 ภาพต่อหนึ่งชนิดของวัสดุ ภาพทั้งหมดจึงมี 800 ภาพ) จากตารางจะเห็นได้ว่าจำนวนข้อมูลทั้งหมดมีจำนวนมิติเป็น 800×1000 อันเนื่องมาจากมีจำนวนภาพ 800 ภาพ และจำนวนนิวรอนในชั้น FC, 1000 นั้นมีจำนวน 1000 นิวรอน โดยค่าของข้อมูลตัวอย่างที่แสดงในตารางที่ 3-2 นั้นแสดงเพียงบางส่วนของ 20 แถวแรก และ 10 คอลัมภ์แรก เช่นเดียวกัน

ข้อมูลที่จัดเก็บเป็น Train Set Feature Vectors ในรูปที่ 3-13 นั้นจะนำไปใช้สำหรับการหาโมเดลที่เหมาะของเครื่องเข้ารหัสอัตโนมัติ (Autoencoder) และเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม (Multi-Class SVM) นั่นคือนำไปใช้สำหรับการฝึกสอน (Train) และการตรวจสอบความสมเหตุสมผล (Validate) สำหรับการหาค่าพารามิเตอร์ต่าง ๆ ที่เหมาะสมให้กับเครื่องเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุน ดังนั้น Train Set Feature Vectors ในรูปที่ 3-13 จึงถูกนำมาแบ่งเป็น 80% แบบสุ่มสำหรับใช้เป็นส่วนของ Train Vectors และอีก 20% สำหรับใช้เป็นส่วนของ Validate Vectors

ตารางที่ 3-1 ตัวอย่างข้อมูลที่ได้หลังจากการแปลงข้อมูลภาพในชุดฝึกสอนจากชุดข้อมูลที่ 2 ที่ศึกษาให้อยู่ในรูปแบบของคุณลักษณะจากชั้น FC, 1000

featuresTrain

1600x1000 single

	1	2	3	4	5	6	7	8	9	10
1	-4.6718	-2.5396	-3.6267	-5.6575	-5.3338	-1.1919	-3.4533	-0.7705	0.1016	-0.2912
2	-2.3654	-1.3609	-1.5086	-3.1686	-2.9223	0.7357	-2.7455	1.3134	3.0705	0.2241
3	-5.7942	-2.5527	-3.2668	-3.7691	-3.4453	-2.5437	-6.0699	1.7038	2.1179	-0.2964
4	-3.8222	-2.0022	-2.2299	-3.2082	-3.2701	-1.7519	-5.6041	0.3113	0.4323	-0.4081
5	-3.3515	-0.8790	-2.9719	-2.7901	-2.2570	4.1235	-1.1685	0.5807	0.8761	0.6275
6	-4.2183	-1.0677	-2.8280	-3.8014	-2.8547	0.2106	-2.0940	1.5635	1.9694	0.8955
7	-6.6546	-2.1150	-4.2649	-4.3143	-3.6154	-2.6601	-5.6967	2.1698	2.4600	-0.3064
8	-4.3366	-2.4854	-2.2893	-3.1619	-3.4367	-2.0860	-6.3082	0.5688	0.4686	-0.8582
9	-3.7928	-0.0470	-1.6360	-2.2496	-1.5840	2.1928	-0.7808	1.1306	0.2680	1.0667
10	-3.3998	-2.6588	-2.7338	-3.1907	-3.0744	1.1410	-3.3052	-0.3191	0.0416	0.8282
11	-5.3945	-1.9581	-3.0940	-3.5783	-3.6536	-1.6501	-4.3335	2.0417	2.9083	1.1827
12	-4.8421	-2.7160	-2.1512	-2.7909	-3.2305	-1.3764	-5.7903	0.5890	0.7092	-0.7355
13	-5.6045	-1.5737	-2.3690	-3.3571	-3.0428	-1.3596	-3.3693	2.0699	1.0468	2.9428
14	-5.0615	-2.1132	-1.4481	-4.2025	-3.4934	0.6577	-3.3721	1.2640	2.7611	0.4349
15	-5.4441	-2.2578	-2.6267	-3.6840	-3.4646	-1.1818	-4.5445	1.8628	3.0981	0.6592
16	-3.8333	-2.0145	-2.3100	-2.5457	-3.0747	-1.1200	-4.9069	0.5242	0.6285	-0.7505
17	-3.9420	-0.3347	-2.0147	-2.9945	-3.2496	1.2488	-1.1768	0.3030	-0.0038	2.2091
18	-3.5612	-1.5403	-2.5854	-3.5944	-2.4526	-0.4367	-3.8035	0.5165	1.8276	0.1030
19	-5.3134	-2.9151	-3.1238	-3.4063	-3.6532	-2.4274	-5.9477	1.7218	2.2177	-0.4110
20	-3.9649	-2.1882	-2.4279	-2.6079	-3.0403	-1.5885	-5.0709	0.7533	0.7878	-0.9045

ตารางที่ 3-2 ตัวอย่างข้อมูลที่ได้หลังจากการแปลงข้อมูลภาพในชุดทดสอบจากชุดข้อมูลที่ 2 ที่ศึกษาให้อยู่ในรูปแบบของคุณลักษณะจากชั้น FC, 1000

featuresTest

800x1000 single

	1	2	3	4	5	6	7	8	9	10
1	-5.2301	-0.3987	-2.8955	-3.7156	-1.6792	2.0015	-0.2784	-0.1721	0.1980	-0.2419
2	-5.8870	-1.9240	-3.0033	-4.8899	-3.6185	-1.3856	-3.8326	1.8708	4.1115	-0.0803
3	-5.8280	-2.5953	-1.1689	-3.2728	-2.5519	-2.0029	-5.6915	-0.6079	1.4634	0.8873
4	-4.3343	-2.1659	-2.7191	-2.7841	-2.5055	-1.1252	-4.3685	0.0124	1.0881	-0.5915
5	-5.8895	-0.9645	-2.7541	-3.6317	-2.2409	0.0983	-2.1991	2.6386	2.3525	1.9726
6	-1.6721	-2.2101	-2.1368	-2.8831	-3.1660	2.0199	-1.6879	-0.8487	0.8337	-1.7489
7	-4.6940	-2.3744	-3.2581	-3.3366	-2.8185	-0.9215	-5.6861	0.3485	1.5493	-1.0280
8	-4.8269	-2.3681	-2.0720	-2.8301	-2.5310	-2.3821	-6.2622	-0.5718	0.4186	-0.5916
9	-5.6866	-0.7174	-2.7380	-4.3251	-3.5292	-0.7783	-3.6426	1.8715	1.2062	1.6387
10	-6.3785	-2.0852	-3.2085	-5.0064	-4.3437	-1.3904	-4.2290	1.8105	2.6811	0.3699
11	-5.8157	-2.9582	-3.2528	-4.2322	-3.4287	-2.6936	-5.7688	1.0587	2.3899	-1.2065
12	-4.7824	-2.2380	-2.6984	-3.4577	-2.5994	-1.7672	-4.7100	0.6471	2.1794	-0.6199
13	-7.8220	-1.1981	-2.6655	-4.4657	-3.4580	-1.9818	-2.7718	3.5351	2.5625	2.7398
14	-6.6659	-2.4155	-2.8888	-5.3887	-3.8764	-2.9836	-5.0182	2.8673	3.2531	0.0412
15	-5.7520	-2.7687	-3.4936	-4.1193	-3.5113	-2.1369	-5.8433	1.2712	2.4679	-1.0003
16	-5.4489	-2.4002	-2.3731	-3.7606	-3.1185	-2.5977	-6.1124	-0.8695	0.3959	-0.8791
17	-4.9538	-1.7540	-1.1067	-2.3160	-1.3728	-1.7179	-1.9967	3.0695	1.5038	4.8131
18	-6.4229	-1.5902	-0.8981	-4.7090	-2.3441	-0.6793	-2.7754	2.9061	3.3187	0.7987
19	-5.7515	-3.2566	-3.0541	-4.2776	-3.2960	-2.4868	-5.5665	1.3760	2.2813	-0.8742
20	-5.0475	-1.9158	-3.0588	-3.8685	-2.2492	-0.8131	-4.4148	1.2840	2.4783	-0.6746

การหาค่าพารามิเตอร์ต่าง ๆ ที่เหมาะสมให้กับเครื่องเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุนนั้นแบ่งออกเป็นสองขั้นตอนหลักคือ ขั้นตอนการลองผิดลองถูก (Trial and Error) และขั้นตอนแบบสุ่ม (Random) โดยขั้นตอนการลองผิดลองถูกใช้สำหรับการกำหนดขอบเขตล่าง (Lower Bound) และขอบเขตบน (Upper Bound) ของพารามิเตอร์แต่ละตัว โดยผลลัพธ์จากขั้นตอนการลองผิดลองถูกคือข้อมูลส่วนของ C ในรูปที่ 3-13 ดังนั้นข้อมูลส่วนของ C ดังกล่าวจึงเป็นขอบเขตล่างและขอบเขตบนของพารามิเตอร์แต่ละตัวที่นำไปใช้สำหรับการสุ่มค่าพารามิเตอร์แต่ละตัวในขั้นตอนแบบสุ่มต่อไปเพื่อนำไปสู่การหาค่าพารามิเตอร์ที่เหมาะสมที่สุด (Optimal Parameters) สำหรับการใช้ในโมเดลของเครื่องเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุน ดังนั้นข้อมูลส่วนของ D ในรูปที่ 3-13 จะเป็นค่าพารามิเตอร์ที่เหมาะสมที่สุดที่ได้จากการใช้กลยุทธ์แบบสุ่ม แล้วค่าพารามิเตอร์ต่าง ๆ ที่เหมาะสมเหล่านั้นจึงนำไปใช้ในโมเดลเครื่องเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุนสำหรับการจำแนกข้อมูลในชุดทดสอบในขั้นตอนของการทดสอบ (Testing) ซึ่งเป็นขั้นตอนสุดท้ายของวิธีการที่นำเสนอต่อไป

ตารางที่ 3-3 แสดงตัวอย่างข้อมูลที่เกิดขึ้นในส่วน E ของรูปที่ 3-13 จากตารางจะเห็นได้ว่าจำนวนข้อมูลทั้งหมดมีจำนวนมิติเป็น 45×800 อันเนื่องมาจากการใช้จำนวนนิวรอน 45 นิวรอนในชั้นซ่อนเร้นของเครื่องเข้ารหัสอัตโนมัติ และมีจำนวนภาพ 800 ภาพจากภาพในชุดทดสอบของชุดข้อมูลที่ 2 ที่ศึกษา โดยค่าของข้อมูลตัวอย่างในตารางที่ 3-3 นั้นแสดงตัวอย่างเพียงบางส่วนของ 20 แถวแรก และ 10 คอลัมภ์แรก

ตารางที่ 3-3 ตัวอย่างข้อมูลในชุดทดสอบจากชุดข้อมูลที่ 2 ที่ศึกษาหลังผ่านการเข้ารหัสด้วยเครื่องเข้ารหัสอัตโนมัติ เมื่อใช้จำนวนนิวรอนในชั้นซ่อนเร้นเป็น 45 นิวรอน

features1T

45x800 double

	1	2	3	4	5	6	7	8	9	10
1	0.0410	0.2790	0.4889	0.4769	0.0796	0.1071	0.5777	0.4858	0.0648	0.1899
2	0.5388	0.1369	0.1060	0.0759	0.2573	0.1609	0.1182	0.0436	0.4025	0.2191
3	0.0462	0.0379	0.0248	0.0150	0.0556	0.0843	0.0202	0.0211	0.0399	0.0344
4	0.0461	0.0447	0.0530	0.1449	0.0502	0.0447	0.0593	0.1030	0.0489	0.0685
5	0.2637	0.0622	0.0600	0.0289	0.1176	0.0333	0.0392	0.0620	0.1623	0.0615
6	0.0916	0.0607	0.3603	0.0315	0.0887	0.4263	0.0430	0.1958	0.0409	0.0723
7	0.1275	0.5509	0.4468	0.1939	0.2678	0.2611	0.3583	0.2923	0.3315	0.6600
8	0.2319	0.1402	0.1331	0.2660	0.1947	0.2722	0.1211	0.1399	0.2546	0.1883
9	0.0477	0.3408	0.8426	0.7852	0.1538	0.4426	0.7994	0.9055	0.2081	0.2614
10	0.2287	0.5100	0.3177	0.1219	0.3321	0.3601	0.2426	0.0742	0.3341	0.6021
11	0.0356	0.1269	0.1009	0.1165	0.1253	0.1503	0.0806	0.1281	0.1366	0.0947
12	0.0537	0.3532	0.7565	0.8490	0.0965	0.0860	0.7838	0.9231	0.0870	0.3939
13	0.4729	0.2711	0.3214	0.3032	0.2023	0.4978	0.2185	0.3742	0.5343	0.2858
14	0.0920	0.0560	0.0686	0.1465	0.0446	0.2779	0.0870	0.1447	0.2771	0.0714
15	0.0373	0.1288	0.1377	0.2663	0.0660	0.1191	0.1170	0.1108	0.0683	0.0567
16	0.0739	0.1159	0.2751	0.1099	0.1652	0.1683	0.0868	0.1149	0.1733	0.0973
17	0.0472	0.1143	0.0766	0.0751	0.2229	0.0578	0.0804	0.0718	0.1366	0.2054
18	0.1118	0.1481	0.0344	0.0700	0.3193	0.1697	0.0800	0.0693	0.3370	0.1288
19	0.3050	0.1598	0.0593	0.1080	0.1792	0.2075	0.0665	0.0377	0.2118	0.1673
20	0.1950	0.1594	0.1309	0.2271	0.3145	0.1486	0.1532	0.0704	0.2594	0.1829

3.3.3 การกำหนดค่าพารามิเตอร์ต่าง ๆ ในวิธีการที่นำเสนอ

วิธีการที่นำเสนอขึ้นสำหรับงานวิจัยนี้มีการนำเทคนิคของโครงข่ายเครื่องเข้ารหัสอัตโนมัติและเทคนิคของเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มเข้ามาใช้เป็นส่วนหนึ่งของวิธีการที่นำเสนอ ด้วยรูปแบบการเรียนรู้ของสองเทคนิคดังกล่าว ในขั้นตอนการฝึกสอนจำเป็นต้องมีการกำหนดค่าพารามิเตอร์ต่าง ๆ ที่สำคัญสำหรับการค้นหาโครงแบบที่เหมาะสมของแต่ละเทคนิค เพื่อให้ได้เป็นโครงแบบที่ดีที่สุดสำหรับการนำไปใช้เป็นโมเดลสำหรับการสร้างตัวแทนข้อมูลและโมเดลสำหรับการจำแนกข้อมูลเพื่อใช้สำหรับการทดสอบต่อไป สำหรับงานวิจัยนี้ค่าพารามิเตอร์ต่าง ๆ ดังกล่าวค้นหามาจากการใช้สองกลยุทธ์ประกอบกันคือ การลองผิดลองถูก (Trial and Error) และกลยุทธ์แบบสุ่ม (Random) การลองผิดลองถูกนำมาใช้เพื่อหาขอบเขตที่เหมาะสมของพารามิเตอร์แบบกว้าง ๆ ส่วนกลยุทธ์แบบสุ่มนำมาใช้เพื่อค้นหาขอบเขตแบบจำกัดที่เหมาะสมหลังจากที่ได้ขอบเขตแบบกว้างจากการลองผิดลองถูกมาแล้ว ซึ่งรายละเอียดของขั้นตอนการหาค่าค่าพารามิเตอร์ที่เหมาะสมดังกล่าวแสดงดังส่วนหนึ่งของผังงานในรูป 3-13 เมื่อพารามิเตอร์ที่ได้จากส่วนของ C ในรูปคือพารามิเตอร์ที่ได้จากการลองผิดลองถูกและพารามิเตอร์ที่ได้จากส่วนของ D ในรูปคือพารามิเตอร์ที่ได้จากกลยุทธ์แบบสุ่ม

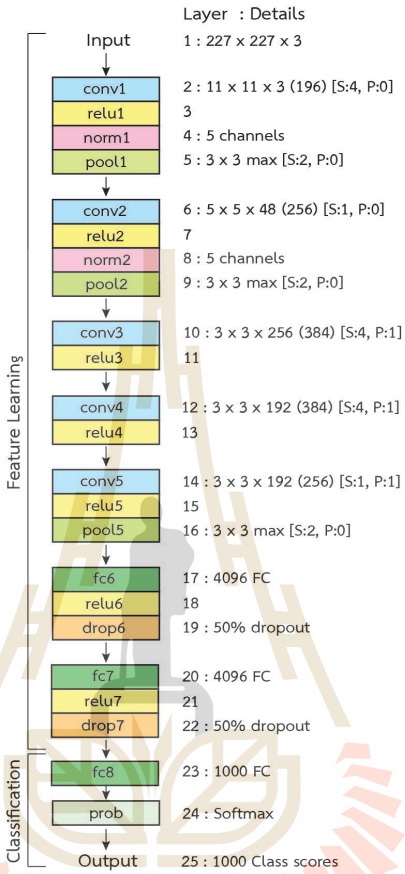
3.4 วิธีการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันในรูปแบบอื่น เพื่อการศึกษาเปรียบเทียบกับวิธีการที่นำเสนอ

นอกเหนือจากวิธีการที่นำเสนอแล้ว งานวิจัยนี้ได้ทำการศึกษาวิธีการในรูปแบบอื่น ๆ ในการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในการก่อสร้าง เพื่อเป็นการศึกษาเปรียบเทียบอีก 4 รูปแบบดังรายละเอียดในข้อ 3.4.1 ถึง 3.4.4

3.4.1 การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet

เป็นการนำโมเดลของ AlexNet ที่ผ่านการฝึกสอนมาก่อนด้วยชุดข้อมูลภาพของ ImageNet เช่นเดียวกันกับโมเดลของ ResNet101 มาศึกษาในรูปแบบการเรียนรู้แบบถ่ายโอน รูปที่ 3-6 แสดงรายละเอียดในแต่ละชั้นย่อยของสถาปัตยกรรม AlexNet ซึ่งเป็นสถาปัตยกรรมที่ชนะการแข่งขันในงาน ILSVRC 2012 และมีการเผยแพร่โมเดลแบบที่ผ่านการฝึกสอนมาก่อน สถาปัตยกรรมของ AlexNet ประกอบด้วยชั้นย่อย ๆ ในรายละเอียดทั้งหมด 25 ชั้น แต่โดยทั่วไปถือว่าเป็นโครงข่ายที่มีความลึกเป็น 8 ชั้น จากลักษณะของโครงข่ายที่ประกอบด้วยชั้นที่มีค่าของ weights และ biases เพียง 8 ชั้น ซึ่งคือชั้นเกี่ยวกับการคอนโวลูชัน (CONV) ที่มีจำนวน 5 ชั้น รวมกับชั้นที่เป็นแบบการเชื่อมถึงกันหมด (FC) ที่มีจำนวน 3 ชั้น ดังแสดงในรูปที่ 3-14 งานวิจัยนี้นำโมเดลแบบที่ผ่านการฝึกสอนมาก่อนของโมเดล AlexNet มาศึกษาในทั้งสองรูปแบบของการเรียนรู้

แบบถ่ายโอน คือแบบยึดคุณลักษณะจากตัวสกัดและแบบปรับแต่งการเรียนรู้ ดังรายละเอียดในหัวข้อ (1) และ (2) ตามลำดับ



รูปที่ 3-14 รายละเอียดสถาปัตยกรรมของโครงข่าย AlexNet

(1) การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet ในแบบยึดคุณลักษณะจากตัวสกัด

การประยุกต์ใช้โมเดลของ AlexNet สำหรับการจำแนกภาพวัสดุในงานก่อสร้างด้วยการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัดสามารถทำได้ในรูปแบบเดียวกันกับการใช้โมเดล ResNet101 ที่มีการกล่าวถึงไปแล้วในหัวข้อ 3.2 และ 3.3 แตกต่างกันเพียงเปลี่ยนขั้นตอนที่ 1 ของรูปที่ 3-10 เป็นการ download โมเดล AlexNet ที่มีการฝึกสอนมาก่อน (Download AlexNet Pre-Trained Model) ดังนั้นในขั้นตอนต่อมาข้อมูลภาพจากชุดข้อมูลที่ศึกษาจึงถูกแปลงเป็นแผนที่คุณลักษณะตามลักษณะของแต่ละชั้นย่อยในสถาปัตยกรรมของ AlexNet ที่แสดงในรูปที่ 3-14 โดยการใช้ค่าของ weights และ biases ที่ถ่ายโอนมาจากโมเดล การแปลงนี้จะทำไปจนถึงชั้น fc8 (ชั้น fully connect ที่ความลึกลำดับที่ 8) ของรูปที่ 3-14 ซึ่งเป็นชั้นที่ใช้จำนวนนิวรอนเป็น 1000 นิวรอน

ดังนั้นหลังจากส่งข้อมูลภาพอินพุตผ่านโมเดล 1 รอบ ภาพแต่ละภาพจะถูกแปลงเป็นเวกเตอร์คุณลักษณะที่มีมิติเท่ากับ 1000

ในส่วนรายละเอียดของวิธีการฝึกสอนและการทดสอบสำหรับการเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet ในแบบยึดคุณลักษณะจากตัวสกัดนั้นก็ทำในรูปแบบเดียวกันกับเมื่อใช้โมเดล ResNet101 นั่นคือตามรายละเอียดในรูปที่ 3-13

(2) การเรียนรู้แบบถ่ายโอนจาก AlexNet ในแบบปรับแต่งการเรียนรู้

อีกรูปแบบหนึ่งของการนำโมเดลที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้ของการเรียนรู้แบบถ่ายโอนคือการนำโมเดลนั้นมาผ่านกระบวนการปรับแต่งการเรียนรู้ หรือ Fine-Tune Learning ที่สามารถทำได้จากการฝึกสอนให้กับค่าของ Weights และ Biases ในบางชั้นของโครงข่ายที่เคยผ่านการฝึกสอนมาก่อนแล้วนั้นใหม่ด้วยข้อมูลชุดฝึกสอนต้องการศึกษา ซึ่งในทางปฏิบัติโดยทั่วไปมักจะยึดค่าของ Weights และ Biases ในชั้นแรก ๆ ของโครงข่ายไว้ตามค่าเดิมของสถาปัตยกรรมนั้น ๆ (มักจะเรียกว่าการแช่แข็ง (Freeze) ค่าพารามิเตอร์เหล่านั้นไว้) และให้มีการปรับแต่ง (Fine-Tune) หรือฝึกสอนใหม่เฉพาะกับค่าในบางชั้นที่อยู่ท้าย ๆ ของโครงข่าย เช่น 10 ชั้นสุดท้าย หรือ 3 ชั้นสุดท้าย ขึ้นอยู่กับว่าสถาปัตยกรรมที่เลือกใช้นั้นมีความลึกเป็นอย่างไร สำหรับการที่จะดูว่าควรฝึกสอนใหม่เพื่อปรับแต่งจำนวนกี่ชั้นนั้นขึ้นอยู่กับแต่ละงานประยุกต์ด้วย ซึ่งจำเป็นต้องผ่านการทดลองตามความเหมาะสมของแต่ละงาน นอกจากนั้นพารามิเตอร์อื่น ๆ ที่ใช้ในโครงข่าย เช่นค่าคงที่การเรียนรู้ จำนวน Epoch ขนาดของ Mini-Batch หรือค่าอื่น ๆ ที่ใช้ในโครงข่ายก็สามารถเลือกปรับได้ตามความเหมาะสมของแต่ละงานประยุกต์เช่นเดียวกัน

สำหรับโมเดลของ AlexNet ที่ผ่านการฝึกสอนมาก่อนแล้วซึ่งมีรายละเอียดสถาปัตยกรรมของโครงข่ายดังรูปที่ 3-14 นั้นเมื่อต้องการนำมาใช้สำหรับการเรียนรู้แบบถ่ายโอนแบบปรับแต่งการเรียนรู้เพื่อจำแนกภาพวัสดุในงานก่อสร้างสามารถทำได้ตามลำดับขั้นตอนต่อไปนี้

1) เปลี่ยนจำนวนนิวรอนของชั้น output ในรูปที่ 3-14 ให้มีจำนวนเท่ากับจำนวนกลุ่มของข้อมูลที่ต้องการคัดแยก (เช่นกำหนดให้เป็น 3 เมื่อต้องการคัดแยกวัสดุในงานก่อสร้างออกเป็น 3 กลุ่มของวัสดุ) ดังนั้นจากจำนวนนิวรอนของชั้น output เดิม 1000 นิวรอนของโครงข่ายรูปที่ 3-14 จึงเปลี่ยนใหม่ให้มีค่าเป็น 3
2) กำหนดช่วงของชั้นย่อยที่ต้องการให้ค่าของ Weights และ Biases มีค่าเหมือนเดิม (ชั้นที่จะ Freeze ค่าของ Weights และ Biases) เช่นชั้นที่ 1-16 ของรูปที่ 3-14

3) กำหนดค่าพารามิเตอร์อื่น ๆ ที่จะใช้สำหรับการฝึกสอนให้กับโครงข่าย เช่น ค่าคงที่การเรียนรู้ จำนวน epoch ขนาดของ mini-batch หรือค่าอื่น ๆ

4) ฝึกสอนโครงข่ายด้วยข้อมูลชุดฝึกสอนที่ต้องการศึกษา (สำหรับงานวิจัยนี้คือข้อมูลชุดฝึกสอนของภาพวัสดุในงานก่อสร้าง) นั่นคือสามารถกล่าวได้ว่าขั้นตอนนี้คือขั้นตอนของการปรับแต่ง

5) ทดสอบประสิทธิภาพของโครงข่ายที่ผ่านการฝึกสอนในข้อ 4 ด้วยข้อมูลชุดทดสอบ

จากขั้นตอนที่กล่าวข้างต้น ขั้นตอนการฝึกสอนนั้นจะทำการปรับแต่งค่าของ Weights และ Biases ในชั้นอื่น ๆ ที่ไม่ใช่ชั้นที่ 1-16 (นั่นคือชั้นที่ 17-25 ของรูปที่ 3-14) ส่วนในชั้นที่ 1-16 นั้นค่าของ Weights Biases จะไม่มีการเปลี่ยนแปลงถึงแม้จะมีการฝึกสอนใหม่ในโครงข่าย

3.4.2 การเรียนรู้แบบถ่ายโอนจากโมเดล GoogleNet

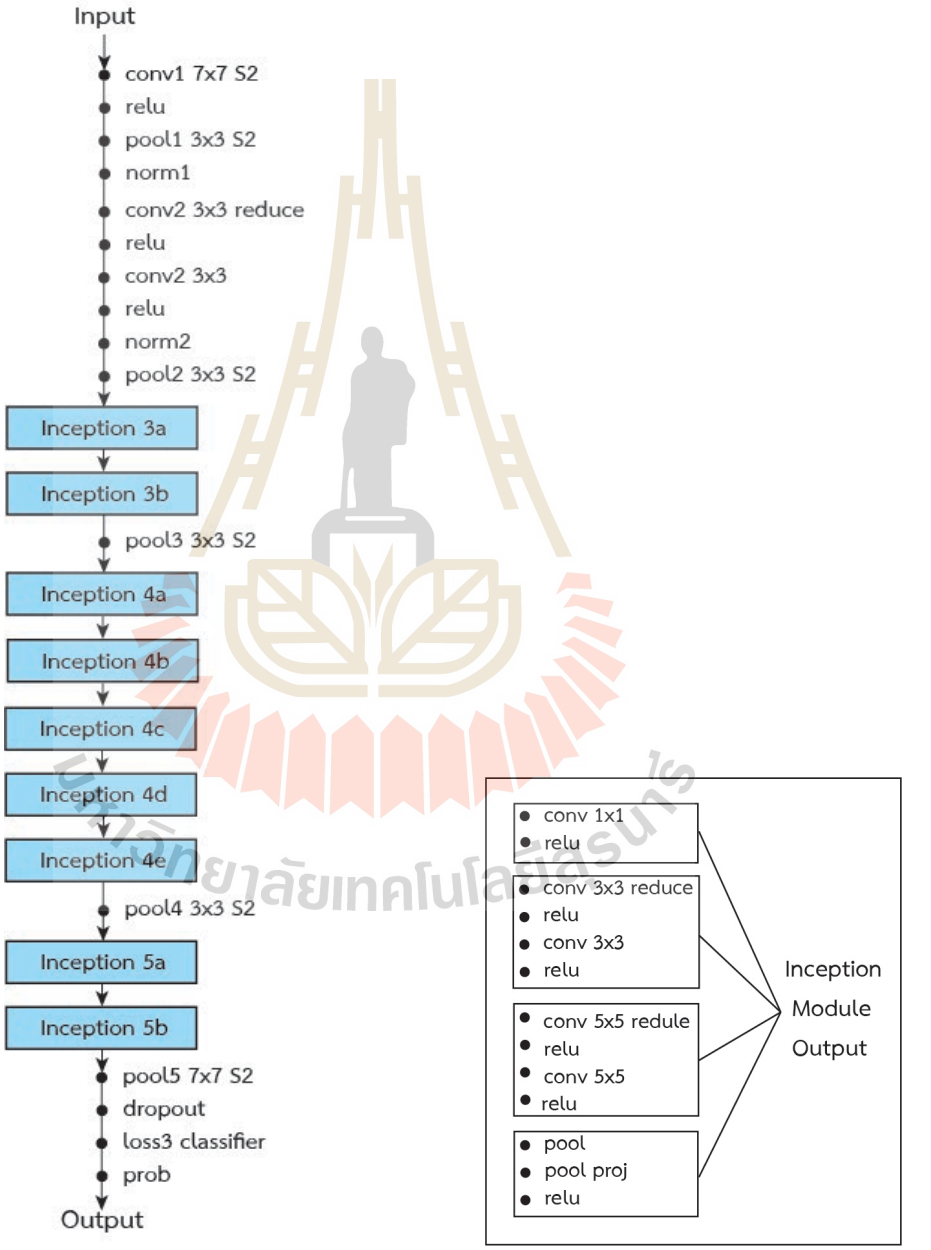
เป็นการนำโมเดลของ GoogleNet ที่ผ่านการฝึกสอนมาก่อนด้วยชุดข้อมูลภาพของ ImageNet เช่นเดียวกันกับโมเดลของ ResNet101 มาศึกษาในรูปแบบการเรียนรู้แบบถ่ายโอน รูปที่ 3-15 แสดงรายละเอียดในแต่ละชั้นย่อยของสถาปัตยกรรม GoogleNet ซึ่งเป็นสถาปัตยกรรมที่ชนะการแข่งขันในงาน ILSVRC 2014 และมีการเผยแพร่โมเดลแบบที่ผ่านการฝึกสอนมาก่อน สถาปัตยกรรมของ GoogleNet ประกอบด้วยชั้นย่อย ๆ ในรายละเอียดทั้งหมด 144 ชั้น (เมื่อนับชั้นย่อย ๆ ในแต่ละ inception module ของรูปที่ 3-15(b) ด้วย) แต่โดยทั่วไปถือว่าเป็นโครงข่ายที่มีความลึกเป็น 22 ชั้น จากลักษณะของโครงข่ายที่ประกอบด้วยชั้นที่มีค่าของ Weights และ Biases เพียง 22 ชั้น งานวิจัยนี้นำโมเดลแบบที่ผ่านการฝึกสอนมาก่อนของโมเดล GoogleNet มาศึกษาในทั้งสองรูปแบบของการเรียนรู้แบบถ่ายโอน คือแบบยึดคุณลักษณะจากตัวสกัดและแบบปรับแต่งการเรียนรู้ ดังรายละเอียดในหัวข้อ (1) และ (2) ตามลำดับ

