

การจำแนกภาพจากธุรกิจร้านอาหารโดยวิธี
การเรียนรู้เชิงลึก
RESTAURANT PHOTO CLASSIFICATION USING
DEEP LEARNING ALGORITHM



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

การจำแนกภาพจากธุรกิจร้านอาหารโดยวิธี
การเรียนรู้เชิงลึก
RESTAURANT PHOTO CLASSIFICATION USING
DEEP LEARNING ALGORITHM



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESTAURANT PHOTO CLASSIFICATION USING DEEP LEARNING ALGORITHM



A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR
OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ

การจำแนกภาพจากธุรกิจร้านอาหารโดยวิธีการเรียนรู้เชิงลึก
Restaurant Photo Classification using Deep Learning
Algorithm

ชื่อนักศึกษา

นาย กุลบุตร สุริยกานนท์ รหัสนักศึกษา 56050240
นางสาว โชเพีย สะรีเดชะ รหัสนักศึกษา 56050240

ปริญญา

วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา

วิทยาการคอมพิวเตอร์

ปีการศึกษา

2559

อาจารย์ที่ปรึกษา

ผศ.ธีระ ศิริธีรากล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการ
คอมพิวเตอร์) ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
ดร.อัคเดช อุดมชัยพร ประธานกรรมการ	อัคเดช
ผศ.ดร.นवलสวาท หิรัญสกุลวงศ์ กรรมการ	นवल
ผศ.ธีระ ศิริธีรากล กรรมการและอาจารย์ที่ปรึกษา	ธีระ

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การจำแนกภาพจากธุรกิจร้านอาหารโดยวิธีการเรียนรู้เชิงลึก
ชื่อนักศึกษา	นายกุลบุตร สุริยกานนท์ รหัสนักศึกษา 56050201 นางสาวโซเฟีย สระรีเดะ รหัสนักศึกษา 56050240
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ผศ.ธีระ ศิริธีรากล

บทคัดย่อ

ปัญหาพิเศษนี้มีวัตถุประสงค์เพื่อศึกษาและนำเสนอวิธีการที่เหมาะสมในการเพิ่มประสิทธิภาพในการจำแนกประเภทภาพ ด้วยเทคนิคการเรียนรู้เชิงลึก โดยมีแหล่งข้อมูลภาพจากธุรกิจร้านอาหารที่นำมาใช้ในการทดลองได้รวบรวมจากเครือข่ายสังคมออนไลน์ของวงในเป็นจำนวน 12,304 ภาพ จากการทดลองพบว่า การเลือกใช้ ADAM ซึ่งเป็นขั้นตอนวิธีการหาค่าที่เหมาะสม ให้ค่าความถูกต้องมากที่สุด บนสถาปัตยกรรม AlexNet และ GoogLeNet อีกทั้งพบว่า การนำโมเดลที่ฝึกสอนบนชุดข้อมูลจาก ILSVRC-12 และ Places365 มาต่อยอดการเรียนรู้ด้วยชุดข้อมูลที่มีอยู่ ให้ค่าความถูกต้องเท่ากับร้อยละ 88.58 และ 88.30 ตามลำดับ ซึ่งสูงกว่าโมเดลที่เรียนรู้เพียงชุดข้อมูลที่มีอยู่

คำสำคัญ : การเรียนรู้เชิงลึก โครงข่ายประสาทเทียม การจำแนกประเภทภาพ

Title	Restaurant Photo Classification using Deep Learning Algorithm
Students	Mr. Kunlabut Suriyakanon Student ID 56050201 Miss Sofia Sarideh Student ID 56050240
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2016
Advisor	Asst. Prof. Teera Siriteerakul

Abstract

For this problem, we aimed to present appropriate solutions to increase the effectiveness of image classification with deep learning techniques. The 12,304 pictures of the restaurant business from Wongnai's social network were used in the experiment, and it was noticed that choosing ADAM, which is an optimal algorithm provides the most accurate on architecture AlexNet and GoogLeNet. Further, performing models that were developed on the ILSVRC-12 and Places365 data sets come at the expense of learning with existing data sets provided the accuracy of up to 88.58 and 88.30 percent, which was higher than the models that learned only the existing data sets.

Keywords : Deep Learning, Artificial Neural Network, Image Classification

กิตติกรรมประกาศ

ปัญหาพิเศษในหัวข้อเรื่องการจำแนกประเภทภาพจากธุรกิจร้านอาหารโดยวิธีการเรียนรู้เชิงลึกเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับการเกื้อหนุนจากบุคคลหลายฝ่ายที่ให้คำปรึกษาและชี้แนะจนสามารถดำเนินงานให้บรรลุเป้าหมายได้เป็นอย่างดี

ขอขอบคุณ ผศ. ธีระ ศิริธีรารกุล อาจารย์ที่ปรึกษาปัญหาพิเศษที่ได้ให้ทั้งคำปรึกษาตลอดจนช่วยเหลือในเรื่องอุปกรณ์การปฏิบัติงาน ขอขอบคุณ ดร.อัคเดช อุดมชัยพร ประธานกรรมการสอบปัญหาพิเศษ ที่ได้คำแนะนำและชี้จุดบกพร่องที่ควรแก้ไข และขอขอบคุณ ผศ.ดร. นवलสวาท หิรัญสกลวงศ์ กรรมการสอบปัญหาพิเศษที่ได้ช่วยแนะนำให้เกิดความสมบูรณ์ให้กับปัญหาพิเศษฉบับนี้

ขอขอบคุณบริษัทวงใน มีเดียจำกัด สำหรับการเอื้อเฟื้อแหล่งข้อมูลที่นำมาใช้ในปัญหาพิเศษนี้
สุดท้ายนี้ ผู้จัดทำปัญหาพิเศษเล่มนี้ขอขอบคุณบิดา มารดา ที่ได้ให้การสนับสนุนด้านทุนการศึกษา ให้คำปรึกษาและเป็นกำลังใจที่สำคัญมาโดยตลอด ผู้จัดทำปัญหาพิเศษจึงใคร่ขอขอบคุณทุกท่านที่มีส่วนเกี่ยวข้องเป็นอย่างสูงไว้ ณ โอกาสนี้

กุลบุตร สุริยกานนท์
ไชเพ็ญ สระเเดะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของงานวิจัย.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.5 ขั้นตอนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1.1 การเรียนรู้ของเครื่องจักร.....	3
2.1.2 การเรียนรู้เชิงลึก.....	6
2.1.3 โครงข่ายประสาทเทียมแบบคอนโวลูชัน.....	7
2.1.4 กระบวนการส่งค่าย้อนกลับ.....	14
2.1.5 สถาปัตยกรรมโครงข่าย.....	16
2.1.6 เฟรมเวิร์คของการเรียนรู้เชิงลึก.....	23
2.1.7 การปรับค่าพารามิเตอร์.....	25
2.1.8 รูปแบบการวัดประสิทธิภาพ.....	29
2.1.9 ดัชนีวัดประสิทธิภาพ.....	30
2.2 ส่วนของโปรแกรมและภาษาที่นำมาใช้.....	32
2.2.1 Nvidia DIGITS.....	32
2.2.2 ส่วนเกี่ยวข้องกับโปรแกรม.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.3 งานวิจัยที่เกี่ยวข้อง.....	36
2.3.1 Using Convolutional Neural Networks and Transfer Learning to Perform Yelp Restaurant Photo Classification.....	36
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน.....	37
3.1 ขั้นตอนก่อนการประมวลผล.....	37
3.1.1 การเตรียมข้อมูล.....	38
3.1.2 การวิเคราะห์ข้อมูล.....	38
3.1.3 การกำจัดสิ่งรบกวน.....	38
3.2 ขั้นตอนการแบ่งข้อมูลด้วยวิธีการ K-Fold Cross Validation.....	38
3.3 ขั้นตอนการแปลงข้อมูลรูปภาพ.....	38
3.4 ขั้นตอนการสร้างฐานข้อมูลของชุดข้อมูล.....	39
3.5 ขั้นตอนการสร้างโมเดล.....	40
3.6 ขั้นตอนการทดลอง.....	41
3.7 ขั้นตอนการวัดประสิทธิภาพ.....	41
บทที่ 4 ผลการดำเนินงาน.....	42
4.1 ผลการทดลองก่อนและหลังการทำกระบวนการกำจัดข้อมูลรบกวนออก.....	42
4.2 ผลการทดลองการเปรียบเทียบโมเดลที่มี solver ที่แตกต่างกัน.....	43
4.3 ผลการทดลองการเปรียบเทียบระหว่างโมเดลที่ฝึกสอนบนชุดข้อมูลและโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่บน AlexNet และ GoogLeNet.....	46
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	60
5.1 สรุปผลการวิจัย.....	60
5.2 ข้อจำกัด.....	60
5.3 ข้อเสนอแนะ.....	61
5.4 ปัญหาและอุปสรรค.....	61
เอกสารอ้างอิง.....	62
ภาคผนวก.....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 ค่าความถูกต้องและค่าเฉลี่ย F-score โดยเปรียบเทียบ solver บนสถาปัตยกรรม AlexNet และ GoogLeNet.....	45
4.2 ค่าความถูกต้องและค่าเฉลี่ย F-score โดยเปรียบเทียบระหว่างโมเดลที่ฝึกสอน บนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูล ที่มีอยู่บนสถาปัตยกรรม AlexNet และ GoogLeNet ในแต่ละรอบ	47



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ขั้นตอนการเรียนรู้ของแมชชีนเลิร์นนิง.....	3
2.2 การทำงานของโครงข่ายการเรียนรู้เชิงลึกและไม่เชิงลึก.....	6
2.3 ตัวแบบของนิเวรอนในคอมพิวเตอร์.....	7
2.4 โครงข่ายแบบปกติและแบบคอนโวลูชัน.....	8
2.5 การทำคอนโวลูชัน.....	8
2.6 รูปแบบการทำงานของคุณสมบัติ local connectivity.....	9
2.7 รูปแบบการทำงานของคุณสมบัติ local connectivity ใน 3 channel.....	9
2.8 คอนโวลูชันที่มี padding.....	10
2.9 แนวพื้นที่การแบ่งพารามิเตอร์ที่ใช้ร่วมกัน.....	11
2.10 ขั้นตอนการทำ max pooling.....	11
2.11 ภาพเปรียบเทียบระหว่าง average pooling และ max pooling.....	12
2.12 กราฟฟังก์ชัน ReLU.....	12
2.13 รูปแบบการเชื่อมโยงในชั้น Fully-connected.....	13
2.14 การทำงานของโครงข่ายก่อนและหลังการใช้ Dropout.....	13
2.15 การทำงานของขั้นตอนการแพร่ข้อมูลไปข้างหน้า.....	15
2.16 แสดงการทำงานของขั้นตอนการแพร่ข้อมูลแบบย้อนกลับ.....	15
2.17 โครงสร้างสถาปัตยกรรมแบบ AlexNet.....	17
2.18 LRN แบบ within channel และแบบ cross channel.....	17
2.19 โครงสร้างการทำงานของสถาปัตยกรรม AlexNet.....	19
2.20 การทำงานของสถาปัตยกรรม GoogLeNet แบบ inception module.....	22
2.21 การทำงานแบบ inception module.....	22
2.22 inception module แบบดั้งเดิม.....	23
2.23 กราฟแสดงการทำงานของ TensorFlow.....	25
2.24 รูปแบบไฮเปอร์พารามิเตอร์ใน learning rate policy.....	28
2.25 การทำงานของ K-Fold Cross Validation.....	29
2.26 ตาราง confusion matrix.....	30
2.27 ตัวอย่างโปรแกรม Nvidia DIGITS.....	33
2.28 กราฟเปรียบเทียบประสิทธิภาพการทำงานเมื่อใช้ cuDNN.....	34
2.29 การทำงานของการจัดเก็บข้อมูล.....	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

2.30 ภาพตัวแทนของแต่ละคลาสของ YelpNet.....	36
3.1 ขั้นตอนวิธีการดำเนินงาน.....	37
3.2 หน้าต่างโปรแกรม Nvidia DIGITS ที่ใช้ในการแปลงรูปภาพ.....	39
4.1 ค่าของความถูกต้องก่อนและหลังการทำกระบวนการกำจัดข้อมูลรบกวนออก.....	42
4.2 กราฟแสดงการเปรียบเทียบค่าความถูกต้องของ solver ที่แตกต่างกันบนสถาปัตยกรรม AlexNet และ GoogLeNet.....	43
4.3 กราฟแสดงการเปรียบเทียบค่าความถูกต้องเฉลี่ยและ error bar เป็นส่วนเบี่ยงเบนมาตรฐานของข้อมูลของ solver ที่แตกต่างกันบน AlexNet และ GoogLeNet.....	44
4.4 กราฟแสดงค่าความถูกต้องโดยเปรียบเทียบระหว่างโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่บนสถาปัตยกรรม AlexNet และ GoogLeNet.....	46
4.5 กราฟแสดงการเปรียบเทียบค่าความถูกต้องเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่ AlexNet และ GoogLeNet.....	46
4.6 พีเจอรืในชั้น convolution ที่ 1 บนสถาปัตยกรรม AlexNet.....	48
4.7 พีเจอรืในชั้น convolution ที่ 1 บนสถาปัตยกรรม GoogLeNet.....	49
4.8 Top-9 ของคลาสบรรยากาศ.....	50
4.9 Top-9 ของคลาสอาหารหรือเครื่องดื่ม.....	50
4.10 Top-9 ของคลาสหน้าร้าน.....	51
4.11 Top-9 ของคลาสป้ายหน้าร้านและสมุดเมนู.....	51
4.12 Top-9 ของคลาสโปสเตอร์.....	52
4.13 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารที่ยังไม่ได้รับการฝึกสอน.....	52
4.14 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารที่ยังไม่ได้รับการฝึกสอน.....	53
4.15 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารที่ยังไม่ได้รับการฝึกสอน.....	53
4.16 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารที่ยังไม่ได้รับการฝึกสอน.....	53
4.17 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารที่ยังไม่ได้รับการฝึกสอน.....	54
4.18 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน.....	54
4.19 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน.....	55
4.20 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน.....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

4.21 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน.....	55
4.22 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน.....	56
4.23 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศที่ยังไม่ได้รับการฝึกสอน.....	56
4.24 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศยังไม่ได้รับการฝึกสอน.....	57
4.25 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศที่ยังไม่ได้รับการฝึกสอน.....	57
4.26 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศที่ยังไม่ได้รับการฝึกสอน.....	57
4.27 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสสมุดเมนูที่ยังไม่ได้รับการฝึกสอน.....	58
4.28 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับโปสเตอร์ที่ยังไม่ได้รับการฝึกสอน.....	59
1 หน้าต่างเว็บไซต์ดาวโหลดไดรเวอร์การ์ดจอ Nvidia	65
2 หน้าต่างเว็บไซต์ดาวโหลดไดรเวอร์การ์ดจอ Nvidia ในเวอร์ชันที่ต้องการ.....	66
3 หน้าต่างโปรแกรม NVIDIA X Server Settings.....	66
4 หน้าต่างเว็บไซต์ดาวโหลด CUDA เวอร์ชัน 8.....	67
5 ผลการรันชุดคำสั่งบนเทอร์มินอล.....	67
6 ผลการรันชุดคำสั่งบนเทอร์มินอล.....	68
7 หน้าต่างเว็บไซต์ดาวโหลด cuDNN เวอร์ชัน 5.1.....	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอินเทอร์เน็ตมีการใช้อย่างแพร่หลายเนื่องจากอุปกรณ์พกพาอย่างเช่น โทรศัพท์มือถือมีราคาที่ถูกลงจึงทำให้ผู้ใช้เกิดการอัปโหลดข้อมูลขึ้นบนเครือข่ายอินเทอร์เน็ตได้ง่าย และเกิดข้อมูลจำนวนมากมหาศาลอยู่ในระบบ การเรียกใช้ข้อมูลนั้นทำได้ยากเนื่องจากข้อมูลยังไม่ถูกจัดกลุ่มหรือไม่มีรูปแบบ ทำให้ต้องสร้างกระบวนการในการวิเคราะห์ข้อมูลเพื่อนำข้อมูลนั้นไปใช้งาน

จากปัญหาพิเศษที่มีชุดข้อมูลรูปภาพจากร้านอาหารที่เกิดจากเครือข่ายสังคมออนไลน์ที่มีการแบ่งปันข้อมูลในแอปพลิเคชันวงในซึ่งมีปริมาณมาก ซึ่งข้อมูลส่วนใหญ่ถูกจำแนกประเภทออกเป็น 5 คลาสประกอบไปด้วย คลาสอาหารหรือเครื่องดื่ม คลาสบรรยากาศ คลาสหน้าร้าน คลาสสมุดเมนูและป้ายหน้าร้าน และคลาสโปรไฟล์ และมีชุดรูปภาพบางส่วนที่ยังขาดการจำแนกประเภท ทำให้ต้องหาวิธีการแก้ปัญหานั้นซึ่งผู้จัดทำได้เลือกใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งเทคนิคนี้กลับมาเป็นที่นิยมอีกครั้งเนื่องจากในปัจจุบันข้อมูลมีเป็นจำนวนมากและฮาร์ดแวร์มีประสิทธิภาพเชิงคำนวณและการจัดเก็บข้อมูลที่ดีขึ้น

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อศึกษาหาวิธีที่ดีที่สุดของการเรียนรู้เชิงลึกในการจำแนกประเภทภาพจากรูกร้านอาหาร

1.3 ขอบเขตของงานวิจัย

- 1) ใช้วิธีการจำแนกรูปภาพจากรูกร้านอาหารด้วยวิธีการเรียนรู้เชิงลึก
- 2) ข้อมูลที่นำมาฝึกสอนแบบจำลองเป็นรูปภาพจากการรีวิวร้านอาหาร เช่น ภาพอาหารหรือเครื่องดื่ม ภาพบรรยากาศร้าน ภาพหน้าร้าน ภาพป้ายและสมุดเมนู และภาพโปรไฟล์
- 3) รูปภาพที่ใช้มีแหล่งที่มาจากรูปภาพของบริษัท วงใน มีเดีย จำกัด

1.4 ประโยชน์ที่คาดว่าจะได้รับ

เพื่อเป็นแนวทางในการศึกษาค้นคว้าแก่ผู้ที่สนใจงานด้านการเรียนรู้เชิงลึกและสามารถนำไปพัฒนาต่อยอดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนการดำเนินงาน

- 1) กำหนดปัญหาและศึกษาความเป็นไปได้
- 2) รวบรวมและคัดกรองข้อมูล
- 3) พัฒนาโดยใช้วิธีการเรียนรู้เชิงลึก
- 4) วิเคราะห์และทดสอบระบบ
- 5) จัดทำเอกสาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในอดีตการเขียนโปรแกรมใดต้องมีการเขียนขั้นตอนวิธีการอย่างชัดเจน ซึ่งการทำเช่นนั้นจะเหมาะกับงานคำนวณทั่วไปจนถึงงานที่ใช้คณิตศาสตร์ขั้นสูง โดยสามารถทำงานซ้ำๆได้อย่างมีประสิทธิภาพ ซึ่งมีความแม่นยำและรวดเร็วกว่าเมื่อเทียบกับมนุษย์ แต่ในงานด้านคอมพิวเตอร์วิทัศน์ เช่น การตรวจจับวัตถุในภาพ หรือการจำแนกประเภทภาพ เป็นงานที่มนุษย์สามารถทำได้ง่ายกว่าและรวดเร็วกว่า แต่กลับเป็นเรื่องที่ยากสำหรับคอมพิวเตอร์

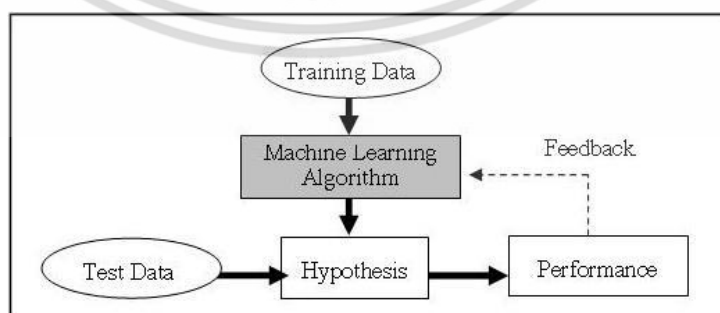
ดังนั้นจึงเกิดความพยายามในการพัฒนาให้คอมพิวเตอร์สามารถทำงานให้ได้เหมือนมนุษย์ โดยวิธีการคือการสอนให้คอมพิวเตอร์เข้าใจข้อมูลอย่างที่มีมนุษย์เข้าใจ แล้วให้คอมพิวเตอร์สร้างชุดโมเดลให้เหมาะกับงานนั้น

คณะผู้วิจัยจึงได้ศึกษาเอกสารต่างๆ และผลงานวิจัยที่เกี่ยวข้องและได้เรียบเรียงแบ่งสาระสำคัญตามลำดับดังนี้

2.1 ทฤษฎีบทและงานวิจัยที่เกี่ยวข้อง

2.1.1 การเรียนรู้ของเครื่องจักร

การเรียนรู้ของเครื่องจักร (Machine Learning) เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่พัฒนาจากการศึกษาการรู้จำแบบ (pattern recognition) และทฤษฎีการเรียนรู้เชิงคณนา (computational learning theory) และอาเธอร์ ซามูเอล ได้ให้นิยามว่า เป็นสาขาที่ให้คอมพิวเตอร์มีความสามารถในการเรียนรู้โดยปราศจากการโปรแกรมให้ชัดเจน นอกจากนี้ ทอม เอ็ม. มิตเชลล์ ได้ขยายนิยามอย่างเป็นทางการไว้ว่า เราจะเรียกคอมพิวเตอร์โปรแกรมว่าได้เรียนรู้จากประสบการณ์ E เพื่อทำงาน T ได้โดยมีประสิทธิภาพ P เมื่อโปรแกรมนั้นสามารถทำงาน T ที่วัดผลด้วย P แล้วพัฒนาขึ้นจากประสบการณ์ E