(1) การเรียนรู้แบบถ่ายโอนจาก GoogleNet ในแบบยึดคุณลักษณะจากตัวสกัด

การประยุกต์ใช้โมเดลของ GoogleNet สำหรับการจำแนกภาพวัสดุในงานก่อสร้างด้วยการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัดสามารถทำได้ในรูปแบบเดียวกันกับการใช้โมเดล ResNet101 ที่มีการกล่าวถึงไปแล้วในหัวข้อ 3.2 และ 3.3 แตกต่างกันเพียงเปลี่ยนขั้นตอนที่ 1 ของรูปที่ 3-10 เป็นการ download โมเดล GoogleNet ที่มีการฝึกสอนมาก่อน (Download GoogleNet Pre-Trained Model) ดังนั้นในขั้นตอนต่อมาข้อมูลภาพจากชุดข้อมูลที่ศึกษาจึงถูกแปลงเป็นแผนที่คุณลักษณะตามลักษณะของแต่ละชั้นย่อยในสถาปัตยกรรมของ GoogleNet ที่แสดงในรูปที่ 3-15 โดยการใช้ค่าของ Weights และ Biases ที่ถ่ายโอนมาจากโมเดล การแปลงนี้จะทำไปจนถึงชั้น loss3 classifier ของรูปที่ 3-15 (ชั้นสุดท้ายก่อนถึงขั้นตอนการจำแนก) ซึ่งเป็นชั้นที่

ใช้จำนวนนิวรอนเป็น 1000 นิวรอน ดังนั้นหลังจากส่งข้อมูลภาพอินพุตผ่านโมเดล 1 รอบ ภาพแต่ละภาพจะถูกแปลงเป็นเวกเตอร์คุณลักษณะที่มีมิติเท่ากับ 1000

ในส่วนรายละเอียดของวิธีการฝึกสอนและการทดสอบสำหรับการเรียนรู้แบบถ่ายโอนจากโมเดล GoogleNet ในแบบยึดคุณลักษณะจากตัวสกัดนั้นก็ทำในรูปแบบเดียวกันกับเมื่อใช้โมเดล ResNet101 นั่นคือตามรายละเอียดในรูปที่ 3-13



(a) โครงข่าย GoogleNet (b) รายละเอียดในแต่ละ Inception Module

รูปที่ 3-15 รายละเอียดสถาปัตยกรรมของโครงข่าย GoogleNet

(2) การเรียนรู้แบบถ่ายโอนจาก GoogleNet ในแบบปรับแต่งการเรียนรู้

วิธีการการประยุกต์ใช้โมเดลของ GoogleNet สำหรับการจำแนกภาพวัสดุในงานก่อสร้างด้วยการเรียนรู้แบบถ่ายโอนในแบบปรับแต่งการเรียนรู้นั้นมีขั้นตอนในรายละเอียดเช่นเดียวกันกับการเรียนรู้แบบถ่ายโอนจาก AlexNet ในแบบปรับแต่งการเรียนรู้ที่ได้อธิบายไปแล้วในหัวข้อ 3.4.2 แตกต่างกันเพียงเป็นการนำโมเดล GoogleNet ที่ผ่านการฝึกสอนมาก่อนมาใช้แทน AlexNet

3.4.3 การนำเทคนิคการเข้ารหัสข้อมูลด้วยวิธีการของ PCA มาใช้แทนเครื่องเข้ารหัสอัตโนมัติ

เป็นการศึกษาเปรียบเทียบเพื่อต้องการเปรียบเทียบผลลัพธ์จากการจำแนกในกรณีที่มีการนำเทคนิคการเข้ารหัสข้อมูลด้วยวิธีการที่แตกต่างจากวิธีการที่นำเสนอในงานวิจัยนี้ ดังนั้นหัวข้อนี้จึงเป็นการศึกษาเปรียบเทียบเมื่อนำเทคนิคการเข้ารหัสข้อมูลด้วยวิธี PCA มาใช้แทนโครงข่ายเครื่องเข้ารหัสอัตโนมัติในวิธีการที่นำเสนอ ดังนั้นขั้นตอนต่าง ๆ ในรายละเอียดการฝึกสอนและการทดสอบจึงเหมือนกันกับที่แสดงในรูปที่ 3-13 แต่เปลี่ยนจาก Autoencoder Network ในรูปดังกล่าวเป็น PCA ซึ่งนอกเหนือจากการศึกษาเปรียบเทียบที่นำ PCA มาใช้ร่วมกับโมเดล ResNet101 แล้ว ในงานวิจัยนี้จะทำการศึกษาเปรียบเทียบเพิ่มเติมในกรณีที่นำ PCA มาใช้ร่วมกันกับโมเดล GoogleNet ด้วย

3.4.4 การสร้างสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันสำหรับการฝึกสอนด้วยชุดข้อมูลที่ศึกษาขึ้นมาเอง (Training from Scratch)

เป็นการสร้างรูปแบบของชั้นต่าง ๆ หรือสถาปัตยกรรมที่เหมาะสมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเองเพื่อใช้สำหรับการฝึกสอนโดยตรงกับชุดข้อมูลที่ศึกษาในงานวิจัยนี้ ซึ่งคือข้อมูลภาพวัสดุในงานก่อสร้าง ตัวอย่างของรูปแบบของสถาปัตยกรรมที่เหมาะสม เช่น การที่จะพิจารณาว่าจะมีจำนวนชั้นการคอนโวลูชันกี่ชั้นในโครงข่าย ขนาดของตัวกรองและจำนวนตัวกรองที่ใช้ในแต่ละชั้นเป็นเท่าไร ใช้วิธีการทำพูลลิ่งแบบใด ลำดับการเรียงของชั้นต่าง ๆ เป็นอย่างไร เป็นต้น ซึ่งจำเป็นต้องมีการพิจารณาเพื่อหาสถาปัตยกรรมที่เหมาะสมดังกล่าวสำหรับนำมาใช้เพื่อฝึกสอนโดยตรงกับชุดข้อมูลที่ศึกษา แล้วนำโมเดลที่ได้จากการฝึกสอนนั้นไปผ่านขั้นตอนการทดสอบด้วยการจำแนกภาพในข้อมูลชุดทดสอบเพื่อวัดประสิทธิภาพของโมเดล

ตารางที่ 3-4 ยกตัวอย่างรายละเอียดในแต่ละชั้นย่อยของสถาปัตยกรรมที่สร้างขึ้นมาเองรูปแบบหนึ่งที่ประกอบด้วยชั้นย่อย ๆ ทั้งหมด 8 ชั้น เมื่อชั้นที่ 1 คือชั้น Input ที่นำภาพสีแต่ละภาพเข้ามาด้วยการใช้ค่าจาก 3 แกนสีของโมเดลสีแบบ RGB ชั้นที่ 2 คือชั้น Conv(9,30) ที่เป็นการคอนโวลูชันด้วยตัวกรองขนาด 9×9 จำนวน 30 ตัวกรอง ด้วยขนาดการ Stride เป็น 1 ชั้นที่ 3 คือชั้น

ReLU เป็นการนำค่าที่ได้จากชั้นที่ 2 มาผ่านการแปลงค่าด้วยฟังก์ชัน ReLU ชั้นที่ 4 คือชั้น Conv(5,20) ที่เป็นการคอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 20 ตัวกรอง ด้วยขนาดการ Stride เป็น 1 ชั้นที่ 5 คือชั้น ReLU เป็นการนำค่าที่ได้จากชั้นที่ 4 มาผ่านการแปลงค่าด้วยฟังก์ชัน ReLU ชั้นที่ 6 คือ MaxPool(2,2) ที่เป็นการทำพูลลิ่งแบบแมกซ์ในรูปแบบ 2×2 ด้วยขนาดการ Stride เป็น 2 ชั้นที่ 7 คือชั้น FC(3) เป็นชั้นที่นำแต่ละค่าจากชั้นก่อนหน้ามาเรียงต่อกันเป็นข้อมูลแบบเวกเตอร์แล้วใช้โครงข่ายแบบเชื่อมถึงกันหมด โดยกำหนดให้เอาต์พุตจากชั้นนี้มีขนาดเป็น 3 ตามจำนวนกลุ่มของข้อมูลที่ต้องการจำแนก และชั้นสุดท้ายคือชั้น SoftMax เป็นการนำค่าจากชั้นก่อนหน้ามาแปลงค่าด้วยฟังก์ชัน SoftMax เพื่อให้ได้ออกมาเป็นผลลัพธ์สุดท้ายจากโครงข่าย

หลังจากกำหนดรายละเอียดภายในโครงข่ายของสถาปัตยกรรมที่สร้างขึ้นมาเองเรียบร้อยแล้ว จึงนำโครงข่ายดังกล่าวไปใช้ฝึกสอนกับชุดข้อมูลที่ศึกษาโดยตรง นั่นคือนำข้อมูลชุดฝึกสอนมาใช้สำหรับการฝึกสอนโครงข่ายและนำข้อมูลชุดทดสอบมาใช้เพื่อการทดสอบหรือประเมินโครงข่ายว่ามีประสิทธิภาพเพียงใด

ตารางที่ 3-4 ตัวอย่างรายละเอียดในแต่ละชั้นย่อยของสถาปัตยกรรม CNN รูปแบบหนึ่งที่สร้างขึ้นมาเพื่อฝึกสอนโดยตรงกับข้อมูลชุดฝึกสอน

ชั้นที่	ชื่อชั้น	รายละเอียด
1	Input	ค่าจาก 3 แกนสีของโมเดลสี RGB
2	Conv(9,30)	คอนโวลูชันด้วยตัวกรองขนาด 9×9 จำนวน 30 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
3	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
4	Conv(5,20)	คอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 20 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
5	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
6	MaxPool(2,2)	ทำพูลลิ่งแบบ 2×2 ด้วยขนาดการ Stride เป็น 2
7	FC(3)	ชั้นการเชื่อมถึงกันหมดที่มี 3 Neurons
8	SoftMax	ชั้นการแปลงค่าด้วยฟังก์ชัน SoftMax

3.4.5 การกำหนดค่าพารามิเตอร์ต่าง ๆ ในวิธีการศึกษาเปรียบเทียบ

สำหรับวิธีการเพื่อการศึกษาเปรียบเทียบที่ศึกษาในงานวิจัยนี้ การกำหนดและการค้นหาค่าพารามิเตอร์สำคัญในวิธีการเหล่านั้น งานวิจัยนี้ค้นหามาจากการใช้สองกลยุทธ์ประกอบ

กันเช่นเดียวกันกับที่ใช้สำหรับวิธีการที่นำเสนอขึ้นในงานวิจัยนี้ นั่นคือใช้ การลองผิดลองถูก (Trial and Error) และกลยุทธ์แบบสุ่ม (Random) การลองผิดลองถูกนำมาใช้เพื่อหาขอบเขตที่เหมาะสมของพารามิเตอร์แบบกว้าง ๆ ส่วนกลยุทธ์แบบสุ่มนำมาใช้เพื่อค้นหาขอบเขตแบบจำกัดที่เหมาะสมหลังจากที่ได้ขอบเขตแบบกว้างจากการลองผิดลองถูกมาแล้ว

3.5 การเปรียบเทียบประสิทธิภาพของการจำแนก

ในการเปรียบเทียบประสิทธิภาพของการจำแนกระหว่างวิธีการสำหรับการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในงานก่อสร้างที่นำเสนอขึ้นสำหรับงานวิจัยนี้กับวิธีการในรูปแบบอื่น ๆ ที่ศึกษาเพื่อเป็นการเปรียบเทียบกันนั้น งานวิจัยนี้ทำการเปรียบเทียบโดยการประเมินประสิทธิภาพของโมเดลจากมาตรวัดต่าง ๆ คือ Accuracy, Precision, Recall และ F-Measure โดย F-Measure นั้นใช้เป็น F_1-Score

บทที่ 4
ผลการศึกษาและการวิเคราะห์ผล

วัตถุประสงค์หลักของงานวิจัยนี้เป็นการศึกษาเพื่อนำเทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยวิธีการที่นำเสนอนั้นเป็นการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนคือโมเดล ResNet101 มาประยุกต์ใช้ เพื่อนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสอัตโนมัติ และนำเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มมาใช้ในการจำแนก นอกเหนือจากวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการศึกษาวิธีการในรูปแบบอื่น ๆ สำหรับการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันเพื่อเป็นการศึกษาเปรียบเทียบด้วย จากผลการศึกษาทั้งหมด สามารถสรุปผลการศึกษาและวิเคราะห์ผลการศึกษาที่ได้ ดังต่อไปนี้

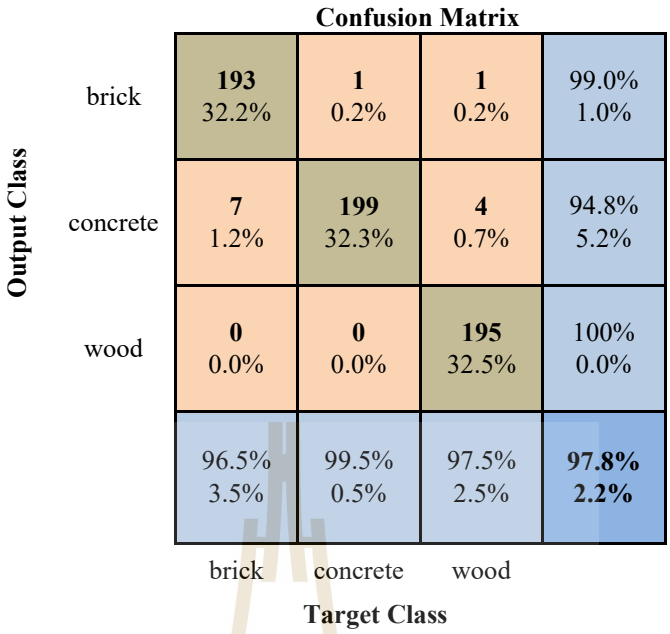
4.1 ผลการศึกษาจากวิธีการที่นำเสนอ

วิธีการที่นำเสนอในงานวิจัยนี้ คือการนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสแบบอัตโนมัติ เพื่อสร้างรูปแบบการแทนข้อมูลที่เหมาะสม และนำเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มมาใช้สำหรับการจำแนก ซึ่งวิธีการที่นำเสนอนี้ได้ทำการทดลองกับภาพวัสดุในงานก่อสร้างทั้งในชุดข้อมูลที่ 1 และชุดข้อมูลที่ 2

4.1.1 ผลการทดลองจากวิธีการที่นำเสนอกับชุดข้อมูลที่ 1

ชุดข้อมูลที่ 1 นั้นประกอบด้วยข้อมูล 3 กลุ่มของวัสดุในงานก่อสร้างคือ อิฐ คอนกรีต และไม้ ซึ่งภาพในชุดทดสอบมีจำนวน 200 ภาพในแต่ละกลุ่มของวัสดุ นั่นคือภาพในชุดทดสอบนี้มีทั้งหมด 600 ภาพ

รูปที่ 4-1 แสดงผลลัพธ์ที่ได้จากการจำแนกภาพในชุดทดสอบจากชุดข้อมูลที่ 1 โดยแสดงผลลัพธ์ในรูปแบบของ Confusion Matrix เมื่อแนวคอลัมน์เป็นเอาต์พุตเป้าหมายที่ต้องการ (ในรูปคือ Target Class) และแนวแถวเป็นเอาต์พุตที่ได้จากวิธีการที่นำเสนอ (ในรูปคือ Output Class) จากผลลัพธ์ที่ได้จะเห็นว่ามีค่า Accuracy เท่ากับ 0.978 หรือคิดเป็น 97.8%



รูปที่ 4-1 Confusion Matrix แสดงผลลัพธ์ที่ได้จากวิธีการที่นำเสนอสำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

จาก Confusion Matrix ในรูปที่ 4-1 สามารถสรุปเป็นค่าในมาตรวัดต่าง ๆ ดังแสดงในตารางที่ 4-1 โดยค่าของมาตรวัดนั้นแสดงด้วยค่าที่เป็นเปอร์เซ็นต์ จะเห็นว่าวิธีการที่นำเสนอสามารถจำแนกกลุ่มของวัสดุที่เป็น Concrete ได้ดีที่สุด นั่นคือมีค่า Recall ของกลุ่ม Concrete เป็น 99.5% ในขณะที่มีความแม่นยำในการจำแนกวัสดุในกลุ่ม Wood สูงมากถึง 100% (นั่นคือค่า Precision ของกลุ่ม Wood เป็น 100%

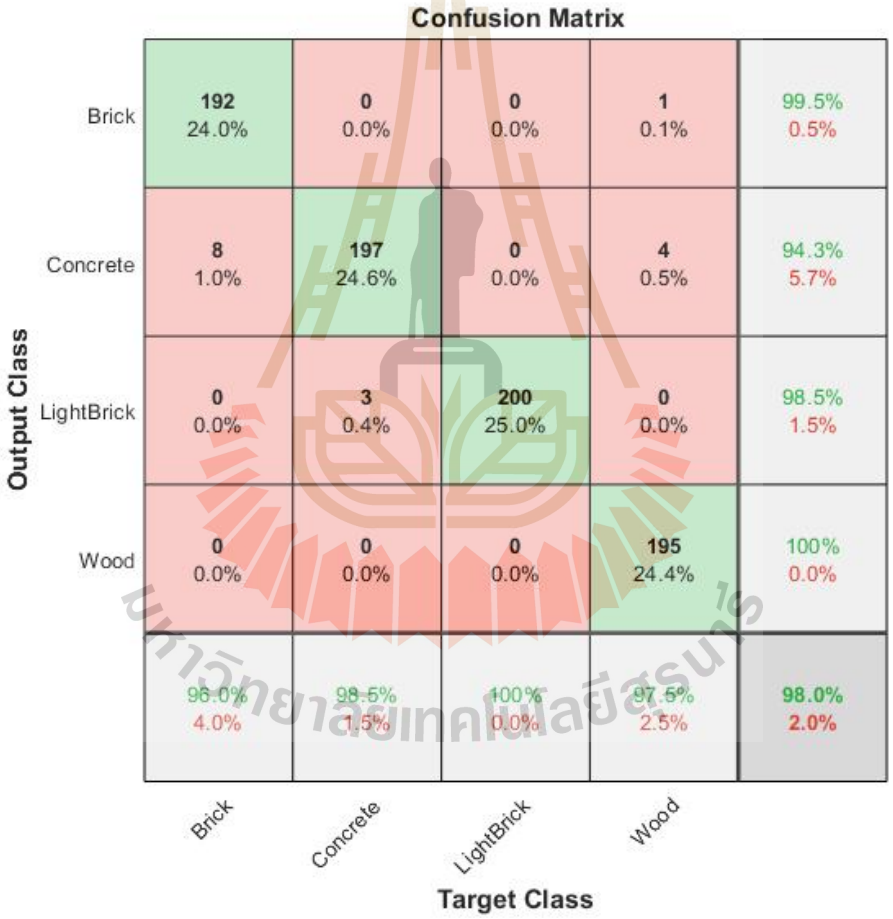
ตารางที่ 4-1 สรุปผลลัพธ์ที่ได้จากวิธีการที่นำเสนอสำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1 โดยใช้ค่าจากมาตรวัดต่าง ๆ

Class	Accuracy (%)	Recall (%)	Precision (%)	F-Measure (%)
brick	***97.8***	96.5	99.0	97.7
concrete		99.5	94.8	97.1
wood		97.5	100.0	98.7
	Average	*97.8*	*97.9*	*97.8*

4.1.2 ผลการทดลองจากวิธีการที่นำเสนอกับชุดข้อมูลที่ 2

ชุดข้อมูลที่ 2 ประกอบด้วยข้อมูล 4 กลุ่มของวัสดุในงานก่อสร้างคือ อิฐ คอนกรีต อิฐมวลเบาคอนกรีตเซลลูล่า และไม้ ซึ่งในชุดทดสอบประกอบด้วยภาพจำนวน 200 ภาพในแต่ละกลุ่มของวัสดุ ที่มีจำนวน 4 กลุ่ม นั่นคือภาพในชุดทดสอบนี้มีทั้งหมด 800 ภาพ

รูปที่ 4-2 แสดงผลลัพธ์ที่ได้จากการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2 โดยรูปที่ 4-2 แสดงผลลัพธ์ในรูปแบบของ Confusion Matrix เมื่อแนวคอลัมน์เป็นเอาต์พุตเป้าหมายที่ต้องการ และแนวแถวเป็นเอาต์พุตที่ได้จากวิธีการที่นำเสนอ จากผลลัพธ์ที่ได้จะเห็นว่ามีค่า Accuracy เท่ากับ 0.980 หรือคิดเป็น 98.0%



รูปที่ 4-2 Confusion Matrix แสดงผลลัพธ์ที่ได้จากวิธีการที่นำเสนอสำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2

จาก Confusion Matrix ในรูปที่ 4-2 สามารถสรุปเป็นค่าในมาตรวัดต่าง ๆ ดังแสดงในตารางที่ 4-2 ซึ่งพบว่าวิธีการที่นำเสนอสามารถจำแนกภาพวัสดุในกลุ่ม Light Brick (อิฐมวลเบาคอนกรีตเซลลูล่า) ด้วยความถูกต้องเป็น 100% นั่นคือค่า Recall ของกลุ่ม Light Brick เป็น 100% และผลลัพธ์ที่ได้มีความแม่นยำสูงมากในการจำแนกวัสดุในกลุ่มของ Brick จากค่า Precision ของกลุ่ม Brick คือ 99.5% ทั้งนี้ โดยรวมแล้ววิธีการที่นำเสนอมีประสิทธิภาพที่สูงสำหรับการจำแนกข้อมูลในทุกกลุ่ม โดยมีค่า Accuracy ที่สูงถึง 98.0% นั่นคือจากภาพในชุดทดสอบทั้งหมด 800 ภาพ วิธีการที่นำเสนอจำแนกผิดเพียง 16 ภาพ ซึ่งกลุ่มที่ถูกจำแนกผิดมากที่สุดคือกลุ่มของ Brick เมื่อทั้ง 8 ภาพของ Brick ที่จำแนกผิดนั้นถูกจำแนกว่าเป็น Concrete ทั้งหมด ดังแสดงใน Confusion Matrix ของรูปที่ 4-2 นอกจากนี้ภาพในกลุ่มของ Wood โดยส่วนใหญ่ที่จำแนกผิดก็ถูกจำแนกว่าเป็น Concrete เช่นกัน ดั้งนั้นจะเห็นว่าค่าความแม่นยำหรือค่า Precision ของกลุ่ม Concrete จึงค่อนข้างน้อยเมื่อเทียบกับค่า Precision ของกลุ่มอื่น ๆ

ตารางที่ 4-2 สรุปผลลัพธ์ที่ได้จากวิธีการที่นำเสนอสำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2 โดยใช้ค่าจากมาตรวัดต่าง ๆ

Class	Accuracy (%)	Recall (%)	Precision (%)	F-Measure (%)
Brick	***98.0***	96.0	99.5	97.7
Concrete		98.5	94.3	96.4
Light Brick		100.0	98.5	99.2
Wood		97.5	100	98.7
	Average	*98.0*	*98.1*	*98.0*

4.1.3 ค่าพารามิเตอร์สำคัญที่ใช้ในวิธีการที่นำเสนอ

จากรายละเอียดการฝึกสอน (Train) และการทดสอบ (Test) ของวิธีการที่นำเสนอ ที่ได้อธิบายในรูปที่ 3-5 ไปแล้วนั้น จะเห็นว่าการนำโครงข่ายเข้ารหัสอัตโนมัติมาใช้เพื่อเข้ารหัสให้กับคุณลักษณะที่สกัดออกมาจากโครงข่าย ResNet101 จะต้องมีการฝึกสอนให้กับโครงข่ายเข้ารหัสอัตโนมัติเพื่อหาโมเดลที่เหมาะสมที่สุด ซึ่งในขั้นตอนการฝึกสอนนั้นเป็นการสร้างโครงแบบ (Configuration) ในรูปแบบต่าง ๆ จากการกำหนดค่าพารามิเตอร์สำคัญที่แตกต่างกันเพื่อหาโครงแบบที่ดีที่สุดสำหรับใช้เป็นโมเดลที่เหมาะสมที่สุดที่จะนำไปใช้กับข้อมูลในชุดทดสอบต่อไป ในงานวิจัยนี้ การกำหนดค่าพารามิเตอร์ต่าง ๆ ดังกล่าวค้นหามาจากการใช้สองกลยุทธ์ประกอบกัน คือ การลองผิดลองถูก (Trial and Error) และกลยุทธ์แบบสุ่ม (Random) การลองผิดลองถูกนำมาใช้

เพื่อหาขอบเขตที่เหมาะสมของพารามิเตอร์แบบกว้างๆ ส่วนกลยุทธ์แบบสุ่มนำมาใช้เพื่อค้นหาขอบเขตแบบจำกัดที่เหมาะสมหลังจากที่ได้ขอบเขตแบบกว้างจากการลองผิดลองถูกมาแล้ว ซึ่งค่าพารามิเตอร์สำคัญที่ผ่านขั้นตอนการค้นหามาแล้วนั้นถูกนำมาใช้ในโมเดลของเครื่องเข้ารหัสอัตโนมัติที่เหมาะสมที่สุดของแต่ละชุดข้อมูล โดยค่าพารามิเตอร์สำคัญที่ค้นหาได้ของแต่ละชุดข้อมูลแสดงดังตารางที่ 4-3

สำหรับค่าพารามิเตอร์สำคัญของเครื่องเวกเตอร์เกื้อหนุนนั้น การทดลองในงานวิจัยนี้ทั้งหมดนำหลักการหาค่าแบบวิธีการ Optimization ที่เป็นค่า Default ของฟังก์ชันสำหรับเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มที่มีให้ใช้งานในโปรแกรม Matlab

ตารางที่ 4-3 ค่าพารามิเตอร์สำคัญที่ใช้ในโมเดลของเครื่องเข้ารหัสอัตโนมัติที่เหมาะสมที่สุดของแต่ละชุดข้อมูล

	No. Hidden Layer	**No. Hidden Neuron**	**No. Epoch**	**Learning Rate**
ชุดข้อมูลที่ 1	1	40	380	0.0001
ชุดข้อมูลที่ 2	1	45	400	0.0001

4.2 ผลการศึกษาจากวิธีการในรูปแบบอื่น ๆ เพื่อการศึกษาเปรียบเทียบ

หัวข้อนี้เป็นการนำเสนอผลการทดลองในส่วนของการศึกษาวิธีการในรูปแบบอื่น ๆ อีก 4 รูปแบบหลักในการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในการก่อสร้างนอกเหนือจากวิธีการที่นำเสนอขึ้นในการวิจัยนี้ เพื่อเป็นการศึกษาเปรียบเทียบกัน โดยการศึกษาเปรียบเทียบนี้ทำการทดลองกับเฉพาะข้อมูลภาพในชุดข้อมูลที่ 1

4.2.1 การศึกษาเปรียบเทียบเมื่อใช้การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet

เป็นการนำสถาปัตยกรรมของโมเดล AlexNet ที่ผ่านการฝึกสอนมาก่อนมาใช้เพื่อศึกษาเปรียบเทียบกับโมเดล ResNet101 ที่ใช้ในวิธีการที่นำเสนอ โดยใช้วิธีการในทั้งสองรูปแบบของการเรียนรู้แบบถ่ายโอนคือแบบยึดคุณลักษณะจากตัวสกัดและแบบปรับแต่งการเรียนรู้ดังรายละเอียดจากการทดลองในข้อ (1) และข้อ (2) ตามลำดับ

(1) การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet ในแบบยึดคุณลักษณะจากตัวสกัด

โดยปกติแล้ว งานวิจัยส่วนใหญ่ที่นำโมเดลของ AlexNet แบบที่ผ่านการฝึกสอนมาก่อนไปใช้ในแนวคิดของการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดนั้นมักจะนำข้อมูลที่ต้องการศึกษามาส่งผ่านโมเดลเพื่อสกัดคุณลักษณะออกมาจากชั้น fc6 หรือ fc8 ของรูปที่ 3-6 แล้วนำคุณลักษณะที่ได้นั้นไปใช้งานต่อหรือนำไปผ่านขั้นตอนการจำแนกถ้าเป็นงานเพื่อการจำแนก

ข้อมูล สำหรับงานวิจัยนี้นอกเหนือจากการพิจารณาคุณลักษณะที่สกัดออกมาจากชั้น fc6 หรือ fc8 ดังกล่าวแล้ว ได้ทดลองในส่วนของชั้นอื่น ๆ ร่วมด้วย ซึ่งพบว่าคุณลักษณะที่สกัดออกมาจากชั้นต่าง ๆ เหล่านั้น นั้นเมื่อนำไปผ่านขั้นตอนการจำแนก ผลลัพธ์จากการจำแนกในบางชั้นน่าพอใจกว่าผลลัพธ์จากชั้น fc6 หรือ fc8 ที่ใช้กันโดยส่วนใหญ่ดังแสดงในตารางที่ 4-4 เมื่อข้อมูลที่ใช้ทดลองคือภาพจากชุดทดสอบของชุดข้อมูลที่ 1 จากตารางจะเห็นว่าคุณลักษณะที่สกัดออกมาจากชั้น pool5 ให้ผลลัพธ์จากการจำแนกดีกว่าชั้น fc6 และ fc8 นั่นคือมีค่า Accuracy เป็น 91.83%

ดังนั้นวิธีการเพื่อการศึกษาเปรียบที่เป็นการนำโมเดลของ AlexNet มาใช้ในแบบยึดคุณลักษณะจากตัวสกัดที่ศึกษาในงานวิจัยนี้จึงใช้คุณลักษณะที่สกัดออกมาจากชั้น pool5 ดังกล่าว นั่นคือเป็นการนำคุณลักษณะที่สกัดออกมาจากชั้น pool5 ไปผ่านขั้นตอนการจำแนกด้วยฟังก์ชัน SoftMax เมื่อส่วน Output ของโมเดล AlexNet ที่แสดงในรูปที่ 3-6 ถูกกำหนดให้เป็น 3 class scores ตามจำนวนกลุ่มของชุดข้อมูลที่ต้องการศึกษาแทนที่จะเป็น 1000 class scores

ตารางที่ 4-4 เปรียบเทียบประสิทธิภาพที่ได้จากการจำแนกเมื่อใช้คุณลักษณะที่สกัดออกมาจากชั้นที่แตกต่างกันของโมเดล AlexNet

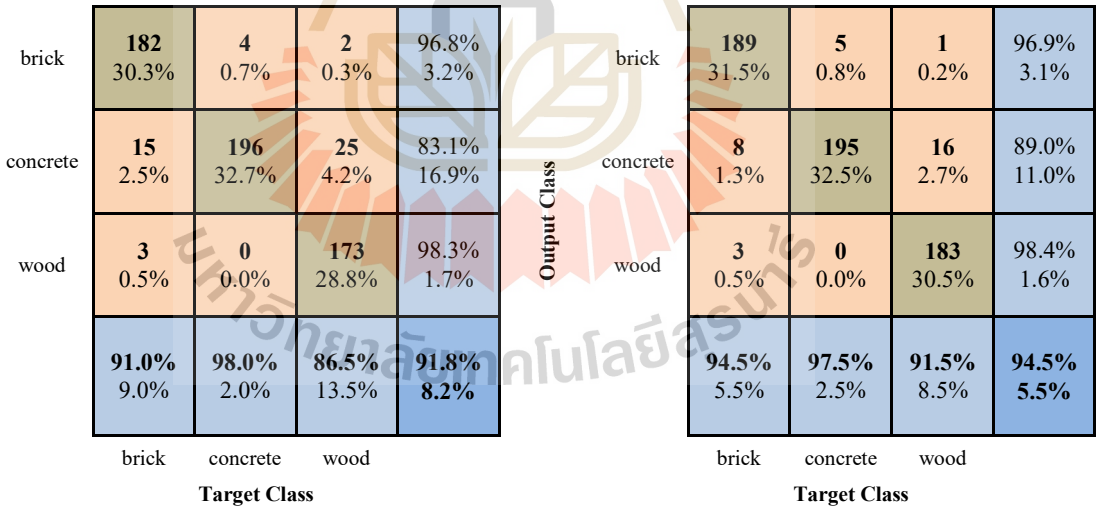
Extracted Feature from Layer	**Accuracy (%)**	**Extracted Feature from Layer**	**Accuracy (%)**
conv4	70.83	relu6	91.17
relu4	87.33	drop6	91.17
conv5	91.00	fc6	89.67
relu5	91.50	relu7	90.67
pool5	**91.83**	drop7	90.67
fc6	91.67	fc8	87.50

(2) การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet ในแบบปรับแต่งการเรียนรู้

การนำโมเดลที่ผ่านการฝึกสอนมาก่อนมาใช้งานในแบบปรับแต่งการเรียนรู้นั้นเป็นการฝึกสอนใหม่ให้กับบางช่วงของชั้นในโครงข่ายด้วยชุดข้อมูลที่ต้องการศึกษา ดังขั้นตอนในรายละเอียดที่กล่าวไปแล้วในข้อย่อยที่ (2) ของหัวข้อ 3.4.1

ผลการทดลองที่ได้เมื่อนำวิธีการที่เป็นการเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet ในทั้งสองรูปแบบมาทดสอบประสิทธิภาพในการจำแนกแสดงดังรูปที่ 4-3 เมื่อทดสอบด้วยข้อมูลในชุดทดสอบของชุดข้อมูลที่ 1 รูปที่ 4-3 แสดงผลในรูปแบบของ Confusion Matrix ที่เป็นผลลัพธ์

จากทั้งสองวิธีการเปรียบเทียบกัน เมื่อรูปที่ 4-3(a) เป็นผลลัพธ์จากวิธีการในแบบยึดคุณลักษณะจากตัวสกัด ที่ได้ค่า Accuracy จากการจำแนกเป็น 91.8% และรูปที่ 4-3(b) เป็นผลลัพธ์จากวิธีการในแบบปรับแต่งการเรียนรู้ มีค่า Accuracy จากการจำแนกเป็น 94.5% ส่วนตารางที่ 4-5 เป็นสรุปผลลัพธ์ที่ได้ของทั้งสองวิธีการด้วยมาตรวัดต่าง ๆ เพื่อเป็นการเปรียบเทียบกัน เมื่อ (1) ในตารางคือวิธีการในแบบยึดคุณลักษณะจากตัวสกัด และ(2) คือวิธีการในแบบปรับแต่งการเรียนรู้ ซึ่งจะเห็นว่าวิธีการในแบบปรับแต่งการเรียนรู้มีประสิทธิภาพดีกว่าวิธีการในแบบยึดคุณลักษณะจากตัวสกัดอย่างชัดเจน จากค่า Accuracy ที่เพิ่มขึ้น 2.7% นั่นคือจาก 91.8% เป็น 94.5% และเมื่อพิจารณาแยกตามแต่ละกลุ่มของวัสดุ จะเห็นว่าวิธีการในแบบปรับแต่งการเรียนรู้สามารถเพิ่มประสิทธิภาพในการจำแนกกลุ่มที่เป็น brick กับ wood ได้มากขึ้นจากค่าของ Recall ของสองกลุ่มนี้ที่เพิ่มขึ้น นั่นคือถ้าพิจารณา Confusion Matrix ดังรูปที่ 4-3 จะพบว่าจำนวนภาพในกลุ่ม brick ที่จำแนกถูกต้องมีเพิ่มขึ้น 7 ภาพ คือจาก 182 เป็น 189 และจำนวนภาพในกลุ่ม wood ที่จำแนกถูกต้องเพิ่มขึ้นมี 10 ภาพ คือจาก 173 เป็น 183 นอกจากนี้การจำแนกจากวิธีการแบบปรับแต่งการเรียนรู้นั้นมีความแม่นยำมากขึ้นในการจำแนกภาพในกลุ่มของ concrete นั่นคือค่า Precision ของกลุ่ม concrete เพิ่มขึ้นจาก 83.1% เป็น 89.0%



(a) จากวิธีการแบบยึดคุณลักษณะจากตัวสกัด (b) จากวิธีการแบบปรับแต่งการเรียนรู้

รูปที่ 4-3 Confusion Matrix ที่ได้การเรียนรู้แบบถ่ายโอนจากโมเดล AlexNet

ตารางที่ 4-5 สรุปผลลัพธ์ที่ได้จากการเรียนรู้แบบถ่ายโอนเมื่อใช้โมเดล AlexNet ในทั้งสองรูปแบบด้วยมาตรวัดต่าง ๆ

Class	Accuracy (%)		Recall (%)		Precision (%)		F-Measure (%)	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
brick	***91.8***	***94.5***	91.0	94.5	96.8	96.9	93.8	95.7
concrete			98.0	97.5	83.1	89.0	89.9	93.1
wood			86.5	91.5	98.3	98.4	92.0	94.8
Average			*91.8*	*94.5*	*92.7*	*94.8*	*91.9*	*94.5*

เมื่อ (1) คือแบบยึดคุณลักษณะจากตัวสกัดและ (2) คือแบบปรับแต่งการเรียนรู้

4.2.2 การศึกษาเปรียบเทียบเมื่อใช้การเรียนรู้แบบถ่ายโอนจากโมเดล GoogleNet

(1) การเรียนรู้แบบถ่ายโอนจากโมเดล GoogleNet ในแบบยึดคุณลักษณะจากตัวสกัด

โดยปกติแล้วงานวิจัยส่วนใหญ่ที่นำโมเดลของ GoogleNet แบบที่ผ่านการฝึกสอนมาก่อนไปใช้ในแบบยึดคุณลักษณะจากตัวสกัดนั้นมักจะนำข้อมูลที่ต้องการศึกษามาส่งผ่านโมเดลเพื่อสกัดคุณลักษณะออกมาจากชั้น pool5 หรือ loss3 ของรูปที่ 3-7 แล้วนำคุณลักษณะที่ได้นั้นไปใช้งานต่อหรือนำไปผ่านขั้นตอนการจำแนกถ้าเป็นงานเพื่อการจำแนกข้อมูล เช่นเดียวกันกับการทดลองที่ทำใน AlexNet สำหรับงานวิจัยนี้นอกเหนือจากการพิจารณาคุณลักษณะที่สกัดออกมาจากชั้น pool5 หรือ loss3 ดังกล่าวแล้ว ได้ทดลองในส่วนของชั้นอื่น ๆ ร่วมด้วย ซึ่งพบว่าคุณลักษณะที่สกัดออกมาจากชั้นต่าง ๆ เหล่านั้น นั้นเมื่อนำไปผ่านขั้นตอนการจำแนก ผลลัพธ์จากการจำแนกในบางชั้นน่าพอใจกว่าผลลัพธ์จากชั้น pool5 หรือ loss3 ที่ใช้กันโดยส่วนใหญ่ดังแสดงในตารางที่ 4-6 เมื่อข้อมูลที่ใช้ทดลองคือภาพจากชุดทดสอบของชุดข้อมูลที่ 1 จากตารางจะเห็นว่าคุณลักษณะที่สกัดออกมาจากชั้น Inception5a ให้ผลลัพธ์จากการจำแนกดีกว่าชั้น pool5 และ loss3 นั่นคือมีค่า Accuracy เป็น 92.33%

ดังนั้นวิธีการเพื่อการศึกษาเปรียบที่เป็นการนำโมเดลของ GoogleNet มาใช้ในแบบยึดคุณลักษณะจากตัวสกัดที่ศึกษาในงานวิจัยนี้จึงใช้คุณลักษณะที่สกัดออกมาจากชั้น Inception5a ดังกล่าว นั่นคือเป็นการนำคุณลักษณะที่สกัดออกมาจากชั้น Inception5a ไปผ่านขั้นตอนการจำแนกด้วยฟังก์ชัน SoftMax เมื่อส่วน Output ของโมเดล GoogleNet ที่แสดงในรูปที่ 3-7 ถูกกำหนดให้เป็น 3 class scores ตามจำนวนกลุ่มของชุดข้อมูลที่ต้องการศึกษาแทนที่จะเป็น 1000 class scores

ตารางที่ 4-6 เปรียบเทียบประสิทธิภาพที่ได้จากการจำแนกเมื่อใช้คุณลักษณะที่สกัดออกมาจากชั้นที่แตกต่างกันของโมเดล GoogleNet

Extracted Feature from Layer	**Accuracy (%)**	**Extracted Feature from Layer**	**Accuracy (%)**
Inception3a	79.17	Inception4e	92.00
Inception3a	79.67	pool4	91.33
pool3	87.33	**Inception5a**	**92.33**
Inception4a	86.33	Inception5b	90.17
Inception4b	89.17	pool5	91.17
Inception4c	89.50	dropout	91.17
Inception4d	92.00	loss3	86.17

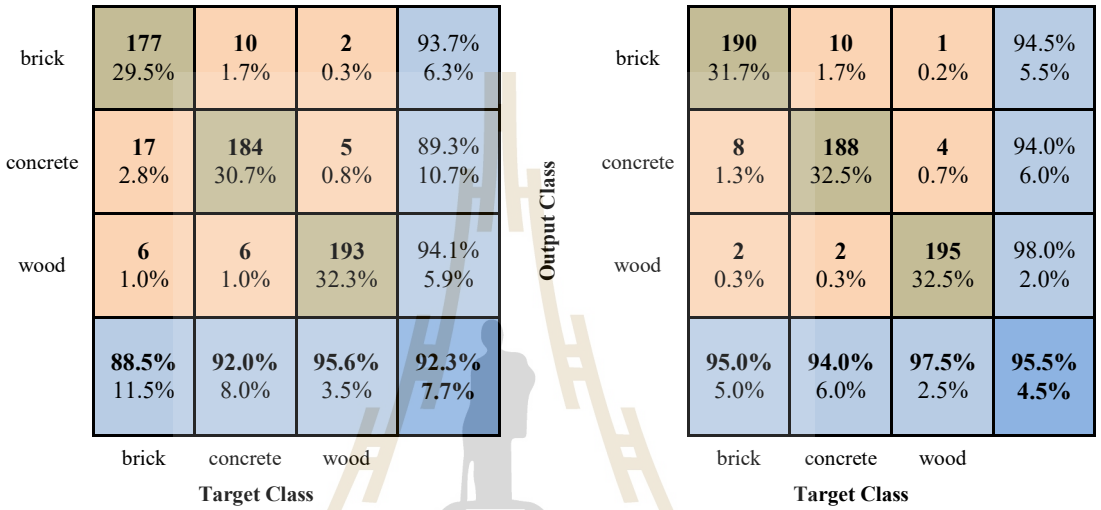
(2) การเรียนรู้แบบถ่ายโอนจาก GoogleNet ในแบบปรับแต่งการเรียนรู้

การนำโมเดลที่ผ่านการฝึกสอนมาก่อนของ GoogleNet มาใช้งานในแบบปรับแต่งการเรียนรู้นั้นเป็นการฝึกสอนใหม่ให้กับบางช่วงของชั้นในโครงข่ายด้วยชุดข้อมูลที่ต้องการศึกษา ดังขั้นตอนในรายละเอียดที่กล่าวไปแล้วในข้อย่อยที่ (2) ของหัวข้อ 3.4.1

ผลการทดลองที่ได้เมื่อนำวิธีการที่เป็นการเรียนรู้แบบถ่ายโอนจากโมเดล GoogleNet ในทั้งสองรูปแบบมาทดสอบประสิทธิภาพในการจำแนกแสดงดังรูปที่ 4-4 เมื่อทดสอบด้วยข้อมูลในชุดทดสอบของชุดข้อมูลที่ 1 และแสดงผลในรูปแบบของ Confusion Matrix ที่เป็นผลลัพธ์จากทั้งสองวิธีการเปรียบเทียบกัน รูปที่ 4-4(a) เป็นผลลัพธ์จากวิธีการในแบบยึดคุณลักษณะจากตัวสกัดที่ได้ค่า Accuracy จากการจำแนกเป็น 92.3% จากการนำคุณลักษณะที่สกัดออกมาได้จากชั้น Inception5a ไปผ่านขั้นตอนการจำแนก ส่วนรูปที่ 4-4(b) เป็นผลลัพธ์จากวิธีการในแบบปรับแต่งการเรียนรู้ มีค่า Accuracy จากการจำแนกเป็น 95.5%

ตารางที่ 4-7 แสดงการสรุปผลลัพธ์ที่ได้ของทั้งสองวิธีการด้วยมาตรวัดต่าง ๆ เพื่อเป็นการเปรียบเทียบกัน เมื่อ (1) ในตารางคือวิธีการในแบบยึดคุณลักษณะจากตัวสกัด และ(2) คือวิธีการในแบบปรับแต่งการเรียนรู้ ซึ่งจะเห็นว่าวิธีการในแบบปรับแต่งการเรียนรู้มีประสิทธิภาพดีกว่าวิธีการในแบบยึดคุณลักษณะจากตัวสกัดอย่างชัดเจน จากค่า Accuracy ที่เพิ่มขึ้น 3.2% นั่นคือจาก 92.3% เป็น 95.5% และเมื่อพิจารณาแยกตามแต่ละกลุ่มของวัสดุ จะเห็นว่าวิธีการในแบบปรับแต่งการเรียนรู้สามารถเพิ่มประสิทธิภาพในการจำแนกกลุ่มที่เป็น Brick กับ Wood ได้มากขึ้น

จากค่าของ Recall ของสองกลุ่มนี้ที่เพิ่มขึ้น นั่นคือถ้าพิจารณา Confusion Matrix ดังรูปที่ 4-4 จะพบว่าจำนวนภาพในกลุ่ม brick ที่จำแนกถูกต้องมีเพิ่มขึ้น 7 ภาพ คือจาก 182 เป็น 189 และจำนวนภาพในกลุ่ม wood ที่จำแนกถูกต้องเพิ่มขึ้นมี 10 ภาพ คือจาก 173 เป็น 183 นอกจากนี้การจำแนกจากวิธีการแบบปรับแต่งการเรียนรู้นั้นมีความแม่นยำมากขึ้นในการจำแนกภาพในกลุ่มของ concrete นั่นคือค่า Precision ของกลุ่ม concrete เพิ่มขึ้นจาก 83.1% เป็น 89.0%



(a) จากวิธีการแบบยึดคุณลักษณะจากตัวสกัด (b) จากวิธีการแบบปรับแต่งการเรียนรู้

รูปที่ 4-4 Confusion Matrix ที่ได้จากการเรียนรู้แบบถ่ายโอนจาก GoogleNet

จาก Confusion Matrix ในรูปที่ 4-4 สามารถสรุปเป็นค่าของมาตรวัดต่าง ๆ เพื่อเปรียบเทียบกันดังตารางที่ 4-7 จากตารางจะเห็นได้ว่าผลลัพธ์จาก GoogleNet ในแบบการปรับแต่งการเรียนรู้นั้นมีประสิทธิภาพมากกว่าแบบยึดคุณลักษณะสำหรับการจำแนกวัสดุในทุก ๆ กลุ่มของวัสดุ นั่นคือค่า Recall ของแต่ละกลุ่มมีค่าเพิ่มขึ้น ซึ่งกลุ่มของ brick มีค่า Recall ที่เพิ่มขึ้นมากที่สุด โดยถ้าดูจาก Confusion Matrix ในรูปที่ 4-4 จะพบว่ากลุ่มของ brick จะจำแนกถูกต้องเพิ่มมากขึ้นถึง 13 รูป (จาก 177 เป็น 190) หลังจากใช้รูปแบบของการปรับแต่งการเรียนรู้

ตารางที่ 4-7 สรุปผลลัพธ์ที่ได้จากการเรียนรู้แบบถ่ายโอนเมื่อใช้โมเดล GoogleNet ในทั้งสองรูปแบบ

Class	**Accuracy (%)**		**Recall (%)**		**Precision (%)**		**F-Measure (%)**	
	(1)	**(2)**	**(1)**	**(2)**	**(1)**	**(2)**	**(1)**	**(2)**
brick	*92.3*	*95.5*	88.5	95.0	93.7	94.5	91.0	94.7
concrete			92.0	94.0	89.3	94.0	90.6	94.0
wood			95.6	97.5	94.1	98.0	94.8	97.7
Average			*92.3*	*95.5*	*92.4*	*95.5*	*92.1*	*95.5*

เมื่อ (1) คือแบบยึดคุณลักษณะจากตัวสกัดและ (2) คือแบบปรับแต่งการเรียนรู้

(3) การกำหนดค่าพารามิเตอร์สำหรับวิธีการในแบบการปรับแต่งการเรียนรู้

ในขั้นตอนการฝึกสอนของวิธีการในแบบการปรับแต่งการเรียนรู้นั้นจำเป็นต้องมีการกำหนดค่าพารามิเตอร์สำคัญต่าง ๆ ที่ต้องใช้ในฝึกสอนใหม่ให้กับโมเดล AlexNet และ GoogleNet ซึ่งการเลือกใช้พารามิเตอร์ที่เหมาะสมที่สุดจะทำให้โมเดลที่ผ่านการปรับแต่งแล้วนั้นมีประสิทธิภาพมากที่สุด ค่าพารามิเตอร์สำคัญต่าง ๆ เหล่านั้นค้นหามาจากการใช้สองกลยุทธ์ประกอบกันคือ การลองผิดลองถูก (Trial and Error) และกลยุทธ์แบบสุ่ม (Random) การลองผิดลองถูกนำมาใช้เพื่อหาขอบเขตที่เหมาะสมของพารามิเตอร์แบบกว้าง ๆ ส่วนกลยุทธ์แบบสุ่มนำมาใช้เพื่อค้นหาขอบเขตแบบจำกัดที่เหมาะสมหลังจากที่ได้ขอบเขตแบบกว้างจากการลองผิดลองถูกมาแล้ว ตารางที่ 4-8 แสดงค่าพารามิเตอร์สำคัญนำมาใช้ในแต่ละโมเดลที่นำมาในการปรับแต่งการเรียนรู้ ซึ่งทำการค้นหามาจากการฝึกสอนด้วยชุดข้อมูลที่ 1

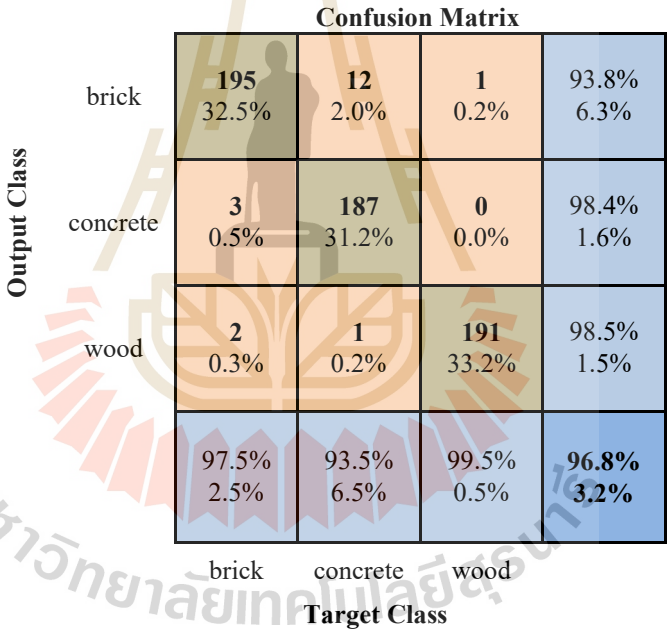
ตารางที่ 4-8 ค่าพารามิเตอร์สำคัญที่นำมาใช้ในแต่ละโมเดลที่นำมาใช้ในการปรับแต่งการเรียนรู้

Parameter/Model	**Freezed Layers**	**Mini-Batch Size**	**No. Epoch**	**Learning Rate**
AlexNet	1-19	5	20	0.0001
GoogleNet	1-110	5	20	0.0001

4.2.3 การศึกษาเปรียบเทียบเมื่อนำวิธีการของ PCA มาใช้แทนเครื่องเข้ารหัสอัตโนมัติ

เป็นการศึกษาเปรียบเทียบเมื่อนำเทคนิคการเข้ารหัสข้อมูลด้วยวิธี PCA มาใช้แทนโครงข่ายเครื่องเข้ารหัสอัตโนมัติในวิธีการที่นำเสนอ ดังนั้นขั้นตอนต่าง ๆ ในรายละเอียดการฝึกสอนและการทดสอบจึงเหมือนกันกับที่แสดงในรูปที่ 3-4 แต่เปลี่ยนจาก Autoencoder Network ในรูปดังกล่าวเป็น PCA

รูปที่ 4-5 แสดงผลลัพธ์ที่ได้จากการจำแนกภาพในชุดทดสอบจากชุดข้อมูลที่ 1 โดยแสดงผลลัพธ์ในรูปแบบของ Confusion Matrix จากผลลัพธ์ที่ได้จะเห็นว่ามีค่า Accuracy เท่ากับ 96.8% ซึ่งจาก Confusion Matrix ในรูปที่ 4-5 สามารถสรุปเป็นค่าในมาตรวัดต่าง ๆ ดังแสดงในตารางที่ 4-9 ซึ่งผลการทดลองด้วยชุดข้อมูลที่ 1 นี้ใช้จำนวนองค์ประกอบหลักทั้งหมด 36 องค์ประกอบจากการค้นหาจำนวนองค์ประกอบที่เหมาะสมด้วยข้อมูลชุดฝึกสอน



รูปที่ 4-5 Confusion Matrix แสดงผลลัพธ์ที่ได้จากวิธีการที่นำคุณลักษณะที่สกัดได้จากโมเดล ResNet101 มาเข้ารหัสด้วยวิธี PCA เมื่อจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

ตารางที่ 4-9 สรุปผลลัพธ์ที่ได้จากวิธีการที่นำคุณลักษณะที่สกัดได้จากโมเดล ResNet101 มาเข้ารหัสด้วยวิธี PCA เมื่อจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

Class	Accuracy (%)	Recall (%)	Precision (%)	F-Measure (%)
Brick	***96.8***	97.5	93.8	95.6
Concrete		93.5	98.4	95.9
Wood		99.5	98.5	99.0
	Average	*96.8*	*96.9*	*96.8*

4.2.4 การศึกษาเปรียบเทียบเมื่อสร้างสถาปัตยกรรมของโครงข่ายสำหรับการฝึกสอนขึ้นมาเอง

เป็นการศึกษาเปรียบเทียบในกรณีที่เป็นการสร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเองเพื่อใช้สำหรับการฝึกสอนโดยตรงกับชุดข้อมูลที่ศึกษา นั่นคือรูปแบบของชั้นต่าง ๆ หรือสถาปัตยกรรมที่แตกต่างกันของโครงข่ายประสาทแบบคอนโวลูชันจะสร้างขึ้นมาเพื่อใช้สำหรับการฝึกสอนด้วยข้อมูลชุดฝึกสอนเพื่อค้นหาสถาปัตยกรรมที่เหมาะสมด้วยการกำหนดค่า Hyperparameters ต่าง ๆ ที่เหมาะสม เมื่อการกำหนดค่า Hyperparameters ดังกล่าวเช่น การพิจารณาว่าจะมีจำนวนชั้นการคอนโวลูชันกี่ชั้นในโครงข่าย ขนาดของตัวกรองและจำนวนตัวกรองที่ใช้ในแต่ละชั้นเป็นเท่าไร ใช้วิธีการทำพูลลิ่งแบบใด ลำดับการเรียงของชั้นต่าง ๆ เป็นอย่างไร เป็นต้น หลักจากการนำสถาปัตยกรรมที่สร้างขึ้นมาไปผ่านขั้นตอนการฝึกสอนแล้วจึงนำข้อมูลชุดทดสอบมาใช้เพื่อการทดสอบหรือประเมินว่าโครงข่ายที่สร้างนั้นมีประสิทธิภาพเพียงใด ตารางที่ 4-10 แสดงรายละเอียดภายในโครงข่ายของสถาปัตยกรรมที่เหมาะสมที่สุดที่ได้จากการสร้างขึ้นมาเองในงานวิจัยนี้ โครงข่ายที่ได้ประกอบด้วยชั้นย่อยทั้งหมด 14 ชั้น โดยโครงข่ายดังกล่าวสร้างขึ้นมาจากการใช้ค่า Hyperparameters ที่กำหนดดังตารางที่ 4-11 เมื่อ “sgdm” คืออัลกอริทึมสำหรับการเรียนรู้ ซึ่งใช้แบบ Stochastic Gradient Descent with Momentum โดยค่าของ Momentum นั้นกำหนดเป็น 0.9 เช่นเดียวกันกับที่ใช้ในงานประยุกต์โดยส่วนใหญ่

รูปที่ 4-6 แสดงผลลัพธ์ที่ได้จากการจำแนกภาพในชุดทดสอบจากชุดข้อมูลที่ 1 เมื่อสร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเอง โดยแสดงผลลัพธ์ในรูปแบบของ Confusion Matrix เมื่อแนวคอลัมภ์เป็นเอาต์พุตเป้าหมายที่ต้องการ และแนวแถวเป็นเอาต์พุตที่ได้จากวิธีการที่นำเสนอ จากผลลัพธ์ที่ได้จะเห็นว่ามีค่า Accuracy เท่ากับ 70.7% ซึ่งจาก Confusion Matrix ในรูปที่ 4-6 สามารถสรุปเป็นค่าในมาตรวัดต่าง ๆ ดังแสดงในตารางที่ 4-12

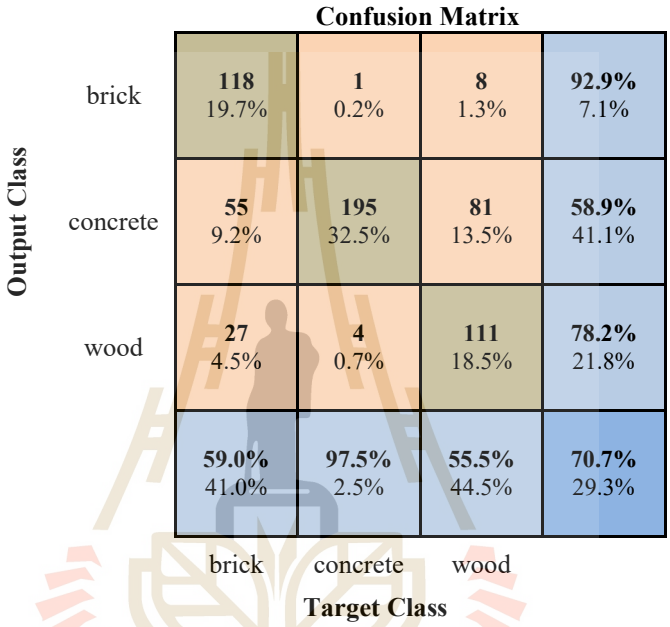
จากรูปที่ 4-6 จะเห็นว่าถึงแม้ค่าความถูกต้องโดยรวมของวิธีการที่เป็นการสร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายขึ้นมาเองนี้จะค่อนข้างน้อยเมื่อเทียบกับวิธีการอื่น ๆ แต่วิธีการนี้สามารถจำแนกวัสดุในกลุ่ม Concrete ได้ด้วยความถูกต้องที่ถือว่าสูงมากเมื่อเทียบกับอีกสองกลุ่มของวัสดุ นั่นคือจะเห็นว่าค่า Recall ของกลุ่ม Concrete ในตารางที่ 4-12 มีค่าสูงถึง 97.5% ในขณะที่อีกสองกลุ่ม มีค่าของ Recall ของแต่ละกลุ่มไม่ถึง 60%

ตารางที่ 4-10 รายละเอียดภายในโครงข่ายของสถาปัตยกรรมที่เหมาะสมที่สุดที่ได้จากการสร้างขึ้นมาเองในงานวิจัยนี้

ชั้นที่	**ชื่อชั้น**	**รายละเอียด**
1	Input	ค่าจาก 3 แกนสีของโมเดลสี RGB
2	Conv(5,16)	คอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 16 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
3	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
4	Conv(5,16)	คอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 16 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
5	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
6	MaxPool(2,2)	ทำพูลลิ่งแบบ 2×2 ด้วยขนาดการ Stride เป็น 2
7	Conv(5,32)	คอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 32 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
8	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
9	Conv(5,32)	คอนโวลูชันด้วยตัวกรองขนาด 5×5 จำนวน 32 ตัวกรอง ด้วยขนาดการ Stride เป็น 1
10	ReLU	ชั้นการแปลงค่าด้วยฟังก์ชัน ReLU
11	MaxPool(2,2)	ทำพูลลิ่งแบบ 2×2 ด้วยขนาดการ Stride เป็น 2
12	FC(3)	ชั้นการเชื่อมถึงกันหมดที่มี 3 Neurons
13	SoftMax	ชั้นการแปลงค่าด้วยฟังก์ชัน SoftMax
14	Output	3 Classes

ตารางที่ 4-11 ค่า Hyperparameters ที่กำหนดในของสถาปัตยกรรมที่เหมาะสมที่สุดที่ได้จากการสร้างโครงข่ายขึ้นมาเอง