รูปที่ 2.1 ขั้นตอนการเรียนรู้ของแมชชีนเลิร์นนิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แมชชีนเลิร์นนิงมีส่วนที่คล้ายกับศาสตร์ในวิชาสถิติการคำนวณ เนื่องจากทั้งสองสาขาศึกษา การวิเคราะห์ข้อมูลเพื่อการทำนายเช่นกัน นอกจากนี้ยังรับทฤษฎีและวิธีการในการหาค่าเหมาะสมที่สุดในทางคณิตศาสตร์มาประยุกต์ใช้อีกด้วย การเรียนรู้ของเครื่องสามารถนำไปประยุกต์ใช้งานได้ หลากหลาย ตัวอย่างเช่น การกรองอีเมลขยะ การรู้จำตัวอักษร เครื่องมือค้นหา และคอมพิวเตอร์วิทัศน์ เป็นต้น

ประเภทของขั้นตอนวิธีการในการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่องสามารถแบ่งได้เป็น 3 ประเภทหลักๆ ตามข้อมูลขาเข้าได้ดังนี้

- 1) การเรียนรู้แบบมีผู้สอน (supervised learning) – เป็นการเรียนรู้แบบที่มีการตรวจคำตอบเพื่อให้แบบจำลองปรับตัว ชุดข้อมูลที่ใช้สอนแบบจำลองจะมีคำตอบไว้คอยตรวจสอบว่าแบบจำลองให้คำตอบที่ ถูกหรือไม่ ถ้าตอบไม่ถูก แบบจำลองก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น
- 2) การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning) – เป็นการเรียนรู้ที่ไม่มีการตรวจคำตอบว่า ถูกหรือผิด จะไม่มีค่าจริงในการฝึกสอน แบบจำลองจะจัดเรียงโครงสร้างของตัวเองตามลักษณะของ ข้อมูล ผลลัพธ์ที่ได้แบบจำลองจะสามารถจัดหมวดหมู่ของข้อมูลได้
- 3) การเรียนรู้แบบเสริมกำลัง (reinforcement learning) - ไม่จำเป็นต้องมีข้อมูลฝึกสอน แต่ใช้ ลักษณะการควบคุมการเรียนรู้โดยที่ระหว่างโปรแกรมคอมพิวเตอร์ทำงานจะมีการให้คำแนะนำ เสมือนเป็นการให้รางวัลเมื่อผลการทำงานถูกต้อง หรือทำโทษเมื่อผลการทำงานไม่ถูกต้อง และระบบ จะทำการเรียนรู้ด้วยการทดลองต่าง ๆ เอง เพื่อให้การทำงานไปในแนวทางที่ถูกต้อง เช่น การนำ เทคนิคนี้ใช้ควบคุมการเดินของหุ่นยนต์

และนอกเหนือไปจาก 3 ประเภทข้างต้น ยังมีการแบ่งที่ย่อยลงไปอีกดังนี้

- 1) การเรียนรู้แบบกึ่งมีผู้สอน (semi supervised learning) - เป็นการเรียนรู้อีกแบบหนึ่งที่ระหว่างการเรียนรู้แบบมีผู้สอนกับการเรียนรู้แบบไม่มีผู้สอน โดยที่ผู้สอนจะไม่สอนอย่างสมบูรณ์ นั่นคือบาง ข้อมูลในเซตการสอนนั้นขาดข้อมูลขาออก
- 2) ทรานสดักชัน (transduction) - เป็นกรณีพิเศษของการเรียนรู้แบบกึ่งมีผู้สอนคือใช้ชุดตัวอย่างที่มี ทั้งฉลากและไม่มีฉลากในการเรียนรู้ แต่จุดประสงค์ไม่ใช่การสร้างแบบจำลอง แต่เป็นการใส่ฉลาก ให้กับตัวอย่างที่ไม่มีฉลากที่ใช้ในการฝึกสอน เนื่องจากการเรียนรู้แบบทรานสดักชันไม่มีแบบจำลอง ผลการเรียนรู้จึงไม่สามารถนำไปใช้กับข้อมูลชุดใหม่ได้โดยตรง
- 3) การเรียนรู้วิธีการเรียน (learning to learn, meta-learning) - เป็นวิธีที่จะเรียนรู้วิธีการเรียนรู้ของ ตนเอง โดยปรับปรุง inductive bias ที่เป็นข้อสมมติฐานที่ใช้ในการเรียนรู้จากประสบการณ์ที่ผ่านมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ การเรียนรู้ของเครื่องยังสามารถแบ่งประเภทของงานได้ตามข้อมูลขาออกจากระบบที่เครื่องจักรได้เรียนรู้แล้วเป็นหลายประเภทดังนี้

- 1) การจำแนกประเภทข้อมูล (classification) - ข้อมูลขาเข้าถูกแบ่งออกเป็นหลายประเภท (class) และผู้เรียนจะต้องสร้างโมเดลที่สามารถกำหนดประเภทให้กับข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้โดยปกติแล้วจะทำโดยวิธีการเรียนรู้แบบมีผู้สอน ตัวอย่างของการแบ่งประเภทข้อมูลได้แก่ การกรองอีเมลขยะ โดยอีเมลจะถูกแบ่งเป็นประเภทขยะและไม่ใช่ขยะ
- 2) การวิเคราะห์การถดถอย (regression) - ใช้หลักการเกี่ยวกับการแบ่งประเภทข้อมูล แต่ข้อมูลขาออกเป็นประเภทค่าต่อเนื่องมากกว่าเป็นประเภทค่าไม่ต่อเนื่อง
- 3) การแบ่งกลุ่มข้อมูล (clustering) - เป้าหมายคือการแบ่งข้อมูลขาเข้าเป็นกลุ่มๆ โดยขั้นตอนวิธีการจะไม่ทราบกลุ่มดังกล่าวล่วงหน้า (ไม่เหมือนกับการจำแนกประเภทข้อมูล) โดยปกติแล้วมักเป็นการเรียนรู้แบบไม่มีผู้สอน
- 4) การประเมินความหนาแน่น (density estimation) - เป็นการหาการกระจายของข้อมูลในบางมิติ
- 5) การลดขนาดของมิติ (dimensionality reduction) - เป็นการเชื่อมโยงข้อมูลหลายมิติไปสู่ปริภูมิที่มีมิติต่ำกว่า

ขั้นตอนวิธีการที่นิยมใช้ใน Machine Learning มีดังต่อไปนี้

- 1) การเรียนรู้ต้นไม้ตัดสินใจ (decision tree learning)
- 2) กฎความสัมพันธ์ (association rule learning)
- 3) โครงข่ายประสาทเทียม (artificial neural networks)
- 4) การโปรแกรมตรรกะเชิงอุปนัย (inductive logic programming)
- 5) ซัพพอร์ตเวกเตอร์แมชชีน (support vector machines)
- 6) การแบ่งกลุ่มข้อมูล (clustering)
- 7) เครือข่ายแบบเบย์ (bayesian networks)
- 8) การเรียนรู้แบบเสริมกำลัง (reinforcement learning)
- 9) การเรียนรู้ด้วยการแทน (representation learning)
- 10) การเรียนรู้ด้วยความคล้าย (similarity and metric learning)
- 11) ขั้นตอนวิธีเชิงพันธุกรรม (genetic algorithms)

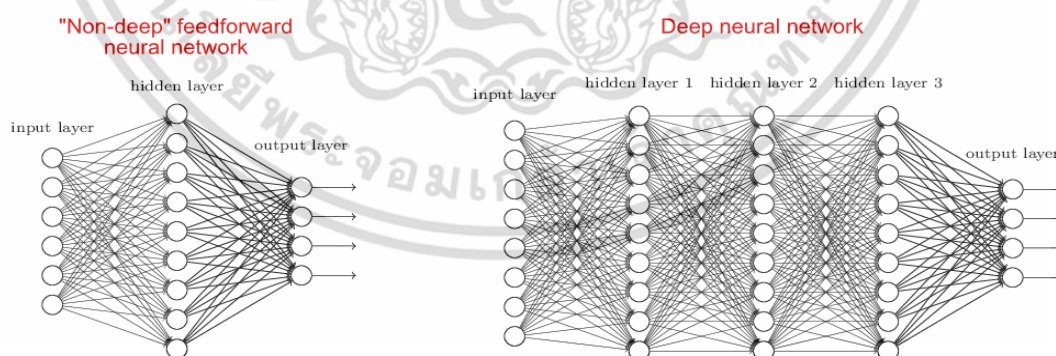
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การเรียนรู้เชิงลึก

การเรียนรู้เชิงลึก (Deep Learning) หรือที่รู้จักกันในนามของ deep structured learning, hierarchical learning หรือ deep machine learning ซึ่งคำศัพท์นี้ถูกนำมาใช้แทนโครงข่ายประสาทเทียม อันเนื่องมาจากการกลับมาเป็นที่นิยมอีกครั้ง โดยจะสื่อความหมายว่า นักวิจัยสามารถฝึกสอนโครงข่ายที่มีจำนวนชั้นมากกว่าในอดีตและมุ่งเน้นศึกษาความสำคัญของจำนวนชั้นในเชิงทฤษฎี การเรียนรู้เชิงลึกนั้นเป็นศาสตร์แขนงหนึ่งของแมชชีนเลิร์นนิง

โครงข่ายประสาทเทียม (Artificial Neural Network-ANN) มีรูปแบบมาจากการทำงานของสมองมนุษย์ที่ประกอบไปด้วย เซลล์ประสาท (neurons) ส่วนของเซลล์ประสาท (dendrite) ไยประสาท (axon) และจุดประสานประสาท (synapses) ซึ่งจะเทียบเท่ากับ โหนด (node) อินพุต (input) เอาต์พุต (output) และค่าถ่วงน้ำหนัก (weight) ตามลำดับ

โครงข่ายประสาทเทียมจะประกอบไปด้วยชั้นหลักๆทั้งหมดอยู่ 3 ชั้น คือ ชั้นอินพุต ชั้นฮิดเดน และชั้นเอาต์พุต โดยจะมีชั้นฮิดเดนหลายๆชั้นเข้าด้วยกันซึ่งจะอยู่ระหว่างชั้นอินพุตและเอาต์พุต ซึ่งจะมีความแตกต่างจากตัวโครงข่ายประสาทเทียมแบบ Shallow เนื่องจาก Shallow มีชั้นฮิดเดนเพียงแค่ชั้นเดียวทำให้ความสามารถในการแก้ปัญหาที่มีประสิทธิภาพที่ต่ำกว่า โครงข่ายประสาทเทียมแบบเชิงลึก ที่มีแบบจำลองข้อมูลที่ซับซ้อนกว่า และนอกจากนี้โครงข่ายประสาทเทียมแบบเชิงลึกยังมีความสามารถในการสร้างแบบจำลองความสัมพันธ์แบบไม่เชิงเส้นได้ด้วยเพื่อใช้ในการสกัดฟีเจอร์ ออกมามีรูปแบบการทำงานที่มีชื่อเรียกว่า feed forward และมีรูปแบบการเรียนรู้ทั้งที่แบบมีผู้สอน และการเรียนรู้แบบที่ไม่มีผู้สอน



รูปที่ 2.2 การทำงานของโครงข่ายการเรียนรู้เชิงลึกและไม่เชิงลึก

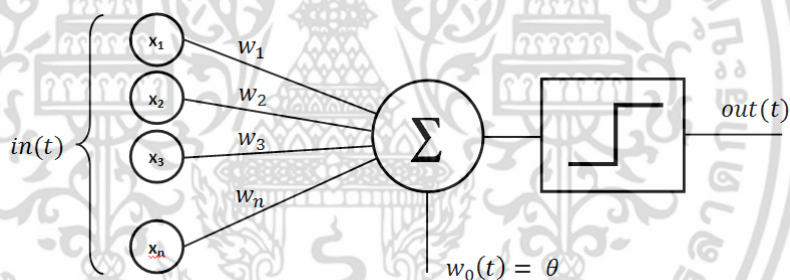
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของโครงข่ายประสาทเทียม

สำหรับในคอมพิวเตอร์ นิวรอนประกอบด้วย อินพุต และ เอาต์พุต เหมือนกัน โดยจำลองให้อินพุตแต่ละอันมีค่าถ่วงน้ำหนักเป็นตัวกำหนดน้ำหนักของอินพุต โดยนิวรอนแต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของอินพุตต้องมากขนาดไหนจึงจะสามารถส่งเอาต์พุตไปยังนิวรอนตัวอื่นได้ เมื่อนำนิวรอนแต่ละหน่วยมาต่อกันให้ทำงานร่วมกัน การทำงานนี้ในทางตรรกะและก็จะเหมือนปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลข เพราะฉะนั้นเมื่อนำการทำงานเข้ามาใช้กับคอมพิวเตอร์สามารถเขียนได้ดังนี้

$$\text{If}(\text{sum}(\text{input} * \text{weight}) > \text{threshold}) \text{ then output}$$

หมายความว่าเมื่อมีอินพุตเข้ามายังโครงข่าย ก็เอาอินพุตมาคูณกับค่าถ่วงน้ำหนักแต่ละค่า ผลที่ได้จากอินพุต ทุกค่าของนิวรอน จะเอามารวมกันแล้วก็เอามาเทียบกับ threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว นิวรอนก็จะส่งเอาต์พุตออกไป เอาต์พุตนี้ก็จะถูกส่งไปยังอินพุตของนิวรอนอื่นๆที่เชื่อมกันในโครงข่าย ถ้าค่าน้อยกว่า threshold ก็จะไม่เกิดเอาต์พุตดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวแบบของนิวรอนในคอมพิวเตอร์

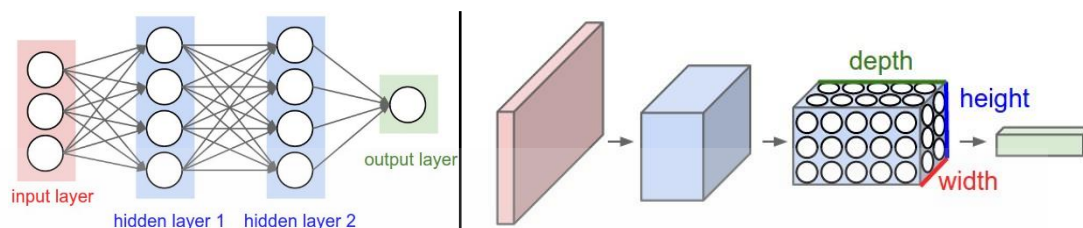
2.1.3 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)

convolutional neural network หรือรู้จักกันในนาม CNN และ ConvNet เป็นโครงข่ายประสาทเทียมแบบ Feed-Forward ซึ่งเป็นรูปแบบการเชื่อมต่อระหว่างเซลล์ประสาทที่ได้รับแรงบันดาลใจมาจากการจัดการโครงสร้างภาพของสมองสัตว์และขั้นตอนการทำงานของความหลากหลายทางชีวภาพ

โดยโครงข่ายแบบคอนโวลูชัน มีรูปแบบที่ไม่ได้แตกต่างจากโครงข่ายประสาทเทียม (Artificial Neural Network) ซึ่งโครงข่ายแบบคอนโวลูชัน จะประกอบไปด้วยค่าถ่วงน้ำหนักที่ได้จากการเรียนรู้ (weight) และค่าไบแอส (biased) แต่ละโหนดจะรับค่าอินพุตเข้ามาและดำเนินการทำตามขั้นตอนที่ได้ถูกกำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายแบบคอนโวลูชันยังถูกรู้จักกันในนามว่า Shift Invariant หรือ Space Invariant Artificial Neural Network (SIANN) โดยถูกตั้งชื่อตามโครงสร้างน้ำหนักและลักษณะการเปลี่ยนแปลงของค่าคงที่



รูปที่ 2.4 (ภาพซ้าย) โครงข่ายแบบปกติที่มีสามชั้น (ภาพขวา) โครงข่ายคอนโวลูชันที่มีการจัดเรียงนิวรอนอยู่ใน 3 มิติ

Convolutional layer

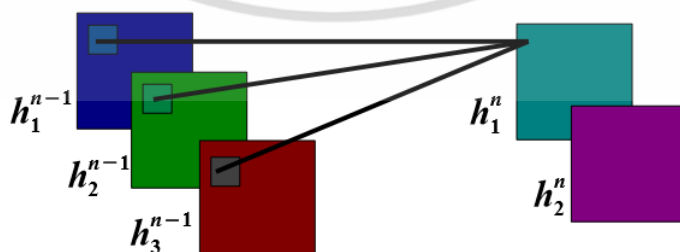
ชั้นคอนโวลูชันประกอบไปด้วยตารางสี่เหลี่ยมหรือฟิลเตอร์ โดยใช้เป็นตัวกรองในการกรองข้อมูลที่รับเข้ามาเพื่อกำหนดค่าถ่วงน้ำหนัก ตัวกรองนี้จะนำไปใช้ทั่วภาพที่ได้รับเข้ามา โดยค่าที่จะได้มาจากการนำตัวกรองไปคำนวณกับภาพ ผลลัพธ์ที่ได้จากตัวกรองจะเรียกว่า Feature Map ซึ่งแต่ละอินพุตของ feature map จะเชื่อมต่อกับเอาต์พุตของแต่ละ feature map โดยมีสมการในการคำนวณ คือ

$$h_j^n = \sum_{k=1}^k h_k^{n-1} * W_{kj}^n$$

กำหนดให้ h_j^n แทนค่าของเอาต์พุตของแต่ละ feature map

h_k^{n-1} แทนอินพุตของแต่ละ feature map

W_{kj}^n แทนฟิลเตอร์



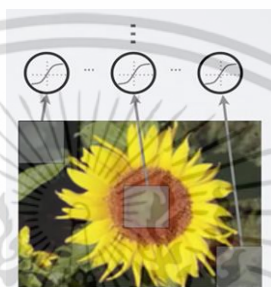
รูปที่ 2.5 การทำคอนโวลูชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

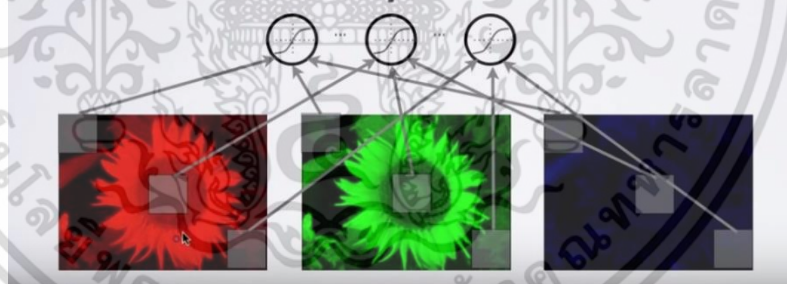
ส่วนประกอบและคุณสมบัติของ Convolutional layer ประกอบไปด้วย

1) Local connectivity

local connectivity เป็นหนึ่งในคุณสมบัติของโครงข่ายแบบคอนโวลูชัน นั้นหมายความว่าทุกนิวรอนในชั้นฮิดเดนจะไม่เชื่อมกับทุกๆนิวรอนในชั้นก่อนหน้าหรือชั้นอินพุต แต่จะเชื่อมแค่บางจุดเท่านั้นซึ่งจะมี receptive field ทำหน้าที่ในการเชื่อม ณ จุดใดๆ โดยคุณสมบัติของ local connectivity จะช่วยในเรื่องของการลดพารามิเตอร์และค่าใช้จ่ายเนื่องจากทุกนิวรอนไม่จำเป็นต้องเชื่อมกับทุกนิวรอนในชั้นก่อนหน้า



รูปที่ 2.6 รูปแบบการทำงานของคุณสมบัติ local connectivity



รูปที่ 2.7 รูปแบบการทำงานของคุณสมบัติ local connectivity ใน 3 channel

2) Spatial arrangement

Spatial arrangement มี 3 พารามิเตอร์ที่เป็นตัวกำหนดขนาดของเอาต์พุตของชั้นคอนโวลูชัน ซึ่งประกอบไปด้วย depth stride และ zero-padding

- Depth คือ ความลึกของปริมาตรของข้อมูล ซึ่งความลึกของเอาต์พุตจะสัมพันธ์กับจำนวนฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Stride คือ การเลื่อนของฟิลเตอร์ไปบนอินพุต โดยระบุค่าในการเลื่อนเป็นหน่วย เมื่อ stride น้อยกว่าขนาดของฟิลเตอร์จะทำให้เกิดการทับซ้อนกัน (overlapping) ของฟิลเตอร์ แต่ถ้า stride มากกว่าหรือเท่ากับขนาดของฟิลเตอร์จะไม่เกิดการทับซ้อนกันทำให้เอาต์พุตที่ได้จะมีขนาดเล็กลง
- Zero-padding คือ การเพิ่มค่าศูนย์ที่ขอบทั้งสี่ด้านของอินพุตที่ได้จากชั้นก่อนหน้า เพื่อคงขนาดของอินพุตไว้

ขนาดของเอาต์พุตสามารถคำนวณด้วยสมการ

$$O = \frac{(W - K + 2P)}{S + 1}$$

- กำหนดให้ W แทนขนาดอินพุต
- O แทนขนาดเอาต์พุต
- K แทนขนาดของฟิลเตอร์ในชั้นคอนโวลูชัน
- S แทน Stride
- P แทนจำนวนของ Zero-padding

จากสูตรถ้าทำการคำนวณโดยให้ padding มีค่าเท่ากับศูนย์แล้วผลลัพธ์ที่ได้มีค่าเป็นทศนิยม ดังนั้นเพื่อให้ขนาดเอาต์พุตเป็นจำนวนเต็มจึงต้องทำการเพิ่ม padding ดังนั้นจะไม่สามารถหาค่าปริมาณอินพุตที่ลงตัวได้ ซึ่งโดยทั่วไปการตั้งค่า zero-padding ให้ $P = (K - 1) / 2$ เมื่อ stride เท่ากับ 1 เพื่อให้แน่ใจว่าปริมาณอินพุตและเอาต์พุตจะมีขนาดพื้นที่เท่าเดิม แม้ว่าโดยทั่วไปจะไม่จำเป็นที่จะต้องใช้นิรอนทั้งหมดของชั้นก่อนหน้านี้ ยกตัวอย่างเช่น อาจจะตัดสินใจในการใช้เพียงส่วนหนึ่งของ zero-padding เท่านั้น