Hyperparameters			
Learning Algorithm	**Mini-Batch Size**	**No. Epoch**	**Learning Rate**
"sgdm"	5	450	0.0001



รูปที่ 4-6 Confusion Matrix แสดงผลลัพธ์ที่ได้จากวิธีการที่สร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเอง เมื่อจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

ตารางที่ 4-12 สรุปผลลัพธ์ที่ได้จากวิธีการที่สร้างสถาปัตยกรรมที่เหมาะสมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเอง เมื่อจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

Class	**Accuracy (%)**	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Brick	***70.7***	59.0	92.9	72.2
Concrete		97.5	58.9	73.4
Wood		55.5	78.2	64.9
	Average	*70.7*	*76.7*	*70.2*

4.3 การทดลองเพิ่มเติมและการเปรียบเทียบผลการศึกษา

จากผลการทดลองของวิธีการที่นำเสนอและวิธีการเพื่อการศึกษาเปรียบเทียบที่ได้แสดงรายละเอียดในหัวข้อ 4.1 และ 4.2 ไปแล้วนั้น พบว่าในวิธีการทั้งหมดที่ศึกษาเมื่อทำการทดลองกับชุดข้อมูลที่ 1 นั้น วิธีการที่นำเสนอขึ้นในงานวิจัยนี้จากการนำการเรียนรู้แบบถ่ายโอนจากโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดมาใช้ร่วมกับโครงข่ายเข้ารหัสอัตโนมัติ (Autoencoder) นั้นมีประสิทธิภาพมากที่สุดจากมาตรวัดที่เป็นค่า Accuracy นั่นคือมีค่าเป็น 97.8% ซึ่งการเปรียบเทียบวิธีการที่ศึกษาทั้งหมดเหล่านั้นเมื่อแสดงตามลำดับของค่า Accuracy สามารถสรุปได้ดังตารางที่ 4-13

ตารางที่ 4-13 เปรียบเทียบวิธีการที่ศึกษาทั้งหมดเมื่อแสดงตามลำดับของค่า Accuracy ที่ได้จากการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

ลำดับที่	วิธีการ	Accuracy (%)
1	ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder (Proposed Work)	97.8
2	ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA	96.8
3	GoogleNet แบบปรับแต่งการเรียนรู้	95.5
4	AlexNet แบบปรับแต่งการเรียนรู้	94.5
5	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด	93.3
6	AlexNet แบบยึดคุณลักษณะจากตัวสกัด	91.8
7	การสร้างสถาปัตยกรรมขึ้นมาฝึกสอนเอง	70.7

จากตารางที่ 4-13 พบว่านอกเหนือจากวิธีการที่นำคุณลักษณะจาก ResNet101 แบบยึดคุณลักษณะจากตัวสกัดมาใช้จะได้ประสิทธิภาพที่น่าพอใจแล้ว วิธีการที่นำคุณลักษณะจาก GoogleNet มาใช้ก็น่าสนใจ ในงานวิจัยนี้จึงได้ทำการทดลองเพิ่มเติมในส่วนของ GoogleNet เพื่อให้ผลลัพธ์ที่ได้จากการเปรียบเทียบครอบคลุมมากยิ่งขึ้น ดังนั้นคุณลักษณะที่ได้จาก ResNet101 และ GoogleNet จึงนำไปทดลองเพิ่มเติมด้วยการนำ PCA และเครื่องเข้ารหัสอัตโนมัติมาใช้ร่วมด้วยในทุกกรณีที่ศึกษา ผลที่ได้จากการทดลองทั้งหมดเมื่อทดลองกับชุดข้อมูลที่ 1 จึงสรุปเพื่อเปรียบเทียบกันได้ดังตารางที่ 4-14

เนื่องจากการนำคุณลักษณะที่ได้จาก GoogleNet ในแบบยึดคุณลักษณะจากตัวสกัดที่นำมาใช้ในการทดลองก่อนหน้านี้ (การทดลองในข้อ (1) ของหัวข้อ 4.2.2) เป็นคุณลักษณะที่สกัด

มาจากชั้น Inception5a ซึ่งถ้าต้องการนำคุณลักษณะที่สกัดจากชั้นดังกล่าวออกมาจากโมเดล GoogleNet แล้วนำไปใช้ต่อ จะทำได้ยากเพราะข้อมูลอยู่ในรูปแบบที่ซับซ้อน (นั่นคือเป็นลักษณะของเทนเซอร์ใน 3 มิติ) สำหรับการทดลองในขั้นตอนนี้จึงนำคุณลักษณะที่สกัดมาจากชั้น pool5 มาใช้แทน ดังนั้นค่า Accuracy ในตารางที่ 4-14 ในส่วนของ GoogleNet ที่เป็นแบบยึดคุณลักษณะจากตัวสกัดจึงเป็น 91.1 สำหรับในส่วนของ ResNet101 นั้นในงานวิจัยนี้ไม่ทำการศึกษาในส่วนของการปรับแต่งการเรียนรู้ (จากการกำหนดค่าเป็น NA ในตาราง) เนื่องจาก ResNet101 มีความซับซ้อนมากจากการที่มีจำนวนชั้นในรายละเอียดมากถึง 347 ชั้น กระบวนการปรับแต่งการเรียนรู้เพื่อให้ได้โมเดลใหม่ที่เหมาะสมที่สุดจะทำได้ลำบาก ในส่วนของตารางที่ 4-14 ที่เป็นคอลัมน์ขวาสุดนั้นเป็นการหาค่าออกมาว่าหลังจากการนำ PCA และ Autocoder มาใช้ร่วมกันกับคุณลักษณะจากฐาน (Based) แต่ละแบบแล้ว ผลที่ได้มีการปรับปรุงดีขึ้นหรือไม่ เพียงใด เช่นเมื่อนำ PCA มาใช้ร่วมกับคุณลักษณะที่ใช้เป็นฐานคือจากแบบปรับแต่งการเรียนรู้ด้วย GoogleNet พบว่าประสิทธิภาพที่ได้ลดลง 1.9% (นั่นคือ 93.6 – 95.5 = –1.9)

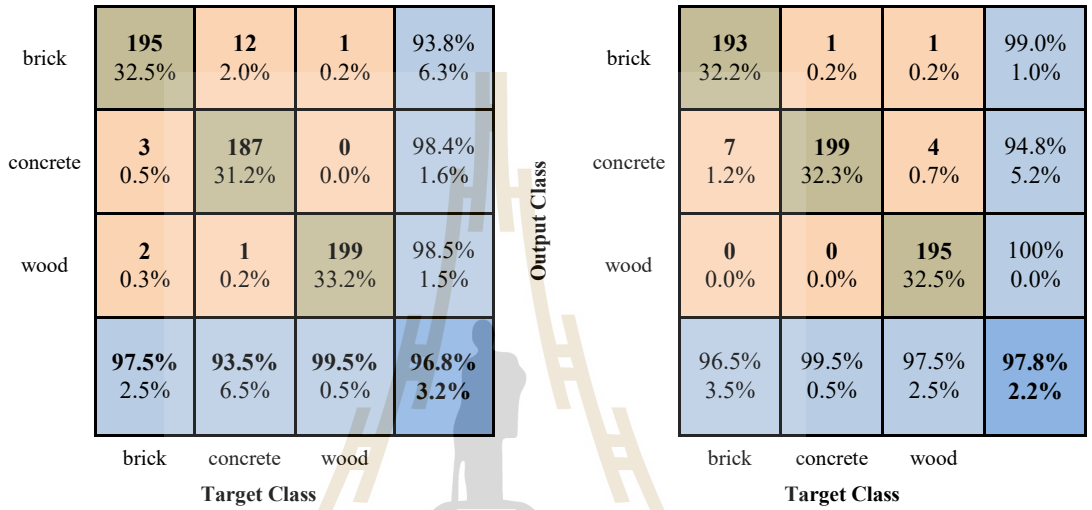
ตารางที่ 4-14 เปรียบเทียบผลการศึกษาจากการวัดค่า Accuracy (%) เมื่อนำเครื่องเข้ารหัสอัตโนมัติและ PCA มาใช้ร่วมด้วยในทุกกรณีที่ศึกษากับ GoogleNet และ ResNet101 ในการจำแนกภาพชุดทดสอบของชุดข้อมูลที่ 1

Model/วิธีการ	**รูปแบบของการเรียนรู้แบบถ่ายโอน**		**Improvement from Based (%)**	
	แบบยึดคุณลักษณะจากตัวสกัด	**แบบปรับแต่งการเรียนรู้**		
GoogleNet	*(91.1)*	*(95.5)*	*(Based)*	
PCA	92.3	93.6	1.2	–1.9
Autoencoder	93.5	95.8	1.4	0.3
ResNet101	*(95.0)*	NA	*(Based)*	
PCA	96.8	NA	1.8	
Autoencoder	**97.8**	NA	2.8	

รูปที่ 4-7 แสดง Confusion Matrix ที่ได้จากสองวิธีการที่มีประสิทธิภาพมากที่สุดสำหรับการจำแนกข้อมูลในชุดทดสอบจากชุดข้อมูลที่ 1 ดังที่แสดงค่าผลลัพธ์จากการจำแนกในตารางที่ 4-14 โดยรูปที่ 4-7(a) คือวิธีการจากการใช้ ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับ PCA มีค่า Accuracy จากการจำแนกข้อมูลชุดที่ 1 เป็น 96.8% ส่วนรูปที่ 4-7(b) เป็นวิธีการจากการใช้

ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับ Autoencoder มีค่า Accuracy จากการจำแนกข้อมูลชุดที่ 1 เป็น 97.8%

สำหรับผลที่ได้จากการทดลองทั้งหมดเมื่อทดลองกับชุดข้อมูลที่ 2 นั้นสรุปเพื่อเปรียบเทียบกันได้ดังตารางที่ 4-15 ซึ่งพบว่า การนำ PCA มาใช้ร่วมกับคุณลักษณะที่สกัดได้จาก GoogleNet ในทั้งสองรูปแบบนั้นกลับทำให้ประสิทธิภาพสำหรับการจำแนกภาพในชุดข้อมูลที่ 2 ต่ำลง



(a) ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA

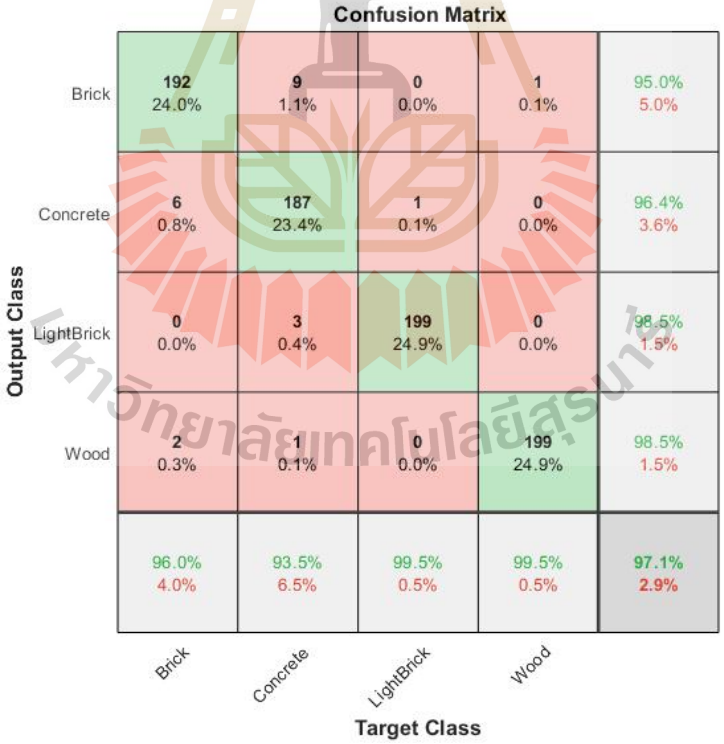
(b) ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder

รูปที่ 4-7 Confusion Matrix ที่ได้จากสองวิธีการที่มีประสิทธิภาพมากที่สุดสำหรับชุดข้อมูลที่ 1

รูปที่ 4-8 และ รูปที่ 4-9 นั้นแสดง Confusion Matrix ที่ได้จากสองวิธีการที่มีประสิทธิภาพมากที่สุดสำหรับการจำแนกข้อมูลในชุดทดสอบจากชุดข้อมูลที่ 2 รูปที่ 4-8 คือวิธีการจากการใช้ ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับ PCA ส่วนรูปที่ 4-9 เป็นวิธีการจากการใช้ ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับเครื่องเข้ารหัสอัตโนมัติ ซึ่งรูปที่ 4-9 นั้นคือรูปเดียวกันกับรูปที่ 4-2 ที่เคยกล่าวถึงไปแล้ว ในที่นี้นำมาใช้อีกเพื่อความชัดเจนในการเปรียบเทียบ

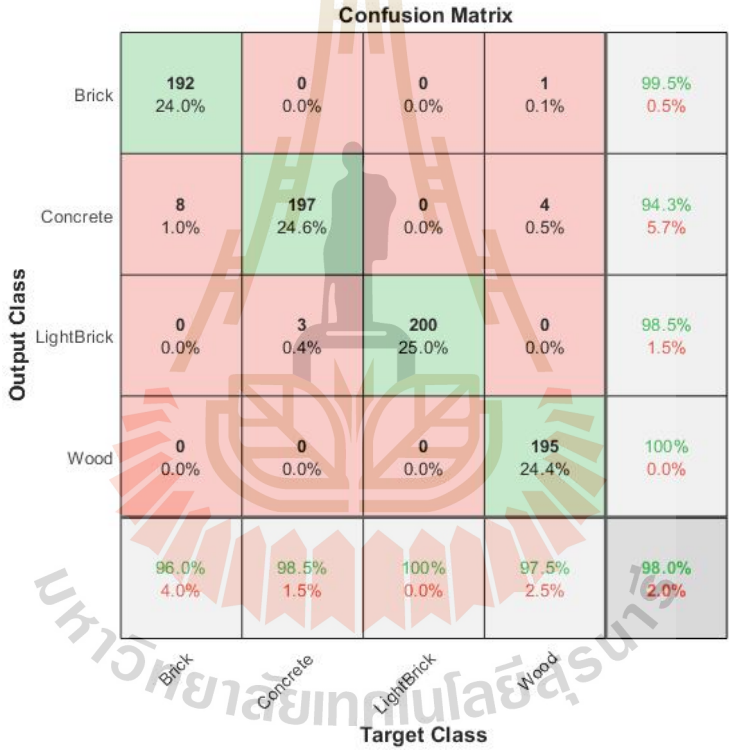
ตารางที่ 4-15 เปรียบเทียบผลการศึกษาจากการวัดค่า Accuracy (%) เมื่อนำเครื่องเข้ารหัสอัตโนมัติและPCA มาใช้ร่วมด้วยในทุกกรณีที่ศึกษากับ GoogleNet และ ResNet101 ในการจำแนกภาพชุดทดสอบของชุดข้อมูลที่ 2

Model/วิธีการ	**รูปแบบของการเรียนรู้แบบถ่ายโอน**		**Improvement from Based (%)**	
	แบบยึดคุณลักษณะจากตัวสกัด	**แบบปรับแต่งการเรียนรู้**		
GoogleNet	*(92.0)*	*(93.6)*	*(Based)*	
PCA	91.6	92.6	–0.4	–1.0
Autoencoder	92.5	94.8	0.5	1.2
ResNet101	*(96.0)*	NA	*(Based)*	
PCA	97.1	NA	1.1	NA
Autoencoder	**98.0**	NA	2.0	NA



รูปที่ 4-8 Confusion Matrix แสดงผลลัพธ์ที่ได้จาก ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับ PCA สำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2

รูปที่ 4-10 แสดงภาพที่จำแนกผิดทั้งหมดด้วยวิธีการที่นำเสนอ เมื่อนำมาจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2 ที่ศึกษาในงานวิจัยนี้ ซึ่งมีจำนวนภาพในชุดทดสอบทั้งหมด 800 ภาพ และมีการจำแนกผิดไป 16 ภาพ แบ่งเป็นการจำแนกผิดจาก Brick เป็น Concrete จำนวน 8 ภาพดังแสดงในรูปที่ 4-10(a) มีการจำแนกผิดจาก Concrete เป็น Light Brick จำนวน 3 ภาพดังแสดงในรูปที่ 4-10(b) มีการจำแนกผิดจาก Wood เป็น Concrete จำนวน 4 ภาพ (แถวบนของ รูปที่ 4-10(c)) และมีจำแนกจาก Wood เป็น Brick จำนวน 1 ภาพ (แถวล่างของ รูปที่ 4-10(c)) จำนวนและรายละเอียดของการจำแนกดังกล่าวนี้สอดคล้องกับผลลัพธ์ที่ได้จาก Confusion Matrix ในรูปที่ 4-9 ที่ได้แสดงไปแล้วก่อนหน้านี้



รูปที่ 4-9 Confusion Matrix แสดงผลลัพธ์ที่ได้จาก ResNet101 แบบยึดคุณลักษณะจากตัวสกัดร่วมกับ Autoencoder สำหรับการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2

(a) จำแนกจาก Brick เป็น Concrete

(b) จำแนกจาก Concrete เป็น Light Brick

(c) จำแนกจาก Wood เป็น Concrete (แถวบน) และ จำแนกจาก Wood เป็น Brick (แถวล่าง)

รูปที่ 4-10 ภาพที่จำแนกผิดทั้งหมดจากชุดทดสอบในชุดข้อมูลที่ 2 ด้วยวิธีการที่นำเสนอ

4.4 การวิเคราะห์ความซับซ้อน (Complexity) ของวิธีการที่นำเสนอ

4.4.1 การวิเคราะห์ Time Complexity และ Space Complexity

วิธีการที่นำเสนอในงานวิจัยนี้เป็นการนำการเรียนรู้แบบถ่ายโอนจากโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดมาใช้งานร่วมกันกับโครงข่ายเครื่องเข้ารหัสอัตโนมัติและเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มสำหรับการจำแนกภาพวัสดุในงานก่อสร้าง ดังนั้นวิธีการที่นำเสนอนั้นประกอบด้วย 3 ส่วนหลักคือ ส่วนของโมเดล ResNet101 ส่วนของโครงข่ายเครื่องเข้ารหัสอัตโนมัติ และส่วนของเครื่องเวกเตอร์เกื้อหนุน ในการวิเคราะห์ความซับซ้อนของวิธีการจึงต้องวิเคราะห์ความซับซ้อนของอัลกอริทึมในแต่ละส่วนแล้วนำทั้งหมดนั้นมารวมกัน ซึ่งรายละเอียดของแต่ละส่วนมีดังต่อไปนี้

(1) ส่วนของโมเดล ResNet101

ในการวิเคราะห์หรือวัดความซับซ้อนทางเวลา (Time Complexity) ของอัลกอริทึมที่เป็นโมเดลแบบเชิงลึกหรือโครงข่ายประสาทแบบคอนโวลูชันเพื่อเป็นการเปรียบเทียบกันนั้น ในทางปฏิบัติจะใช้การนับจำนวน FLOPs (Floating Point Operations) ทั้งหมดที่ต้องใช้ในโมเดล ซึ่งจำนวน FLOPs โดยส่วนใหญ่ของโครงข่ายประสาทแบบคอนโวลูชันมาจากการทำ Operation สำหรับการคูณและการรวมผลคูณ (Multiply and Accumulates, MACs) ที่ต้องใช้ทั้งหมดในโครงข่าย ดังนั้นถ้าโมเดลใดมีจำนวนของ FLOPs มากกว่าอีกโมเดลหนึ่งก็สามารถวิเคราะห์ได้ว่าโมเดลนั้นมี Time Complexity มากกว่าอีกโมเดลหนึ่ง ส่วนการวัดซับซ้อนในการใช้พื้นที่หน่วยความจำ (Space Complexity) นั้นจะพิจารณาจากจำนวน Parameters ทั้งหมดที่ใช้ในโมเดล ซึ่งคือ Parameters สำหรับการเก็บค่าของ Weights และ Parameters สำหรับการเก็บค่าของแผนที่คุณลักษณะ (Feature Maps) ในแต่ละชั้น เพราะถ้าจำนวน Parameters ทั้งหมดที่ใช้มีมาก ปริมาณของหน่วยความจำที่ต้องใช้สำหรับการจัดเก็บ Parameters เหล่านั้นก็ต้องมากด้วย

ตารางที่ 4-16 ยกตัวอย่างการคำนวณในส่วนของการนับจำนวน Parameters ทั้งหมดที่ใช้ในโมเดลของ AlexNet โดยในส่วนของ ResNet101 ก็นับด้วยหลักการเดียวกัน (ในที่นี้ยกตัวอย่างจาก AlexNet เพราะโมเดลไม่ซับซ้อนมากเกินไปสำหรับการนำเสนอในรายละเอียดการนับเพื่อเป็นตัวอย่าง) จากตารางที่ 4-16 ในส่วนของคอลัมน์ “in” ของแต่ละชั้นคือจำนวน Channel ของอินพุตที่เข้าสู่ชั้นนั้น (นั่นคือ Depth ของเทนเซอร์อินพุต) และส่วนของคอลัมน์ “out” คือจำนวน Kernel (Mask) ที่ใช้ในชั้นนั้น จากตารางจะเห็นว่าจำนวน Parameters ของแต่ละชั้นคำนวณมาจาก (Kernel Size) × (จำนวน Channel) × (จำนวน Kernel) ดังนั้นจำนวน Parameters ของชั้นแรก (นั่นคือ Layer “conv1”) ในตารางที่ 4-16 จึงมีจำนวนเป็น $11\times\mathbf{11}\times3\times\mathbf{96}=\mathbf{34944}$ ส่วนจำนวน Parameters ของชั้นอื่น ๆ ก็คิดให้รูปแบบเดียวกัน สำหรับ AlexNet เมื่อนำ Parameters ที่คำนวณได้ในแต่ละชั้นมารวมกันจึงได้จำนวน Parameters ทั้งหมดเป็น 62,388,344 หรือประมาณ 62M (Million)

ตารางที่ 4-17 แสดงจำนวน Parameters และจำนวน FLOPs ทั้งหมดที่ใช้ในแต่ละ Model ที่ศึกษาในงานวิจัยนี้เปรียบเทียบกัน จะเห็นว่า AlexNet นั้นมี Time Complexity ที่ดีที่สุด (มี # of FLOPs น้อยที่สุด) แต่มี Space Complexity ที่ด้อยที่สุด (มี # of Parameters มากที่สุด) ส่วน GoogleNet นั้นมี Space Complexity ที่ดีกว่า AlexNet ประมาณ 10 เท่า และดีกว่า ResNet101 ประมาณ 7 เท่า ส่วน Time Complexity นั้นด้อยกว่า AlexNet แต่ดีกว่า ResNet เกือบ 4 เท่า ทั้งนี้ในส่วนของ Time Complexity นั้นจะเห็นว่ามีความซับซ้อนที่แปรตามความลึก (Depth) ของโมเดล

ดังนั้น ResNet101 ซึ่งมีความลึกมากกว่า AlexNet และ GoogleNet มากจึงมี Time Complexity ที่ด้อยกว่า แต่ ResNet101 ก็ยังใช้จำนวน Parameters ที่น้อยกว่า AlexNet มาก

ตารางที่ 4-16 ตัวอย่างการนับจำนวน Parameters ทั้งหมดที่ต้องใช้ใน AlexNet (Anwar, 2019)

AlexNet Network - Structural Details													
Input			Output			Layer	Stride	Pad	Kernel size		in	out	# of Param
227	227	3	55	55	96	conv1	4	0	11	11	3	96	34944
55	55	96	27	27	96	maxpool1	2	0	3	3	96	96	0
27	27	96	27	27	256	conv2	1	2	5	5	96	256	614656
27	27	256	13	13	256	maxpool2	2	0	3	3	256	256	0
13	13	256	13	13	384	conv3	1	1	3	3	256	384	885120
13	13	384	13	13	384	conv4	1	1	3	3	384	384	1327488
13	13	384	13	13	256	conv5	1	1	3	3	384	256	884992
13	13	256	6	6	256	maxpool5	2	0	3	3	256	256	0
						fc6			1	1	9216	4096	37752832
						fc7			1	1	4096	4096	16781312
						fc8			1	1	4096	1000	4097000
						Total							**62,378,344**

ตารางที่ 4-17 จำนวน Parameters และจำนวน FLOPs ทั้งหมดที่ใช้ในแต่ละ Model ที่ศึกษาในงานวิจัยนี้ (Anwar, 2019; Zagoruyko and Komodakis, 2019; He, et al., 2015)

Model	**Depth**	**# of Parameters**	**# of FLOPs**
AlexNet	8	62M (Million)	1.5B (Billion)
GoogleNet	22	6.4M	2.0B
ResNet101	101	44.5M	7.6B

วิธีการที่นำเสนอในงานวิจัยนี้เป็นการนำโมเดลของ ResNet101 มาใช้ในรูปแบบของการเรียนรู้แบบถ่ายโอนในแบบยึดคุณลักษณะจากตัวสกัด ซึ่งไม่ต้องมีการฝึกสอนเพิ่มเติมใด ๆ ให้กับโครงข่าย ดังนั้นการที่โครงข่าย ResNet101 มี Time Complexity ที่ค่อนข้างสูงนั้นไม่ได้มีผลกระทบกับเวลาในการทำงานของวิธีการที่นำเสนอมาก เพราะข้อมูลแต่ละตัวถูกส่งผ่านโครงข่ายเพียง 1 รอบ ไม่ว่าจะเป็นขั้นตอนการฝึกสอนหรือการทดสอบ

(2) ส่วนของโครงข่ายเครื่องเข้ารหัสอัตโนมัติ

จริง ๆ แล้วโครงข่ายเครื่องเข้ารหัสอัตโนมัติก็คือโครงข่าย MLP (Multi-Layer Perceptron) รูปแบบหนึ่ง การวิเคราะห์ความซับซ้อนจึงใช้หลักการเดียวกันกับอัลกอริทึมของ MLP (Fredenslund, 2018) นั่นคือมี Time Complexity โดยประมาณ (Approximately) ในรูปของ Big-O เป็น

$$O(\#\text{ of Epochs} \times \#\text{ of Neurons} \times \#\text{ of Features} \times \#\text{ of Examples})$$

เมื่อ # of Epochs คือจำนวน Epoch ทั้งหมดที่ใช้

# of Examples คือจำนวนข้อมูลทั้งหมด (n)

# of Features คือจำนวนมิติของข้อมูล

# of Neurons คือจำนวนนิวรอนทั้งหมดที่ใช้ในโครงข่าย

ดังนั้น สำหรับงานวิจัยนี้ ในขั้นตอนการฝึกสอนโครงข่ายเข้ารหัสอัตโนมัติที่เป็นการทดลองกับชุดข้อมูลที่ 1 ซึ่งใช้โครงข่ายที่มี 1 ชั้นซ่อนเร้นที่มีจำนวนนิวรอนในชั้นนั้นเป็น 40 และฝึกสอนจำนวน 380 Epochs สามารถคำนวณได้ว่ามี Time Complexity เป็น

$$O(380 \times 2{,}040 \times 1000 \times \boldsymbol{n})$$

เมื่อ # of Examples (n) สำหรับการฝึกสอนคือ 1,200

# of Neurons ในชั้นอินพุตเท่ากันกับในชั้นเอาต์พุต (เท่ากับ 1,000)

# of Features คือ 1,000

(3) ส่วนของเครื่องเวกเตอร์เกื้อหนุน

อัลกอรึทึมของเครื่องเวกเตอร์เกื้อหนุนนั้นมี Time Complexity เป็น

$$O(\#\text{ of Examples}^{\mathbf{3}})$$

โดยเวลาส่วนใหญ่ที่ใช้มาจากขั้นตอนการคำนวณ Invert Matrix ที่มีขนาดเป็น $R \times R$ เมื่อ R คือจำนวน Support Vector (Bordes, et al., 2005) นั่นคือถ้าข้อมูลทุกตัวถูกมองว่าเป็น Support Vector เวลาในการคำนวณจะเป็น $n^{\mathbf{3}}$ ($R = n$)

4.4.2 การวัด Running Time

งานวิจัยนี้ทำการทดลองเพื่อวัด Running Time ของวิธีการที่นำเสนอจากการใช้โปรแกรม Matlab เวอร์ชัน R2018a แบบ 64 bits ด้วยคอมพิวเตอร์ที่ใช้สำหรับการวิจัยซึ่งใช้หน่วยประมวลผลกลาง คือ Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz ใช้หน่วยประมวลผลกราฟิก คือ NVIDIA GeForce GTX 1060 3GB มีหน่วยความจำหลัก 16 GB และใช้ระบบปฏิบัติการเป็น Windows 7 Ultimate 64 bits จากการทดลองกับชุดข้อมูลที่ 2 ที่ศึกษาในงานวิจัยนี้ ซึ่งประกอบด้วยภาพในชุดฝึกสอนทั้งหมด 1600 ภาพ และภาพในชุดทดสอบทั้งหมด 800 ภาพ พบว่าในขั้นตอนการฝึกสอนมี Running Time เป็น 91.4716 วินาที (Seconds) และในขั้นตอนการทดสอบมี Running Time เป็น 12.8061 วินาที

4.5 การอภิปรายผลการศึกษา

เป้าหมายของงานวิจัยนี้คือการนำโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยนอกเหนือจากวิธีการที่นำเสนอขึ้นสำหรับงานวิจัยนี้แล้วยังได้นำเสนอการศึกษาเกี่ยวกับวิธีการอื่นในหลากหลายรูปแบบของการนำโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้เพื่อเป็นการเปรียบเทียบกัน ผลจากการศึกษาที่ได้สามารถสรุปเป็นประเด็นต่าง ๆ ที่น่าสนใจได้หลายประเด็น ดังต่อไปนี้

(1) วิธีการที่มีประสิทธิภาพมากที่สุดสำหรับการจำแนกภาพวัสดุในงานก่อสร้างคือวิธีการที่นำเสนอขึ้นในงานวิจัยนี้ นั่นคือเป็นการนำการเรียนรู้แบบถ่ายโอนจากโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดมาใช้ร่วมกับโครงข่ายเครื่องเข้ารหัสอัตโนมัติและใช้เครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มสำหรับขั้นตอนการจำแนก โดยวิธีที่นำเสนอสามารถจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1 ได้ด้วยค่าความถูกต้อง 97.8% และสามารถจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2 ได้ด้วยค่าความถูกต้อง 98.0% ซึ่งค่าความถูกต้องที่ได้ในทั้งสองชุดข้อมูลนั้นสูงกว่าวิธีการอื่น ๆ ที่นำมาศึกษาเปรียบเทียบกัน นั่นคือวิธีการที่นำเสนอมีความเหมาะสมที่สุดสำหรับการจำแนกภาพวัสดุในงานก่อสร้าง