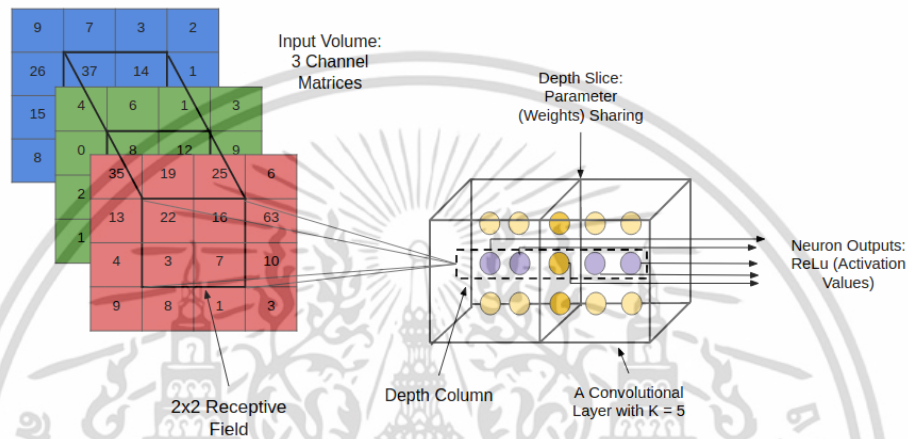
0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

รูปที่ 2.8 คอนโวลูชันที่มี padding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Parameter sharing

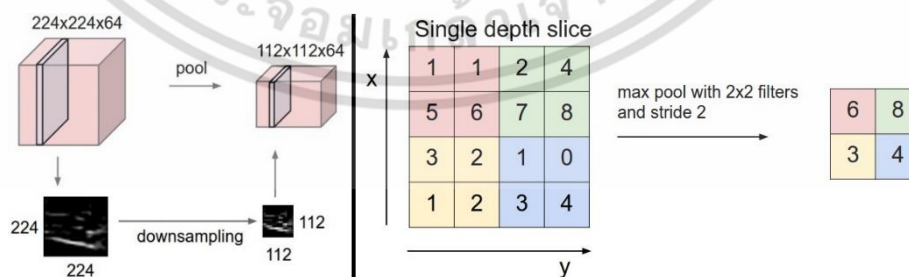
คือรูปแบบการแบ่งพารามิเตอร์ที่นำมาใช้ในชั้นของคอนโวลูชันซึ่งมีเพื่อควบคุมจำนวนของพารามิเตอร์อิสระที่มากเกินไป โดยมีสมมติฐานที่สมเหตุสมผลดังนี้ คือถ้าพีเจอร์หนึ่งพีเจอร์ใดมีประโยชน์ต่อการคำนวณตำแหน่งพื้นที่บางตำแหน่งแล้ว ดังนั้นพีเจอรันั้นก็จะมีประโยชน์ต่อการคำนวณตำแหน่งของพื้นที่อื่นอีกด้วย หรือสามารถกล่าวได้อีกนัยหนึ่งว่า จากรูปที่ 2.9 กำหนดให้นิวรอนของแต่ละ depth slice ใช้ค่าถ่วงน้ำหนักและค่าไบแอสร่วมกัน



รูปที่ 2.9 แนวพื้นที่การแบ่งพารามิเตอร์ที่ใช้ร่วมกัน

Pooling layer

ชั้น pooling ทำหน้าที่ในการลดขนาดพื้นที่เพื่อเป็นการลดจำนวนพารามิเตอร์และการคำนวณของโครงข่ายรวมถึงเป็นการควบคุมการเกิด overfit ซึ่งทั่วไป pooling จะใช้ตัวดำเนินการที่เรียกว่า Max และใช้ฟิลเตอร์ขนาด 2x2 stride เท่ากับ 2



รูปที่ 2.10 ขั้นตอนการทำ max pooling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

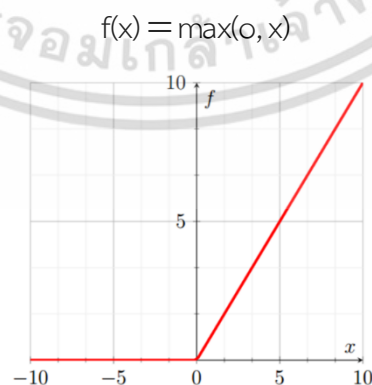
ตัวดำเนินการในชั้น pooling นอกจาก max แล้วยังมี average ซึ่งเคยถูกใช้อย่างแพร่หลายมาก่อน ซึ่งการเลือกใช้นั้นขึ้นอยู่กับลักษณะของข้อมูล เนื่องจาก average pooling จะใช้สก็ดพีเจอร์ที่ค่อนข้าง smooth เหมาะกับงาน เช่น การเรียนรู้ค่าคอนทราสต์ของภาพ ส่วน max pooling จะเหมาะสมกว่าสำหรับสก็ดพีเจอร์ที่ extreme เช่น ขอบภาพและในบางกรณีที่มีความแปรปรวนในฟิลเตอร์ไม่ใช่สิ่งสำคัญทั้งสองจะให้ผลลัพธ์ชนิดเดียวกัน



รูปที่ 2.11 (ก) ภาพต้นฉบับ (ข) ภาพที่ใช้ average pooling (ค) ภาพที่ใช้ max pooling

ReLU layer

ReLU ย่อมาจาก Rectified Linear Units เป็นฟังก์ชันหนึ่งที่ทำหน้าที่ในการคำนวณค่าถ่วงน้ำหนักหลังจากผ่านชั้นคอนโวลูชัน โดยจัดเป็นฟังก์ชันการกระตุ้นที่ไม่อิ่มตัว ฟังก์ชัน ReLU ถูกคิดค้นขึ้นในปี 2000 โดย Hahnloser และได้รับความนิยมในงานของการเรียนรู้เชิงลึกในปี 2015

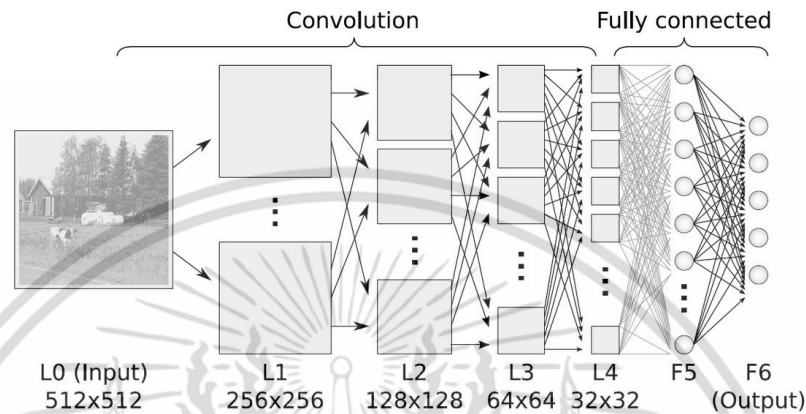


รูปที่ 2.12 กราฟฟังก์ชัน ReLU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fully Connected layer

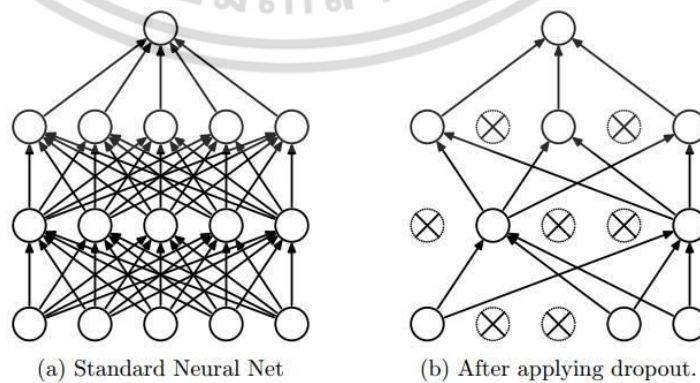
นิวรอนในชั้น fully-connected จะเชื่อมโยงถึงกันหมดกับทุกนิวรอนในชั้นก่อนหน้าและแต่ละการเชื่อมโยงมีค่าถ่วงน้ำหนักเป็นของตัวเอง และในชั้นนี้ค่อนข้างที่จะมีค่าใช้จ่ายที่สูงในเทอมของความจำและการคำนวณ



รูปที่ 2.13 รูปแบบการเชื่อมโยงในชั้น Fully-connected

Dropout layer

เทคนิคในการป้องกันและลดอัตราการเกิด overfitting โดยวิธีการสุ่มนิวรอนบางตัวในชั้นอินพุตหรือเรียกว่าชั้น visible กับชั้นฮิดเดนของโครงข่ายให้มีค่าเป็นศูนย์ชั่วคราวระหว่างการฝึกสอน ในระหว่างการส่งข้อมูลไปข้างหน้า แล้วจะไม่นำค่าถ่วงน้ำหนักมาใช้ในการส่งข้อมูลย้อนกลับ ซึ่งการสุ่มนิวรอนนั้นจะกำหนดให้ทุกตัวมีค่าความน่าจะเป็น p ที่เป็นอิสระต่อกัน โดย p จะเท่ากับ 0.5 โดยข้อดีของ dropout คือจะช่วยให้โครงข่ายที่ต้องใช้ในการคำนวณมีขนาดเล็กลงแต่ยังคงประสิทธิภาพไว้ทำให้ช่วยในเรื่องของความจุข้อมูล



รูปที่ 2.14 การทำงานของโครงข่ายก่อนและหลังการใช้ Dropout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Softmax layer

ชั้น softmax โดยทั่วไปจะอยู่ในชั้นสุดท้ายซึ่งเป็นส่วนของเอาต์พุตของโครงข่ายประสาทเทียมและจะมีการดำเนินการกับการจำแนกคลาสแบบหลายคลาส ยกตัวอย่างเช่น การจดจำวัตถุ โดย softmax มีชื่อมาจากฟังก์ชัน softmax ซึ่งจะนำค่าการทำนายของอินพุต $z_k, k = 1, 2, \dots, K$ ป้อนใส่เข้าไปในคู่อันดับระหว่าง 0 ถึง 1 และจะมีผลรวมเท่ากับหนึ่ง โดย softmax มีหน้าที่ในการเป็นตัวแทนการกระจายของค่าความน่าจะเป็นสำหรับ label ของแต่ละคลาส

$$\sigma(z_c) = \frac{e^{z_c}}{\sum_{j=1}^K e^{z_j}}$$

รูปแบบเลเยอร์ของโครงข่ายแบบคอนโวลูชัน

รูปแบบทั่วไปของสถาปัตยกรรมโครงข่ายแบบคอนโวลูชันจะประกอบไปด้วย ชั้นคอนโวลูชัน ชั้น RELU จากนั้นตามด้วยชั้น pooling และชั้น fully-connected เป็นชั้นสุดท้าย ซึ่งมีรูปแบบดังนี้

$$[[\text{CONV} \rightarrow \text{RELU}] * N \rightarrow \text{POOL}] * M \rightarrow [\text{FC} \rightarrow \text{RELU}] * K \rightarrow \text{FC}$$

โดยที่ * หมายถึงการทำซ้ำ ซึ่ง N M และ K เป็นจำนวนเต็ม

N มีค่าอยู่ระหว่าง 0 ถึง 3

M มีค่ามากกว่าหรือเท่ากับ 0

K มีค่าอยู่ระหว่าง 0 ถึง 2

2.1.4 กระบวนการทำงานแบบแพร่ย้อนกลับ (Backpropagation)

Backpropagation เป็นขั้นตอนวิธีของการเรียนรู้ของโครงข่ายวิธีหนึ่งที่นิยมใช้ในมัลติเลเยอร์เพอร์เซปตรอน (Multilayer perceptron) หรือที่เรียกอีกอย่างว่า Widrow-Hoff learning rule ซึ่งโครงข่ายจะทำการปรับค่าถ่วงน้ำหนัก หลังจากให้ข้อมูลสำหรับการฝึกสอนโครงข่ายแต่ละรอบ(iteration) แล้วเสร็จ ค่าเอาต์พุตของโครงข่าย จะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งจะนำค่าไปใช้ในการปรับปรุงค่าถ่วงน้ำหนัก

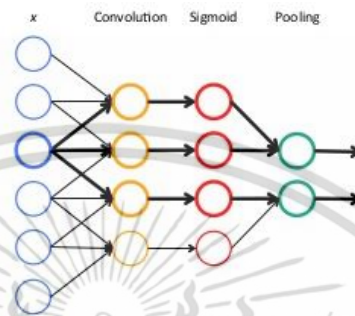
การปรับปรุงค่าเกรเดียนต์ มีอยู่ 2 วิธีที่แตกต่างกัน ได้แก่แบบ Incremental และแบบ Batch ในโหมด Incremental นั้นค่าของเกรเดียนต์ จะมีการคำนวณค่าและจะปรับค่าถ่วงน้ำหนักหลังจากที่มีการป้อนอินพุตแต่ละค่าเสร็จแล้ว ส่วนในโหมด Batch mode นั้นจะต้องป้อนอินพุตทั้งหมดให้กับโครงข่ายก่อนถึงจะมีการปรับค่าถ่วงน้ำหนักและค่าไบแอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการวิธีการของ backpropagation

1) การแพร่ข้อมูลไปข้างหน้า

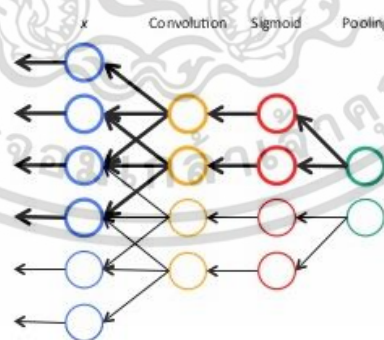
ส่วนนี้จะส่งค่าเข้าไปในทุกโหนดของชั้นอินพุต และทำการคำนวณชั้นฮิดเดนถัดไปโดยใช้ค่าที่ได้จากชั้นก่อนหน้าหรือเรียกอีกอย่างว่าชั้นเอาต์พุตก่อนหน้าหลังจากทำการคำนวณแล้วจะส่งค่าไปคำนวณเรื่อยไปจนถึงชั้นเอาต์พุตของโครงข่าย



รูปที่ 2.15 การทำงานของขั้นตอนการแพร่ข้อมูลไปข้างหน้า

2) การแพร่ข้อมูลแบบย้อนกลับ

ขั้นตอนนี้จะเป็นการปรับค่าถ่วงน้ำหนักเพื่อให้โครงข่ายได้ค่าที่ถูกต้องมากที่สุด โดยการหาค่าที่เหมาะสมที่สุดนี้จำเป็นที่จะต้องใช้ขั้นตอนวิธีการหนึ่ง ที่มีชื่อว่าเกรเดียนต์เดสเซนทในการปรับค่าถ่วงน้ำหนัก ซึ่งจะต้องมีการใช้กฎลูกโซ่เข้ามาช่วย โดยจะเริ่มจากการคิดและปรับค่าถ่วงน้ำหนักจากข้างหลังมาข้างหน้า



รูปที่ 2.16 การทำงานของขั้นตอนการแพร่ข้อมูลแบบย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

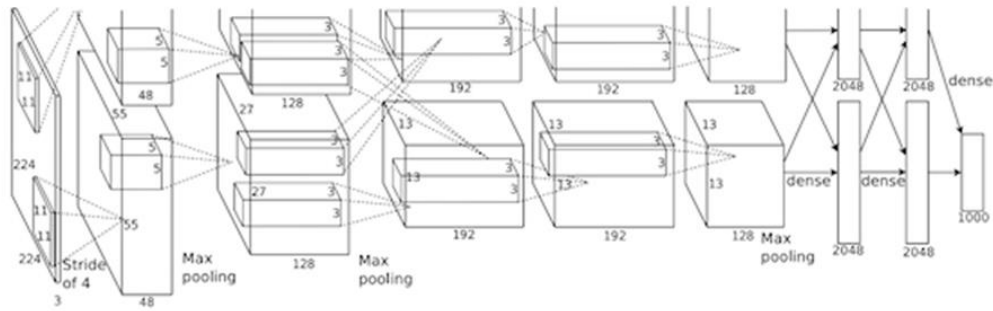
ขั้นตอนการฝึกโครงข่ายประสาทเทียมเพื่อให้เกิดการเรียนรู้

- 1) สุ่มค่าถ่วงน้ำหนักให้กับส่วนอินพุตของแต่ละนิวรอน ซึ่งค่าถ่วงน้ำหนักที่สุ่มได้อาจมีค่าอยู่ในช่วง $[-1.0, 1.0]$ หรือ $[-0.5, 0.5]$ โดยสามารถเปลี่ยนแปลงได้
- 2) กำหนดค่าไบแอสหรือไบแอส กำหนดเป็นค่าเริ่มต้นในขั้นตอนแรก ซึ่งอาจมีค่าเท่ากับ 1 หรืออาจจะทำการสุ่มขึ้นมา โดยสามารถเปลี่ยนแปลงได้
- 3) นำข้อมูลเข้าไปในโครงข่ายผ่านทางชั้นอินพุต
- 4) คำนวณค่าอินพุตและเอาต์พุตในแต่ละชั้น
- 5) คำนวณค่าความคลาดเคลื่อนในแต่ละชั้น โดยคำนวณที่ชั้นเอาต์พุตก่อนจากนั้นจึงคำนวณที่ชั้นฮิดเดน ซึ่งค่าความคลาดเคลื่อนสามารถคำนวณได้จากการเปรียบเทียบผลลัพธ์ที่คำนวณได้กับค่าเป้าหมายที่ตั้งไว้
- 6) ทำการปรับค่าถ่วงน้ำหนักและค่าไบแอสใหม่จากค่าความคลาดเคลื่อนที่คำนวณได้
- 7) ทำซ้ำกระบวนการเดิมไปเรื่อยๆจนกว่าจะป้อนข้อมูลฝึกจนหมด

2.1.5 สถาปัตยกรรมโครงข่ายประสาทเทียม

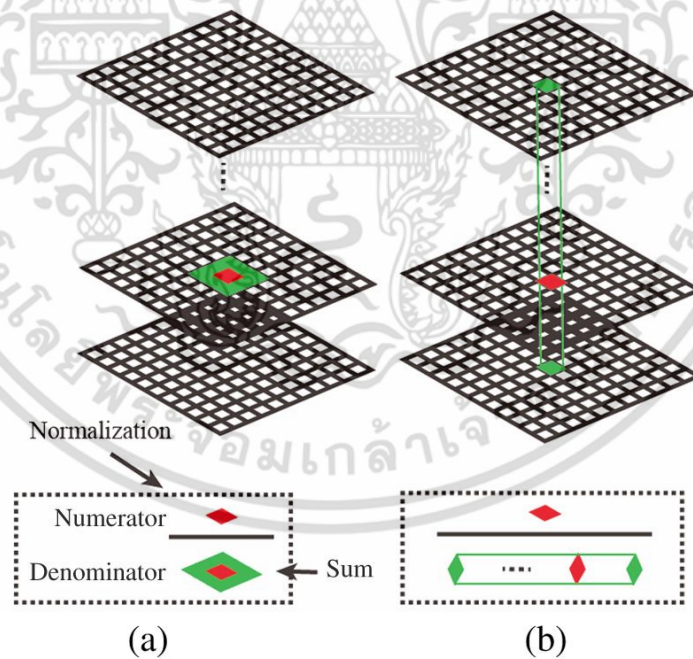
1) AlexNet

AlexNet เป็นสถาปัตยกรรมโครงข่ายแรกๆ ที่ได้รับความนิยมในงานของ CNN ในด้านคอมพิวเตอร์วิทัศน์ซึ่งได้รับการพัฒนาโดย Alex Krizhevsky Ilya Sutskever และ Geoff Hinton ทางทีมพัฒนาได้ส่งโมเดลเข้าร่วมการแข่งขัน ILSVRC ที่จัดขึ้นโดย ImageNet ในปี 2012 ซึ่งปรากฏว่า AlexNet ได้ชนะเลิศการแข่งขันและมีประสิทธิภาพสูงที่สุด โดยเปรียบเทียบกับค่า top-5 error โดย AlexNet อยู่ที่ 16% ในขณะที่อันดับสองอยู่ที่ 26% สถาปัตยกรรมของ AlexNet ค่อนข้างที่จะมีโครงสร้างคล้ายคลึงกับ LeNet แต่มีความลึกกว่าและใหญ่กว่า อีกทั้งมีพีเจอร์ของชั้นคอนโวลูชันหลายชั้นซ้อนอยู่ด้านบน ซึ่งปกติจะมีชั้นคอนโวลูชันเพียงชั้นเดียวแล้วตามด้วยชั้น Pooling เสมอ โดยสถาปัตยกรรมจะประกอบไปด้วยจำนวนพารามิเตอร์ 60 ล้านพารามิเตอร์ จำนวนนิวรอนทั้งสิ้น 650,000 นิวรอน และจำนวนชั้น 11 ชั้น ซึ่งประกอบไปด้วย ชั้นคอนโวลูชัน 5 ชั้น ชั้น max pooling 5 ชั้น และชั้น fully-connected 3 ชั้น ทำให้การฝึกสอนเป็นไปอย่างรวดเร็ว AlexNet ได้ใช้ฟังก์ชันการกระตุ้นแบบ non-saturating และรองรับการฝึกสอนด้วย GPU พร้อมกันสองตัว อีกทั้งมีชั้น dropout เป็นตัวทำการลดค่า overfitting ในชั้น fully-connected



รูปที่ 2.17 โครงสร้างสถาปัตยกรรมแบบ AlexNet

นอกจากนี้ยังมีชั้น LRN (Local Response Normalization) ทำหน้าที่ในการช่วยเพิ่มความเด่นชัดให้กับพีเจอร์ ซึ่งแบ่งออกเป็น 2 แบบ คือ within channel และ cross channel โดย within channel จะคำนวณผลรวมของเอาต์พุตกำลังสองในบริเวณที่มีขนาด $n \times n$ ที่อยู่บนชั้นลิกเดียวกัน โดยมีตำแหน่ง ซึ่ง (x,y) เป็นศูนย์กลางของบริเวณดังกล่าว ส่วน cross channel จะคำนวณผลรวมของเอาต์พุตกำลังสองในตำแหน่ง (x,y) ที่ตรงกันในชั้นลิกที่ติดกัน คือ ชั้นลิกที่ $\max(0, i - \frac{n}{2})$ ถึงชั้นลิกที่ $\min(N - 1, i + \frac{n}{2})$



รูปที่ 2.18 (a) LRN แบบ within channel (b) LRN แบบ cross channel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนบริเวณสี่เหลี่ยมแทนค่าที่นำมาคำนวณหาผลรวม และบริเวณสีแดงแสดงค่าที่ศูนย์กลาง
การคำนวณสามารถดำเนินการได้ดังสมการ

$$b_{x,y}^i = a_{x,y}^i / \left(k + \left(\frac{\alpha}{n} \right) \sum_{\max(0, i-n/2)}^{\min(N-1, i+n/2)} (a_{x,y}^i)^2 \right)^\beta$$

โดยที่ $b_{x,y}^i$ คือ ผลลัพธ์ที่ตำแหน่ง (x,y) ของชั้นลึกที่ i ซึ่ง $i = 0,1,2, \dots, N$

$a_{x,y}^i$ คือ เอาต์พุตที่ตำแหน่ง (x,y) ของชั้นลึกที่ i ในชั้นคอนโวลูชันหลังจากผ่านชั้น ReLU

n คือ จำนวนชั้นที่อยู่ติดกับชั้นลึกที่ i ซึ่งเราจะกำหนดให้ $n = 5$

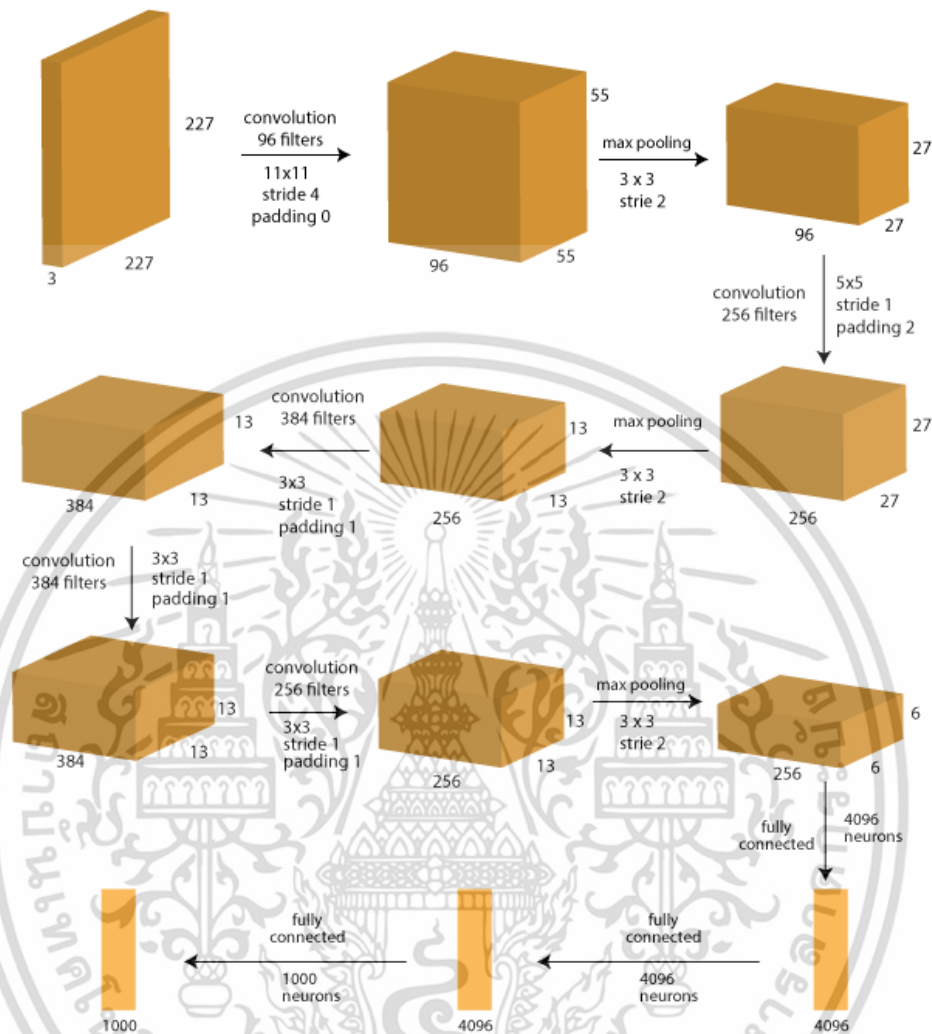
N คือ จำนวนชั้นลึกทั้งหมด (depth)

k, α, β คือ ค่าคงที่ ซึ่งเรียกว่า hyper-parameter ซึ่งเราจะกำหนดให้ $k = 1$ $\alpha = 1$

และ $\beta = 0.75$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณขนาดอินพุตและเอาต์พุตของสถาปัตยกรรม AlexNet



รูปที่ 2.19 โครงสร้างการทำงานของสถาปัตยกรรม AlexNet

ชั้นที่ 0 : นำเข้าภาพ

- ขนาด : $227 \times 227 \times 3$

ชั้นที่ 1 : Convolution ใช้ 96 ฟิลเตอร์ ขนาด 11×11 ค่า stride = 4 และค่า padding = 0

- $\frac{(227-11)}{4} + 1 = 55$ เป็นขนาดของผลลัพธ์
- ขนาดเอาต์พุต : $55 \times 55 \times 96$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นที่ 2 : Max-Pooling ใช้ 1 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 2

- $\frac{(55-3)}{2} + 1 = 27$ เป็นขนาดของผลลัพธ์
- ขนาดเอาต์พุต : $27 \times 27 \times 96$

ชั้นที่ 3 : Convolution ใช้ 256 ฟิลเตอร์ ขนาด 5×5 และค่า stride = 1

- หาค่าการขยายช่อง (padding) $\frac{(5-1)}{2} = 2$ จะทำให้สามารถรักษาขนาดของอินพุตไว้ได้
- ขนาดเอาต์พุต : $27 \times 27 \times 256$

ชั้นที่ 4 : Max-Pooling ใช้ 1 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 2

- $\frac{(27-3)}{2} + 1 = 13$ เป็นขนาดของผลลัพธ์
- ขนาดเอาต์พุต : $13 \times 13 \times 256$

ชั้นที่ 5 : Convolution ใช้ 384 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 1

- หาค่าการขยายช่อง (padding) $\frac{(3-1)}{2} = 1$ จะทำให้สามารถรักษาขนาดของอินพุตไว้ได้
- ขนาดเอาต์พุต : $13 \times 13 \times 384$

ชั้นที่ 6 : Convolution ใช้ 384 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 1

- หาค่าการขยายช่อง (padding) $\frac{(3-1)}{2} = 1$ จะทำให้สามารถรักษาขนาดของอินพุตไว้ได้
- ขนาดเอาต์พุต : $13 \times 13 \times 384$

ชั้นที่ 7 : Convolution ใช้ 256 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 1

- หาค่าการขยายช่อง (padding) $\frac{(3-1)}{2} = 1$ จะทำให้สามารถรักษาขนาดของอินพุตไว้ได้
- ขนาดเอาต์พุต : $13 \times 13 \times 256$

ชั้นที่ 8 : Max-Pooling ใช้ 1 ฟิลเตอร์ ขนาด 3×3 และค่า stride = 2

- $\frac{(13-3)}{2} + 1 = 6$ เป็นขนาดของผลลัพธ์
- ขนาดเอาต์พุต : $6 \times 6 \times 256$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นที่ 9 : Fully-Connected กับ 4,096 นิวรอน

- ภายหลังแต่ละ 9216 พิกเซลจะถูกป้อนเข้าไปในแต่ละนิวรอนและค่าถ่วงน้ำหนักทั้งหมด 4,096 ค่า ซึ่งถูกกำหนดโดย back-propagation

ชั้นที่ 10 : Fully-Connected กับ 4,096 นิวรอน

- ทำเหมือนชั้นที่ 9

ชั้นที่ 11 : Fully-Connected กับ 1,000 นิวรอน

- ชั้นนี้เป็นชั้นสุดท้ายและมีนิวรอนทั้งหมด 1,000 เพราะ ImageNet มีข้อมูล 1,000 คลาสในการทำนาย

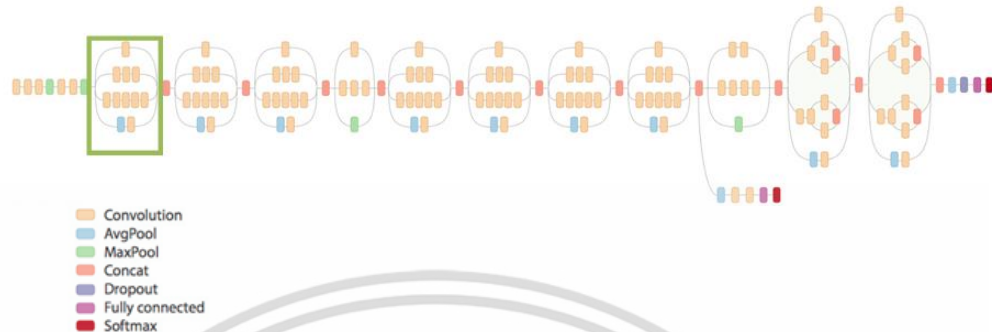
2) GoogLeNet

GoogLeNet เป็นโครงข่ายแบบคอนโวลูชันที่ถูกพัฒนาขึ้นมาโดย Szegedy และคณะจากกูเกิล และได้รับรางวัลชนะเลิศในการแข่งขัน ILSVRC ในปี 2014 ซึ่งตัวสถาปัตยกรรมได้ออกแบบต่างจากสถาปัตยกรรมก่อนๆที่มีรูปแบบการทำงานเป็นลำดับชั้นดังนั้นจึงถือว่าเป็นคุณสมบัติเด่นของ GoogLeNet โดยเรียกว่าการพัฒนาแบบ Inception Module ซึ่งจะประกอบไปด้วย inception module ทั้งหมด 9 โมดูลและมีเลเยอร์ทั้งหมดมากกว่า 100 เลเยอร์ และจะใช้ค่าเฉลี่ยของ pooling แทนที่จะเป็นชั้น fully connected โดยระหว่างที่ทำการทดสอบ จะทำการครอปรูปภาพหลายๆอันจากรูปภาพอันเดียวกันแล้วนำเข้าสู่โครงข่ายและ softmax จากนั้นจะหาค่าเฉลี่ยที่เป็นผลคำตอบสุดท้ายออกมา

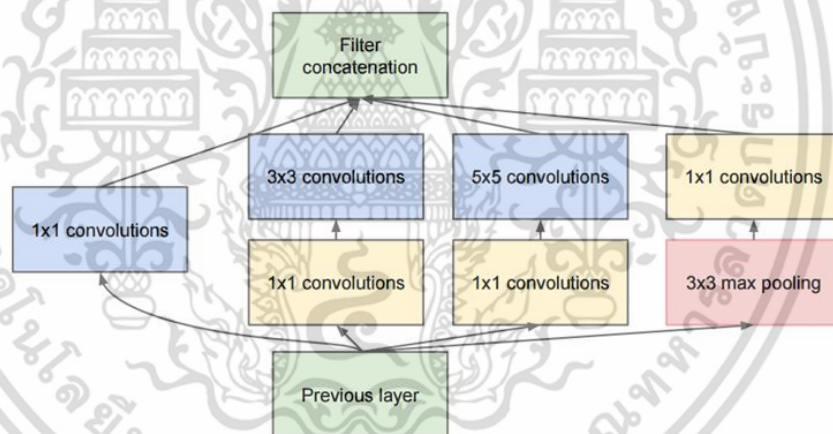
GoogLeNet ได้ให้ความสำคัญในเรื่องของหน่วยความจำและการใช้งานของแบตเตอรี่เป็นอย่างมาก จึงมีการใช้พารามิเตอร์ไปเพียงไม่มากหรือน้อยกว่า 12 เท่าเมื่อเทียบกับ AlexNet การฝึกสอนจะดำเนินไปบน GPU ที่มีคุณภาพสูงเพียงไม่กี่ตัวภายในเวลาเพียงหนึ่งอาทิตย์เท่านั้น

Inception Module

การทำงานเป็นแบบพาราเรล หรือกล่าวคือทุกอย่างจะไม่เกิดขึ้นตามลำดับ



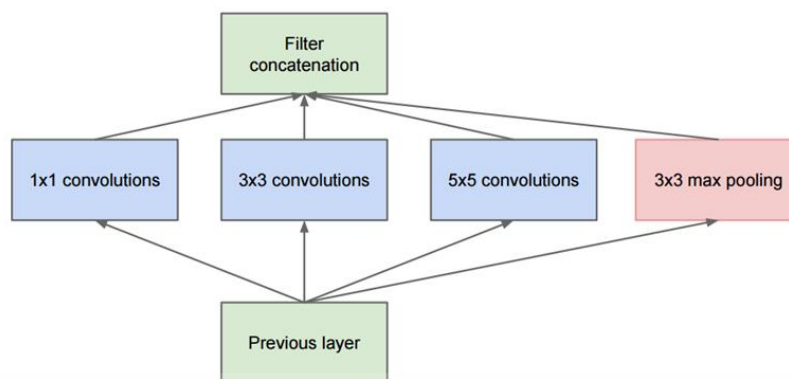
รูปที่ 2.20 การทำงานของสถาปัตยกรรม GoogLeNet แบบ inception module



รูปที่ 2.21 การทำงานแบบ inception module

จากรูปที่ 2.21 กล่องสี่เหลี่ยมด้านล่างคืออินพุตและด้านบนสุดคือเอาต์พุตของโมเดล โดยทั่วไปแล้วแต่ละเลเยอร์ของโครงข่ายคอนโวลูชันดั้งเดิม จะต้องเลือกว่าต้องการใช้งาน pooling หรือว่า convolution แต่ inception module อนุญาตให้ใช้ได้พร้อมกันในรูปแบบพาราเรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 inception module แบบดั้งเดิม

แนวคิดที่นักพัฒนาได้ออกแบบเกี่ยวกับ inception module แบบดั้งเดิมคือออกแบบให้หนึ่งชั้นประกอบไปด้วย 1x1 convolution ซึ่งจะมาก่อน 3x3 และ 5x5 โดย 1x1 convolution จะมีหน้าที่ในการจัดหาวิธีการในการลดมิติให้เล็กลง ยกตัวอย่าง เช่นถ้าอินพุตมีขนาด 100x100x60 และฟิลเตอร์มีขนาด 20 เมื่อนำเข้าสู่ 1x1 convolution จะได้เป็น 100x100x20 โดยจะหมายความว่า 3x3 และ 5x5 convolution จะไม่ต้องเจอกับขนาดที่ใหญ่จนเกินไป ซึ่งสามารถเปรียบเสมือนเป็น pooling อย่างหนึ่งได้เพราะมีการลด depth เกิดขึ้นซึ่งจะคล้ายคลึงกับชั้น max pooling ของสถาปัตยกรรมทั่วไป

โดยในโมดูลของชั้นของโครงข่ายประกอบไปด้วยฟิลเตอร์ขนาดกลางและขนาดใหญ่มารวมถึงมีการดำเนินการ pooling อยู่ด้วย ซึ่งโครงข่ายในชั้นคอนโวลูชันสามารถที่จะสกัดข้อมูลที่ค่อนข้างเฉพาะ ในขณะที่ฟิลเตอร์ขนาด 5x5 สามารถที่จะครอบคลุม receptive field ของอินพุตที่มีขนาดใหญ่ได้และดังนั้นจึงสามารถที่จะสกัดข้อมูลได้เป็นอย่างดี อีกทั้งยังมีตัวดำเนินการ pooling ที่จะช่วยลดขนาดของพื้นที่เชิงมิติและแก้ปัญหา overfit และบนด้านบนสุดนั้นจะมี ReLU หลังจากชั้นคอนโวลูชันซึ่งจะช่วยปรับปรุงฟังก์ชันที่ไม่เป็นเชิงเส้นของโครงข่าย

2.1.6 เฟรมเวิร์คของการเรียนรู้เชิงลึก

การใช้โปรแกรมไลบรารีของการเรียนรู้เชิงลึกเป็นวิธีที่ดีเพื่อให้กระบวนการพัฒนาเกิดง่ายขึ้น โดยที่ไม่ต้องสร้างทุกอย่างขึ้นมาใหม่ เพียงแค่เรียกใช้จากแหล่งที่มีอยู่แล้ว คำว่าไลบรารีหมายถึงชุดของฟังก์ชันและโมดูลที่สร้างไว้แล้วคุณสามารถเรียกผ่านโปรแกรมของคุณได้ โดยสามารถแบ่งไลบรารีของการเรียนรู้เชิงลึกได้ในสองรูปแบบ คือ

- 1) ไลบรารีในเชิงพาณิชย์ เช่น deeplearning4j Torch และ Caffe
- 2) ไลบรารีในเชิงการศึกษาหรือวิทยาศาสตร์ เช่น Theano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Caffe

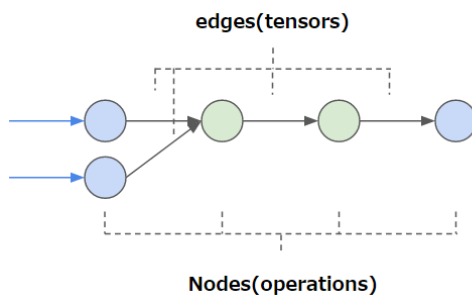
เป็นไลบรารีที่สร้างโดย Yangqing Jia ซึ่งขณะนั้นกำลังศึกษาในระดับปริญญาเอก ที่มหาวิทยาลัยแคลิฟอร์เนีย เบิร์กลีย์ โดยไลบรารีนี้เขียนด้วยภาษา CUDA C/C++ ทำให้สามารถสลับการทำงานระหว่าง CPU และ GPU ได้ง่าย อีกทั้งรองรับการสร้างโครงข่ายทั้งแบบ convolution และแบบ recurrent นอกจากนี้ยังมีส่วนติดต่อกับภาษา Matlab และ Python ซึ่งเรียกว่า matcaffe และ pycaffe ตามลำดับ ทำให้สามารถเขียนสคริปต์เรียกใช้ caffe ได้ จุดเด่นแรกของ caffe คือสามารถปรับค่าไฮเปอร์พารามิเตอร์ต่างๆได้ แต่มีความซับซ้อนมาก ทำให้กลุ่มผู้ใช้ส่วนมาก จำกัดในวงนักวิชาการ นักวิจัย และผู้เชี่ยวชาญ ซึ่งจะพบว่าส่วนมากใช้ในงานวิจัยที่ตีพิมพ์ด้านการเรียนรู้เชิงลึก ซึ่งทราบได้จากการเปิดเผย source code บน GitHub Repository และอีกหนึ่งจุดแข็งของไลบรารีมาจากมีชุมชนนักพัฒนา ขนาดใหญ่ อันก่อให้เกิดแนวคิด Model Zoo ที่นักพัฒนาสามารถแบ่งปันโมเดลสำเร็จรูปกันได้บน GitHub Repository ช่วยให้นักพัฒนาไม่จำเป็นต้องฝึกสอนโมเดลเองตั้งแต่ต้น

2) Theano

ไลบรารีมีชุดฟังก์ชันเพื่อสร้างโครงข่ายการเรียนรู้เชิงลึกที่ช่วยให้การฝึกสอนเป็นไปได้อย่างรวดเร็ว Theano สร้างขึ้นโดยกลุ่มที่มาจากมหาวิทยาลัยมอนเทรียล โดยภาษาที่ใช้พัฒนาคือ python โดย Theano จะช่วยนิยามและหาค่าพจน์ทางคณิตศาสตร์ที่มีเวกเตอร์และเมทริกซ์ได้อย่างรวดเร็ว แต่ข้อเสียคือต้องเขียนโค้ดเองในหลายๆด้าน เช่น การเรียกใช้การกระตุ้น วิธีการฝึกสอน และวิธีพิเศษใดๆเพื่อป้องกัน overfitting

3) Tensorflow

เป็นไลบรารีสำหรับงานด้าน machine learning ซึ่งมีชื่อเดิมว่า DistBelief เป็นส่วนหนึ่งของ Google Brain Project ของกูเกิล ไลบรารีนี้พัฒนาขึ้นด้วยภาษา C++ และ Python โดยมีเป้าหมายในการสร้างระบบที่ง่ายต่อการติดตั้งซอฟต์แวร์ที่มีโมเดลด้าน machine learning บนฮาร์ดแวร์ที่หลากหลาย เช่น สมาร์ทโฟน เซิร์ฟเวอร์เดี่ยว และเครือข่ายเซิร์ฟเวอร์ขนาดใหญ่ ที่มีเครื่องลูกข่ายกว่า 100 เครื่อง เทียบเท่าหน่วยประมวลผลกราฟิกกว่า 1000 ชิป เป็นต้น ส่งผลให้เพิ่ม portability ของซอฟต์แวร์ดังกล่าว จุดเด่นของไลบรารีคือมีส่วนแสดงผลข้อมูลของโมเดลบนบราวเซอร์เรียกว่า TensorBoard ตัวอย่างเช่น ภาพโครงสร้างของโมเดลแสดงด้วย computational graph โดยจัดรูปที่ 2.23 ซึ่งกราฟจะประกอบด้วยโหนด แทนการดำเนินการทางคณิตศาสตร์ และลูกศร แทนทิศทางการไหลของข้อมูลระหว่างโหนด สำหรับข้อมูลจะอยู่ในรูปแบบของอาร์เรย์หลายมิติ เรียกว่า tensor จึงเป็นที่มาของชื่อไลบรารีนี้



รูปที่ 2.23 กราฟแสดงการทำงานของ TensorFlow

4) Torch

Torch เป็นไลบรารี deep learning คุณภาพสูงที่ใช้ในงานทางด้านเชิงวิทยาศาสตร์และมีการสนับสนุนจากชุมชนนักพัฒนาขนาดใหญ่มากมาย โดย Torch ถูกพัฒนาขึ้นด้วยภาษา C สามารถทำการปรับค่าไฮเปอร์พารามิเตอร์ได้เหมือนไลบรารีอื่นๆ มีการคำนวณคณิตศาสตร์และเก็บค่าทางคณิตศาสตร์ที่ใช้กับงานที่เกิดขึ้นเป็นประจำเพื่อเพิ่มประสิทธิภาพและช่วยให้มีความเร็วขึ้น

โดย Torch มีเป้าหมายคือการทำให้งานปรับแต่งค่าใช้งานง่ายที่สุดแต่จะคงความยืดหยุ่นไว้ให้มากที่สุดในการใช้งานกับโครงข่ายที่ซับซ้อนและรวมถึงทำให้ขั้นตอนวิธีการนั้นทำงานได้อย่างรวดเร็ว ปัจจุบันถูกใช้งานทางด้านวิจัยของเฟซบุ๊กและทวิตเตอร์

2.1.7 การปรับค่าพารามิเตอร์

1) ฟังก์ชันการสูญเสีย (Loss function)