(2) วิธีการที่เป็นการนำเครื่องเข้ารหัสอัตโนมัติมาใช้ร่วมกันกับคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนสามารถเพิ่มประสิทธิภาพในการจำแนกได้ดีกว่าวิธีการที่เป็นการนำ PCA มาใช้ในทุก ๆ กรณีที่ได้ทำการศึกษา นั่นคือเครื่องเข้ารหัสอัตโนมัติมีประสิทธิภาพที่ดีกว่า PCA สำหรับการใช้เป็นวิธีการเพื่อสร้างตัวแทนข้อมูลที่เหมาะสำหรับงานการจำแนกภาพวัสดุในงานก่อสร้าง

(3) การนำ PCA มาใช้ร่วมกับโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดสามารถช่วยเพิ่มประสิทธิภาพในการจำแนกได้ในทั้งสองชุดข้อมูลที่ศึกษาแต่เพิ่มขึ้นได้ไม่สูงเท่ากับการใช้เครื่องเข้ารหัสอัตโนมัติ ในขณะที่การนำ PCA มาใช้ร่วมกับโมเดล GoogleNet ในแบบปรับแต่งการเรียนรู้นั้นทำให้ประสิทธิภาพในการจำแนกแย่ลงในทั้งสองชุดข้อมูลที่ศึกษา (จากค่าที่ติดลบในตารางที่ 4-14 และ 4-15)

(4) จากการนำโมเดล AlexNet และ GoogleNet มาศึกษาเปรียบเทียบกันพบว่า วิธีการที่เป็นการนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนในแบบปรับแต่งการเรียนรู้มาใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างนั้นมีประสิทธิภาพเหนือกว่าวิธีการนำคุณลักษณะในแบบยึดคุณลักษณะจากตัวสกัดมาใช้ ดังผลการทดลองที่ได้จาก จากรูปที่ 4-3 และรูปที่ 4-4 แต่กระบวนการในการเรียนรู้เพื่อการปรับแต่งการเรียนรู้นั้นมี

ความซับซ้อนในขั้นตอนการกำหนดและการค้นหาค่าพารามิเตอร์สำคัญต่าง ๆ เพื่อให้ได้มาซึ่งโมเดลที่ผ่านการปรับแต่งแล้วที่เหมาะสมที่สุด โดยเฉพาะอย่างยิ่งถ้าสถาปัตยกรรมที่ผ่านการฝึกสอนมาก่อนที่นำมาใช้นั้นมีความซับซ้อนหรือมีจำนวนชั้นที่ลึกมาก ซึ่งจากการทดลองในการวิจัยนี้เห็นได้อย่างชัดเจนว่ากระบวนการในการเรียนรู้เพื่อการปรับแต่งการเรียนรู้ให้กับ GoogleNet นั้นมีความยุ่งยากและซับซ้อนกว่า AlexNet มาก เนื่องมาจากจำนวนชั้นในรายละเอียดของ GoogleNet ที่มีทั้งหมด 144 ชั้น ในขณะที่ AlexNet มีจำนวนชั้นทั้งหมด 25 ชั้น

(5) งานวิจัยนี้ไม่ได้นำเสนอการศึกษาที่เป็นการนำโมเดล ResNet101 ในแบบปรับแต่งการเรียนรู้มาใช้ อันเป็นผลสืบเนื่องมาจากการอภิปรายในข้อ (4) ที่ผ่านมา เพราะ ResNet101 นั้นมีจำนวนชั้นในรายละเอียดมากถึง 347 ชั้น การทดลองเพื่อการปรับแต่งการเรียนรู้ในรูปแบบต่าง ๆ ให้ครอบคลุมเพื่อให้ได้มาซึ่งโมเดลที่ผ่านการปรับแต่งแล้วที่เหมาะสมที่สุดจะยิ่งมีความซับซ้อนกว่า GoogleNet มาก นอกจากนี้การฝึกสอนใหม่เพิ่มเติมให้กับโครงข่ายก็ต้องใช้เวลานานกว่ามาก

(6) จากผลการทดลองด้วยชุดข้อมูลที่ 1 พบว่า โมเดล ResNet101 ที่นำมาใช้แบบยึดคุณลักษณะจากตัวสกัด (เมื่อไม่มีการนำ Autoencoder มาใช้ร่วมด้วย) มีประสิทธิภาพที่ใกล้เคียงกันกับ โมเดล GoogleNet ที่เป็นแบบปรับแต่งการเรียนรู้ สำหรับการจำแนกภาพวัสดุในงานก่อสร้าง นั่นคือพบว่าเมื่อนำคุณลักษณะในแต่ละแบบดังกล่าวไปผ่านขั้นตอนการจำแนกโดยไม่ผ่านวิธีการเข้ารหัสข้อมูลก่อน ผลที่ได้จากทั้งสองวิธีดังกล่าวใกล้เคียงกัน ซึ่ง ResNet101 มีค่า Accuracy เป็น 95% ในขณะที่ GoogleNet ได้เป็น 95.5% แต่ในชุดข้อมูลที่ 2 พบว่า ResNet101 ที่นำมาใช้แบบยึดคุณลักษณะจากตัวสกัดมีประสิทธิภาพมากกว่า GoogleNet ที่เป็นแบบปรับแต่งการเรียนรู้ นั่นคือคุณลักษณะที่ได้จากโมเดล ResNet101 แบบที่ผ่านการฝึกสอนมาก่อนด้วยชุดข้อมูลภาพ ImageNet มีความเหมาะสมสำหรับการนำไปใช้จำแนกภาพวัสดุในงานก่อสร้างโดยไม่จำเป็นต้องผ่านกระบวนการปรับแต่งที่ต้องมีการฝึกสอนใหม่เพิ่มเติมให้กับโมเดล ซึ่งมีความซับซ้อนและใช้เวลานาน ดังที่ได้อภิปรายในข้อ (5)

(7) วิธีที่นำเสนอขึ้นในงานวิจัยนี้มีการนำโครงข่ายเข้ารหัสอัตโนมัติมาใช้ร่วมกันกับโมเดล ResNet101 แบบยึดคุณลักษณะจากตัวสกัด ที่ต้องมีการฝึกสอนให้กับโครงข่ายเข้ารหัสอัตโนมัติเพื่อหาโมเดลที่เหมาะสม แต่การกำหนดพารามิเตอร์สำหรับการฝึกสอนให้กับโครงข่ายเข้ารหัสอัตโนมัตินั้นไม่ได้มีความซับซ้อน เพราะจากการทดลองพบว่าการใช้จำนวนชั้นซ่อนเร้นเป็น 2 ชั้น กับ 1 ชั้นนั้นมีประสิทธิภาพที่

ใกล้เคียงกัน งานวิจัยนี้จึงเลือกใช้โครงข่ายที่เป็น 1 ชั้นซ่อนเร้นโดยใช้จำนวนนิวรอนเพียง 40 นิวรอนในชั้นซ่อนเร้นนั้นสำหรับชุดข้อมูลที่ 1 และ 45 นิวรอนสำหรับชุดข้อมูลที่ 2 ส่วนชั้นอินพุตและเอาต์พุตนั้นกำหนดเป็น 1000 นิวรอนตามจำนวนมิติของเวกเตอร์คุณลักษณะที่สกัดได้จากโมเดล ResNet101

(8) วิธีการที่เป็นการสร้างสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันขึ้นมาเองเพื่อฝึกสอนโดยตรงกับชุดข้อมูลที่ศึกษานั้นสามารถทำได้จากกรอบของเครื่องมือที่ใช้สำหรับงานวิจัยนี้ แต่ประสิทธิภาพที่ได้ยังถือว่าไม่น่าพอใจเมื่อเทียบกับวิธีการอื่น ๆ รวมทั้งต้องใช้เวลานานกว่าวิธีการอื่น ๆ ด้วย ทั้งนี้ด้วยเหตุผลหลักเพราะจำนวนข้อมูลที่ใช้สำหรับการฝึกสอนยังมีไม่มากพอที่จะทำการฝึกสอนโดยตรงให้กับโครงข่าย มีผลทำให้เกิดการ Overfit ได้ง่าย ประสิทธิภาพที่ได้จากการจำแนกข้อมูลในชุดทดสอบจึงไม่ดี

(9) จากผลการทดลองในชุดข้อมูลที่ 1 เมื่อพิจารณาผลที่ได้แบบเจาะจงกลุ่มข้อมูลพบว่าวิธีการที่นำเสนอ (นั่นคือ ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder) สามารถจำแนกภาพในกลุ่มที่เป็นคอนกรีตได้ดีที่สุดโดยมีค่า Recall ของกลุ่มคอนกรีตเป็น 99.5% แต่จำแนกภาพในกลุ่มอิฐได้ด้อยที่สุด ในขณะที่วิธีการที่เป็น ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA นั้นสามารถจำแนกภาพในกลุ่มที่เป็นไม้ได้ดีที่สุดโดยมีค่า Recall ของกลุ่มไม้เป็น 99.5% แต่จำแนกภาพในกลุ่มคอนกรีตได้ด้อยที่สุด

(10) จากผลการทดลองในชุดข้อมูลที่ 2 เมื่อพิจารณาผลที่ได้แบบเจาะจงกลุ่มของวัสดุพบว่า วิธีการที่นำเสนอ สามารถจำแนกภาพในกลุ่มที่เป็นอิฐมวลเบาคอนกรีตเซลลูล่าได้ถูกต้องทั้งหมด นั่นคือค่า Recall ของกลุ่มนี้เป็น 100% และรองลงมาคือกลุ่มคอนกรีต ในขณะที่วิธีการที่เป็น ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA นั้นสามารถจำแนกภาพในกลุ่มที่เป็นอิฐมวลเบาคอนกรีตเซลลูล่าได้ดีพอ ๆ กับกลุ่มที่เป็นไม้ โดยมีค่า Recall ของทั้งสองกลุ่มดังกล่าวเป็น 99.5% เท่ากัน

บทที่ 5
บทสรุป

งานวิจัยนี้เป็นการนำเสนอการศึกษาเกี่ยวกับการนำเทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยวิธีการที่นำเสนอนั้นเป็นการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันของ ResNet101 แบบที่ผ่านการฝึกสอนมาก่อนมาประยุกต์ใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้าง ซึ่งมีการนำเสนอการศึกษาในส่วนวิธีการอื่น ๆ เพื่อเป็นการเปรียบเทียบกันในหลากหลายรูปแบบของการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันอีกด้วย จากผลการศึกษาทั้งหมดสามารถสรุปผลการวิจัยได้ดังนี้

5.1 สรุปผลการวิจัย

วัตถุประสงค์ของการพัฒนางานวิจัยนี้คือ การนำเสนอการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกคือโครงข่ายประสาทแบบคอนโวลูชันเพื่อการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ โดยต้องการนำเสนอแนวคิดในการประยุกต์ใช้ในรูปแบบของการเรียนรู้แบบถ่ายโอน ที่เป็นการนำสถาปัตยกรรมของโครงข่ายประสาทแบบคอนโวลูชันแบบที่ผ่านการฝึกสอนมาก่อนคือโมเดล ResNet101 มาประยุกต์ใช้ เพื่อนำคุณลักษณะที่ได้จากการเรียนรู้แบบถ่ายโอนแบบยึดคุณลักษณะจากตัวสกัดด้วยโมเดล ResNet101 มาใช้ร่วมกันกับเทคนิคการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสอัตโนมัติ เพื่อสร้างรูปแบบการแทนข้อมูลที่เหมาะสม และนำเทคนิคการจำแนกข้อมูลด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มมาใช้สำหรับการเพิ่มประสิทธิภาพในการจำแนกภาพวัสดุในงานก่อสร้างแบบอัตโนมัติ ซึ่งวิธีการที่นำเสนอประกอบด้วยขั้นตอนหลัก 4 ขั้นตอนดังต่อไปนี้

(1) การ download โมเดลของ ResNet101 ที่ผ่านการฝึกสอนมาก่อนและนำเข้ามาในโปรแกรม Matlab เพื่อนำมาใช้ในลักษณะของการเรียนรู้แบบถ่ายโอน
(2) การแปลงข้อมูลภาพไปเป็นเวกเตอร์คุณลักษณะจากการใช้คุณลักษณะที่สกัดมาจากโมเดลของ ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัด
(3) การเข้ารหัสเวกเตอร์คุณลักษณะด้วยเครื่องเข้ารหัสอัตโนมัติ
(4) การจำแนกด้วยเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่ม

นอกเหนือจากวิธีการที่นำเสนอซึ่งประกอบด้วยขั้นตอนหลักข้างต้นแล้วแล้ว งานวิจัยนี้ได้ทำการศึกษาวิธีการในรูปแบบอื่น ๆ สำหรับการประยุกต์ใช้โครงข่ายประสาทแบบคอนโวลูชันเพื่อเป็นการศึกษาเปรียบเทียบในหลากหลายรูปแบบอีกด้วย โดยเมื่อนำวิธีการที่นำเสนอและวิธีการในรูปแบบอื่น ๆ มาแยกเป็นกรณีต่าง ๆ ที่ได้ทำการศึกษา สามารถสรุปได้ดังตารางที่ 5-1 ในที่นี้วิธีการต่าง ๆ เหล่านั้นแสดงตามลำดับเปอร์เซ็นต์ของค่า Accuracy ที่ได้จากการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1 โดยข้อมูลที่ใช้สำหรับงานวิจัยนี้ประกอบด้วยข้อมูล 2 ชุดคือ

ชุดข้อมูลที่ 1: เป็นข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะซึ่งประกอบด้วย 3 กลุ่มคือ อิฐ ไม้ และคอนกรีต

ชุดข้อมูลที่ 2: เป็นข้อมูลภาพที่เผยแพร่ในงานวิจัยของ DeGol และคณะ (นั่นคือชุดข้อมูลที่ 1) รวมข้อมูลภาพวัสดุอิฐมวลเบาคอนกรีตเซลลูล่า ดังนั้นจึงประกอบด้วย 4 กลุ่มคือกลุ่มคือ อิฐ ไม้ คอนกรีตและอิฐมวลเบาคอนกรีตเซลลูล่า

ส่วนตารางที่ 5-2 เป็นการสรุปวิธีการที่ใช้สำหรับการจำแนกข้อมูลในชุดข้อมูลที่ 2 โดยการแสดงผลลัพธ์จากแต่ละวิธีการเรียงตามลำดับเปอร์เซ็นต์ของค่า Accuracy เช่นเดียวกัน จากทั้งสองตาราง จะเห็นชัดเจนว่าวิธีการที่นำเสนอมีประสิทธิภาพมากที่สุดสำหรับการจำแนกภาพวัสดุในงานก่อสร้างในทั้งสองชุดข้อมูลที่ศึกษา ซึ่งวิธีการที่นำเสนอนั้นเป็นการนำสถาปัตยกรรมของโครงข่าย ResNet101 ที่ผ่านการฝึกสอนมาก่อนแล้วจากชุดข้อมูลภาพของ ImageNet มาใช้สำหรับงานวิจัยนี้ในรูปแบบของการเรียนรู้แบบถ่ายโอน นั่นคือค่าพารามิเตอร์ต่าง ๆ ที่ได้มาจากการฝึกสอนด้วยชุดข้อมูลภาพของ ImageNet ถูกถ่ายโอนมาใช้สำหรับการจำแนกภาพวัสดุในงานก่อสร้าง

ในการถ่ายโอนการเรียนรู้ (Transfer Learning) นั้น สามารถนำสถาปัตยกรรมของโครงข่ายที่ผ่านการฝึกสอนมาก่อนแล้วมาใช้ได้ในสองรูปแบบ คือแบบยึดคุณลักษณะจากตัวสกัด (Fixed Feature Extractor) และแบบการปรับแต่งการเรียนรู้ (Fine-Tune Learning) ซึ่งวิธีที่นำเสนอในงานวิจัยนี้เป็นการใช้ในแบบยึดคุณลักษณะจากตัวสกัด ที่มีความสะดวกและรวดเร็วในการนำคุณลักษณะที่สกัดออกมาจากโครงข่ายมาใช้งานต่อมากกว่าการปรับแต่งการเรียนรู้ เพราะไม่ต้องมีการฝึกสอนเพิ่มเติมให้กับโครงข่ายที่นำมาใช้ด้วยชุดข้อมูลใหม่ที่ศึกษา ซึ่งคือภาพวัสดุในงานก่อสร้างที่ศึกษาในงานวิจัยนี้ ในขณะที่การถ่ายโอนการเรียนรู้แบบการปรับแต่งการเรียนรู้นั้นจะต้องมีการฝึกสอนให้กับบางช่วงของชั้นในโครงข่ายที่นำมาใช้เพิ่มเติมด้วยชุดข้อมูลที่ศึกษาเพื่อเป็นการปรับค่าของพารามิเตอร์บางส่วนของโครงข่ายใหม่ให้มีความเหมาะสมกับชุดข้อมูลที่ศึกษามากยิ่งขึ้น ซึ่งในหลาย ๆ งานประยุกต์การปรับแต่งการเรียนรู้ก็จะมีประสิทธิภาพในการนำไปใช้งานถ้าโครงข่ายที่เลือกนำมาใช้นั้นไม่ซับซ้อนมากจนเกินไป และสามารถที่จะสร้างรูปแบบของการปรับแต่งเพื่อหาพารามิเตอร์ที่เหมาะสมที่สุดจากการฝึกสอนเพิ่มเติมได้

สำหรับงานวิจัยนี้พบว่า จากการเลือกใช้โมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัด แล้วนำคุณลักษณะที่สกัดออกไปได้นั้นมาผ่านการเข้ารหัสข้อมูลด้วยโครงข่ายเข้ารหัสอัตโนมัติแล้วใช้วิธีการจำแนกจากเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มนั้น มีประสิทธิภาพมากกว่าวิธีการในแบบการปรับแต่งการเรียนรู้ที่ได้ทำการทดลองเปรียบเทียบด้วยการใช้ GoogleNet เป็นการแสดงให้เห็นว่า ถ้าสามารถเลือกใช้โครงข่ายที่ผ่านการฝึกสอนมาก่อนที่เหมาะสมกับข้อมูลที่ต้องการศึกษา เราสามารถนำคุณลักษณะที่สกัดออกมาในแบบยึดคุณลักษณะจากตัวสกัดไปผ่านวิธีการบางอย่างเพิ่มเติมเพื่อให้สามารถเพิ่มประสิทธิภาพในการจำแนกได้ โดยไม่จำเป็นต้องไปใช้วิธีการในแบบของการปรับแต่งการเรียนรู้ ที่การปรับแต่งการเรียนรู้นั้นจะยิ่งมีความยุ่งยากถ้าโครงข่ายที่เลือกมาใช้นั้นมีความลึกมากหรือซับซ้อนมาก

ตารางที่ 5-1 สรุปวิธีการย่อย ๆ ที่ศึกษาทั้งหมดเมื่อแสดงตามลำดับของค่า Accuracy ที่ได้จากการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 1

ลำดับที่	**วิธีการ**	**Accuracy (%)**
1	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder (วิธีการที่นำเสนอ)*	*97.8*
2	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA*	*96.8*
3	GoogleNet แบบปรับแต่งการเรียนรู้ + Autoencoder	95.8
4	GoogleNet แบบปรับแต่งการเรียนรู้	95.5
5	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด*	*95.0*
6	AlexNet แบบปรับแต่งการเรียนรู้	94.5
7	GoogleNet แบบปรับแต่งการเรียนรู้ + PCA	93.6
9	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder	93.5
10	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด + PCA	92.3
11	AlexNet แบบยึดคุณลักษณะจากตัวสกัด	91.8
12	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด	91.1
13	การสร้างสถาปัตยกรรมขึ้นมาฝึกสอนเอง	70.7

ตารางที่ 5-2 สรุปวิธีการย่อย ๆ ที่ศึกษาทั้งหมดเมื่อแสดงตามลำดับของค่า Accuracy ที่ได้จากการจำแนกภาพในชุดทดสอบของชุดข้อมูลที่ 2

ลำดับที่	**วิธีการ**	**Accuracy (%)**
1	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + Autoencoder (วิธีการที่นำเสนอ)*	*98.0*
2	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด + PCA*	*97.1*
3	*ResNet101 แบบยึดคุณลักษณะจากตัวสกัด*	*96.0*
4	GoogleNet แบบปรับแต่งการเรียนรู้ + Autoencoder	94.8
5	GoogleNet แบบปรับแต่งการเรียนรู้	93.6
6	GoogleNet แบบปรับแต่งการเรียนรู้ + PCA	92.6
7	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด+ Autoencoder	92.5
8	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด+ PCA	92.0
9	GoogleNet แบบยึดคุณลักษณะจากตัวสกัด	91.6

สำหรับการศึกษาทั้งหมดในงานวิจัยนี้ค่าพารามิเตอร์สำคัญต่าง ๆ ที่ต้องใช้สำหรับโครงข่ายประสาทแบบคอนโวลูชันและโครงข่ายเครื่องเข้ารหัสอัตโนมัตินั้นค้นหามาจากการใช้สองกลยุทธ์ประกอบกันคือ การลองผิดลองถูก (Trial and Error) และกลยุทธ์แบบสุ่ม (Random) การลองผิดลองถูกนำมาใช้เพื่อหาขอบเขตที่เหมาะสมของพารามิเตอร์แบบกว้าง ๆ ส่วนกลยุทธ์แบบสุ่มนำมาใช้เพื่อค้นหาขอบเขตแบบจำกัดที่เหมาะสมหลังจากที่ได้ขอบเขตแบบกว้างจากการลองผิดลองถูกมาแล้ว สำหรับค่าพารามิเตอร์สำคัญของเครื่องเวกเตอร์เกื้อหนุนนั้น การทดลองในงานวิจัยนี้ทั้งหมดนำหลักการหาค่าแบบวิธีการ Optimization ที่เป็นค่า Default ของฟังก์ชันสำหรับเครื่องเวกเตอร์เกื้อหนุนแบบหลายกลุ่มที่มีให้ใช้งานในโปรแกรม Matlab

5.2 ข้อเสนอแนะ

จากผลจากการศึกษาที่ได้ในงานวิจัยนี้สามารถสรุปเป็นประเด็นต่าง ๆ ที่เป็นการเสนอแนะเพื่อการต่อยอดงานวิจัยได้ดังต่อไปนี้

(1) ถึงแม้ว่าจากผลการศึกษาพบว่าวิธีการที่นำเสนอซึ่งเป็นการนำโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดมาใช้ร่วมกับ Autoencoder จะมีประสิทธิภาพดีกว่าวิธีการที่เป็นการนำโมเดล ResNet101 ในแบบยึดคุณลักษณะจากตัวสกัดมาใช้ร่วมกับ PCA แต่พบว่าสองวิธีการนี้มีความสามารถในการจำแนกภาพในแต่ละกลุ่มย่อย ๆ ที่

แตกต่างกัน นั่นคือวิธีแบบที่ใช้ Autoencoder สามารถจำแนกภาพในกลุ่มของคอนกรีตได้ดีมาก ในขณะที่วิธีแบบที่ใช้ PCA สามารถจำแนกภาพในกลุ่มของไม้ได้ดีมาก ดังนั้น ด้วยความสามารถที่แตกต่างกันของทั้งสองวิธีการนี้ เราอาจจะขยายผลการศึกษาเพิ่มเติมสำหรับการนำสองวิธีการนี้มาใช้งานร่วมกันเพื่อการจำแนกภาพวัสดุในงานก่อสร้าง เช่นวิธีในรูปแบบของ Ensemble หรืออื่น ๆ

(2) วิธีการที่นำเสนอในงานวิจัยนี้ จากการทดลองพบว่ามีประสิทธิที่น่าพอใจสำหรับการใช้จำแนกภาพวัสดุในงานก่อสร้างที่ประกอบด้วย 3 กลุ่มของวัสดุในชุดข้อมูลที่ 1 และประกอบด้วย 4 กลุ่มของวัสดุในชุดข้อมูลที่ 2 ดังนั้นวิธีการที่นำเสนอขึ้นนี้สามารถนำไปปรับปรุงหรือขยายผลการศึกษาต่อสำหรับการจำแนกภาพวัสดุในงานก่อสร้างที่ประกอบด้วยจำนวนกลุ่มของวัสดุที่มากขึ้น

(3) ถึงแม้ว่าวิธีการจากการสร้างสถาปัตกรรมของโครงข่ายแบบคอนโวลูชันขึ้นมาเองเพื่อฝึกสอนโดยตรงกับชุดข้อมูลภาพวัสดุในงานก่อสร้างจากการศึกษาพบว่ามีประสิทธิภาพน้อยกว่าวิธีการอื่น ๆ แต่วิธีการดังกล่าวมีความสามารถในการจำแนกวัสดุในกลุ่มของคอนกรีตได้ดีมากเมื่อเทียบกับกลุ่มอื่น ๆ แสดงว่าการสร้างสถาปัตกรรมของโครงข่ายแบบคอนโวลูชันขึ้นมาเองเพื่อฝึกสอนโดยตรงนั้นก็มีคุณสมบัติที่น่าสนใจ ซึ่งการศึกษาต่อยอดเพิ่มเติมในส่วนของวิธีการนี้ สามารถทำได้ด้วยการเพิ่มจำนวนและความหลากหลายของข้อมูลภาพที่ใช้สำหรับการฝึกสอน อาจจะด้วยการมองหาชุดข้อมูลภาพที่มีขนาดใหญ่ขึ้นหรือสร้างข้อมูลภาพขึ้นมาเองเพิ่มขึ้น รวมทั้งอาจจะนำวิธีการสำหรับเพิ่มข้อมูลภาพจากชุดข้อมูลที่มีอยู่แล้วด้วยเทคนิคของการทำ Data Augmentation มาใช้ร่วมด้วย เป็นต้น

รายการอ้างอิง

วิสูตร จิระดำเกิง. (2554). การบริหารงานก่อสร้าง. **ปทุมธานี : วรรณกวี**.

อาทิตย์ ศรีแก้ว. (2558). ปัญญาเชิงคำนวณ. **นครราชสีมา : มหาวิทยาลัยเทคโนโลยีสุรนารี**.

Anatomylibrary. (2015). Neuron Synapse (Online). Available: http://www.anatomylibrary.us/diagram-neural-communication/neuron-synapse-2/ [2018, February 5]

Anwar, A. (2019). Difference between AlexNet, VGGNet, ResNet and Inception. Available: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96 [2019, September 15]

Bengio, Y., De Mori, R., Flammia, G., and Kompe, R. (1991). Phonetically motivated acoustic parameters for continuous speech recognition using artificial neural networks. **In Proceedings of Euro Speech'91**.

Brilakis, I.K., Soibelman, L., and Shinagawa, Y. (2006). Construction site image retrieval based on material cluster recognition. **Adv. Eng. Informat.** 20: 443-452.

Boonmarueng, B. (2019) Principal Component Analysis (PCA). Available: https://medium.com/@boonritboonmarueng/principal-component-analysis-pca-f0ef43a8c489 [2019, April 18]

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast Kernel Classifiers with Online and Active Learning. **The Journal of Machine Learning Research**. 6(Sep): 1579-1619, 2005.

Canziani, A., Paszke, A., and Culurciello, E. (2017). An Analysis of Deep Neural Network Models for Practical Applications. **CVPR 2017**.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). Semi-Supervised Learning. **MIT Press**. Cambridge, MA.

Chen, K., Lu, W., Peng, Y., Rowlinson, S., and Huang, G.Q. (2015). Bridging BIM and building: From a literature review to an integrated conceptual framework. **International Journal of Project Management**. 33: 1405-1416.

Cimpoi, M., Maji, S. and Vedaldi, A., 2015, June. Deep filter banks for texture recognition and segmentation. **In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference**. 3828-3836.

Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep big simple neural nets for handwritten digit recognition. **Neural Computation**. 22: 1-14.

Coates, A. and Ng, A. Y. (2011). The importance of encoding versus training with sparse coding and vector quantization. **In ICML'2011**.

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., and Andrew, N. (2013). Deep learning with COTS HPC systems. In S. Dasgupta and D. McAllester, editors. **Proceedings of the 30th International Conference on Machine Learning (ICML-13)**. 28 (3): 1337-1345.

DeGol, J., Golparvar-Fard, M., and D. Hoiem. D. (2016). Geometry-Informed Material Recognition. **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. 1554-1562.

DeGol, J., Golparvar-Fard, M., and Hoiem, D. (2016). Geometry-informed material recognition. **In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. 1554-1562.

Demuth, H., and Beale, M. (2004). Neural Network Toolbox for Use with MATLAB®. **The MathWorks, Inc**.

Deng, L., and Yu, D. (2014). Deep Learning: Methods and Applications. **Foundations and Trends in Signal Processing**. 7: 197-387.

Deshpande, A. (2016). The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3). Available: https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html [2018, February 5]

Dimitrov, A., and Golparvar-Fard, M. (2014). Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collection. **Advanced Engineering Informatics**. 28: 37-49.

Fredenslund, K. (2018). Computational Complexity of Neural Networks. Available: https://kasperfred.com/series/introduction-to-neural-networks/computational-complexity-of-neural-networks. [2019, September 15]

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**. 36: 193-202.

Gonzalez, R.C., Woods, R.E., and Eddins, S.L. (2004). Digital Image Processing Using MATLAB. **Pearson Education**.

Goodfellow, l., Bengio, Y., and Courville, A. (2016). Deep Learning. **MIT Press**. http://www.deeplearningbook.org.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. **Neurocomputing**. 187: 27-48.

Hagan, M.T., Demuth, H.B., Beale, M., and Jesús, O.D. (1996). Neural Network Design. **PWS Publishing Company**.

Haykin, S. (2009). Neural Networks and Learning Machines. 3rd edition, **Pearson Education, Inc**.

He, K. (2017). Learning Deep Features for Visual Recognition (Slide). **CVPR 2017 Tutorial**. Available: http://deeplearning.csail.mit.edu/cvpr2017_tutorial_kaiminghe.pdf

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition, **CVPR 2015**.

Herrero, S. (2010). Multiclass Classification using Massively Threaded Multiprocessors. (Slide). Available: https://www.slideshare.net/sergherrero/multiclass-classification-using-massively-threaded-multiprocessors.

Hinton, G. E., Osindero S., Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. **Neural Computation**. 18(7): 1527-1554.

Ilango, G. (2017). Why Deep Learning so Popular? (Online). Available: https://gogul09.github.io/software/why-deep-learning-so-popular. [2018, February 5]

Image-net. (2012). ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). Available at: http://www.image-net.org/challenges/ILSVRC/2012/ [2018, February 5]

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. **Science**. 304(5667): 78–80.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? **In ICCV'09**.