เป็นที่รู้จักในชื่อฟังก์ชันความคลาดเคลื่อน (Error function) หรือฟังก์ชันต้นทุน (Cost function) หรือฟังก์ชันวัตถุประสงค์ (Objective function) โดยเป็นฟังก์ชันทางคณิตศาสตร์ที่นำมาใช้ในงานจำแนกข้อมูลโดยมีวิธีคำนวณค่าหลายแบบ เช่น แบบ Square แบบ Hinge แบบ Logistic และแบบ Cross entropy เป็นต้น โดยฟังก์ชันการสูญเสียนิยมใช้อย่างแพร่หลายในโครงข่ายเชิงลึกสมัยใหม่ คือ ฟังก์ชันการสูญเสียแบบ Cross entropy ซึ่งมีสูตรคำนวณดังนี้

$$E = \frac{-1}{N} \sum_{n=1}^N \log(\hat{p}_n, l_n)$$

โดยที่ l_n คือ เซตของแท็กภาพ สามารถเขียนได้ดังนี้ $l_n \in [0, 1, 2, \dots, K - 1]$ ซึ่ง K แทนจำนวนคลาส และ \hat{p}_n คือเซตของค่าทำนายที่คำนวณด้วยฟังก์ชัน softmax

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) **Solver** คือตัวแก้ปัญหาที่จะช่วยปรับประสิทธิภาพของโมเดลโดยการหาค่า loss ให้ต่ำที่สุดซึ่งจะหาได้จากการแพร่ข้อมูลไปข้างหน้าตามโครงข่ายหรือเป็นการคำนวณค่าฟังก์ชันแล้วแพร่ไปข้างหน้า จากนั้นแพร่ย้อนกลับมาเพื่อหาค่าเกรเดียนต์เพื่อที่จะปรับค่าพารามิเตอร์สำหรับการปรับปรุงค่า loss ซึ่งในตัว solver หรือเกรเดียนต์เดสเซนท์ที่สามารถเรียกใช้ได้แก่

- Stochastic Gradient Descent (SGD)
- Adaptive Moment Estimation (Adam)
- Nesterov's Accelerated Gradient (NAG)
- RMSprop (RMSProp)

หลักการทำงานของ solver

- 1) สร้างโครงข่ายการฝึกสอนสำหรับเรียนรู้และสร้างโครงข่ายทดสอบสำหรับการประเมินผล
- 2) ทำการปรับค่าให้เหมาะสมด้วยการเรียกใช้ขั้นตอนวิธีการไปแพร่ข้างหน้า-ย้อนกลับ แล้วทำการปรับปรุงค่าพารามิเตอร์ โดยที่การย้อนกลับจะต้องรวมค่าเกรเดียนต์เข้าไปในพารามิเตอร์ซึ่งเรียกว่า เมธอด solver
- 3) ประเมินผลของโครงข่ายสำหรับทดสอบหรือชุดข้อมูลสำหรับทดสอบ
- 4) ปรับเพิ่มประสิทธิภาพใหม่จนกว่าจะได้ประสิทธิภาพที่ต้องการ

การปรับค่าถ่วงน้ำหนัก (Weight Updates)

ทำการหาคำนวณหา average loss ได้ดังสมการ

$$L(W) = \frac{1}{|D|} \sum_i f_w(X^{(i)}) + \lambda r(W)$$

โดย $|D|$ แทนจำนวนตัวอย่างของชุดข้อมูล

f_w แทนฟังก์ชัน loss

$X^{(i)}$ แทนตัวอย่างที่ i

$r(W)$ แทน regularization term

λ แทนค่าถ่วงน้ำหนักของ regularization term

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGD

ย่อมาจาก Stochastic Gradient Descent เป็นขั้นตอนวิธีการในการปรับหาค่าจุดที่เหมาะสม โดยในแต่ละการคำนวณเกรเดียนต์ จะต้องทำการสุ่มข้อมูลบางส่วนเพื่อใช้ในการอัปเดตซึ่งจะไม่ได้ใช้ข้อมูลทั้งหมดและหลังจากนั้นสมการจะถูกรวบรวมเข้าสู่คำตอบที่ใกล้เคียง โดยวิธีแบบ SGD นั้นเร็วกว่าวิธีการหาเกรเดียนต์แบบดั้งเดิม โดยใช้สูตรดังนี้

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t)$$

$$W_{t+1} = W_t + V_{t+1}$$

RMSprop

เป็นวิธีปรับอัตราการเรียนรู้ที่มีประสิทธิภาพตัวหนึ่ง โดยในปัจจุบันยังไม่ได้ถูกตีพิมพ์แต่อย่างใด เป็นเพียงการนำเสนอของ Geoff Hinton ในห้องเรียนของ Coursera เท่านั้น ซึ่งการอัปเดต RMSProp จะปรับมาจากวิธีการ Adagrad ด้วยวิธีการที่ง่าย โดยจะทำการลดอัตราการเรียนรู้ลงเรื่อยๆ โดยใช้สูตรดังนี้

$$MS((W_t)_i) = \delta MS((W_{t-1})_i) + (1 - \delta) (\nabla L(W_t))_i^2$$

$$(W_{t+1})_i = (W_t)_i - \alpha \frac{(\nabla L(W_t))_i}{\sqrt{MS((W_t)_i)}}$$

Adam

ย่อมาจาก Adaptive Moment Estimation เป็นอีกหนึ่งวิธีที่มีการคำนวณอัตราการเรียนรู้ โดยปรับให้เข้ากับแต่ละพารามิเตอร์ Adam จะแก้ปัญหาค่าไบแอสโดยการคำนวณค่าที่ถูกต้องก่อนในตอนแรกและจากนั้นทำการประเมิน หลังจากนั้นจะนำพารามิเตอร์ที่ได้มาทำการปรับปรุงพารามิเตอร์ใหม่ โดยใช้สูตรดังนี้

$$(m_t)_i = \beta_1 (m_{t-1})_i + (1 - \beta_1) (\nabla L(W_t))_i,$$

$$(v_t)_i = \beta_2 (v_{t-1})_i + (1 - \beta_2) (\nabla L(W_t))_i^2$$

NAG

ย่อมาจาก Nesterov Acceleration Gradient เป็นอีกหนึ่งวิธีสำหรับการคำนวณอัตราการเรียนรู้ โดยใช้สูตรดังนี้

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t + \mu V_t)$$

$$W_{t+1} = W_t + V_{t+1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เกี่ยวข้องกับระหว่าง gradient descent และ โปรแกรม

- 1) **Learning Rate** หรืออัตราการเรียนรู้ คือเป็นค่าคงที่ซึ่งจะบอกค่าถ่วงน้ำหนักจะปรับค่าไปมากน้อยแค่ไหน ยิ่งเยอะแสดงว่าแต่ละรอบค่าถ่วงน้ำหนักจะปรับไปเยอะ ซึ่งก็อาจจะทำให้ ลู่เข้าเร็วขึ้นใช้เวลาฝึกสอนน้อยลง แต่บางครั้งถ้าค่าถ่วงน้ำหนักโดนปรับมากไปก็อาจจะทำให้ข้ามค่าถ่วงน้ำหนักที่ดีที่สุดไปได้ ดังนั้นแล้วโดยปกติจะเริ่มจากอัตราการเรียนรู้ค่าหนึ่งที่น้อยๆแล้วค่อยๆเพิ่มมาเมื่อจำนวนรอบมากขึ้น
- 2) **Momentum** คือ เป็นตัวสำหรับควบคุมอัตราการเรียนรู้เพื่อช่วยในกรณีที่เกิดค่าความผิดพลาดมากขึ้นเรื่อยๆ ดังนั้นการนำโมเมนตัมเข้าไปช่วยจะทำให้ค่าถ่วงน้ำหนักก้าวกระโดดเร็วขึ้นเพื่อข้ามค่าพวกนั้นไป หรือกล่าวคือการช่วยให้โครงข่ายหลุดจาก local minima
- 3) **Epoch** คือ รอบการทำงานของ การส่งข้อมูลแบบ backpropagation ของชุดข้อมูลฝึกสอนทั้งหมด โดยหนึ่งรอบหมายถึงชุดข้อมูลทั้งหมดเข้าไปในโครงข่ายด้วยขั้นตอนที่ส่งข้อมูลไปข้างหน้าแล้วส่งค่า loss กลับมาแบบย้อนกลับ
- 4) **Batch size** คือ การกำหนดจำนวนตัวอย่างที่จะผ่านเข้าไปในโครงข่าย และการใช้ batch size จะมีผลต่อความเร็วที่ใช้ในการฝึกสอนและใช้พื้นที่หน่วยความจำน้อย
- 5) **Iteration** คือ จำนวนรอบในการผ่านเข้าโครงข่าย โดยหนึ่งรอบหมายถึงผ่านการทำงานแบบ backpropagation แล้ว
- 6) **Learning rate policy** คือ รูปแบบการปรับค่าอัตราการเรียนรู้ซึ่งไฮเปอร์พารามิเตอร์หลายตัวที่สามารถปรับได้

```

base_lr: 0.01 # begin training at a learning rate of 0.01 = 1e-2
lr_policy: "step" # learning rate policy: drop the learning rate in "steps"
# by a factor of gamma every stepsize iterations
gamma: 0.1 # drop the learning rate by a factor of 10
# (i.e., multiply it by a factor of gamma = 0.1)
stepsize: 100000 # drop the learning rate every 100K iterations
max_iter: 350000 # train for 350K iterations total
momentum: 0.9

```

รูปที่ 2.24 รูปแบบไฮเปอร์พารามิเตอร์ใน learning rate policy

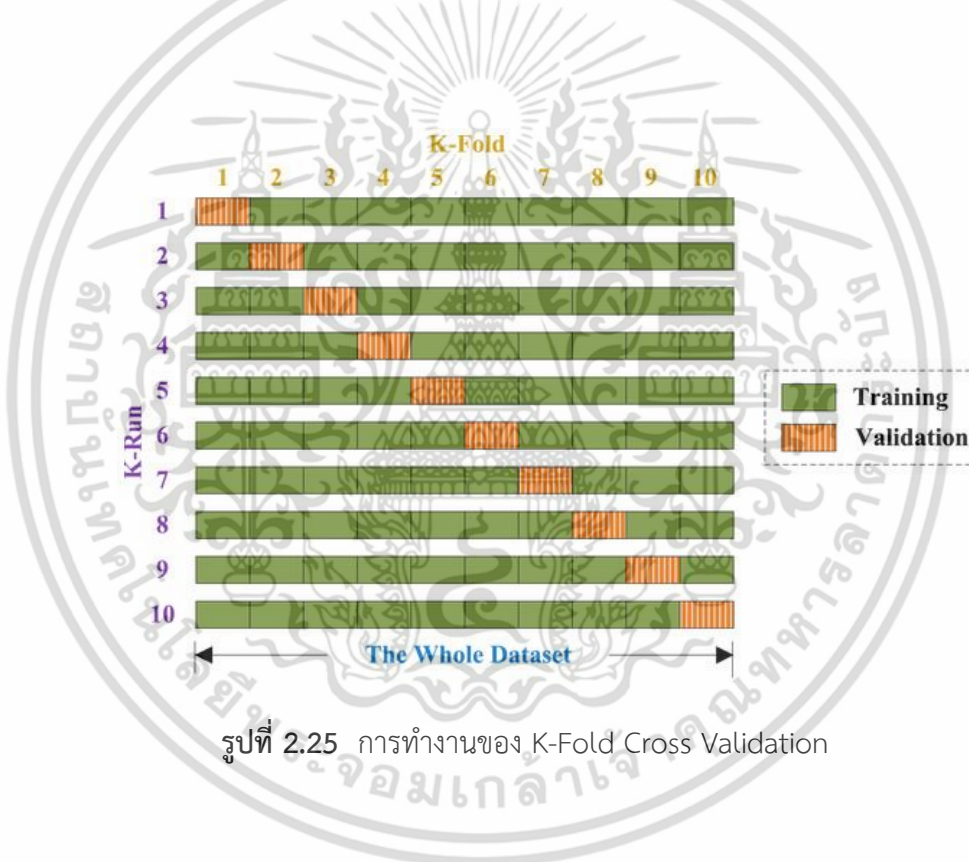
- 7) **Step decay** อัตราการเรียนรู้จะลดลงทุกๆ stepsize และจะลดลงเป็นจำนวน gamma เท่า โดย stepsize หมายถึงจำนวน iteration
- 8) **Exponential decay** อัตราการเรียนรู้จะลดลงตามสูตรต่อไปนี้ $\alpha = \alpha_0 e^{-kt}$ โดยที่ α_0, k เป็นไฮเปอร์พารามิเตอร์ และ t เป็นจำนวน iteration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 รูปแบบการวัดประสิทธิภาพ

K-Fold Cross Validation เป็นวิธีที่ใช้ในการทดสอบประสิทธิภาพของโมเดลเนื่องจากผลที่ได้มีความน่าเชื่อถือ การวัดประสิทธิภาพด้วยวิธี Cross Validation นี้จะทำการแบ่งข้อมูลออกเป็น K ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่าๆกันหลังจากนั้นนำข้อมูล K-1 ส่วนไว้สำหรับฝึกสอนโครงข่าย และ 1 ส่วนที่เหลือมาทำการทดสอบทำซ้ำจนครบ K รอบ โดยประสิทธิภาพโดยรวมของโมเดล ได้จากการหาค่าเฉลี่ย K รอบ

ยกตัวอย่างเช่น 10-Fold Cross Validation โดยพฤติกรรมของการทำ 10-Fold จะทำการแบ่งข้อมูลออกเป็น 10 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หลังจากนั้นจะนำข้อมูลจำนวน 9 ส่วนไปสร้างโมเดล เมื่อฝึกสอนโมเดลเสร็จก็จะใช้ข้อมูล 1 ส่วน เป็นตัวทดสอบประสิทธิภาพของโมเดล ทำวนไปเช่นนี้จนครบจำนวนที่แบ่งไว้ นั่นหมายความว่ามีการฝึกสอนและทดสอบถึง 10 ครั้ง



รูปที่ 2.25 การทำงานของ K-Fold Cross Validation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9 ดัชนีวัดประสิทธิภาพ

ขั้นตอนสุดท้ายในการสร้างโมเดลจำเป็นอย่างยิ่งที่ต้องทำการวัดประสิทธิภาพของโมเดลเพื่อทดสอบว่าโมเดลที่เราสร้างนั้นมีผลการทำงานที่มีค่าความถูกต้องเป็นเท่าไร หรือเพื่อเป็นการทดสอบและเปรียบเทียบความแม่นยำของแต่ละโมเดล โดยจะมีตัววัดประสิทธิภาพดังนี้

1) **Confusion matrix** คือตารางที่มีการประเมินผลลัพธ์การทำนายเปรียบเทียบกับผลลัพธ์ที่เกิดขึ้นจริง โดยใช้กับการวิเคราะห์ปัญหาที่เป็นรูปแบบของการจำแนกแบบไบนารี ซึ่งในตารางจะประกอบไปด้วย True Positive (TP) False Positive (FP) False Negative (FN) และ True Negative (TN)

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

รูปที่ 2.26 ตาราง confusion matrix

- True Positive (TP) คือ จำนวนที่ทำนายตรงกับข้อมูลจริงในคลาสที่กำลังพิจารณา
- True Negative (TN) คือ จำนวนที่ทำนายตรงกับข้อมูลจริงในคลาสที่ไม่ได้กำลังพิจารณา
- False Positive (FP) คือ จำนวนที่ทายผิดเป็นคลาสที่กำลังพิจารณา
- False Negative (FN) คือ จำนวนที่ทายผิดเป็นคลาสที่ไม่ได้กำลังพิจารณา

2) **Accuracy** คือ ค่าที่บอกว่าโมเดลสามารถทำนายได้แม่นยำขนาดไหน ซึ่งหาได้จาก

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) **Precision** คือ ค่าที่บอกว่าโมเดลทำนายได้ว่าจริงถูกต้องเท่าไรของข้อมูลทั้งหมดในคลาสที่กำลังพิจารณา ซึ่งหาได้จาก

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

4) **Recall** หรือ True Positive Rate (TPR) หรือ Sensitivity คือ ค่าที่บอกว่าโมเดลทำนายได้ว่าจริง เป็นอัตราส่วนเท่าไรของข้อมูลจริงทั้งหมด ซึ่งหาได้จาก

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

5) **True Negative Rate (TNR)** หรือ Specificity คือ ค่าที่บอกว่าโมเดลทำนายได้ว่าไม่จริง เป็นอัตราส่วนเท่าไรของข้อมูลจริงทั้งหมด ซึ่งหาได้จาก

$$TNR = \frac{(TN)}{(TN + FP)}$$

6) **False Positive Rate (TPR)** หรือ Fall-out คือ ค่าที่บอกว่าโมเดลทำนายได้ว่าจริงเป็นอัตราส่วนเท่าไรของข้อมูลไม่จริงทั้งหมด ซึ่งหาได้จาก

$$TPR = \frac{(FP)}{(TN + FP)}$$

7) **False Negative Rate (FNR)** คือ ค่าที่บอกว่าโมเดลทำนายว่าไม่จริงเป็นอัตราส่วนเท่าไรของข้อมูลจริงทั้งหมด ซึ่งหาได้จาก

$$FNR = \frac{(FN)}{(TP + FN)}$$

การหาค่า True Positive Rate และ False Positive Rate จะขึ้นอยู่กับค่าของ Threshold ที่เรากำหนด ซึ่งค่าที่ว่ามีเหมือนเป็นเกณฑ์การทำนายที่ระบุว่าถ้าค่ามากกว่าให้อยู่ในคลาสนี้หรือค่าน้อยกว่าอยู่ในอีกคลาสหนึ่งทำให้สามารถจำแนกออกมาได้ เป็นต้น โดยสิ่งที่ต้องคำนึงคือต้องให้ค่า

True Positive Rate สูงสุด และ False Positive Rate ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) **F-score** หรือ F-measure เป็นการวัดค่าความสัมพันธ์ระหว่าง Precision และ Recall ในเชิงฮาร์โมนิก เหมาะสำหรับฐานข้อมูลสารสนเทศที่มีขนาดใหญ่ และมักจะไม่ทราบข้อมูลภาพที่ถูกต้องทั้งหมดมีอยู่เท่าใด ทำให้ต้องทำการประมาณโดยใช้การสุ่มตัวอย่างตามหลักทางสถิติหรือด้วยวิธีอื่นด้วย โดยทั่วไปจะเป็นการหาค่า F-score ซึ่งแสดงสูตรได้ดังนี้

$$F - score = \frac{(2 * precision * recall)}{(precision + recall)}$$

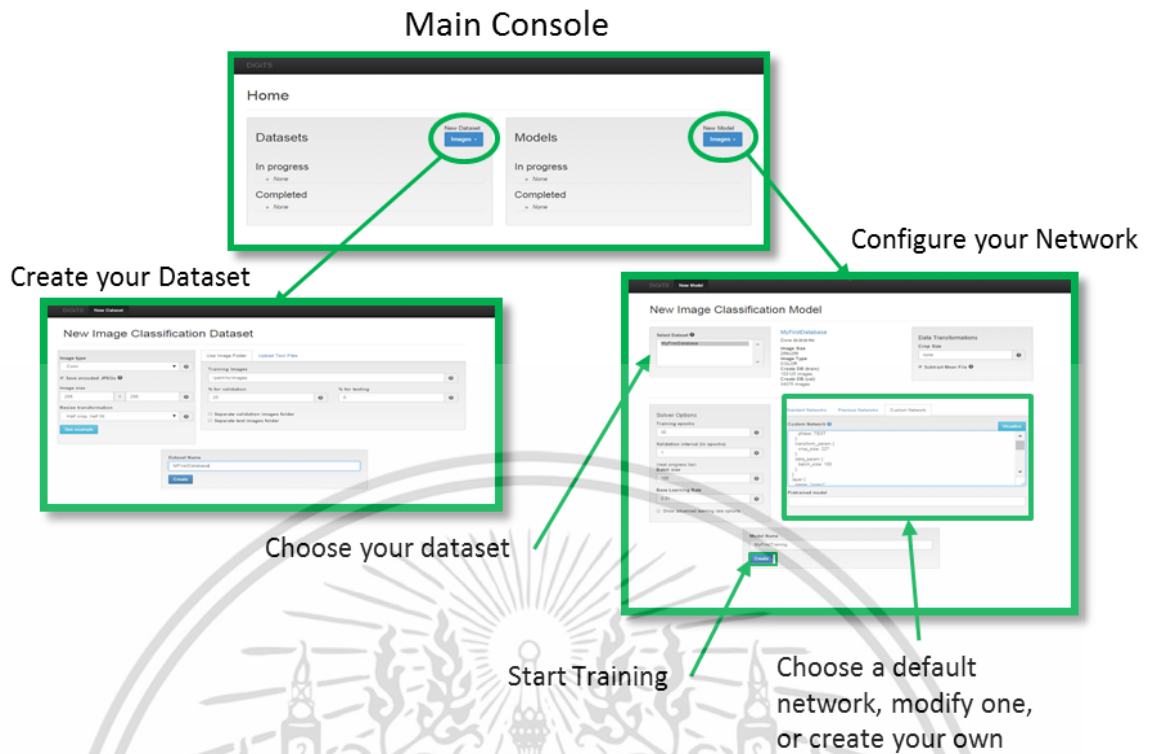
2.2 ส่วนของโปรแกรมและภาษาที่นำมาใช้

2.2.1 Nvidia DIGITS

Nvidia DIGITS เป็นซอฟต์แวร์สำหรับฝึกโครงข่าย Deep Learning ซึ่งมาจากชื่อเต็มว่า Deep Learning GPU Training System (DIGITS) เป็นซอฟต์แวร์เพื่อการสร้างโมเดลโครงข่ายประสาทเทียมแบบเชิงลึก (deep neural network - DNN)

DIGITS ทำให้การออกแบบโครงข่ายประสาทเทียมด้วยตัวเองเป็นไปได้ง่ายด้วยการคอนฟิกเครือข่ายว่าต้องการจำรูปร่างแบบใด แล้วใส่ชุดข้อมูลเพื่อฝึกโครงข่ายประสาทเทียมเพื่อให้ได้โมเดลไปใช้งาน ซึ่ง DIGITS จะให้บริการในหลายๆด้าน เช่น งานจำพวกจัดการข้อมูล การออกแบบการฝึกสอนโครงข่ายประสาทเทียมบนระบบที่มีหลายจีพียู การตรวจสอบประสิทธิภาพตามเวลาที่เกิดขึ้นจริงด้วยการทำให้เห็นภาพขั้นสูง และสามารถเลือกโมเดลที่มีประสิทธิภาพมากที่สุดมาแสดงผลเพื่อการใช้งาน

DIGITS มีการปฏิสัมพันธ์กับผู้ใช้อย่างสมบูรณ์เพื่อที่จะช่วยนักวิทยาศาสตร์โฟกัสไปที่การออกแบบและการฝึกสอนโครงข่ายมากกว่าที่จะไปสนใจการเขียนโปรแกรมและการดีบัก



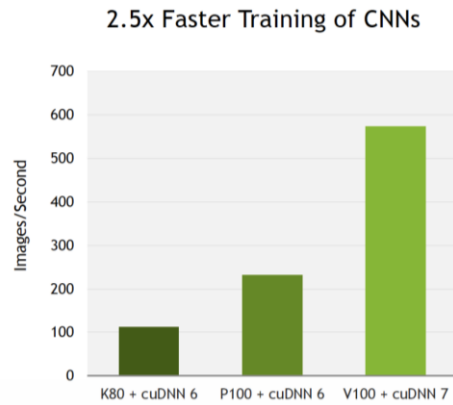
รูปที่ 2.27 ตัวอย่างโปรแกรม Nvidia DIGITS

2.2.2 ส่วนที่เกี่ยวข้องกับโปรแกรม

1) **CUDA** คือสถาปัตยกรรมหนึ่งที่ถูกพัฒนาขึ้นโดย Nvidia ซึ่งจะประกอบไปด้วยกราฟิกการ์ด รูปแบบการเขียนโปรแกรม และคอมไพเลอร์ ที่จะช่วยในการเขียนโปรแกรมแบบขนาน (parallel programming) โดยจะเข้ามาช่วย CPU ไม่ให้ทำงานหนักจนเกินไป โดยสามารถสร้างชุดคำสั่งได้หลากหลายภาษา เช่น Java Python เป็นต้น

2) **cuDNN** เป็นไลบรารีขั้นพื้นฐานของตัวเร่งการแสดงผลกราฟิก (GPU-accelerated) โดยจะจัดเตรียมเครื่องมือที่ถูกปรับแต่งไว้อย่างดีแล้วให้สำหรับงานหลักที่เกิดขึ้นเป็นประจำ อย่างเช่น forward และ backward โดย cuDNN เป็นส่วนหนึ่งของเครื่องมือพัฒนา deep learning หรือ Nvidia Deep Learning SDK ซึ่งนักพัฒนาเชื่อมั่นใน cuDNN ว่ามีประสิทธิภาพของตัวเร่งการแสดงผลกราฟิกที่สูง โดย cuDNN จะดึงเฟรมเวิร์กมาใช้ร่วมด้วย เช่น Caffe TensorFlow เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



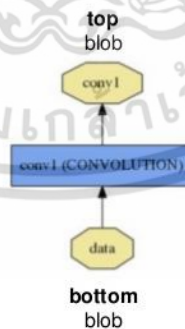
รูปที่ 2.28 กราฟเปรียบเทียบประสิทธิภาพการทำงานเมื่อใช้ cuDNN

3) การจัดเก็บข้อมูล (Data storage) การจัดเก็บข้อมูลและการสื่อสารข้อมูลจะเป็นแบบอาร์เรย์ 4 มิติซึ่งเรียกว่า blob หรือ binary large object โดยแทนข้อมูลประเภทต่างๆ เช่น ภาพ เสียง ของวัตถุ้มัลติมีเดียต่างๆ ซึ่งเป็นการทำงานร่วมกันระหว่างหน่วยประมวลผลกลางในการทำงานของคอมพิวเตอร์ (CPU) และ หน่วยประมวลผลด้านกราฟิก 3 มิติ (GPU) จะเป็นการโหลดข้อมูลจากการจัดเก็บข้อมูลไปให้ blob ในการแปลรหัสใน CPU เพื่อประมวลผลเรียกว่า CUDA โดยจะมีการเข้ามาช่วย CPU ประมวลผลชุดคำสั่งบางชนิด เพื่อแบ่งเบาภาระของ CPU ไม่ให้ทำงานหนักจนเกินไป แทนที่การ์ดจอจะประมวลผลภาพหรือแสดงผลภาพเพียงอย่างเดียว โดยสามารถสร้างชุดคำสั่งผ่านโปรแกรมต่างๆ เช่น Java Fortran Python C C++ C# ดังรูป 2.29

```

name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
... definition ...

```



รูปที่ 2.29 การทำงานของการจัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) **LMDB** หรือ **Lightning Memory-Mapped Database** เป็นฐานข้อมูลชนิดหนึ่งที่ได้รับการออกแบบมาเพื่ออำนวยความสะดวกในหน่วยความจำเสมือนที่พบได้ในระบบปฏิบัติการที่ทันสมัย มีวิธีการควบคุมการใช้งานพร้อมกัน หรือ **Multi-Version Concurrency Control** และหน่วยความจำคอมพิวเตอร์หรือที่เรียกว่า **single-level store** โดย **LMDB** เป็นหน่วยความจำที่มีขนาดเล็ก แต่มีความสามารถค่อนข้างมาก เช่น

- การทำทรานแซกชันที่เต็มรูปแบบ
- สามารถจับคู่หน่วยความจำเพื่อให้สามารถค้นหา
- ไม่มีการแคชในระดับแอปพลิเคชันสามารถใช้ประโยชน์จากแคชบัฟเฟอร์ได้อย่างเต็มที่
- **Reader** และ **Writer** จะไม่ปิดกั้นกันและกันและการจะไม่มีการหยุดชะงักหรือที่เรียกว่า **deadlock**
- สนับสนุน **multi-thread** และ **multi-process concurrency** สภาพแวดล้อมมาจถูกเปิดโดยใช้โปรเซสเซอร์หลายตัวบนโฮสต์เดียวกัน

5) **HDF** หรือ **Hierarchical Data Format** เป็นชุดของรูปแบบไฟล์ถูกออกแบบมาเพื่อจัดเก็บและจัดการข้อมูลจำนวนมาก การพัฒนาครั้งแรกเกิดขึ้นที่ศูนย์แห่งชาติสำหรับการประยุกต์ใช้เครื่องคำนวณที่มีความสามารถสูง ซึ่งได้รับการสนับสนุนจากกลุ่ม **HDF** โดยบริษัทที่ไม่แสวงผลกำไรซึ่งมีภารกิจหลักคือเพื่อให้แน่ใจว่าจะพัฒนาเทคโนโลยี **HDF5** อย่างต่อเนื่องและการเข้าถึงข้อมูลที่เก็บใน **HDF** อย่างต่อเนื่อง เพื่อให้สอดคล้องกับเป้าหมายนี้ ไลบรารีของ **HDF** และเครื่องมือที่เกี่ยวข้องสามารถใช้งานได้ภายใต้ใบอนุญาตแบบเสรีนิยม **BSD** สำหรับการใช้งานทั่วไป **HDF** ได้รับการสนับสนุนโดยแพลตฟอร์มซอฟต์แวร์เชิงพาณิชย์และซอฟต์แวร์ที่ไม่ใช่เชิงพาณิชย์มากมายรวมถึง **Java** **MATLAB** **Scilab** **Octave** **Mathematica** **IDL** **Python** **R** และ **Julia** ซึ่ง **HDF5** เป็นเวอร์ชันในปัจจุบัน

6) **BLAS** หรือ **Basic Linear Algebra Subroutines** คือ การจัดเตรียมสมการพีชคณิตเชิงเส้นในการดำเนินการแบบ **matrix-matrix** แบบ **matrix-vector** **BLAS** และแบบ **ESSL** สามารถช่วยประหยัดเวลาในการปรับการคำนวณทางคณิตศาสตร์ และยังคงจัดเตรียมผลการทำงานที่ดีกว่าหรือการทำให้เหมาะสมที่สุด การคำนวณทางคณิตศาสตร์ที่เขียนโค้ดขึ้น สามารถเรียกฟังก์ชันได้จากทั้งไลบรารีจาก **FORTRAN** **C** และ **C++**

7) **Boost** คือ ทำหน้าที่ในการเร่งความเร็วของโปรเซสเซอร์และเพิ่มประสิทธิภาพระบบกราฟิกเพื่อรองรับภาระงานสูงสุด ทำให้แกนโปรเซสเซอร์สามารถเร่งความเร็วให้เร็วกว่าอัตราความถี่ปฏิบัติการอัตโนมัติ

8) **Protobuf** **Glog** และ **Gflags** คือสิ่งที่ช่วยในการแก้จุดบกพร่องและการใช้งานขั้นสูง

9) **Ubuntu** เป็นระบบปฏิบัติการคอมพิวเตอร์ที่เป็นระบบปฏิบัติการแบบเปิดซึ่งมีพื้นฐานบนลินุกซ์ดิสทริบิวชัน โดยจะนำมาใช้ร่วมกับ **Caffe** **CUDA** **BLAS** และ **cuDNN**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 10) Job Information คือ อธิบายข้อมูลต่างๆของฐานข้อมูลที่ได้ทำการสร้างชุดข้อมูลไว้
- 11) Job status คือ แสดงสถานะการทำงานและการประมวลผลของชุดข้อมูล
- 12) Create DB (Train) คือ ขั้นตอนการสร้างฐานข้อมูลหรือการฝึกสอนจะแสดงผลออกมาเป็นกราฟแท่งแสดงจำนวนของชุดข้อมูลที่ได้ทำการอัปโหลดเข้าไป

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 Using Convolutional Neural Networks and Transfer Learning to Perform Yelp Restaurant Photo Classification

วิจัยนี้เกี่ยวข้องกับการจำแนกร้านอาหารจากรูปที่ผู้ใช้งาน Yelp อัปโหลดเข้าไปในเว็บไซต์ โดยจะสกัดข้อมูลออกมาเป็นเลเบลทั้ง 9 แบบดังนี้

- | | |
|----------------------|----------------------------|
| 1) Good for Lunch | 6) Has-table service |
| 2) Good for dinner | 7) Ambiance-is-classy |
| 3) Takes Reservation | 8) Good for kids |
| 4) Outdoor-seating | 9) Restaurant is expensive |
| 5) Has-alcohol | |

ซึ่งใช้ช่วยในการแนะนำร้านอาหารที่ผู้ใช้อาจจะชอบ โดยจะใช้ VGGNet เป็น feature extractor และใช้ SVM เป็น classifier บนชุดข้อมูลฝึกสอนมีทั้งหมด 234,842 ภาพ จากร้านอาหารประมาณ 2,000 ร้าน โดย validation set จะใช้ภาพจาก 100 ร้าน และปรับขนาดภาพฝึกสอนเท่ากับ $224 \times 224 \times 3$ เนื่องจากเป็นขนาดที่เหมาะสม VGGNet ใช้เวลาฝึกสอน 14 ชม. และตั้งชื่อว่า YelpNet ซึ่งจากการวัดผลประสิทธิภาพจะได้ค่า F1-score เท่ากับ 0.658



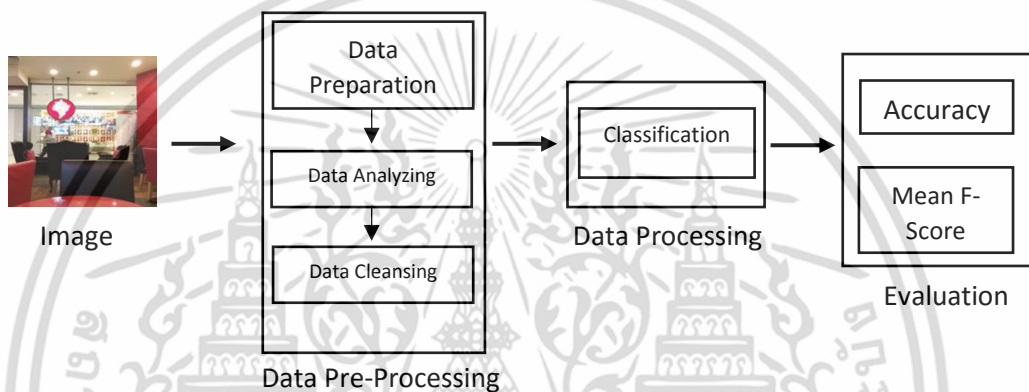
รูปที่ 2.30 ภาพตัวแทนของแต่ละคลาสของ YelpNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

ในการศึกษาปัญหาพิเศษในครั้งนี้ได้มีการดึงข้อมูลรูปภาพเพื่อเตรียมเข้าสู่ขั้นตอนการดำเนินงานโดยมีการแบ่งเป็น 3 ขั้นตอนหลักดังต่อไปนี้ คือ ขั้นตอนการเตรียมข้อมูล (Data Pre-Processing) ขั้นตอนการประมวลผล (Data Processing) และขั้นตอนการการวัดประสิทธิภาพ (Evaluation) โดยขั้นตอนทั้งหมดแสดงไว้ในแผนภาพด้านล่าง



รูปที่ 3.1 ขั้นตอนวิธีการดำเนินงาน

3.1 ขั้นตอนก่อนการประมวลผล (Data Pre-Processing)

ขั้นตอนก่อนการประมวลผลประกอบไปด้วย 3 ขั้นตอนย่อยดังนี้

3.1.1 การเตรียมข้อมูล (Data Preparation)

เริ่มต้นด้วยการดาวน์โหลดชุดข้อมูลจากฐานข้อมูลลงใน จำนวน 12,304 รูปภาพ ซึ่งมีจำนวนทั้งหมด 5 คลาส ที่ประกอบไปด้วย คลาสอาหารหรือเครื่องดื่ม (food or drink) คลาสสมุดเมนู (menu) คลาสบรรยากาศร้าน (inside) คลาสหน้าร้าน (storeFront) และคลาสโปรไฟล์ (profile) และชุดข้อมูลรูปภาพที่ต้องการนำมาจำแนก 9,919 รูปภาพ

3.1.2 การวิเคราะห์ข้อมูล (Data Analyzing)

จากการวิเคราะห์ชุดข้อมูลทั้งหมดพบว่า ภาพในคลาสบรรยากาศมีกลุ่มที่มีการทับซ้อนกับคลาสหน้าร้าน อีกทั้งมีภาพที่ถูกแท็กผิดกลุ่มอย่างชัดเจน อันเนื่องจากการกำหนดคำศัพท์ที่ใช้แท็กภาพจากแหล่งคลังภาพมีขอบเขตความหมายที่กว้างและไม่ชัดเจน เช่นคำว่า บรรยากาศร้าน สามารถตีความหมายได้ว่าบรรยากาศทั้งภายในและภายนอก และเมื่อมีการนำรูปภาพที่เป็นหน้าร้านเข้าสู่ระบบแต่แท็กไปที่บรรยากาศร้าน จึงทำให้ภาพที่ถูกแท็กไม่อยู่ตรงตามคลาส ซึ่งชุดข้อมูลที่ได้มานั้นมีภาพเหล่านี้อยู่จำนวนหนึ่งจึงต้องดำเนินการกับภาพให้มีความหมายตรงตามที่จะควรจะเป็นโดยจะใช้วิธีในข้อ 3.1.3 เพื่อให้ได้โมเดลการจำแนกที่ตอบโจทย์การนำไปใช้งานจริง

3.1.3 การกำจัดสิ่งรบกวน (Data Cleansing)

ทำการตัดรูปภาพออกจากคลาสบรรยากาศตามที่ได้วิเคราะห์ไว้ใน 3.1.2

3.2 ขั้นตอนการแบ่งข้อมูลด้วยวิธีการ K-Fold Cross Validation

ทำการแบ่งรูปภาพตามวิธีการของ cross validation โดยกำหนด $k = 10$ ซึ่งมีวิธีการคือ นำรูปภาพหนึ่งส่วนจากทุกๆคลาสมารวมกันเป็นชุดข้อมูลย่อยที่ 1 และกระทำเช่นเดียวกันจนถึงชุดข้อมูลย่อยที่ 10 โดยในแต่ละชุดข้อมูลย่อยจะต้องมีรูปภาพที่ไม่ซ้ำกัน ดังนั้นแล้วจะได้ผลลัพธ์ทั้งสิ้น 10 ชุดข้อมูลย่อย ซึ่งนำมาใช้งานในแต่ละรอบ ทั้งหมดเป็น 10 รอบและแต่ละรอบจะถูกแบ่งเป็นชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลทดสอบ (validation set) ตามหลักการของของ cross validation

3.3 ขั้นตอนการแปลงข้อมูลรูปภาพ

ขั้นตอนการแปลงข้อมูลรูปภาพประกอบไปด้วย 4 ขั้นตอนย่อยดังนี้

3.3.1 การเลือกชนิดรูปภาพ

มีสองตัวเลือกคือ ภาพสีและภาพขาวดำซึ่งในการทดลองนี้ เลือกใช้ภาพสี

3.3.2 การกำหนดให้ขนาดภาพ

การกำหนดให้ขนาดภาพหลังจากปรับขนาดภาพเท่ากับ 256x256 พิกเซล

3.3.3 การปรับขนาดภาพ

มีรูปแบบให้เลือกทั้งหมด 4 ชนิด คือ

- crop ยึดความกว้างและตัดขอบบนและล่างให้เท่ากับขนาดที่ได้กำหนดไว้
- squash ยึดความสูงและยืดรูปให้กว้างให้เท่ากับขนาดที่ได้กำหนดไว้
- fill ยึดความสูงและเพิ่มขอบด้านซ้ายและขวาให้เท่ากับขนาดที่ได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- half fill half crop คือการรวมกันของ crop และ fill โดยจะทำการตัดขอบบนและล่าง และเพิ่มขอบด้านซ้ายและขวาให้เท่ากับขนาดที่ได้กำหนดไว้ ซึ่งในการทดลองนี้เลือกใช้ squash

3.3.4 การเลือกนามสกุลให้กับรูปภาพ

โดยมีให้เลือกสองรูปแบบคือ .PNG และ .JPG ซึ่งในการทดลองจะเลือกเป็น .JPG

รูปที่ 3.2 หน้าต่างโปรแกรม Nvidia DIGITS ที่ใช้ในการแปลงรูปภาพ

3.4 ขั้นตอนสร้างฐานข้อมูลของชุดข้อมูล

ทำการเลือกสร้างฐานข้อมูลโดยมีให้เลือกใช้สองรูปแบบคือ LMDB และ HDF5 แต่ในการทดลองนี้จะเลือกใช้ LMDB ในการเก็บชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ขั้นตอนการสร้างโมเดล

สำหรับขั้นตอนของการสร้างโมเดลประกอบไปด้วย 2 ขั้นตอนย่อย คือ การเลือกชุดข้อมูลฝึกสอนในรอบที่ต้องการ และ การตั้งค่าตัวเลือกต่างๆให้กับโมเดล

3.5.1 การเลือกชุดข้อมูลฝึกสอนในรอบที่ต้องการ

เริ่มต้นโดยการเลือกชุดข้อมูลฝึกสอนในรอบต่างๆที่ต้องการ

3.5.2 การตั้งค่าให้กับโมเดลโดยมีให้เลือกดังนี้

- 1) เลือกสถาปัตยกรรมของโครงข่ายโดยการทดลองนี้จะเลือกใช้ดังนี้
 - AlexNet
 - GoogLeNet
- 2) การตั้งค่า solver
 - จำนวน training epoch เท่ากับ 30
 - snapshot interval เท่ากับ 1 ซึ่งเป็นค่า default
 - validation interval เท่ากับ 1 ซึ่งเป็นค่า default
 - random seed เท่ากับ none ซึ่งเป็นค่า default
 - จำนวน batch size กำหนดใช้เป็นค่า default ของแต่ละโครงข่าย
 - กำหนดชนิด solver โดยที่จะเลือกใช้ในการทดลองมีทั้งหมด 4 ชนิด ดังนี้
 - Stochastic Gradient Descent (SGD)
 - Nesterov Accelerated Gradient (NAG)
 - Adaptive Moment Estimation (ADAM)
 - RMSprop
 - กำหนดค่าอัตราการเรียนรู้เริ่มต้น
 - ถ้าเลือกใช้ SGD หรือ NAG จะกำหนดอัตราการเรียนรู้ให้เป็น 0.01 ซึ่งเป็นค่า default
 - ถ้าเลือกใช้ ADAM หรือ RMSprop จะกำหนดอัตราการเรียนรู้ ให้เป็น 0.0001
 - ค่า gamma ใช้เป็นค่า default คือ 0.1
 - learning rate policy จะเลือกใช้เป็นค่า default คือ แบบ Step Down
 - step size กำหนดร้อยละการลดลงของอัตราการเรียนรู้เท่ากับ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ขั้นตอนการทดลอง

ขั้นตอนการทดลองประกอบไปด้วย 3 ขั้นตอนย่อยดังต่อไปนี้

3.6.1 การเปรียบเทียบโมเดลก่อนและหลังการทำการกระบวนการ cleansing ข้อมูล

นำโมเดลที่ยังไม่กำจัดข้อมูลรบกวนและโมเดลที่ผ่านการกำจัดข้อมูลรบกวนแล้วมาเปรียบเทียบค่าความถูกต้องเฉลี่ยว่าโมเดลไหนมีประสิทธิภาพที่ดีกว่า

3.6.2 การเปรียบเทียบโมเดลที่มี solver ที่แตกต่างกัน

นำโมเดลที่ใช้ solver ที่แตกต่างกัน คือ SGD NAG ADAM และ RMSProp มาเปรียบเทียบค่าความถูกต้องเฉลี่ยและค่า Mean F-score บนสถาปัตยกรรมเดียวกัน เพื่อหาโมเดลที่ให้ค่าที่มีประสิทธิภาพที่ดีกว่า

3.6.3 การเปรียบเทียบโมเดลที่ผ่านการ pretrained และ non-pretrained

นำโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 แล้วมาต่อยอดการเรียนรู้ด้วยกับชุดข้อมูลที่มีอยู่และที่เรียนรู้ซึ่งใช้เพียงชุดข้อมูลที่มีอยู่และมาทำการเปรียบเทียบค่าความถูกต้องเฉลี่ยและ Mean F-score เพื่อหาโมเดลที่ให้ค่าที่มีประสิทธิภาพที่ดีกว่า