Jordan, M. I. (2004). The Kernel Trick. Available: https://people.eecs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf [2019, May 5]

Karpathy, A. (2016). Deep Learning for Computer Vision (Slide). Available: https://tensorflowkorea.files.wordpress.com/2016/09/bay-area-deep-learning-school-presentation.pdf

Karpathy, A. (2018). Transfer Learning. CS231n Convolutional Neural Networks for Visual Recognition. Available: http://cs231n.github.io/transfer-learning/ [2018, February 5]

Kopsida, M., Brilakis, I., and Vela, P.A. (2015). A Review of Automated Construction Progress Monitoring and Inspection Methods. **Proceeding of the 32nd CIB W78 Conference 2015**. 27th-29th 2015, Eindhoven, Netherlands.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. **In NIPS'2012.**

Kumar, S. (2005). Neural Networks: A Classroom Approach. **Tata McGraw-Hill Education.**

Le, Q., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J., and Ng, A. (2012). Building high-level features using large scale unsupervised learning. **In ICML'2012.**

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient based learning applied to document recognition. **Proc. IEEE**. 86(11): 2278-2324.

Li, F. F., Johnson, J. and Yeung, S. (2017). Lecture slides. Available: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf [2018, February 5]

McAndrew, A. (2004). An Introduction to Digital Image Processing with MATLAB. Lecture Note. **School of Computer Science and Mathematics, Victoria University of Technology**.

McCulloch, B. (1997). Automating field data collection in construction organizations. **In Construction Congress V: Managing Engineered Construction in Expanding Global Markets. ASCE.** 957-963.

Mulc, T. (2016). Inception modules: explained and implemented. Available: https://hacktilldawn.com/2016/09/25/inception-modules-explained-and-implemented/ [2018, February 5]

Nielsen, M. (2017). Deep Learning. Available: http://neuralnetworksanddeeplearning.com/chap6.html [2018, February 5]

Victor Powell, V. (2019) Principle Component Analysis. Available: http://setosa.io/ev/principal-component-analysis/ [2019, May 20]

Raina, R., Madhavan, A., and Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. **In L. Bottou and M. Littman, editors, Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML'09)**. 873-880.

Rashidi, A., Sigari, M. H., Maghiar, M., and Citrin, D. (2016). An analogy between various machine-learning techniques for detecting construction materials in digital images. **KSCE Journal of Civil Engineering**. 20(4): 1178-1188.

Rashidi, A., Sigari, M.H., Maghiar, M., and Citrin, D. (2016). An Analogy between Various Machine-learning Techniques for Detecting Construction Materials in Digital Images. **KSCE Journal of Civil Engineering**. 20(4): 1178-1188.

Robinson, A. J. and Fallside, F. (1991). A recurrent error propagation network speech recognition system. **Computer Speech and Language**. 5(3): 259-274.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**. 65: 386-408.

Rosenblatt, F. (1962). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. **Washington : Spartan Books**.

Rubashkin, M. (2017). Getting Started with Deep Learning. Available: https://www.kdnuggets.com/2017/03/getting-started-deep-learning.html

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). Learning representations by back-propagation errors. **In Nature**. 323: 533-536.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Learning internal representations by error propagation. **Parallel Distributed Processing**. 1(8): 318–362. MIT Press, Cambridge.

Salakhutdinov, R. and Hinton, G. (2009a). Deep Boltzmann machines. In Proceedings of the **International Conference on Artificial Intelligence and Statistics.** 5: 448–455.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. **Journal of Artificial Intelligence Research**. 4, 61-76.

Saxena, A. (2016). “Convolutional Neural Networks (CNNs): An Illustrated Explanation.” From ACM--XRDS: the ACM Magazine for Students. Available: https://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/ [2018, February 5]

Sifre, L. and Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. **In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference**. 1233-1240.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. **ICLR 2014**.

Son, H., Kim, C., Hwang, N., Kim, C., and Kang, Y. (2014). Classification of major construction materials in construction environments using ensemble classifiers. **Advanced Engineering Informatics**. 28(1): 1-10.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014a). Going deeper with convolutions. Technical report, **arXiv:1409.4842.**

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions, **CVPR 2014.**

Teizer, J. (2015). Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure construction sites. **Advanced Engineering Informatics**. 29: 225-238.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. **Springer-Verlag**.

Varma, M., and Zisserman, A. (2009). A statistical approach to material classification using image patch exemplars. **IEEE Trans. Pattern Anal. Mach. Intell**. 31: 2032-2047.

Veličković, P. (2017). Deep learning for complete beginners: convolutional neural networks with keras. Available: https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html [2018, February 5]

Weisstein, E.W. (2018). “Convolution.” From MathWorld--A Wolfram Web Resource. Available: http://mathworld.wolfram.com/Convolution.html [2018, February 5]

Widrow, B. (1987). Adaline and madaline - 1963, plenary speech. **In Proceedings of First IEEE International Conference on Neural Networks**. 1: 145-158.

Widrow, B., and Hoff, M.E. (1960). Adaptive switching circuits. **In 1960 IRE WESCON Convention Record**. 4: 96-104. IRE, New York.

Zagoruyko, S. and Komodakis, N. (2019). Wide Residual Networks. In Richard C. Wilson, Edwin R. Hancock and William A. P. Smith, editors, **Proceedings of the British Machine Vision Conference (BMVC)**. pages 87.1-87.12. BMVA Press, September 2016.

Zeiler, M., and Fergus R. (2013). Visualizing and Understanding Convolutional Networks. **ECCV 2013**. 818-833

Zhu, Z., and Brilakis, I. (2010). Concrete column recognition in images and videos. **Journal of computing in civil engineering**. 24(6): 478-487.

ภาคผนวก

บทความวิชาการที่ได้รับการตีพิมพ์ในระหว่างศึกษา

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์ในระหว่างศึกษา

Bunrit, S., Kerdprasop, N., and Kerdprasop, K. (2018). Construction Material Image Classification using Deep Learning Technique of Convolution Neural Network. **The 4th International Conference on Computer, Communication and Control Technology (I4CT)**, Krabi, Thailand, 20-22 April 2018.

Bunrit, S., Kerdprasop, N., and Kerdprasop, K. (2019) Evaluating on the Transfer Learning of CNN Architectures to a Construction Material Image Classification Task, **International Journal of Machine Learning and Computing**, vol. 9, no. 2, pp. 201-207.

Bunrit, S., Kerdprasop, N., and Kerdprasop, K. (2019). Improving the Representation of CNN Based Features by Autoencoder for a Task of Construction Material Image Classification, **International Journal of Machine Learning and Computing**, Accepted 5 January 2019.

Construction Material Image Classification Using Deep Learning Technique of Convolution Neural Network

S. Bunrit[1], N. Kerdprasop[2], and K. Kerdprasop[3]
[1, 2, 3] *School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Thailand*
sbunrit@sut.ac.th

***Abstract*— Technology progress in computer and internet incorporate to the data acquisition technology can be supported many line of works in construction management system. For automatic construction progress monitoring task, there inevitability involves the evaluating part according to the construction material usage. Thereby, automatic process for classifying the difference between each of construction material from an image is necessary in the preliminary procedure. The best of the correctness in classifying of material categories resulted for the best of the preciseness in evaluating of the progress in the further procedures. In this work, therefore, we investigate on the classification of construction material image by using the state of the art technique in deep leaning, which is convolution neural network (CNN). Our proposed work explores in diversified practical aspects of CNN applied for construction material images dataset. Both of transfer learning from CNN public pre-trained architecture and the generating of self-train CNN architecture are investigated. Experiment conducts on three prominent classes of construction material, which are brick, wood, and concrete. The result reveals the best practical scheme is the transfer learning aspect by fine-tuning of the pre-trained architecture to the studied dataset. It exposes in 94.17% of classification accuracy.**

***Index Terms*—Classification; Construction Material; Deep Learning; Convolution Neural Network (CNN).**

I. Introduction

Achievement in management and controlling of modern-day construction project require extremely in planning and handling. Entirely construction projects have the restriction in time, cost and the quality of work. Early know of any problem during the construction process, the problem can be solved in time. As a result, quickly and timely in evaluating and solving each of the construction task are very important in controlling the project status. Because construction project is the continuous process, a bit in delay may damage a lot. Automatic construction progress monitoring is, consequently, a major required task in modern construction project management.

Most of nowadays construction progress monitoring use manual assessment even for the large project. By such method, human are mainly done in monitoring and evaluating of the progress. Certainly, high degree of error and be tardy [1]. Moreover, Mccullouch [2] reported that by monitoring and evaluating of the progress manually, project manager wasted of the time around 30-50% of the whole time for recording and analyzing the site data. Such time should be used for the others more essential works. Semi-automatic or automatic in construction progress monitoring at present day, therefore, invasively explore by both researcher groups and construction investor groups.

The benefit from technology progress in computer and internet incorporate to the data acquisition technology can be supported many of construction management tasks. From surveying the data acquisition technologies used in the application of BBB (Bridging BIM and Building, BIM: Building Information Modeling) by Chen et al., [3] revealed laser scanning was used by the most of surveying applications, following by RFID and digital camera, respectively. RFID was suitable for only using with steel and prefabricated component materials because tag locations were needed to install. Whereas, digital camera was low cost and very low computing of information compared to laser scanning. It is a generalize usage technology of which may use in CCTV, smart phone, or installing on some kinds of unmanned aerial vehicles such as drone. Acquisition information from digital camera may collect in the form of image or video.

Both of the progress on computer and digital camera technology, There are tentatively need to develop the potential methods in extracting the useful information from image or video. Therefore, digital image processing and computer vision, at this moment, are the progressive research direction in architecture, engineering, construction, and facilities management (AEC/FM) [4]. Especially, in construction project management, digital image processing and computer vision can support in many of individual task. For automatic construction progress monitoring task, there inevitability involves the evaluating part according to the construction material usage. Thereby, automatic process for classifying the difference between each of construction material is necessary in the preliminary procedure. Acquisition information for the difference of each material must be done solely by camera because information from laser scanning could not indicate the difference among materials [5]. Whereas, information from RFID is very restriction from which it could be used for only some materials and need to invest more on attaching tag position. Besides, it is burdensome in managing such tag

through the construction site environment that change all the time [6].

As a result, acquisition information about the material difference from the construction site must be taken by camera. Such information is then extracting by some kind of methods in image processing, computer vision or any others. Finally, some of the suitable technique is used for classifying such extracted material information. The result from the classification step is the category of each material image patch. Such material categories will further use in identifying of the material locations in a construction site image and support for the evaluating of material usage in the task of construction progress monitoring. Consequently, the best of the correctness in classifying of material categories resulted for the best of the preciseness in evaluating of the further procedures.

In this work, therefore, we investigate on the classification of construction material image by using the state of the art technique in deep leaning, which is convolution neural network (CNN). CNN is a novel technique in machine learning that revealed remarkable results in various applications especially for image information. Nevertheless, in literature, CNN have not directly explored in diversified practical aspects for construction material images classification.

II. Literature Review

The methods used for construction image classification can be thought that it closed to or related to general material classification [7] and texture classification tasks [8][9]. Of which there exist many of outstanding methods proposed for this fields. However, surveying by Dimitrov and Golparvar-Fard [5] exposed when applied the methods used in general material or texture classification to the construction material images taken form the real construction site, the accuracy drop from 90% to 70% and may be more than this for some dataset. Because the nature of the construction materials, those are difference in detail from the general material and texture. The particularly methods suitable to the construction material dataset are, therefore, need to be specifically explored.

In literature, the former of work involved the classification of construction material image intended for the application of material image retrieval. Brilakis et al., [10] proposed automatic method for material image retrieval using content based. First, image is decomposed into color, texture, and structure features by applying a series of filters. Then, clustering is used to divide an image into each region and computed feature signature of each cluster by comparing with knowledge database. In this work, Threshold is need for dividing the intervals of feature signatures in knowledge database. The resulted materials are identified in the original image includes as the attributes before retrieving. Euclidean distance is measures for comparing with images in database. Experiments conducted on collection of 1025 images of which grouped into 20 materials. The results were evaluated on the graph of precision and recall. In 2010, Zhu and Brilakis [11] considered machine-learning techniques for identifying concrete material regions. By the method, image segmentation was first divided the construction site image into regions. Then, color and texture features were used as visual features. Such visual features were classified by Support Vector Machine (SVM) compared to Artificial Neural Network (ANN). The model was trained with 114 of image samples divided into 63 positive concrete and 51 negative concrete samples. The validation was done on 167 samples. The performance from ANN was better than SVM of which the average of precision and recall were around 80%. Rashidi et al., [4] studied an analogy between various machine-learning techniques for detecting construction material of building. The materials included concrete, red brick, and OSB (Oriented Strand Board). The classifiers they studied are Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), and SVM. This research used RGB histogram, HSV histogram, and histogram of dominant edges as the features. They studied each classifier in two-class problem classification that is target and non-target class of materials. Experiments conducted on 750 images in total. It exposed SVM with RBF kernel could classify with the best accuracy.

Ensemble classifiers were also explored for major construction materials by Son et al., [12]. They investigated the performance of six single classifiers and potential ensemble classifies on three data set of concrete, steel, and wood. In this work, voting based ensemble classifier was created by six different types which are SVM, ANN, Commercial version 4.5 (C4.5), Naïve Bayes (NB), Logistic regression (LR), and k-Nearest neighbors (KNN). Only three features from HSI color space are used. The classification results compared by measuring the accuracy, precision, sensitivity, and average score values. Their proposed ensemble classifier is significantly outperformed the single classifiers. In 2014, Dimitrov and Golparvar-Fard [5] proposed a vision-based method for material classification for an image with unknown viewpoint and site illumination conditions. They modeled the material appearance by joint probability distribution of response from a filter bank and principle HSV color values and classified by SVM. The main technical proposed in this work are: (1) a Bag of Words (BoW) pipeline for forming statistical distributions; (2) a multiple binary SVM classifier; and (3) a construction material library and validation metrics. They created a new comprehensive Construction Material Library (CML) comprised 20 major construction material types with more than 150 images per category. The evaluation performance measured by precision, recall, and accuracy values. Overall, the average accuracy is 97.1% for 200×200 pixel image patches and 90.8% for 30×30 pixel image patches.

DeGol et al., [13] investigated on how 3D geometry information could be used with 2D features for material image classification. The studied 3D geometries are surface normal, camera intrinsic and extrinsic parameters and the 2D features are texture and color. Such 2D features were extracted by RFS and MR8 filter bank, fisher vector, HSV color, and CNN feature from pre-trained VGG-M network. The classifier used a one vs. all SVM scheme. In this work, they introduced a new dataset named GeoMat which provide both image and geometry data. The experiment consisted of various combinations of 2D and 3D features. The highest overall accuracy is 73.84% when surface normal is used incorporated

to fisher vector and CNN feature. In a case that considered only 2D features, the combination of fisher vector and CNN features got the highest accuracy of 68.92%.

Almost all the mentioned literature methods for material image classification used the traditional scheme of machine learning which based on feature-based methods. By this scheme, hand-designed features must be determined and extracted in the specific way before the classification process. As a result, the automatic features extracted by some of deep learning techniques did not yet directly investigated for the construction material dataset. Although DeGol et al., [13] used CNN feature from a pre-trained CNN network in their work, such CNN feature only explored incorporated to other features in order to study about the important of 3D geometry. They did not focus the studied directly to the feature trained from CNN network for construction material dataset. In our work, in order to explore the suitable applied methods of the state of the art CNN network to the construction material dataset, we investigate on diversified practical aspects of CNN network for construction material images classification. Therefore, both of the transfer learning from CNN public pre-trained architecture and the generating of self-train CNN architecture are investigated in our proposed work.

III. Convolution Neural Network (CNN)

A. CNN Network

Deep learning technique named CNN is one kind of a particular neural network that processing the data in form of grid-like topology [14]. The name "Convolution Neural Network" tells this network use the convolution operation as the major processing operation. Emerging of CNN came from three of concepts, which are sparse interaction, parameter sharing, and equivariant representation. Such concepts transform to the network configuration depicts in Figure 1. The network consists mainly of two process; feature learning process and classification process. In feature learning process, three of principle stages are used in order to learn and extract the features from input, which are convolution stage follow by ReLU (Rectified Linear Unit) stage and pooling stage. Such many of these stages are consecutively used as layer-by-layer aimed at automatically extracting the features in deep. The features extracted from feature learning process will be further used for the classification process of the Fully Connected (FC) network as the traditional neural network manner. Finally, softmax function is applied for the last layer of the network in order to give the output in probability fashion.

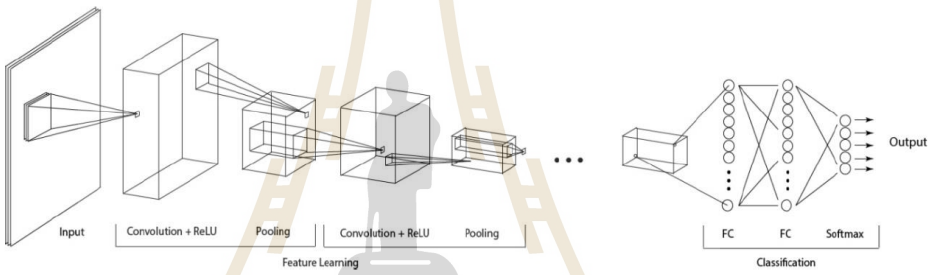


Figure 1: Configuration of Convolution Neural Network

The convolution operation used in CNN network is in the formed of 2D convolution with 3D input as shown in Figure 2. Many filters of size $k \times k \times D$ are used once at a time to convolute with the 3D input of size $W \times H \times D$ in a form of sliding windows. The convolution result from one filter is one of the feature map output. As a result, when N filters are applied, the whole output from the convolution stage is the stacks of N feature maps as depict in the right side of Figure 2. Thereby, the information in each layer of CNN feature-learning process is viewed as the features in 3D.

Since the convolution is a linear operation, the results from the convolution stage of CNN network will pass through a non-linear ReLU function in order to extract the non-linear property of the features. ReLU function is shown in Equation 1. The function converts all the negative values to zero where keeps the others as the original.

$$f(x) = \begin{cases} x, x \gg 0 \\ 0, x < 0 \end{cases} \quad (1)$$

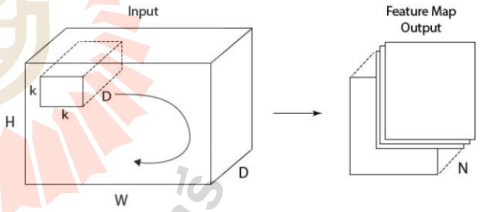


Figure 2: 2D Convolution with 3D Input using N Filters

The pooling stage of the network used for subsampling of the data. After the pooling stage, the dimension of width (W) and height (H) of the data will decrease. As shown in Figure 1, the width and height of cube in each pooling layer are smaller. Difference of pooling size can be used. If use the pooling size of 2×2, it means only one value from four values is selected as the subsampling value. Therefore, max pooling, average pooling, or any others pooling types can be applied to selected one subsampling value from such four values.

The extracted features from the last layer of the feature learning process will be arranged to vector data to be the input of the classification process. In this process, some of FC layers the same as traditional Multi-Layer Perceptron (MLP) are used. For the last layer, softmax function is applied in order to transform the output of the network to be the values in term of probability. Softmax function is shown in Equation 2.

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{j} e^{y_i}} \quad (2)$$

where: $S(y_i)$ is the softmax result of each y_i,
y_i is an output of each neuron i,
e^{y_i} is the exponential value of vector y,
and j is the component of vector y

B. *CNN Pre-Trained Architecture*

Although it was well known that the pioneer work of CNN architecture is LeNet [15], the name CNN is very popular after Krizhevsky et al., used AlexNet [16] for ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012. The architecture details of AlexNet shows in Table 1. In total, it used five of convolution layers, two of max pooling layers, two of normalization layers, and three of FC layers. Nowadays, pre-trained architecture of AlexNet using ImageNet dataset is public. Many applications applied this pre-trained architecture to their tasks. In our work, we also employ this architecture in the concept of transfer learning.

Table 1
Details of Each Layer in AlexNet [16]

Layer No.	Feature Maps	Layer Name	Details
1	[227×227×3]	INPUT	
2	[55×55×96]	CONV1	96 11×11 filters, stride 4, pad 0
3	[27×27×96]	MAX POOL1	3×3 filters, stride 2
4	[27×27×96]	NORM1	Normalization layer
5	[27×27×256]	CONV2	256 5×5 filters, stride 1, pad 2
6	[13×13×256]	MAX POOL2	3×3 filters, stride 2
7	[13×13×256]	NORM2	Normalization layer
8	[13×13×384]	CONV3	384 3×3 filters, stride 1, pad 1
9	[13×13×384]	CONV4	384 3×3 filters, stride 1, pad 1
10	[13×13×256]	CONV5	256 3×3 filters, stride 1, pad 1
11	[6×6×256]	MAX POOL3	3×3 filters, stride 2
12	[4096]	FC6	4096 neurons
13	[4096]	FC7	4096 neurons
14	[1000]	FC8	1000 neurons (class scores)

IV. Research Methodology

When we want to apply CNN network to our application domain we can do in two different ways. The first is generating the new CNN architecture appropriated to the studied dataset. The second is applying some of the pre-trained architectures trained from other dataset to the studied dataset. The second way is known in the term "transfer learning" [14]. By exploring in diversified practical aspects of CNN for classifying the construction material images. In our work, both of transfer learning from CNN public pre-trained architecture and the generating of self-train CNN architecture are investigated. Therefore, overall of our proposed methods are shown in Figure 3. Three schemes of the proposed methods are applied and compared. Scheme P1 and P2 use transfer learning from AlexNet pre-trained architecture. P1 is a fixed feature extractor scheme and P2 is a fine-tuning scheme. For P3, we generate a new CNN architecture appropriated to the studied material image dataset and name it as self-train architecture.

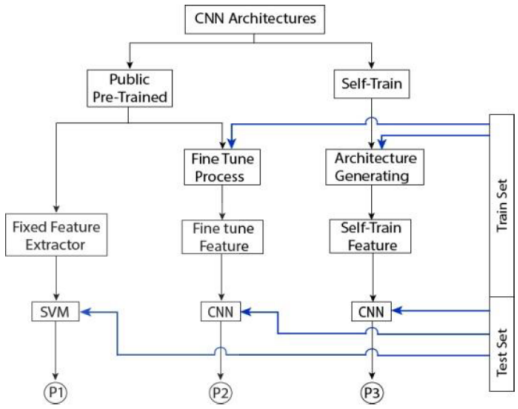


Figure 3: Proposed Methods

A. *Transfer Learning of Pre-Trained CNN Architecture*

Transfer learning is one of the very popular applied methods for CNN in practice. Because training of CNN architecture used quite a huge of computer resource. Besides, the training data should be used as much as possible. Transfer leaning of CNN can be divided into two scenarios, which are fixed feature extractor and fine-tuning. For fixed feature extractor, no need to retrain of the network. The feature trained by the original dataset of the pre-trained network will directly apply to the test set of an application task. In a case of fine-tuning, we need to fine tune in some parts of the pre-trained network. It means we need to retrain some layers of the pre-trained network by our studied dataset in order to fine-tune the network appropriated to a new dataset. In practice, always fine-tune on some of the last layers where the early layers are fixed.

For our proposed work, P1 in Figure 3 is the proposed method when fixed feature extractor from AlexNet of layer name "FC7" in Table 1 is used as the feature. The studied testing set, therefore, can directly transform to such feature and classify by SVM. In P2, some of the last layers from the pre-trained network are fine-tuning by the studied training set. In our experiments, we fine-tune of the last three layers and the last five layers of AlexNet in Table 1 for comparison. That mean if fine-tuning is done for the last three layers, the layer no. 12-14 of Table 1 will be retrained using the studied training set. Where the parameters of the remaining layers are fixed to be the values from the original pre-trained network. Result from retraining in fine tune process is the fine-tuned feature that can be used to classify the image in the test set.

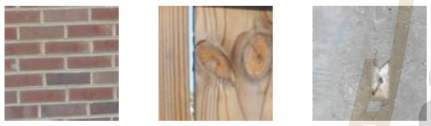
B. Generating of Self-Train CNN Architecture

In our work, we also generate a new CNN architecture directly trained on the studied training set of the construction material. Feature extracted from the training process is self-train feature and used for classifying the test set images. This scheme is proposed as P3 in Figure 3. By this approach, the appropriated hyperparameters of the generating CNN architecture as shown in Table 1 are needed to consider. Therefore, hyperparameters such as number of layers, layer ordering, filter sizes or number of filters of each layer must be identified and explored for the generating architecture.

V. Experimental Results and Discussion

A. Experimental Results

In experiment, we use three classes of the prominent materials, which are brick, wood, and concrete. There are the public images in DeGol et al., [13] work. Example of an image in each class shows in Figure 4. The left column is brick, the middle is wood, and the right is concrete, respectively. The training set consists of 400 images of each class and the testing set is 200 images each. All images are 100×100 pixels resolution.



brick wood concrete

Figure 4: Example of an image in each class

Table 2 shows the accuracy from the experimental results when all three of the proposed methods mentioned in Section IV are experimented and compared. The result reveals the best of the practical scheme is the transfer learning aspect by fine-tuning of the pre-trained architecture. It exposes in 94.17% of classification accuracy.

Table 2
Accuracy from the Experimental Results

Transfer Learning of Public Pre-Trained Architecture		Self-Train Architecture (Generating)
Fixed Feature Extractor	Fine-Tuning	
88.67 %	*94.17 %*	70.17 %

B. Discussion

Fine-tuning of the pre-trained architecture to the studied dataset that propose in P2 scheme result in the highest accuracy compared to the other practical aspects. It can be informed that features of the construction materials are quite related to the features of ImageNet dataset. Such 94.17 % is the accuracy when we fine-tune the last three layer of AlexNet. We also experiment on five last layers fine-tuning, the accuracy result is less than when the fixed feature extractor is applied in scheme P1. For the self-train architecture, if some kind of heuristic search uses to find the suitable hyperparameters, the result could be improved.

VI. Conclusion

We investigate on the classification of construction material image by using the state of the art CNN network. Both of transfer learning from CNN public pre-trained architecture and the generating of self-train CNN architecture are explored. Three schemes of practical aspects when applied CNN for construction material images dataset are proposed, experimented, and compared. Experiment conducts on three classes of material, which are brick, wood and concrete. The result reveals transfer learning by fine-tuning of the pre-trained architecture to the studied dataset expose the best in accuracy of 94.17%. The accuracy result may increase by including other hand-designed features accomplish to CNN feature for the future work.

References

[1] J. Teizer. (2015). Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure construction sites. Advanced Engineering Informatics. 29: 225-238.

[2] B. McCulloch. (1997). Automating field data collection in construction organizations. In Construction Congress V: Managing Engineered Construction in Expanding Global Markets. ASCE. 957-963.

[3] K. Chen, W. Lu, Y. Peng, S. Rowlinson, and G. Q. Huang. (2015). Bridging BIM and building: From a literature review to an integrated conceptual framework. International Journal of Project Management. 33: 1405-1416.

[4] A. Rashidi, M. H. Sigari, M. Maghiar, and D. Citrin. (2016). An analogy between various machine-learning techniques for detecting construction materials in digital images. KSCE Journal of Civil Engineering. 20(4): 1178-1188.

[5] A. Dimitrov, and M. Golparvar-Fard. (2014). Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collection. Advanced Engineering Informatics. 28: 37-49.

[6] M. Kopsida, I. Brilakis, and P. A. Vela. (2015). A Review of Automated Construction Progress Monitoring and Inspection Methods. Proceeding of the 32nd CIB W78 Conference 2015. 27th-29th 2015.

[7] M. Varma, and A. Zisserman. (2009). A statistical approach to material classification using image patch exemplars. IEEE Trans. Pattern Anal. Mach. Intell. 31: 2032-2047.

[8] L. Sifre, and S. Mallat. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference. 1233-1240.

[9] M. Cimpoi, S. Maji, and A. Vedaldi. (2015). Deep filter banks for texture recognition and segmentation. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference. 3828-3836.

[10] I.K. Brilakis, L. Soibelman, and Y. Shinagawa. (2006). Construction site image retrieval based on material cluster recognition. Adv. Eng. Informat. 20: 443-452

[11] Z. Zhu, and I. Brilakis. (2010). Concrete column recognition in images and videos. Journal of computing in civil engineering. 24(6): 478-487.

[12] H. Son, C. Kim, N. Hwang, C. Kim, and Y. Kang. (2014). Classification of major construction materials in construction environments using ensemble classifiers. Advanced Engineering Informatics. 28(1): 1-10.

[13] J. DeGol, M. Golparvar-Fard, and D. Hoiem. (2016). Geometry-Informed Material Recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1554-1562.

[14] I. Goodfellow, Y. Bengio, and A. Courville. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. (1998). Gradient based learning applied to document recognition. Proc. IEEE. 86(11): 2278-2324.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton. (2012). ImageNet classification with deep convolutional neural networks. In NIPS' 2012.

International Journal of Machine Learning and Computing, Vol. 9, No. 2, April 2019

Evaluating on the Transfer Learning of CNN Architectures to a Construction Material Image Classification Task

Supaporn Bunrit, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Various sub-tasks on modern construction management system require automatic or semi-automatic processes in handling the operation inside. Especially for construction progress monitoring task, the automatic process in classifying the difference of each construction material from an image is necessary in the preliminary stage. The more the preciseness in automatic classifying, the more the exactness in assessment of each material had been used. Subsequently, the progress of the construction can be evaluated with the highest degree of reliability. As a result, classification of construction material images is very essential process for automatic progress monitoring. Whereas, the similarities in material image appearances are the major classifying challenges. All most all existing related works have been studied based on hand-designed features of which the classified accuracy still not much appreciated from different studied datasets. In our work, automatic feature extracted method from the prominent technique in deep learning, convolution neural network (CNN), is proposed. The pre-trained CNN architectures of AlexNet and GoogleNet are adopt with the task of construction material images classification in the concept of transfer learning. Both of fixed feature extractor and fine-tuning schemes of transfer learning are technically implemented and evaluated. Analyzing results from the two pre-trained architectures expose very impressive and interesting circumstances to the studied dataset. Entirely, fine-tuning scheme of GoogleNet reveals the highest classification result by 95.50 percent of accuracy.**

***Index Terms*—Convolution neural network (CNN), deep learning, transfer learning, construction material, image classification.**

I. Introduction

At present, digital image processing and computer vision are the progressive research directions in architecture, engineering, construction, and facilities management (AEC/FM) [1]. Such directions when incorporate to the advancement of machine learning techniques can be solved for many tentative applications. In a field of construction management, automatic or semi-automatic systems for various sub-tasks are needed. Particularly for construction progress monitoring task, automatic classifying the difference between the construction materials is essential in the preliminary procedure. Where source data of construction materials must be acquired by camera in a form of image or video cause, other technologies could not indicate the difference among materials [2]. The useful information from image or video, therefore, must be extracted and identified by some efficient methods. As a result, in an application of construction progress monitoring, the classification of construction material from images must be as precise as possible. In order that the subsequent steps in evaluating the progress of the construction can perform with the highest degree of reliability.