3.7 ขั้นตอนการวัดประสิทธิภาพ

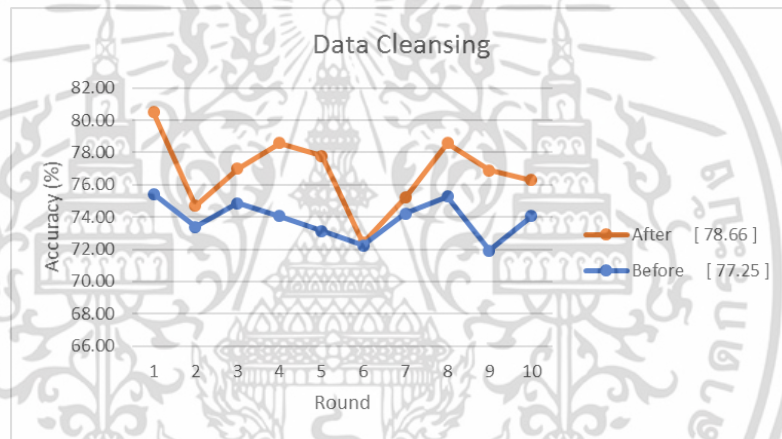
ขั้นตอนการวัดประสิทธิภาพเป็นขั้นตอนสุดท้ายของกระบวนการจำแนกข้อมูลรูปภาพจะนำเอาผลการจำแนกของข้อมูลรูปภาพมาทำการวัดประสิทธิภาพด้วยตัวชี้วัดคือค่าความถูกต้องและ Mean F-Score โดยนำค่าทั้งหมดมาแปลผลและประเมินผลลัพธ์ที่ได้ว่ามีความเหมาะสม หรือตรงกับวัตถุประสงค์ที่ต้องการหรือไม่ ในรูปแบบที่สามารถเข้าใจได้ง่าย เช่น ตาราง หรือ กราฟ

บทที่ 4

ผลการดำเนินการ

ในการทำวิจัยครั้งนี้ได้นำเสนอวิธีการจำแนกประเภทภาพโดยใช้วิธีการเรียนรู้เชิงลึก ซึ่งได้มีการสกัดข้อมูลพีเจอร์และจำแนกด้วยโครงข่ายแบบคอนโวลูชันจากกระบวนการดังกล่าวแล้วในบทที่ 3 ได้แสดงผลการทดลองดังนี้

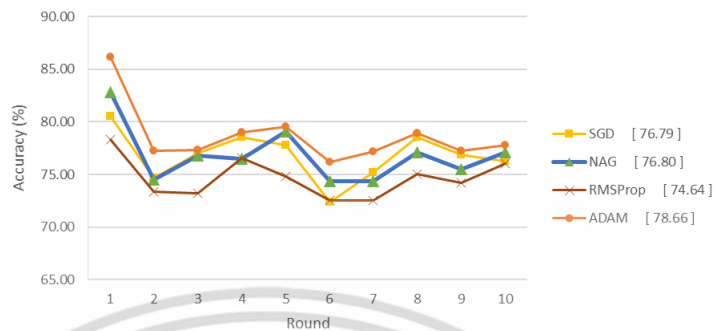
4.1 ผลการทดลองก่อนและหลังการทำกระบวนการกำจัดข้อมูลรบกวนออก (Data Cleansing)



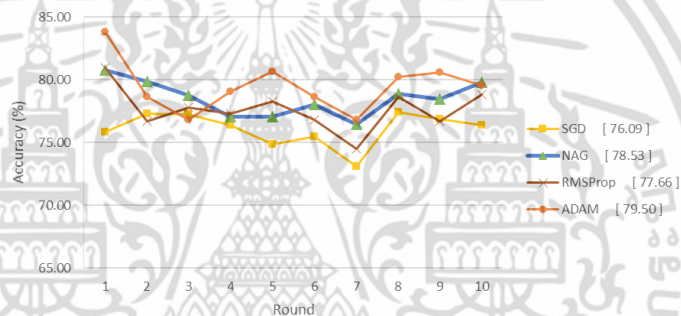
รูปที่ 4.1 ค่าของความถูกต้องก่อนและหลังการทำกระบวนการกำจัดข้อมูลรบกวนออก โดยแสดงค่าความถูกต้องเฉลี่ยอยู่ในวงเล็บ

จากผลการทดลองที่ 4.1 จากการตรวจสอบข้อมูลก่อนและหลังการทำกระบวนการกำจัดข้อมูลรบกวนออก โดยก่อนทำมีค่าความถูกต้องเฉลี่ย 77.25% และหลังทำมีค่าความถูกต้องเฉลี่ย 78.66% ดังนั้นจะพบว่ามีค่าความถูกต้องเฉลี่ยที่เพิ่มขึ้นถึง 2.93% จึงสรุปว่าการกำจัดข้อมูลรบกวนออกช่วยให้โมเดลมีประสิทธิภาพที่ดีขึ้น

4.2 ผลการทดลองการเปรียบเทียบโมเดลที่มี solver ที่แตกต่างกันคือ SGD NAG ADAM และ RMSProp บนสถาปัตยกรรมระหว่าง AlexNet และ GoogLeNet



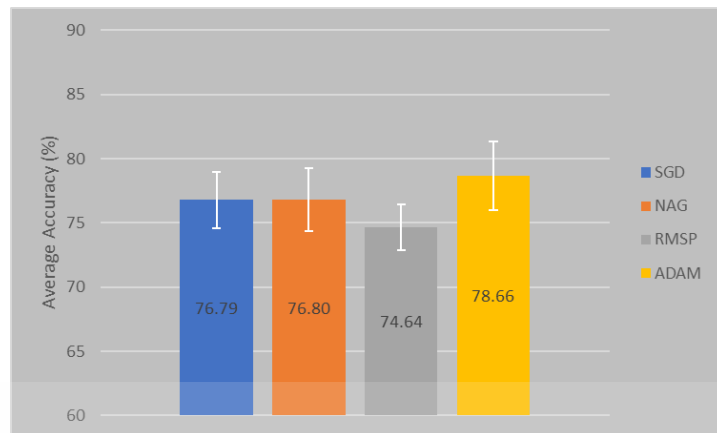
(ก)



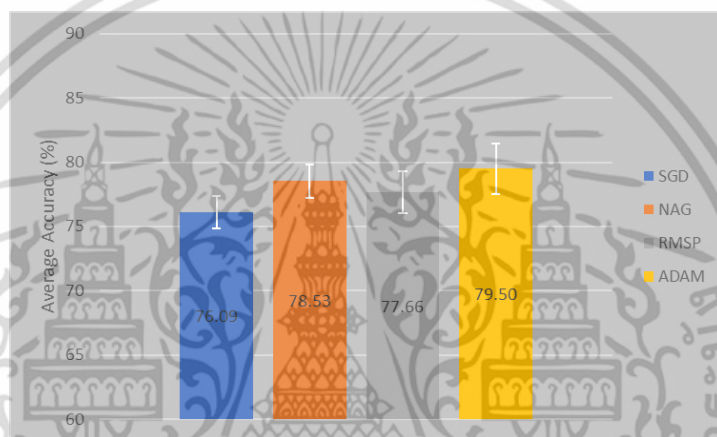
(ข)

รูปที่ 4.2 (ก) กราฟแสดงการเปรียบเทียบค่าความถูกต้องของ solver ที่แตกต่างกันบนสถาปัตยกรรม AlexNet และแสดงค่าความถูกต้องเฉลี่ยภายในวงเล็บ

(ข) กราฟแสดงการเปรียบเทียบค่าความถูกต้องของ solver ที่แตกต่างกันบนสถาปัตยกรรม GoogLeNet และแสดงค่าความถูกต้องเฉลี่ยภายในวงเล็บ



(ก)



(ข)

รูปที่ 4.3 (ก) กราฟแสดงการเปรียบเทียบค่าความถูกต้องเฉลี่ยและ error bar เป็นส่วนเบี่ยงเบนมาตรฐานของข้อมูลของ solver ที่แตกต่างกันบนสถาปัตยกรรม AlexNet
 (ข) กราฟแสดงการเปรียบเทียบค่าความถูกต้องเฉลี่ยและ error bar เป็นส่วนเบี่ยงเบนมาตรฐานของข้อมูลของ solver ที่แตกต่างกันบนสถาปัตยกรรม GoogLeNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าความถูกต้องและค่าเฉลี่ย F-score โดยเปรียบเทียบ solver ระหว่าง SGD NAG ADAM และ RMSProp บนสถาปัตยกรรม AlexNet และ GoogLeNet

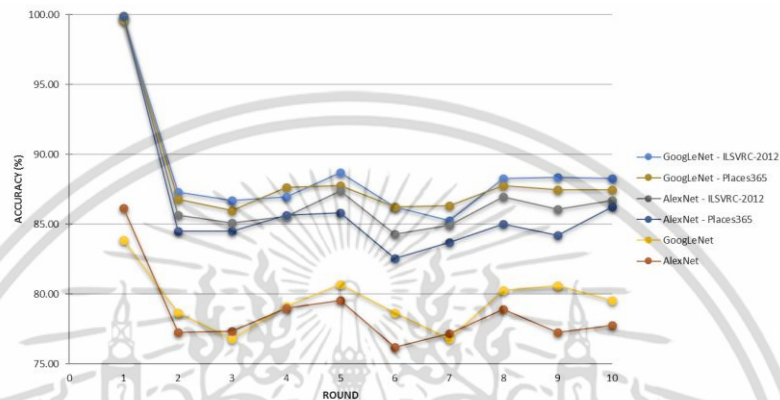
TCV	Accuracy (%)								Mean F-Score							
	AlexNet				GoogLeNet				AlexNet				GoogLeNet			
	SGD	NAG	RMSProp	ADAM	SGD	NAG	RMSProp	ADAM	SGD	NAG	RMSProp	ADAM	SGD	NAG	RMSProp	ADAM
1	80.51	82.80	78.31	86.16	75.84	80.78	80.95	83.86	0.8049	0.8279	0.7823	0.8621	0.7564	0.8065	0.8101	0.8394
2	74.69	74.51	73.37	77.25	77.34	79.89	76.72	78.66	0.7473	0.7450	0.7338	0.7720	0.7724	0.7985	0.7677	0.7863
3	76.98	76.81	73.19	77.34	77.25	78.75	77.78	76.81	0.7723	0.7686	0.7336	0.7744	0.7717	0.7884	0.7734	0.7684
4	78.57	76.46	76.54	79.01	76.37	77.07	77.34	79.10	0.7866	0.7642	0.7654	0.7920	0.7643	0.7714	0.7756	0.7929
5	77.78	79.10	74.78	79.54	74.87	77.07	78.31	80.69	0.7775	0.7910	0.7477	0.7964	0.7460	0.7707	0.7846	0.8078
6	72.40	74.34	72.49	76.19	75.49	78.04	76.81	78.66	0.7262	0.7455	0.7262	0.7638	0.7567	0.7813	0.7714	0.7905
7	75.22	74.34	72.49	77.16	73.10	76.46	74.51	76.81	0.7526	0.7445	0.7234	0.7713	0.7304	0.7637	0.7471	0.7687
8	78.57	77.07	75.04	78.92	77.43	78.92	78.66	80.25	0.7724	0.7724	0.7491	0.7903	0.7899	0.7885	0.8029	0.8029
9	76.90	75.49	74.16	77.25	76.90	78.48	76.72	80.60	0.7701	0.7555	0.7400	0.7737	0.7692	0.7854	0.7675	0.8068
10	76.28	77.07	76.01	77.78	76.37	79.81	78.84	79.54	0.7636	0.7710	0.7616	0.7791	0.7666	0.8010	0.7901	0.7965
AVG	76.79	76.80	74.64	78.66	76.10	78.53	77.66	79.50	0.7688	0.7686	0.7463	0.7875	0.7609	0.7857	0.7776	0.7960
SD	2.182	2.462	1.798	2.684	1.279	1.324	1.618	1.956	0.0215	0.0243	0.0179	0.0268	0.0133	0.0134	0.0160	0.0196

จากผลการทดลองในตารางที่ 4.1 จะได้ว่า SGD NAG RMSProp และ ADAM บนสถาปัตยกรรม AlexNet มีค่าความถูกต้องเฉลี่ยเป็น 76.79% 76.80% 74.64% และ 78.66% ตามลำดับ และมีค่า Mean F-score เป็น 0.7688 0.7686 0.7463 และ 0.7875 ตามลำดับ ส่วนบนสถาปัตยกรรม GoogLeNet มีค่าความถูกต้องเป็น 76.10% 78.53% 77.66% และ 79.50% ตามลำดับ และมีค่า Mean F-score เป็น 0.7609 0.7857 0.7776 และ 0.7960 ตามลำดับ ซึ่งจะเห็นว่าค่าความถูกต้องเฉลี่ยและ Mean F-score มีความสอดคล้องกัน ดังนั้นจะขอกกล่าวถึงเพียงความถูกต้องเฉลี่ย โดย GoogLeNet มีค่าความถูกต้องเฉลี่ยสูงกว่า AlexNet ทุก solver ยกเว้น SGD และมีส่วนเบี่ยงเบนมาตรฐานน้อยกว่า AlexNet ในทุก solver แสดงให้เห็นถึงค่าความถูกต้องมีการกระจายตัวน้อยกว่า

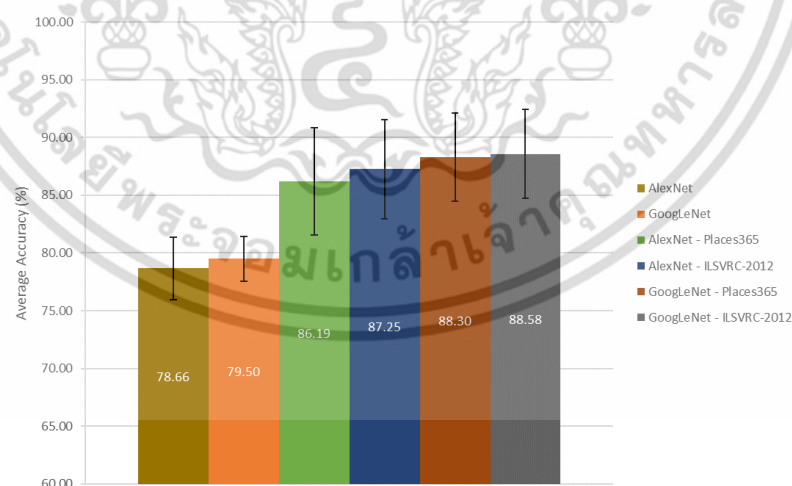
โดยพบว่า solver ที่ให้ค่าความถูกต้องเฉลี่ยมากที่สุดของทั้งสองสถาปัตยกรรมคือ ADAM และเมื่อเปรียบเทียบระหว่างทั้งสองสถาปัตยกรรมที่ใช้ ADAM พบว่าสถาปัตยกรรมของ GoogLeNet นั้นให้ค่าที่ดีกว่าถึง 0.84%

4.3 ผลการทดลองการเปรียบเทียบระหว่างโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 มาต่อยอดการเรียนรู้ด้วยชุดข้อมูลที่มีอยู่ (pretrained) และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่ (non-pretrained) บนสถาปัตยกรรม AlexNet และ GoogLeNet

4.3.1 ผลการแสดงความถูกต้อง



รูปที่ 4.4 กราฟแสดงความถูกต้องโดยเปรียบเทียบระหว่างโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 มาต่อยอดการเรียนรู้ด้วยชุดข้อมูลที่มีอยู่ (pretrained) และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่ (non-pretrained) บนสถาปัตยกรรม AlexNet และ GoogLeNet ในแต่ละรอบ



รูปที่ 4.5 กราฟแสดงการเปรียบเทียบค่าความถูกต้องเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของข้อมูลของโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 มาต่อยอดการเรียนรู้ด้วยชุดข้อมูลที่มีอยู่ (pretrained) และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่ (non-pretrained) บนสถาปัตยกรรม AlexNet และ GoogLeNet ในแต่ละรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

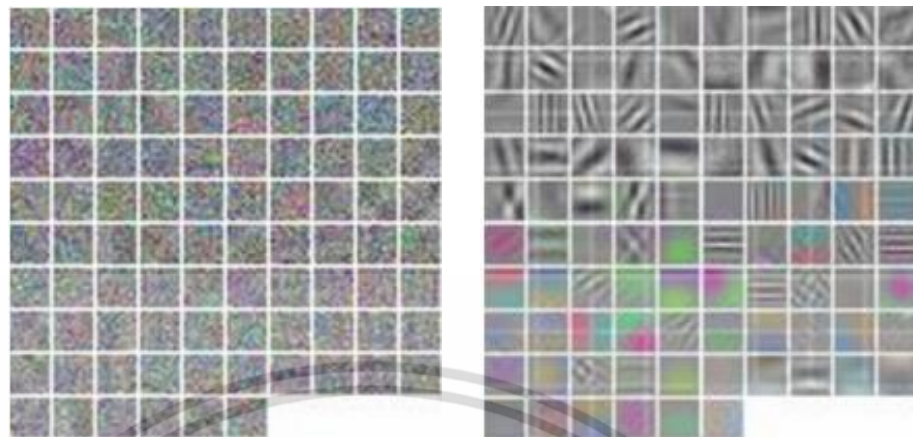
ตารางที่ 4.2 ค่าความถูกต้องและค่าเฉลี่ย F-score โดยเปรียบเทียบระหว่างโมเดลที่ฝึกสอนบนชุดข้อมูลจาก Places365 กับ ILSVRC-2012 มาต่อยอดการเรียนรู้ด้วยชุดข้อมูลที่มีอยู่ (pretrained) และโมเดลที่เรียนรู้ใช้เพียงชุดข้อมูลที่มีอยู่ (non-pretrained) บนสถาปัตยกรรม AlexNet และ GoogLeNet ในแต่ละรอบ

TCV	Accuracy (%)						Mean F-Score					
	Pretrained				Non-Pretrained		Pretrained				Non-Pretrained	
	Places356		ILSVRC-2012				Places356		ILSVRC-2012			
	AlexNet	GoogLeNet	AlexNet	GoogLeNet	AlexNet	GoogLeNet	AlexNet	GoogLeNet	AlexNet	GoogLeNet	AlexNet	GoogLeNet
1	99.82	99.56	99.91	99.74	86.16	83.86	0.9982	0.9956	0.9991	0.9974	0.8621	0.8394
2	84.48	86.77	85.63	87.30	77.25	78.66	0.8444	0.8679	0.8561	0.8736	0.7720	0.7863
3	84.48	85.98	85.10	86.68	77.34	76.81	0.8445	0.8595	0.8508	0.8674	0.7744	0.7684
4	85.63	87.65	85.54	86.95	79.01	79.10	0.8561	0.8775	0.8552	0.8694	0.7920	0.7929
5	85.80	87.74	87.39	88.71	79.54	80.69	0.8585	0.8775	0.8738	0.8877	0.7964	0.8078
6	82.54	86.24	84.30	86.24	76.19	78.66	0.8258	0.8638	0.8445	0.8637	0.7638	0.7905
7	83.69	86.33	84.92	85.27	77.16	76.81	0.8354	0.8628	0.8480	0.8537	0.7713	0.7687
8	85.01	87.74	86.95	88.27	78.92	80.25	0.8498	0.8777	0.8697	0.8830	0.7903	0.8029
9	84.22	87.48	86.07	88.36	77.25	80.60	0.8414	0.8746	0.8609	0.8835	0.7737	0.8068
10	86.24	87.48	86.68	88.27	77.78	79.54	0.8620	0.8754	0.8671	0.8830	0.7791	0.7965
AVG	86.19	88.30	87.25	88.58	78.66	79.50	0.8616	0.8832	0.8725	0.8862	0.7875	0.7960
SD	4.658	3.808	4.317	3.862	2.684	1.956	0.047	0.038	0.043	0.038	0.027	0.020

จากตารางที่ 4.2 เมื่อพิจารณาระหว่างการใช้ชุดข้อมูลภายนอกมาปรับใช้กับโมเดลและการใช้เพียงชุดข้อมูลจากปัญหาพิเศษบนระหว่างสองสถาปัตยกรรมนั้นคือ AlexNet และ GoogLeNet ได้ให้ผลลัพธ์ของค่าความถูกต้องเฉลี่ยที่แตกต่างกัน ซึ่งสำหรับ non-pretrained ค่าที่ได้จาก AlexNet เป็น 78.66% และจาก GoogLeNet เป็น 79.50% และมีค่า Mean F-score เท่ากับ 0.7875 และ 0.7690 ตามลำดับ ในขณะที่ pretrained จาก Places365 บนสถาปัตยกรรม AlexNet และ GoogLeNet ให้ค่าเป็น 86.19% และ 88.30% และมีค่า Mean F-score เท่ากับ 0.8616 และ 0.8832 ตามลำดับ และ pretrained บน ILSVRC-2012 ให้ค่าเป็น 87.25% และ 88.58% ในขณะที่ Mean F-score เท่ากับ 0.8725 และ 0.8862 ตามลำดับ ซึ่งจะเห็นว่าค่าความถูกต้องเฉลี่ยและ Mean F-score มีค่าที่สอดคล้องกัน ดังนั้นจึงเลือกใช้ในการอ้างอิงประสิทธิภาพโมเดลด้วยค่าความถูกต้องเฉลี่ย และพบว่าส่วนเบี่ยงเบนมาตรฐานของ pretrained มีค่าสูงกว่า non-pretrained เนื่องจากค่าความถูกต้องของ pretrained ในรอบที่ 1 มีค่าสูงกว่าค่าเฉลี่ยมากเมื่อเทียบกับ non-pretrained

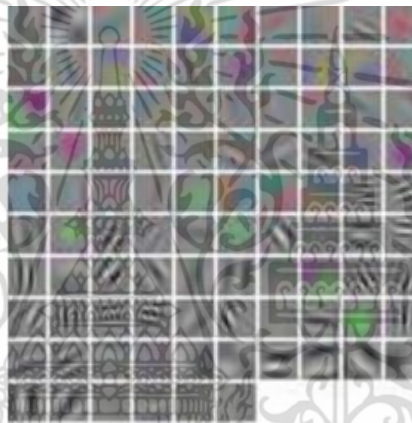
จะพบว่าโมเดลที่มี pretrained ที่ให้ค่าสูงที่สุดคือ pretrained จาก ILSVRC-2012 บนสถาปัตยกรรม GoogLeNet เท่ากับ 88.58% และค่าสูงที่สุดของ non-pretrained คือค่าบนสถาปัตยกรรม GoogLeNet เท่ากับ 79.50% เมื่อเปรียบเทียบค่าระหว่างโมเดลที่ pretrained และ non-pretrained ได้ให้ค่าความถูกต้องที่สูงกว่าโมเดล non-pretrained ถึง 9.08%

4.3.2 ผลลัพธ์ฟีเจอร์ชั้น convolution ที่ 1 บนสถาปัตยกรรม AlexNet และ GoogLeNet



(ก)