In literatures, the methods involved construction material image classifications were studied based on hand-designed features. Where the prominent algorithms in digital image processing or computer vision were applied to extract the expected features and the suitable classifier was selected to classify such features. Therefore, the classification accuracy depends on manual selection of the feature-extracted algorithm. For our proposed work, automatic feature extraction method by a novel CNN in deep learning technique is employed. By the way, CNN based methods can separate into two scenarios, which are learning from scratch and transfer learning. In this work, we explore on the transfer learning. Two of the difference pre-trained architectures which are AlexNet [3] and GoogleNet [4] are technically transferred to a construction material image classification task. Both fixed feature extractor and fine-tuning schemes of transfer learning are evaluated from such two architectures.

II. Literature Reviews

Material images classification method initiatively studied by Brilakis *et al*., [5] in an application of material image retrieval. A series of content-based filters were employed in such work to decompose an image into color, texture, and structure features. They used knowledge database to compare the computed feature signature of each cluster after dividing an image into cluster region. The interval of each feature signature was done by threshold and comparing was measured by Euclidean distance. Zhu and Brilakis [6] also considered machine-learning techniques for identifying concrete material regions. Firstly, segmentation was applied to divide the construction site image into regions. Then, visual features from color and texture were used to classify by support vector machine (SVM) against artificial neural network (ANN). Experiment revealed the performance from ANN was better than SVM of which the average of precision and recall were around 80%. In 2016, Rashidi *et al*., [1], proposed an analogy between various machine-learning techniques for detecting construction material of building. The studied materials were concrete, red brick, and OSB (Oriented Strand Board). The comparison classifiers were multi-layer perceptron (MLP), radial basis function (RBF), and SVM. Where RGB histogram, HSV histogram, and histogram of dominant edges were extracted as the features.

Manuscript received August 25, 2018; revised November 1, 2018. This work was supported by grants from Suranaree University of Technology (SUT), Thailand.

The authors are with the School of Computer Engineering, SUT, Thailand (corresponding author: S. Bunrit; Tel.: +66944961244; e-mail: sbunrit@sut.ac.th, nittaya@sut.ac.th, kerdpras@sut.ac.th).

doi: 10.18178/ijmlc.2019.9.2.787

Experiments conducted based on two-class of problem classification; target and non-target class of materials. The best accuracy was from SVM with RBF kernel.

Son *et al*., [7] explored the performance of six classifiers and the potential of ensemble classifies on three materials, which are concrete, steel, and wood. Voting based ensemble was created by six different classifiers which are SVM, ANN, Commercial version 4.5 (C4.5), Naïve Bayes (NB), Logistic regression (LR), and k-Nearest neighbors (KNN). Three values from HSI color space are used as features. The accuracy, precision, sensitivity, and average score values were measuring and comparing. The ensemble classifier was significantly better than each single classifier. In 2014, Dimitrov and Golparvar-Fard [2] technically proposed a bag of words (BoW) pipeline for forming statistical distributions of materials and multiples of binary SVM were used as the classifiers. The material appearances were modeled by joint probability distribution of response from a filter bank and principle HSV color values. In this work, they also proposed the prototype of the construction material library and the validation metrics. 3D geometry information of materials was investigated incorporated to 2D features in a work of DeGol *et al*., [8]. Considering features of 3D geometries were surface normal, camera intrinsic, and extrinsic parameters. 2D features were fisher vector, HSV color, and CNN feature from pre-trained VGG-M network. A one vs. all SVM scheme was used as the classifier. New dataset, which provide both images and geometry data, had been public in this work. They experimented on various combinations of 2D and 3D features. The results revealed the combination of surface normal, fisher vector, and CNN feature got the highest accuracy of which 73.84%. Whereas, when considered only 2D features the best accuracy was 68.92% from fisher vector incorporated to CNN feature.

All mentioned existing methods for the classification task of material image were proposed based on hand-designed features. That means the specific ways of the extracted features must be identified before the classification process. For such methods, none of the automatic feature extracted method such as deep learning technique has been directly studied. Although DeGol *et al*., [8] used CNN feature in their work, such feature only explored incorporated to other features in order to study about the important of 3D geometry. They did not focus the studied in particular to CNN network applying for construction material dataset. For our proposed work, as a result, a new notable scenario of CNN based method which is transfer learning is applied and evaluated for material image classification task. Two of pre-trained architectures trained on ImageNet dataset (based task) are explored in order to evaluate that if there are two architectures pre-trained on the same based task, which one is suitable to our task specific (construction material dataset). We select the two distinct pre-trained architectures of which both differences in deep and in their layer details; AlexNet and GoogleNet.

III. Theories

A. Convolution Neural Networks (CNNs)

CNN is a particular neural network model of which the convolution is employed as a key operation in a network. The network for classification task can separate into feature learning process and classification process as shows in Fig. 1. In feature learning process, three principles of stage are used in order to learn and extract the features from input, which are convolution stage follow by nonlinearity stage using rectified linear unit function (ReLU) and subsampling stage name by pooling. Such many of these stages are consecutively used as layer-by-layer aimed at automatically extracting the features in deep. Therefore, each of stage may views as each of a layer in the network. The features learned by feature learning process will further send forward to the classification process where the fully connected (FC) manner as the traditional multilayer perceptron (MLP) is used. Finally, softmax function is employed for a layer before output layer in order to gain the output in a probability fashion. The layer details of a network shows in Fig. 1 is an architecture of AlexNet that won on ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. It views as an architecture that consists of eight weighted layers (when count only the layers that has weights to be adapted). If count all of the detailed layers, it can separate into 25 consecutive layers shows on the right side of Fig. 1.

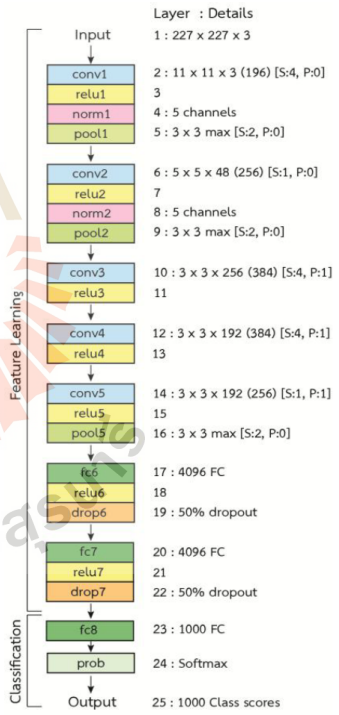


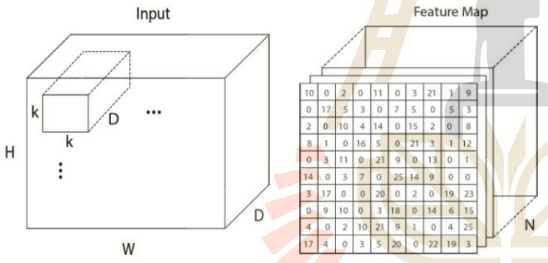
Fig. 1. CNN architecture details. Example of AlexNet [3] trained on ImageNet dataset.

Each convolution layer of CNN use many filters of size $k \times k \times D$ to convolute with the incoming input in the formed of 2D convolution with 3D input as shown in Fig. 2(a). Such filters are used once at a time to convolute with the 3D input of size $W \times H \times D$ in a form of sliding windows. Therefore, an input image of RGB color will has D equal to three from three-color channel of R, G, and B. The convolution result from one filter is one of the feature map output. As a result,

when N filters are employed, total output from a convolution stage is the stacks of N feature maps as depict Fig. 2(b). Actually the output from convolution may result in negative values, in Fig. 2(b) shows the values when ReLU function is already applied. Because convolution is a linear operation, the results from the convolution stage of CNN network will pass through a non-linear ReLU function in order to extract the non-linear property of the features. ReLU function is shown in (1). The function converts all the negative values to zero where keeps the others as the original.

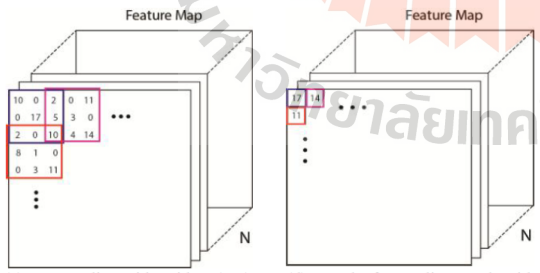
$$\phi(x) = \max(0, x) \tag{1}$$

Another key operation in CNN is pooling. These stage uses for subsampling to the previous stage data. After pooling, the dimension of width (W) and height (H) of the data will decrease. Fig. 2(c) shows an example of pooling operation when subsampling of the data by windows size of 3×3 and stride (slide to the right or to the bottom) by two positions. Its mean only one value from nine values is selected as the subsampling value. Thereby, average pooling, max pooling, or any others pooling types can be used to select one subsampling value from such nine values. Fig. 2(d) is the results after max pooling is applied to the 3×3 windows of the region marked in Fig. 2(c). After pooling stage, as a result, the width and height of the feature map are smaller.



(a). Tensor input and filters (weight)

(b). Example of feature map output after applies "conv" with N filters follows by "relu".



(c). 3×3 pooling with stride 2 (S:2).

(d). Result after applies "pool" with 3×3 max to location in figure c.

Fig. 2. The operations in major layers of CNN.

In feature learning process, many layers consisting of the convolution (conv), nonlinearity by ReLU (relu), and pooling (pool) stages are consecutively arranged to form a network. Some other stages may include such as in AlexNet shown in Fig. 1, the stage of normalization (norn) and dropout (drop) are also used.

For the classification process, some of fully connected layers (fc) are used in order to classify the extracted features from the last layer of the feature learning process. For the last layer of CNN before output, softmax function is employed to transform the output of the network to be the values in term of probability. Softmax function shows in (2).

$$S(y_i) = \frac{\exp^{y_i}}{\sum_{j=1}^{J} \exp^{y_j}} \tag{2}$$

where

(y_i) : the softmax result of each y_i,

y_i : an output of each i,

$\exp^{y_i}$: the exponential value of y_i,

and j : the component of vector y

B. CNN Based Methods

When we want to apply CNN network to our application domain we can do in two different CNN based methods show in Fig. 3. The first method knows in term "learning from scratch" when the CNN network that appropriated to the studied dataset (task specific dataset) are generated and fully train on such dataset. The second method is the transferring of knowledge (in term of weight and bias values) from some of the pre-trained architectures trained by other dataset (based task). Such knowledge from pre-trained architecture is transferred to the task specific dataset. For this way knows in the term "transfer learning" [9]. Transfer learning of knowledge can also apply by two schemes of which fixed feature extractor and fine-tuning depicted in Fig. 3. Fixed feature extractor directly uses the pre-trained weights and bias transferred to a task specific by no need to retain the network. Opposite to the fine-tuning, the network must be retrain on some parts of a network using a task specific dataset with weights and bias initialized from transferring pre-trained weights and bias.

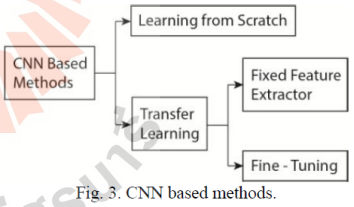


Fig. 3. CNN based methods.

C. CNN Pre-trained Architectures

The name of CNN has been well known since 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC 2012) when CNN Architecture of AlexNet [3] by Krizhevsky *et al.*, won the competition. Since then, year-by-year different deep learning architectures of CNN still were the winner of ILSVRC. There were ZF Net [10] in 2013, GoogleNet in 2014 [4], and ResNet in 2015 [11], respectively. After that, deeper and deeper architectures are proposed by the combination of GoogleNet and ResNet concepts. It has been known in machine learning community that fully trains of CNN network to a task specific dataset needs huge of computer resources and take time. Most of all, dataset size must effect to the performance. By the reasons, transfer of learning approach has been come up and many of well-known CNN pre-trained architectures are public.

In this research, two of pre-trained architectures trained on

based task of ImageNet dataset are explored. We select the two distinct pre-trained architectures of which both differences in deep and in their layer details; AlexNet and GoogleNet. Such architectures are public as the pre-trained networks with natural images in ImageNet dataset. Both were per-trained by around 1.2M images of 1000 classes of everyday used images. Architecture detailed of each explained as follow:

AlexNet

The architecture details of AlexNet already shown in Fig. 1. In total, it used five of convolution (conv) layers, two of max pooling (pool) layers, two of normalization (norm) layers, and three of fully connected (fc) layers. Nowadays, pre-trained architecture of AlexNet using ImageNet dataset is public and transfers to many application domains.

GoogleNet

In 2014, Szegegy *et al*., from Google research term developed architecture of CNN shown in Fig. 4 for ILSVRC 2014 and won the competition. The network quite deeper than AlexNet of which view as consisting of 22 weighted layers. They proposed a network under an improvement on the calculation resources. The efficient of a network came from both wider and deeper by incorporating nine modules of "inception module" on some parts of a network as shows in Fig. 4(a). Details of each inception module are in Fig. 4(b). Only small filter size of 1×1, 3×3, and 5×5 are used in the module. Each block in a module can do in parallel and the results from all blocks are concatenated to be inception module output send to the next layer.

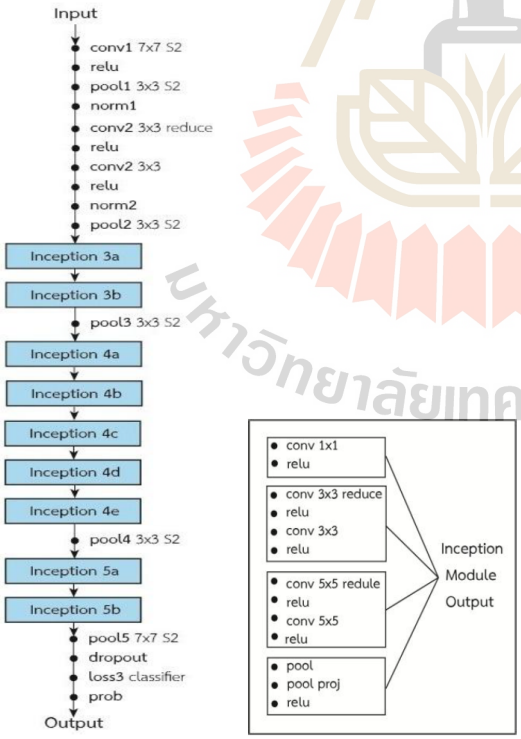


Fig. 4. Pre-trained GoogleNet [4] architecture trained on ImageNet dataset.

If the architecture of GoogleNet is pictured as AlexNet in Fig. 1 it will consist of around 144 detailed layers. Therefore, its architecture is more complex both in deep and in details.

IV. Proposed Method

A. Contributions

Our research is the first work that directly explore about transferring the pre-trained CNN architecture to a task specific on construction material images classification. Under our study, transfer-learning scenario of CNN based method is technically adopts in order to evaluate as follow:

1) If there are two of CNN architectures pre-trained on the same based-task, which one is the most suitable for transferring to our task specific?
2) For fixed featured extractor scheme, related works (from many task specifics) always explore just the feature from the last layer of feature learning. Is it can reveal any interesting result if we explore and evaluate on other learned features from the intermediated layers?
3) For fine-tuning scheme, what are the suitable important parameters selected for each pre-trained architecture? Such parameters must use to retrain in the fine-tuning process, e.g. learning rate, size of mini-batch, and number of epoch.
4) Can the transfer learning from the explored pre-trained CNN architectures achieve an attractive result to our task specific?

Under the evaluation according to our contribution, the two architectures we select to study are AlexNet depicted in Fig. 1 and GoogleNet shown in Fig. 4. These two distinct pre-trained architectures are difference in their deep and their layer details. Both pre-trained on the same based-task, which is consisting of around 1.2M images of 1000 classes from ImageNet dataset.

B. Transfer Learning by Fixed Feature Extractor

Fixed feature extractor scheme is a method that the transferred weights and bias from the pre-trained architecture are directly used for the classification process by no need to retrain the network with the training set of the task specific data set. That mean, the task specific data can directly transform to the pre-trained features and any classifier can further classify such transformed features related to the target class of the task specific dataset.

Our study task specific dataset of construction material image, there consist three classes of materials, which are brick, concrete, and wood. In order to evaluate the performance of fixed feature extractor, we use support vector machine (SVM) as a classifier. According to the contribution number two mentions previously, instead of explore only the last layer of feature learning process as done on most works, we investigate on many of intermediated featured layers.

C. Transfer Learning by Fine-Tuning

In order to fine-tuning the pre-trained architecture to a task specific dataset, we can implement by retraining some part of the pre-trained network with the training set of the task specified dataset. Fine-tuning process use the following steps:

1) Replace the output layer of a pre-trained architecture to match the number of target class exist on the studied

dataset. Therefore, fine-tuning of AlexNet and GoogleNet to our studied dataset, output layer in Fig. 1 and Fig. 4 must be changed from 1000 class scores to 3 classes scores (because our studied dataset has 3 target classes). As a result, our fine-tuning network has only three output neurons.

2) Set the initial values of all weights and bias for part of the fine-tuning network with the transferred pre-trained weights and bias.
3) Set the training parameters of CNN, which are learning rate, mini-batch size, number of epoch or number of iteration to be learned. This may include the momentum and regularization parameters.
4) Train the fine-tuning network with the training set of the task specified dataset.
5) Evaluate the fine-tuning performance by the testing set of the task specified dataset.

The suitable important parameters used in step 3 above come from the stochastic gradient descent (SGD) optimization algorithm used in CNN learning. Equation 3 [12] expresses the empirical loss with regularization term we want to minimize. Such minimization is done by the training samples $(X_i, Y_i)_{1\le i\le n}$ to estimate the parameters θ (all weights and bias).

$$L_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(f(X_i,\theta), Y_i) + \lambda\Omega(\theta) \quad (3)$$

where,

$L_n(\theta)$: the empirical loss,

$l(f(X_i,\theta), Y_i)$: the loss function,

and $\lambda\Omega(\theta)$: the regularization term

In order to minimize $L_n(\theta)$ in (3), stochastic gradient descent algorithm is used. Such algorithm performs by adapting the parameters θ as (4).

$$\theta^{k+1} = \theta^k - \varepsilon\frac{1}{m}\sum_{i\in B}[\nabla_\theta l(f(X_i,\theta), Y_i) + \lambda\nabla_\theta\Omega(\theta)] \quad (4)$$

where, k: iteration number, ε: leaning rate, m : mini-batch size, B : samples in each mini-batch, and ∇_θ: gradient of θ.

According to (4), when train CNN network, the gradient for the loss function do not compute at each iteration, but only on a set B. Where size of B equal to mini-batch size (m). This procedure called mini-batch learning, which is an approach always use in deep learning algorithms including CNN. Therefore, the important parameter in (4) needed to set is leaning rate, mini-batch size, and total numbers of epoch, where one epoch counted for a pass of all samples to the network. In our work, these parameters are observed based on empirical experiments.

V. Experimental Results and Discussions

A. Data Sets

Task specific dataset use in this study consists of three prominent classes of material image, which are brick, wood, and concrete. There are parts of the public images in a work of DeGol *et al.*, [8]. Examples of some images in each class show in Fig. 5. The left column is brick, the middle is concrete, and the right is wood, respectively. All images are 100×100 pixels resolution. The training set consists of 400 images of per class and the testing set is 200 images per class.

brick concrete wood

Fig. 5. Examples of some images in each class.

B. Experimental Results

1) Fixed feature extractor

Following our contributions mentioned in part *A* of Section IV, we set the experiments related to such contributions. Table I shows the result of fixed feature extractor from AlexNet when we investigate on many of intermediated layers. Most research of many application tasks always used the feature from layer "fc7" marked by underline. For our work, feature from such layer reveal a bit poor result compare to the others. The highest classification accuracy is form the feature in layer "pool5" with 91.83% of accuracy labeled by bold font of Table I.

TABLE I: Classification Accuracy Using Fixed Feature Extractor of **AlexNet** from Difference Layers (Layer Name Refer to Fig. 1)

Extracted Feature from Layer	Accuracy (%)	Extracted Feature from Layer	Accuracy (%)
conv4	70.83	fc6	91.17
relu4	87.33	drop6	91.17
conv5	91.00	fc7	89.67
relu5	91.50	relu7	90.67
pool5	**91.83**	drop7	90.67
fc6	91.67	fc8	87.50

Table II are the results from GoogleNet when experiment on fixed feature extractor scheme. Most research used the feature from layer "loss3" marked by underline. Nevertheless, for our task, such layer exposes very poor result compare to the layer "Inception5a" marked by bold font. Pre-trained feature from Inception5a layer get 92.33% of classification accuracy.

TABLE II: Classification Accuracy Using Fixed Feature Extractor of **GoogleNet** from Difference Layers (Layer Name Refer to Fig. 4)

Extracted Feature from Layer	Accuracy (%)	Extracted Feature from Layer	Accuracy (%)
Inception3a	79.17	Inception4e	92.00
Inception3a	79.67	pool4	91.33
pool3	87.33	**Inception5a**	**92.33**
Inception4a	86.33	Inception5b	90.17
Inception4b	89.17	pool5	91.17
Inception4c	89.50	dropout	91.17
Inception4d	92.00	loss3	86.17

2) Fine-tuning

The experiments on fine-tuning scheme are conducted based on the parameters setting for each architecture according to Table III. Such parameters are observed based on an empirical experiments follow a work in [13]. Highest classification accuracy from the test set data are got when use parameters shown in Table III. Besides, both architectures we set the momentum term to be 0.9 and the regularization term to be 0.5. As such, we are fine-tuning the network by the stochastic gradient descent with momentum (SGDM) algorithm.

TABLE III: PARAMETERS SETTING IN FINE-TUNING PROCESS OF BOTH ARCHITECTURES

Architecture	Learning rate	Mini-batch size	No. of epoch
AlexNet	0.0001	5	20
GoogleNet	0.0001	5	15

After fine-tuning, the performance of both AlexNet and GoogleNet are improved when compared to the fixed feature extractor scheme. Fig. 6 shows the results from AlexNet by using confusion matrix. In total, the test set consists of 600 images from three classes. There are 200 image or 33.33% for each class. Overall accuracy from AlexNet fine-tuning is 94.5% marked by bold font. When consider on per class classification, it can classify concrete class with the highest accuracy of which 97.5% that label by italic bold font.

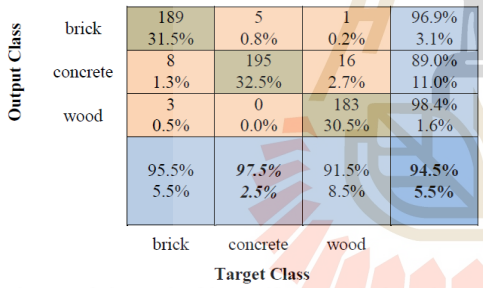


Fig. 6. Confusion matrix of the classification results from AlexNet with fine-tuning.

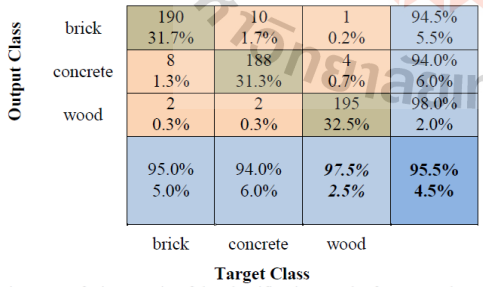


Fig. 7. Confusion matrix of the classification results from GoogleNet with fine-tuning.

Fig. 7 is the confusion matrix results from GoogleNet fine-tuning. Overall accuracy is 95.5% marked as bold font. For per class classification, class of wood can classify with the highest accuracy of which 97.5% represent as italic bold font.

Table IV shows the overall of the classification results from both schemes of transfer learning and depicts to compare for both architectures. Entirely, fine-tuning scheme of GoogleNet exposes the best classification accuracy of which 95.5%. After fine-tuning, the performance from both architectures are improved. Where, the performance from GoogleNet improves higher than AlexNet around 0.5%.

TABLE IV: OVERALL TRANSFER LEARNING RESULTS FROM BOTH PRE-TRAINED ARCHITECTURES

Architecture	Accuracy	Improvement after fine-tuning
AlexNet		
Fixed Featured Extractor	91.83	NA
Fine-tuning	94.50	2.67%
GoogleNet		
Fixed Featured Extractor	92.33	NA
Fine-tuning	**95.50**	3.17%

C. Evaluations and Discussions

When transfer-learning scenario is adopt to a task specific on construction image classification task it exhibit very interesting circumstances to the studied dataset. First, it achieves very attractive result from 95.5% of the classification accuracy by fine-tuning GoogleNet. Second, for fixed featured extractor scheme, when we investigate on the transferred features from the intermediated layers, the results from such some layers are higher than the layer always used by many existing applications. Third, based on our empirical experiment on the parameters setting for the fine-tuning process, the most performance-affected parameter is the learning rate. These means, if we set the learning rate to 0.01, the performance is much worse than 0.0001 shown in Table III. Finally, from the confusion matrix in Fig. 6, it reveals AlexNet can classify the best for concrete class. While, in Fig. 7, GoogleNet can do the best for class of wood. In addition, both architectures can do quite the same for a class of brick. By this result, it discloses us for the further study whether we can use both the extracted features from both architectures in a combination way such as ensemble or any others for the future work.

VI. CONCLUSION

In this work, a new notable scenario of CNN based method by transfer learning is applied and evaluated for construction material image classification task. Two of pre-trained architectures trained on based task of ImageNet dataset, which are AlexNet and GoogleNet are explored. Both of fixed feature extractor and fine-tuning schemes of transfer learning are technically implemented and evaluated. Analyzing results from the two pre-trained architectures expose very impressive and interesting circumstances to the studied dataset. Best of all, fine-tuning scheme of GoogleNet reveals the highest classification result by 95.50 percent of accuracy.

REFERENCES

[1] A. Rashidi, M. H. Sigari, M. Maghiar, and D. Citrin, "An analogy between various machine-learning techniques for detecting construction materials in digital images," *KSCE Journal of Civil Engineering*, vol. 20, no. 4, pp. 1178-1188, 2016.

[2] A. Dimitrov and M. Golparvar-Fard, "Vision-based material recognition for automated monitoring of construction progress and

generating building information modeling from unordered site image collection," *Advanced Engineering Informatics,* vol. 28, pp. 37-49, 2014.
[3] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS 2012*, pp. 1106-1114, 2012.
[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CVPR 2014*, 2014.
[5] I. K. Brilakis, L. Soibelman, and Y. Shinagawa, "Construction site image retrieval based on material cluster recognition," *Advanced Engineering Informatics,* vol. 20, pp. 443-452, 2006
[6] Z. Zhu and I. Brilakis, "Concrete column recognition in images and videos," *Journal of Computing in Civil Engineering*, vol. 24, no. 6, pp. 478-487, 2010.
[7] H. Son, C. Kim, N. Hwang, C. Kim, and Y. Kang, "Classification of major construction materials in construction environments using ensemble classifiers," *Advanced Engineering Informatics*, vol. 28, no. 1, pp. 1-10, 2014.
[8] J. DeGol, M. Golparvar-Fard, and D. Hoiem, "Geometry-informed material recognition," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1554-1562.
[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[10] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *ECCV 2013*, pp. 818-833, 2013.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition*, 2015.
[12] Neural networks and introduction to deep learning. [Online]. Available: https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf
[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural network?" *NIP 2014*, 2014.

S. Bunrit is a lecturer with computer engineering school, SUT. She received her bachelor degree in science (mathematics) from Kasetsart University, Thailand, in 1997, the master degree in science (computer science) from Chulalongkorn University, Thailand, in 2001. Her research of interest includes artificial neural network, deep learning, machine learning, digital image processing, computer vision, and time series analysis.

N. Kerdprasop is an associate professor with computer engineering school, SUT. She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and the doctoral degree in computer science from Nova Southeastern University, U.S.A, in 1999. Her research of interest includes data mining, artificial intelligence, and intelligent databases.

K. Kerdprasop is an associate professor and chair of the School of Computer Engineering, SUT. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and the doctoral degree in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes data mining, artificial intelligence, computational statistics.

Improving the Representation of CNN Based Features by Autoencoder for a Task of Construction Material Image Classification

Supaporn Bunrit[1], Nittaya Kerdprasop[1], and Kittisak Kerdprasop[1]
[1] School of Computer Engineering, Institute of Engineering,
Suranaree University of Technology, Nakhon Ratchasima, Thailand
Email: sbunrit@sut.ac.th; {nittaya; kerdpras}@sut.ac.th

Abstract— **Deep learning based model named Convolution Neural Network (CNN) has been extensively employed by diversified applications concerned images or videos data. Because training a specific CNN model for an application task consumes enormous machine resources and need many of the training data, consequently pre-trained models of CNN have been broadly used as the transfer-learning scenario. By the scenario, features had been learned from a pre-trained model by one source task can be proficiently sent further to another specific task in a concept of knowledge transferring. As a result, a task specific can be directly employed such pre-trained features or further train more by setting the pre-trained features as a starting point. Thereby, it takes not much time and can improve the performance from many referenced works. In this work, with a task specific on construction material images classification, we investigate on the transfer learning of GoogleNet and ResNet101 that pre-trained on ImageNet dataset (source task). By applying both of the transfer-learning schemes, they reveal quite satisfied results. The best for GoogleNet, it gets 95.50 percent of the classification accuracy by fine-tuning scheme. Where, for ResNet101, the best is of 95.00 percent by using fixed feature extractor scheme. Nevertheless, after the learning based representation methods are further employed on top of the transferred features, they expose more appeal results. By Autoencoder based representation method reveals the performance can improve more than PCA (Principal Component Analysis) in all cases. Especially, when the fixed feature extractor of ResNet101 are used as the input to Autoencoder, the classified result can be improved up to 97.83%. It can be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.**

Index Terms— **Convolution Neural Network (CNN), Transfer Learning, Autoencoder, Construction Material, Image Classification.**

I. Introduction

Since the emerging of deep learning, Convolution Neural Network (CNN) based learning has been extensively employed by diversified applications. Especially, for the tasks concerned images or videos data. Due to the constructing and learning of a specific CNN model for an application task consumes enormous machine resources and need many of the training data, consequently pre-trained models of CNN have been published and appreciation by many application domains. The features had been learned from a pre-trained model by one source task can be proficiently sent further to another specific task in a concept of transfer learning. By transfer learning, a task specific can be directly employed such pre-trained transfer features or further train more by setting the pre-trained transfer features as a starting point. Thereby, it takes not much time and can improve the performance from many referenced works.

Transfer learning of CNN model can be applied by two schemes, which are fixed feature extractor and fine-tuning. Fixed feature extractor directly transfers pre-trained features to a task specific by just project (activate) the task specified data to such features. Another one popular scheme is fine-tuning. It means the pre-trained transfer features from a source task are fine-tuned to a task specific by training more with a task specific dataset. The result features after retrain are then utilize. Naturally, fixed featured extractor can be process faster than fine-tuning, especially when the pre-trained model is very deep. The deeper of the model, the longer of the fine-tune process. In addition, fine-tuning process need to set many of hyper-parameters. Searching for such suitable and optimal hyper-parameters also take much of time and complex.

In this research, aim at looking for the best performance in construction material images classification task, the novel suitable approaches are then explored. Previous works concerned construction material image classifications were studied based on hand-designed features, of which the outstanding algorithms in image analysis were applied to extract the features and then some classifiers were selected to classify such features. Therefore, the classification accuracy depends on manual selection of the feature-

Manuscript received November 13, 2018; revised January 5, 2019; accepted January 5, 2019.
Corresponding author: Supaporn Bunrit (email: sbunrit@sut.ac.th)

extracted algorithm. In our study, the state of the art approach based on the transfer learning of CNN pre-trained models/architectures is investigated. A set of construction material images is act as a task specific dataset in the transfer-learning scenario. The two selected architectures are GoogleNet [1] and ResNet101 [2] pre-trained on a source task of ImageNet dataset. These two architectures are differences both in deep and in detailed layers.

After the preliminary experiments conduct just on the transfer-learning scenario, we encounter the cumbersome process in fine-tuning scheme. Such fine-tuning process always takes so long time and it is complicated in searching for the optimal hyper-parameters. Especially for ResNet101 which consists of up to 101 weighted layers in deep. Instead of wasting too much of time for fine-tuning, feature learning based representation methods of Autoencoder and PCA are considered in our work. Such two representation methods are applied on top of the transferred pre-trained features. By Autoencoder, the representation features learned from CNN pre-trained based features can improve the performance more than PCA in all cases.

II. Related Works

Many of construction management tasks can be supported by technology progress in computer and internet incorporate to the data acquisition technology. Surveying of the data acquisition technologies used in the construction management applications by Chen et *al.*, [3] indicated laser scanning was used by the most of surveying applications, following by RFID and digital camera, respectively. For construction management tasks involved details of the construction material, acquisition information for the difference of each material must be done solely by camera because information from laser scanning could not indicate the difference among materials [4]. Therefore, digital image processing and computer vision, at this moment, are the progressive research direction in architecture, engineering, construction, and facilities management (AEC/FM) [5].

Concerning the construction material classification, in literature Brilakis *et al.*, [6] imitatively explored the method for material images classification in an application of material image retrieval. They employed a series of content-based filters to decompose an image into color, texture, and structure features. Knowledge database was created and used for comparing the feature signature of each cluster when and an image was divided into cluster region. The interval of each feature signature was done by threshold and the comparing was measured by Euclidean distance. Machine learning techniques were considered in a work of Zhu and Brilakis [7] for identifying concrete material regions. Firstly, segmentation was applied to divide the construction site image into regions. Then, visual features from color and texture were used to classify by support vector machine (SVM) against artificial neural network (ANN). Experiment revealed the performance from ANN was better than SVM of which the average of precision and recall were around 80%. Rashidi *et al.*, [5] investigated an analogy between various machine-learning techniques for detecting construction material of building. The studied materials were concrete, red brick, and OSB (Oriented Strand Board). The studied classifiers were multi-layer perceptron (MLP), radial basis function (RBF), and SVM. Where RGB histogram, HSV histogram, and histogram of dominant edges were extracted as the features. Experiments conducted based on two-class of problem classification; target and non-target class of materials. The best accuracy was from SVM with RBF kernel.

The potential of ensemble classifies were explored by Son *et al.*, [8] They explored the performance of six classifiers on three materials, which are concrete, steel, and wood. Voting based ensemble was created by six different classifiers which are SVM, ANN, Commercial version 4.5 (C4.5), Naïve Bayes (NB), Logistic regression (LR), and k-Nearest neighbors (KNN). Features used are three values from HSI color space. The accuracy, precision, sensitivity, and average score values were measuring and comparing. The ensemble classifier was significantly better than each single classifier. Dimitrov and Golparvar-Fard [4] proposed a bag of words (BoW) pipeline for forming statistical distributions of materials and multiples of binary SVM were used as the classifiers. The material appearances were modeled by joint probability distribution of response from a filter bank and principle HSV color values. They also proposed the prototype of the construction material library and the validation metrics. In a work of DeGol *et al.*, [9] 3D geometry information of materials was investigated incorporated to 2D features. The considered features of 3D geometries were surface normal, camera intrinsic, and extrinsic parameters. The 2D features were fisher vector, HSV color, and CNN feature from pre-trained VGG-M network. A one vs. all SVM scheme was used as the classifier. New dataset, which provide both images and geometry data, had been public in this work. They experimented on various combinations of 2D and 3D features. The results revealed the combination of surface normal, fisher vector, and CNN feature got the highest accuracy. When only 2D features were considered the best accuracy was from fisher vector incorporated to CNN feature.

Related works on construction material images classification were studied based on hand-designed features. Whereas, the specific ways of the extracted features must be identified before the classification process and the classification accuracy depends on manual selection of the feature-extracted algorithm. None of the automatic feature extracted method such as deep learning technique has been focus the studied for construction material images. Although DeGol *et al.*, [9] used CNN feature in their work, such feature only explored incorporated to other features in order to study

about the important of 3D geometry. They did not focus the studied in particular to CNN network applying for construction material dataset. In our proposed work, therefore, a new notable transfer learning scenario of CNN based method with its improvement is investigated for material image classification task. Where, two of pre-trained architectures that are GoogleNet and ResNet101 trained on ImageNet dataset are employed. Moreover, Autoencoder and PCA are also applied incorporated to the pre-trained features from GoogleNet and ResNet101.

III. CNN Based Methods

A. CNN Based Model

Emerging of CNN model comes from three of concepts, which are sparse interaction, parameter sharing, and equivariant representation [10]. Such concepts transform to the network configuration demonstrates in Fig. 1. The network may view as it consists mainly of two processes that are feature learning process and classification process. In feature learning process, the principle stages, which are convolution, nonlinearity, and pooling stages incorporated to fully connected stage, are used in order extract the features in deep. Such stages are named respectively in Fig. 1 as *CONV*, *ReLU*, *POOL*, and *FC*. Where *ReLU* means the stage uses Rectified Linear Unit (ReLU) function as a nonlinearity function. In CNN model, many of these principle stages are consecutively arranged as layer-by-layer aimed at automatically learning the deep features from input. The features extracted from feature learning process will be further used for the classification process by Softmax function. It applied for the last layer of the network in order to give the output in probability manner.

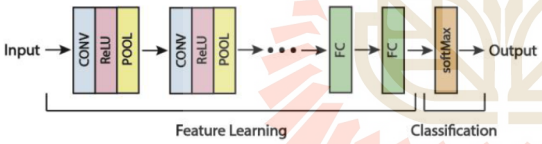


Fig. 1. Principle layers of CNN Model.

1) Convolution Layer

In CNN model, convolution is a major operation. It is in the formed of 2D convolution with 3D input, where many filters of size $k \times k \times D$ are used once at a time to convolute with the 3D input of size $W \times H \times D$ by sliding windows manner. The convolution result from one filter is one of the feature map output. Therefore, when N filters are applied in a convolution layer, the entire output from the convolution stage is a stack of N feature maps. That means the information in each convolution layer of CNN model is viewed as the features in 3D.

2) ReLU Layer

Because convolution is a linear operation, the feature maps result from the convolution layer always pass through a non-linear ReLU function in order to extract the non-linear property of the features. ReLU function is shown in (1). It simply but work very well by converting all the negative values of input to zero where keeps the others as the original.

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \gg 0 \end{cases} \quad (1)$$

3) Pooling Layer

The pooling layer of CNN model used for subsampling to each feature map input. After the pooling stage, the dimension of width (W) and height (H) of the feature map will decrease. By subsampling, therefore, difference of pooling size can be used. If use the pooling size of 2×2, it means only one value from four values is selected as the subsampling value. Whereas, max pooling, average pooling, or any others pooling types can be employed to selected the one subsampling value from such four values.

4) Fully Connected Layer

Most CNN models use some of fully connected (FC) layers at the position near output. In FC layer, the feature maps from previous layer will be arranged as a vector data to be the input of the FC layer and connect together in the same way as a Multi-Layer Perceptron (MLP) network used.

5) Softmax Layer

In feature learning process, many layers consisting of the convolution, nonlinearity by ReLU function, and pooling are consecutively arranged to form a network. Some other stages may include such as the stage of normalization or dropout that are also used in AlexNet.

For the classification process, the softMax function is employed to transform the output of the network to be the values in term of probability. Its function shows in (2).

$$softMax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}} \quad (2)$$

Where: $softMax(y_i)$ is the softmax result of each y_i,

y_i is an output of each neuron i,
e^{y_i} is the exponential value of y_i,
and k is the component of vector y

When we want to apply CNN based method to our application task we can do in two different ways. The first way knows in term *learning from scratch*. By this scheme, the CNN structure that appropriates to the studied dataset (task specific dataset) is created and fully trains on such dataset. It may works very well if our task specific dataset is large enough. The second way is *transfer learning*. Where the knowledge (in term of weight and bias parameters) from some of the pre-trained architectures trained by other dataset that big enough (source task dataset) is transferred to the task specific dataset. It has been known in machine learning community that fully trains of CNN network to a task specific dataset needs huge of computer resources and take time. Most of all, dataset size must effect to the performance. By the reasons, transfer-learning approach has been come up and many of well-known CNN pre-

trained architectures are public and appreciation by the researchers in the fields.

B. CNN Pre-Trained Models

Nowadays, many so named CNN architectures pre-trained on some source tasks are public. Most of all, architectures involved each year ILSVRC are well known. Table I shows some of ILSVRC architectures that pre-trained by ImageNet dataset. A table focuses to compare their complexity in term of depth and total number of parameters. Where, number of parameters of CNN model means total number of weights and bias employed in a network.

TABLE I: COMPARE PRE-TRAINED ARCHITECTURES DETAIL THAT USED IMAGENET DATASET AS A SOURCE TASK.

Architecture	No. Weighted Layers	No. Total Layers	No. of Parameters (million)
AlexNet	8	25	62.4
GoogleNet	22	144	6.8
ResNet50	50	177	25.6
ResNet101	101	347	44.5

For GoogleNet, Its weighted layers has around 3 times deeper than AlexNet but total number of parameters has 10 times less than. ResNet50 and ResNet101 were the two architectures based on ResNet structure. ResNet101 is deeper and as a result has more parameters. In this work, we select architectures of GoogleNet and ResNet101. These two architectures are differences both in deep and in detailed layers. They were per-trained by around 1.2 million images of 1000 classes of everyday used images. Details for each architecture explain as follow:

1) GoogleNet

Szegegy *et al.*, [1] from Google research term developed an architecture of GoogleNet for ILSVRC 2014 and won the competition. The network quite deeper than AlexNet of which view as consisting of 22 weighted layers. They proposed a network under an improvement on the calculation resources. The efficient of a network came from both wider and deeper by incorporating nine modules of *inception module*. Each module used only small filter size of 1×1, 3×3, and 5×5. Each block in a module can do in parallel and the results from all blocks are concatenated to be the inception module output send to the next layer in a network. GoogleNet used total number of parameters around ten times fewer than AlexNet as showed in Table I.

2) ResNet101

If compare by the layer in deep, ResNet101 is five times deeper than GoogleNet when count for its weighted layers. ResNet101 employed the same inside details as ResNet152 that won ILSVRC 2015. All ResNet architectures construct based on the core idea of introducing a so-called *identity shortcut connection* that skips one or more layers. Such an idea was then transformed to a *deep residual learning* framework [2].

C. Transfer Learning of CNN

Transfer learning of knowledge from CNN based models can apply by two schemes, which are *fixed feature extractor* and *fine-tuning*. Fixed feature extractor directly uses the pre-trained weights and bias and transferred to a task specific by no need to retain the network. Opposite to the fine-tuning, the network must be retrain on some parts using a task specific dataset with weights and bias initialized from transferring pre-trained weights and bias.

In practice, both fixed featured extractor and fine-tuning are popular for the image classification tasks. Due to CNN features are more generic in early layers and more original-dataset-specific in later layers, There are some common rules for navigating the following 4 scenarios [13]:

1) *Task specific dataset is small and similar to source task dataset:* employ fixed features extractor by training a linear classifier from the activation features at the top layer.

2) *Task specific dataset is large and similar to source task dataset:* employ fine-tuning through the full pre-trained network.

3) *Task specific dataset is small but very different from the source task dataset:* employ fixed features extractor by training a linear classifier from the activation features somewhere earlier in the network.

4) *Task specific dataset is large and very different from the source task dataset:* employ fine-tuning through the full pre-trained network. Actually, we can afford to train a CNN model from scratch.

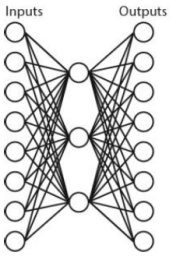
IV. FEATURE REPRESENTATION METHODS

A. Autoencoder

Autoencoders are a type of neural network architecture that take in an input vector, compress (encode) the input to a reduced set of dimensions and then reconstruct (decode) the compressed data back to its original form. Therefore, a lossy transformation is applied to the data that may be used in applications like image compression [14]. Although concept of Autoencoders is as old as neural network, it comes up of interest since the emerging of deep learning. Autoencoders of many consecutive hidden layers is named as stack-Autoencoders for deep learning.

Example of the learning by Autoencoder shows in Fig. 2. This Autoencoder is forced to encode a 8 bits input to be 3 bits by setting an output of the network the same as

an input. That mean, by Autoencoder it is forced to learn f(x) = x, where x is an input. Its weight values act as the encoding function (i. e. group of segments on the left of Fig. 2(a).) and decoding function are the weighted on the right side of Fig. 2(a). For the encoded values, just encoding weights are used. After activated such encoding weight to each input, we get the encoded 3 bits results by the values show as Hidden Values of Fig. 2(b).



(a). Autoencoder Model

Input	Hidden Values	Output
10000000	.89 .04 .08	10000000
01000000	.15 .99 .99	01000000
00100000	.01 .97 .27	00100000
00010000	.99 .97 .71	00010000
00001000	.03 .05 .02	00001000
00000100	.01 .11 .88	00000100
00000010	.80 .01 .98	00000010
00000001	.60 .94 .01	00000001

(b). Activated features learned at the hidden layer of (a).

Fig. 2. Autoencoder model and its example [15].

B. *Principal Component Analysis (PCA)*

PCA is a non-parametric method of extracting relevant information from data by reducing a complex data to a lower dimension. It is an unsupervised learning method for dimensionality-reduction.

PCA process consists the five steps are as follows [16]:

1) Subtract the mean from each of the dimensions.

2) Calculate the covariance matrix.

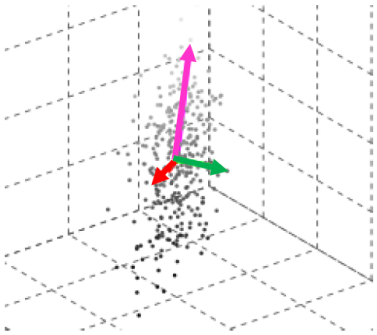
3) Calculate the eigenvectors V and eigenvalues D of the covariance matrix.

4) Reduce dimensionality and form feature vector.Where, the eigenvector with the highest eigenvalue is the principal component of the data.

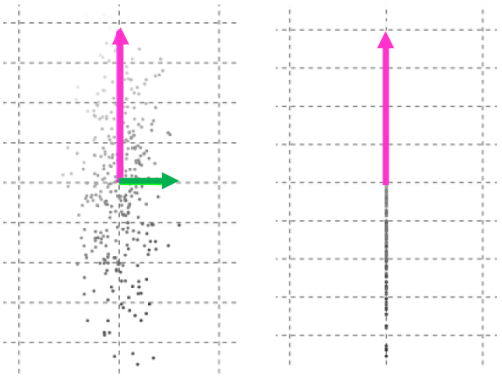
5) Derive the new data as FinalData. Where,

FinalData = RowFeatureVector ×RowZeroMeanData

Example of PCA shows in Fig. 3. The original data with 3D feature space is in Fig. 3(a). Each principal component shows by arrows represented its eigenvector. After applied PCA to this data, if select the two largest components, will get the result as in Fig. 3(b). Whereas, if select only the one largest component, result will be as shown Fig. 3(c).



(a). Original data with 3D feature space. Each principal component shows by arrows represented its eigenvector.



(b). Project the original data in (a) onto the two largest eigenvectors and obtain a 2D feature space.

(c). Project the original data in (a) onto the largest eigenvector and obtain a 1D feature space.

Fig. 3. Example of PCA. Edit from [17].

V. Proposed Works

Our proposed works can be divided into two parts. The first part, we explore on the transfer learning of GoogleNet and ResNet101 by both fixed feature extractor scheme and fine-tuning scheme. The second part, we employ Autoencoder and PCA as the feature representation methods to the transferred pre-trained feature results from the first part. Details of each part explain in Subsection A and Subsection B, respectively.

A. *Transfer Learning of GoogleNet and ResNet101*

1) *Fixed Feature Extractor*

By a scheme of fixed feature extractor, the weights and bias from the pre-trained architecture are directly transferred and used for the classification process by have no need to retrain the network with the training set of the task specific data set. Therefore, the task specific data can directly transform to the pre-trained features and any classifier can further classify such transformed features related to the target class of the task specific dataset.

Our task specific dataset is the construction material images that consists three classes of materials, which are brick, concrete, and wood. In order to evaluate the performance of fixed feature extractor by both GoogleNet and ResNet101, we use support vector machine (SVM) as a classifier. We employ the feature from layer "pool5" for GoogleNet. For ResNet101 we use the last layer of feature learning process. The last layer before softMax layer.

2) *Fine-Tuning*

By fine-tuning scheme, we observed the fine-tuning parameters based on an empirical experiments follow a work of [18]. Both architectures, we set the momentum term to be 0.9 and the regularization term to be 0.5. As such, we are fine-tuning the network by the stochastic gradient descent with momentum (SGDM) algorithm.

B. *Feature Representation of the transferred pre-trained features*

For each CNN pre-trained architecture, the feature result from each transfer-learning scheme of subsection A is used as an input to Autoencoder and PCA. Actually, we should have four Autoencoders and four PCAs. However, we experiment on only three Autoencoders and three PCAs. We do not apply the feature representation method to the fine-tuning feature of ResNet101 because we believe we still not reach the optimal fine-tuning result due to the complex in setting the hyper-parameters in the fine-tuning process.

VI. Experimental Results and Discussions

A. *Task Specific Dataset*

Our studied dataset or task specific dataset is a collection of construction material images consists of three prominent classes of materials, which are brick, wood, and concrete. There are parts of the public images in a work of DeGol et al., [9]. Examples of some images for each class show in Fig. 4. Class of brick is shown in Fig. 4(a), Concrete is Fig. 4(b), and Fig. 4(c) is wood, respectively. All images are 100×100 pixels resolution. The training set consists of 400 images per class and the testing set is 200 images per class.

(a). brick (b). concrete (c). wood

Fig. 4. Examples of some images in each class

B. *Experimental Results*

Table II shows percent of accuracy results from the transfer learning scenarios using GoogleNet and ResNet101 pre-trained architectures. The best accuracy from GoogleNet obtains when fine-tuning is applied; it is of 95.50%. By fine-tuning, the performance improves 3.33% from the fixed feature extractor scheme. In case of ResNet101, it exposes quite high in accuracy when just fixed feature extractor scheme is used; it is of 95.00%. On the other hand, by fine-tuning of ResNet101, it is difficult to tune this pre-trained model because of its depth. In total, it consists of 347 detail layers as shown in Table I. As a result, the fine-tuning process is complex in searching for the optimal hyper-parameters. Based on our experiences, the best result we get by fine-tuning is just 93.00% as shows in Table II. By the result, we expect it is not the optimal one.

TABLE II: Accuracy Results (%) from the Transfer Learning of GoogleNet and ResNet101 Architectures.

Pre-Trained Architecture	Transfer Learning Scheme		Improvement after fine-tuning
	Fixed Feature Extractor	Fine-Tuning	
GoogleNet	91.17	95.50	3.33
ResNet101	95.00	93.00 *	-2.00

Table III compares the accuracy results after the CNN based features from the transfer learning features of GoogleNet and ResNet101 are applied by PCA and Autoencoder. When the transferred fixed feature extractor of GoogleNet is employed as an input to PCA and Autoencoder, the classification results can improve to be 92.33 and 93.50, respectively. For ResNet101, PCA can improve the fixed feature extractor based feature to be 96.83%, whereas by Autoencoder it can improve up to 97.83% as shows by bold font value in Table III.

It is note that applying the feature representation methods to the transferred fine-tuning feature cannot significant improved the performance. Thus, show by the result of Table II when fine-tuning feature of GoogleNet is employed. By PCA, the result is less than when the based feature is used. Although by Autoencoder reveal a bit higher result, it is not the significant improvement. For fine-tuning feature of ResNet101, we do not conduct the experiment on feature representation methods because we believe the based fine-tuning features still not the optimal result.

TABLE III. Compare Accuracy Results (%) after Features from GoogleNet and Resnet101 are Applied by PCA and Autoencoder

Model/Representation Method	Based Features Used		Best Improvement from Based
	Fixed Feature Extractor	Fine Tuning	
GoogleNet (Based)	*(91.17)*	*(95.50)*	
PCA	92.33	93.67	
Autoencoder	93.50	95.83	2.33
ResNet101 (Based)	*(95.00)*	NA	
PCA	96.83	NA	
Autoencoder	**97.83**	NA	2.83

Fig. 5 represents the confusion matrix result from the testing set when Autoencoder applied to ResNet101 fixed feature extractor based features. The test set consists of 600 images from three classes. There are 200 image or 33.33% for each class. Target class labels in Fig. 5 is an actual class and output class is a class from the classification result of the proposed method. Overall accuracy is 97.83% (a matrix shows 97.80 by ceiling) marked by bold font. It is the highest classification result from our proposed work. When only per class classification is considered, it can classify concrete class with the highest accuracy of which 99.50% that label by italic bold font in Fig. 5. Out of 199 images from 200 images for a class of concrete can be correctly classified after Autoencoder is applied.

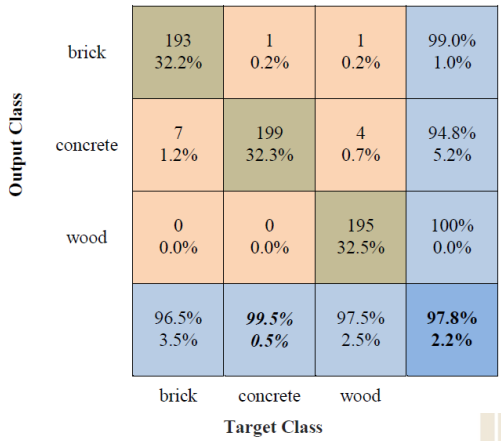


Fig. 5. Confusion matrix result when Autoencoder applied to ResNet101 fixed feature extractor.

The confusion matrix shows in Fig. 6 is the result from the testing set when PCA is applied to ResNet101 of which fixed feature extractor is used as a based feature. Overall accuracy is 96.83% (a matrix shows 96.8 by ceiling) marked as bold font. For per class classification, class of wood can classify with the highest accuracy of which 99.50% represent as italic bold font.

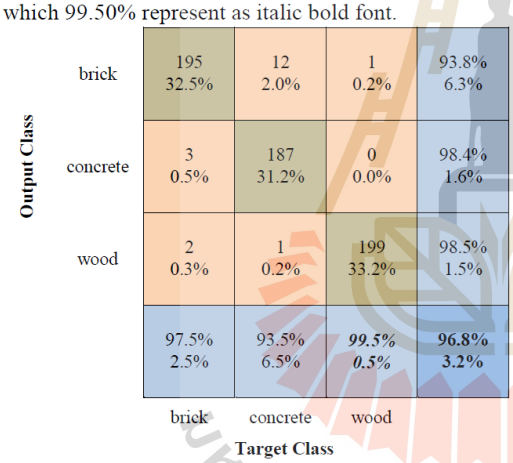


Fig. 6. Confusion matrix result when PCA applied to ResNet101 fixed feature extractor.

C. *Discussions*

Our task specific dataset is a small one. It consists of only 1,200 training images and 600 testing images. According to the common rules for navigating the 4 scenarios of transfer learning in Subsection C of Section III (CNN Based Method), just applying fixed feature extraction scheme, it should get the good result. From our experiment, according the results show in Table II. When used fixed feature extractor of ResNet101, the classification result is better than fixed feature extractor of GoogleNet. Where, GoogleNet can improve its performance by the fine-tuning scheme. In a case of ResNet101, we believe we still not reach the optimal fine-tuning result due to the complex in setting the hyper-parameters in the fine-tuning process.

By the way, the performance from fixed feature extractor of both pre-trained model can be further improved when apply Autoencoder and PCA to such feature. By Autoencoder based representation method reveals the performance can improve more than PCA in all cases as shown in Table III. The best improvement of GoogleNet is of 2.33%, whereas for ResNet101 it is of 2.83%. Best of all, Autoencoder when use ResNet101 fixed feature extractor as a based feature get the highest performance for our task specific dataset of construction material image classification. It can also be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.

From the classification results of per class classification shown by the confusion matrices; Fig. 5 and Fig. 6, we can see that Autoencoder applied to ResNet101 fixed feature extractor can classify very well for a class of concrete. Whereas, when PCA applied to ResNet101 fixed feature extractor it is good for classifying the class of wood. For our further research, we therefore, may investigate on combining of the two cases of features.

VII. Conclusion

In our study, the state of the art approach based on the transfer learning of CNN pre-trained architectures is investigated. A set of construction material images is act as a task specific dataset in the transfer-learning scenario. We investigate on the transfer learning of GoogleNet and ResNet101 that pre-trained on ImageNet dataset (source task). By applying both of the transfer-learning schemes, they reveal quite satisfied results. The best for GoogleNet, it gets 95.50 percent of the classification accuracy by fine-tuning scheme. Where, for ResNet101, the best is of 95.00 percent by using fixed feature extractor scheme. Nevertheless, after the learning based representation methods are further employed on top of the transferred features, they expose more appeal results. By Autoencoder based representation method reveals the performance can improve more than PCA (Principal Component Analysis) in all cases. Especially, when the fixed feature extractor of ResNet101 are used as the input to Autoencoder, the classified result can be improved up to 97.83%. It can be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," CVPR 2014, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Computer Vision and Pattern Recognition, 2015.

[3] K. Chen, W. Lu, Y. Peng, S. Rowlinson, and G. Q. Huang., "Bridging BIM and building: From a literature review to an integrated conceptual framework," International Journal of Project Management. vol.33, pp 1405-1416, 2015

[4] A. Dimitrov, and M. Golparvar-Fard, "Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collection," Advanced Engineering Informatics, vol. 28, pp. 37-49, 2014.
[5] A. Rashidi, M. H. Sigari, M. Maghiar, and D. Citrin, "An analogy between various machine-learning techniques for detecting construction materials in digital images," KSCE Journal of Civil Engineering, vol. 20, no. 4, pp. 1178-1188, 2016.
[6] I. K. Brilakis, L. Soibelman, and Y. Shinagawa, "Construction site image retrieval based on material cluster recognition," *Advanced Engineering Informatics,* vol. 20, pp. 443-452, 2006
[7] Z. Zhu, and I. Brilakis, "Concrete column recognition in images and videos," *Journal of computing in civil engineering*, vol. 24 no.6, pp.478-487, 2010.
[8] H. Son, C. Kim, N. Hwang, C. Kim, and Y. Kang, "Classification of major construction materials in construction environments using ensemble classifiers," Advanced Engineering Informatics. vol. 28, no.1, pp. 1-10, 2014.
[9] J. DeGol, M. Golparvar-Fard, and D. Hoiem, "Geometry-Informed Material Recognition," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1554-1562, 2016.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," NIPS 2012, pp. 1106-1114, 2012
[12] M. Zeiler, and R. Fergus R, "Visualizing and Understanding Convolutional Networks," *ECCV 2013*, pp. 818-833, 2013.
[13] Transfer Learning, http://cs231n.github.io/transfer-learning/
[14] T. M. Dahan, "PCA and Autoencoders," Technical Report. Concordia University, INSE 6220 - Fall 2017.
[15] T. M. Mitchell, Machine Learning, McGraw-Hill, 1997, ch. 4, pp. 106-108.
[16] S. Y. Elhabian and A. Farag. "A Tutorial on Data Reduction: Principal Component Analysis Theoretical Discussion," Technical Report. Computer Vision and Image Processing Laboratory, CVIP Lab, University of Louisville, September 2009.
[17] V. Spruyt. Feature extraction using PCA. Avalable at: http://www.visiondummy.com/2014/05/feature-extraction-using-pca/
[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural network?," *NIP 2014*, 2014.

S. Bunrit is a lecturer with computer engineering school, SUT. She received her bachelor degree in Science (Mathematics) from Kasetsart University, Thailand, in 1997, master degree in Science (Computer Science) from Chulalongkorn University, Thailand, in 2001. Her research of interest includes Artificial Neural Network, Deep Learning, Machine Learning, Digital Image Processing, Computer Vision, and Time Series Analysis.

N. Kerdprasop is an associate professor with computer engineering school, SUT. She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A, in 1999. Her research of interest includes Data Mining, Artificial Intelligence, and Intelligent Databases.

K. Kerdprasop is an associate professor and chair of the School of Computer Engineering, SUT. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in Computer Science from the Prince of Songkla University, Thailand,in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A., in 1999. His current research includes Data mining, Artificial Intelligence, Computational Statistics.

ประวัติผู้เขียน

นางสาวสุภาพร บุญฤทธิ์ เกิดเมื่อวันที่ 29 เดือนตุลาคม พ.ศ. 2517 ณ อำเภอพิปูน จังหวัดนครศรีธรรมราช สำเร็จการศึกษาระดับชั้นมัธยมศึกษาจากโรงเรียนกัลยาณีศรีธรรมราช อำเภอเมือง จังหวัดนครศรีธรรมราช ในปีการศึกษา 2536 ได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ และสำเร็จการศึกษาเมื่อปี พ.ศ. 2540 ภายหลังสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าศึกษาในระดับปริญญาโท สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และสำเร็จการศึกษาในปี พ.ศ. 2544 หลังจากนั้นได้ทำงานเป็นผู้ช่วย Instructor ประจำศูนย์ Kumon ก่อนจะมาเป็นอาจารย์พิเศษสอนวิชา Computer Graphics ที่สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี พ.ศ. 2549 ถึง ปี พ.ศ. 2550 และได้เป็นอาจารย์ประจำที่สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ตั้งแต่ปี พ.ศ. 2550 จนถึงปัจจุบัน

ประสบการณ์และความรู้ที่ได้จากการทำงานช่วยให้ผู้วิจัยสามารถนำมาใช้ต่อยอดสำหรับการศึกษาค้นคว้าเพิ่มเติมในระหว่างการศึกษาและได้มีผลงานตีพิมพ์เผยแพร่บทความทางวิชาการในระหว่างศึกษาจำนวน 3 เรื่อง ดังรายละเอียดในภาคผนวก