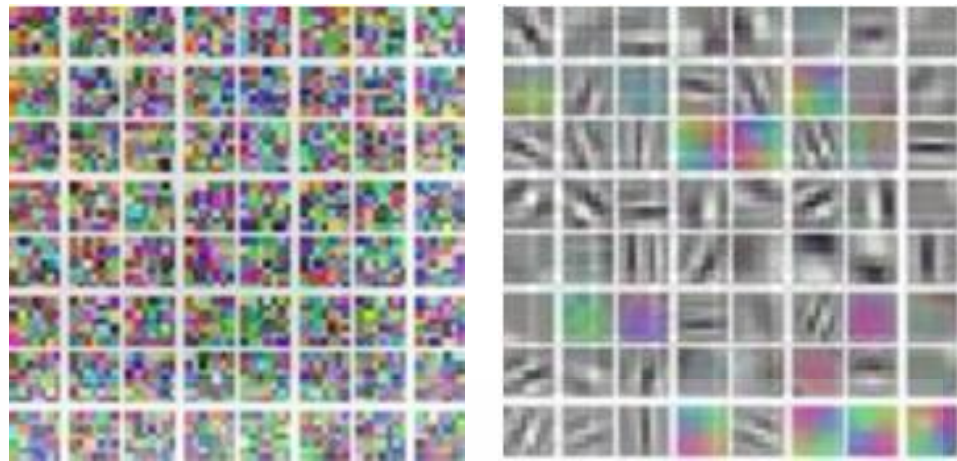
(ข)



(ค)

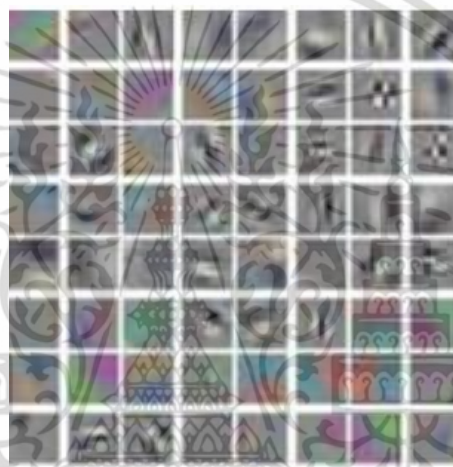
รูปที่ 4.6 (ก) ฟีเจอร์ในชั้น convolution ที่ 1 ของ non-pretrained บนสถาปัตยกรรม AlexNet
 (ข) ฟีเจอร์ในชั้น convolution ที่ 1 ของ pretrained จาก Places365 บนสถาปัตยกรรม AlexNet
 (ค) ฟีเจอร์ในชั้น convolution ที่ 1 ของ pretrained จาก ILSVRC-2012 บนสถาปัตยกรรม AlexNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

(ข)



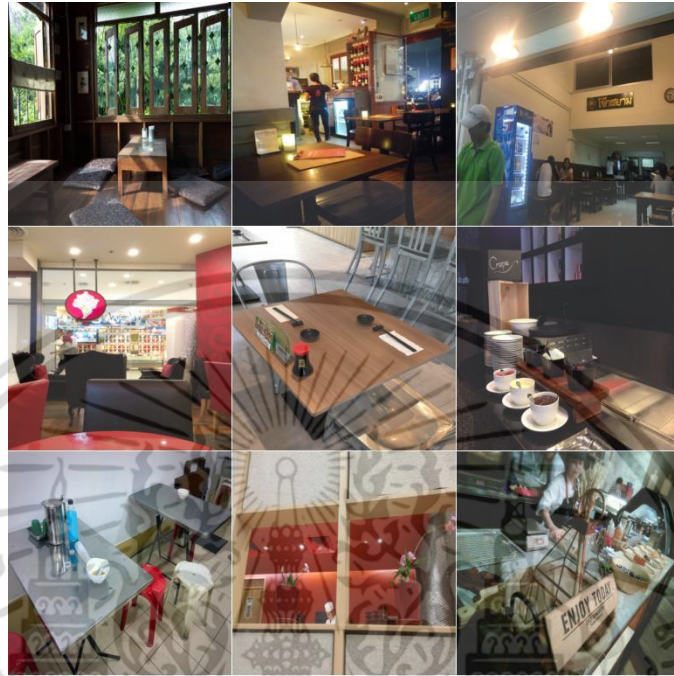
(ค)

รูปที่ 4.7 (ก) ฟีเจอร์ในชั้น convolution ที่ 1 ของ non-pretrained
 บนสถาปัตยกรรม GoogLeNet
 (ข) ฟีเจอร์ในชั้น convolution ที่ 1 ของ pretrained จาก Places365
 บนสถาปัตยกรรม GoogLeNet
 (ค) ฟีเจอร์ในชั้น convolution ที่ 1 ของ pretrained จาก ILSVRC-2012
 บนสถาปัตยกรรม GoogLeNet

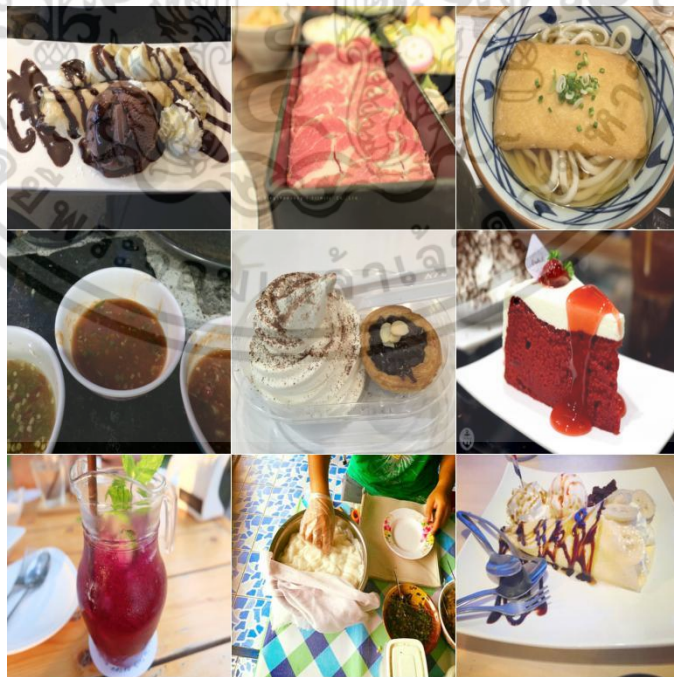
จากรูปที่ 4.6-4.7 พบว่าฟิลเตอร์ของ non-pretrained ทั้งสองสถาปัตยกรรมมีฟีเจอร์ที่มีรูปแบบไม่เด่นชัดหมายถึงมีนอยซ์หรือสิ่งรบกวน ซึ่งเกิดจากการฝึกสอนที่ยังไม่เพียงพอต่อโครงข่าย โดยในกรณีนี้หมายความว่าชุดข้อมูล non-pretrained มีคุณภาพน้อยกว่าชุดข้อมูล pretrained หลายเท่า ส่งผลให้ได้ค่าความถูกต้องที่ต่ำกว่าแบบ pretrained ที่มีฟีเจอร์ที่มีรูปแบบที่ชัดเจนกว่า มีสิ่งรบกวนน้อยกว่าอย่างเห็นได้ชัดในชั้น convolution ที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลลัพธ์การแสดงผลภาพที่ได้ค่าทำนายดีที่สุด 9 อันดับแรกในแต่ละคลาส (Top-9) จากโมเดลที่ให้ค่าความถูกต้องดีที่สุด

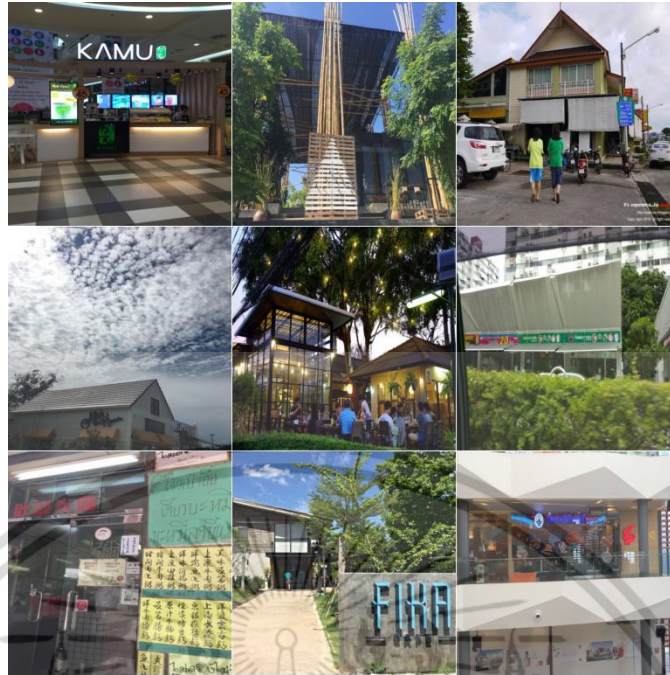


รูปที่ 4.8 Top-9 ของคลาสบรรยากาศ



รูปที่ 4.9 Top-9 ของคลาสอาหารหรือเครื่องดื่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

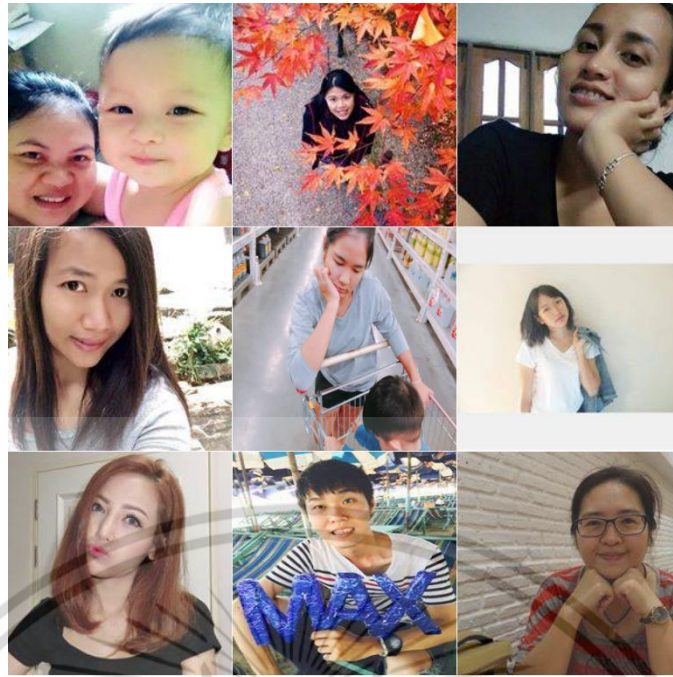


รูปที่ 4.10 Top-9 ของคลาสหน้าร้าน



รูปที่ 4.11 Top-9 ของคลาสป้ายหน้าร้านและสมุดเมนู

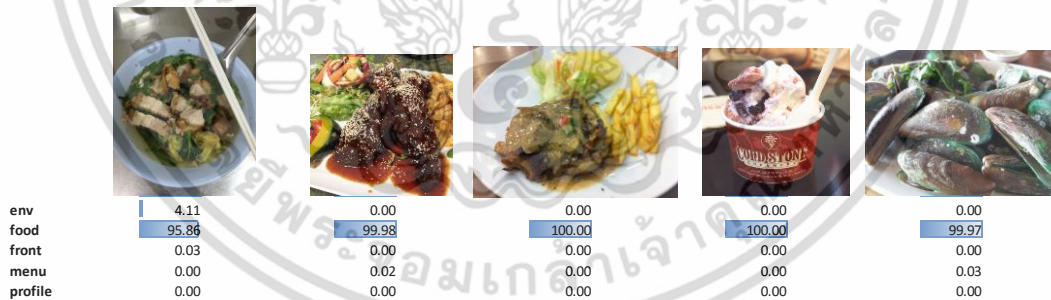
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 Top-9 ของคลาสโปรไฟล์

4.3.4 ผลการทดสอบด้วยชุดข้อมูลที่ยังไม่ได้รับการฝึกสอน

1) ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน



รูปที่ 4.13 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน

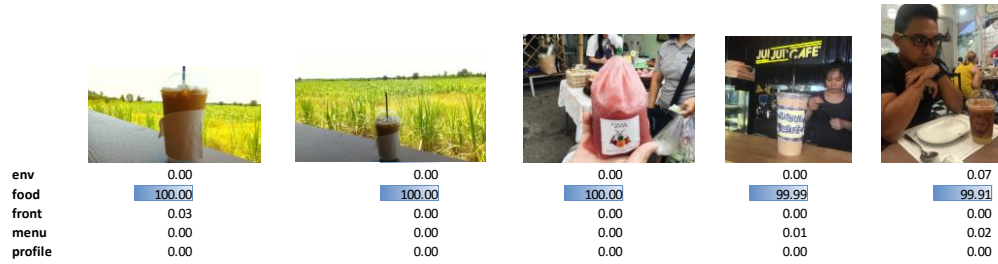


รูปที่ 4.15 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน



รูปที่ 4.16 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสอาหารหรือเครื่องดื่มที่ยังไม่ได้รับการฝึกสอน

จากรูปที่ 4.13-4.17 พบว่าชุดข้อมูลมีผลการทำนายที่แตกต่างกัน โดยรูปที่ 4.13 ให้ผลการทำนายว่าอยู่ในคลาสอาหารทั้งหมดเนื่องจากเป็นรูปที่ไม่ซับซ้อนและมองออกได้ง่าย รูปที่ 4.14 ให้ผลการทำนายว่าอยู่ในคลาสอาหารทั้งหมดด้วยความน่าจะเป็นร้อยละ 100 เนื่องจากเป็นภาพอาหารที่ถ่ายซูมเข้าไปเห็นรายละเอียดของอาหารเพียงอย่างเดียว รูปที่ 4.15 ให้ผลการทำนายว่าอยู่ในคลาสอาหารทั้งหมด แต่มีค่าทำนายบางส่วนที่มองว่าภาพอยู่ในคลาสสมุดเมนูและป้ายหน้าร้าน ซึ่งมีเหตุผลจากในรายละเอียดของบางภาพจะเห็นว่ามีรูปสี่เหลี่ยมที่ชัดเจนอยู่ในภาพทำให้โครงข่ายเข้าใจว่าเป็นสมุดเมนู รูปที่ 4.16 ให้ผลการทำนายว่าอยู่ในคลาสบรรยากาศเป็นส่วนใหญ่ และบางส่วนของภาพให้ผลการทำนายว่าเป็นคลาสสมุดเมนูและป้ายหน้าร้าน โดยเนื่องจากว่าภาพนั้นมีส่วนของอาหารอยู่ก็จริงแต่จะมีบางส่วนของภาพที่มีขอบบนเป็นลักษณะที่ไม่มีรูปแบบทำให้โครงข่ายมองว่าเป็นบรรยากาศและบางภาพที่ทำนายว่าเป็นเมนูนั้นเนื่องจากเห็นรายละเอียดในภาพเป็นรูปสี่เหลี่ยมอย่างชัดเจนตามเหตุผลที่ได้อธิบายมาข้างต้น และรูปที่ 4.17 ให้ผลการทำนายว่าอยู่ในคลาสอาหารหรือเครื่องดื่มเป็นส่วนใหญ่ เนื่องจากโครงข่ายได้เรียนรู้ลักษณะของแก้วที่ถูกจัดอยู่ในคลาสอาหารหรือเครื่องดื่ม

2) ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับหน้าร้านที่ยังไม่ได้รับการฝึกสอน



รูปที่ 4.18 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



env	0.00	0.01	0.00
food	0.00	0.00	0.00
front	100.00	99.99	100.00
menu	0.00	0.00	0.00
profile	0.00	0.00	0.00

รูปที่ 4.19 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน



env	0.00	0.00	0.00
food	0.00	0.00	0.00
front	100.00	100.00	99.99
menu	0.00	0.00	0.00
profile	0.00	0.00	0.00

รูปที่ 4.20 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน



env	0.00	0.00	0.00	0.00	0.00	0.00
food	0.00	0.00	0.00	0.00	0.00	0.00
front	99.99	99.88	96.49	99.98	99.94	100.00
menu	0.01	0.11	3.54	0.02	0.06	0.00
profile	0.00	0.00	0.00	0.00	0.00	0.00

รูปที่ 4.21 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				
env	100.00	100.00	96.70	98.03
food	0.00	0.00	0.00	0.00
front	0.00	0.00	3.30	1.97
menu	0.00	0.00	0.00	0.00
profile	0.00	0.00	0.00	0.00

รูปที่ 4.22 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสหน้าร้านที่ยังไม่ได้รับการฝึกสอน

จากรูปที่ 4.18-4.21 พบว่าชุดข้อมูลมีผลการทำนายที่คล้ายกันคือเป็นคลาสหน้าร้านด้วยความน่าจะเป็นร้อยละ 99 หรือ 100 มีรูปแบบของการเป็นหน้าร้านที่เด่นชัด ส่วนรูปที่ 4.22 ให้ผลการทำนายเป็นคลาสบรรยากาศเนื่องจากมีรูปแบบรายละเอียดภาพที่ไม่ชัดเจนคือประกอบไปด้วยหลายวัตถุโดยสรุปว่าการทำนายภาพในคลาสหน้าร้านให้ผลความถูกต้องที่ค่อนข้างแม่นยำแต่ถ้าหากมีรูปที่เป็นได้ทั้งสองคลาสอย่างรูปที่ 4.22 จะทำให้การทำนายเป็นไปได้ยาก

3) ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับบรรยากาศที่ยังไม่ได้รับการฝึกสอน

			
env	100.00	100.00	96.70
food	0.00	0.00	0.00
front	0.00	0.00	3.30
menu	0.00	0.00	0.00
profile	0.00	0.00	0.00

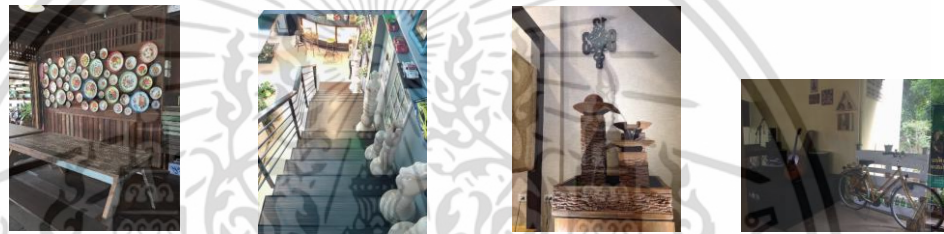
รูปที่ 4.23 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับบรรยากาศที่ยังไม่ได้รับการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



env	100.00	100.00	100.00
food	0.00	0.00	0.00
front	0.00	0.00	0.00
menu	0.00	0.00	0.00
profile	0.00	0.00	0.00

รูปที่ 4.24 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศยังไม่ได้รับการฝึกสอน



env	99.55	100.00	100.00	100.00
food	0.00	0.00	0.00	0.00
front	0.45	0.00	0.00	0.00
menu	0.00	0.00	0.00	0.00
profile	0.00	0.00	0.00	0.00

รูปที่ 4.25 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศที่ยังไม่ได้รับการฝึกสอน



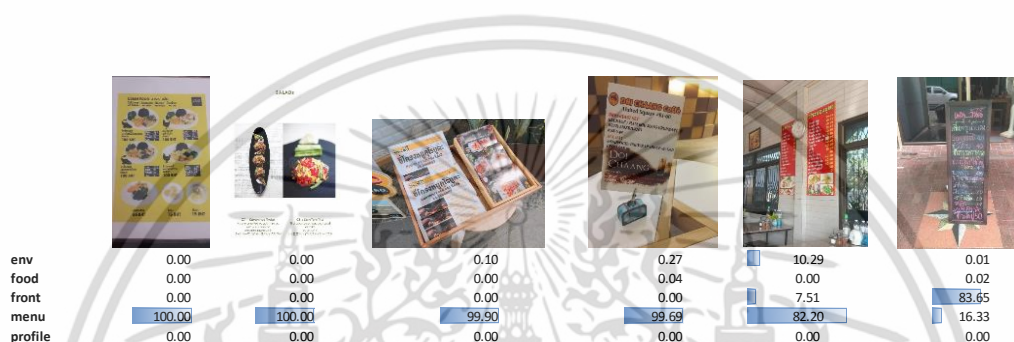
env	0.38	21.84	55.14	2.43
food	99.62	31.10	44.86	0.00
front	0.45	2.87	0.00	0.00
menu	0.00	42.89	0.00	97.56
profile	0.00	1.30	0.00	0.00

รูปที่ 4.26 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสบรรยากาศที่ยังไม่ได้รับการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.21-4.25 พบว่าชุดข้อมูลมีผลการทำนายที่คล้ายกันคือเป็นคลาสบรรยากาศด้วยความน่าจะเป็นร้อยละ 99 หรือ 100 ซึ่งการทำนายคลาสนี้คือภาพไม่ได้มีรูปแบบที่แน่นอนแต่จะประกอบด้วยวัตถุต่างๆที่หลากหลาย และจากรูปที่ 4.26 จะพบว่าให้ผลการทำนายที่ไม่ดีเนื่องจากในรายละเอียดบางภาพประกอบไปด้วยขวดน้ำซึ่งให้ค่าทำนายเป็นอาหารหรือเครื่องดื่ม มีกรอบสี่เหลี่ยมที่ชัดเจนทำให้ได้ผลการทำนายอยู่ในคลาสเมนูด้วย โดยสรุปได้ว่าการทำนายคลาสบรรยากาศให้ผลความถูกต้องที่แม่นยำ

4) ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับสมุดเมนูและป้ายหน้าร้านที่ยังไม่ได้รับการฝึกสอน



รูปที่ 4.27 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับคลาสสมุดเมนูที่ยังไม่ได้รับการฝึกสอน

จากรูปที่ 4.27 พบว่าภาพที่เป็นป้ายที่เป็นภาพซูมให้ผลการทำนายได้ถูกต้อง แต่เมื่อมีอย่างอื่นเข้ามาประกอบจะทำให้ประสิทธิภาพในการทำนายว่าเป็นคลาสสมุดเมนูได้น้อยลง เช่น การถ่ายภาพป้ายแต่ให้เห็นรายละเอียดหน้าร้าน โครงข่ายก็จะทำนายว่าเป็นคลาสหน้าร้านและสมุดเมนู

5) ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับโปรไฟล์ที่ยังไม่ได้รับการฝึกสอน

					
env	0.00	0.00	0.00	0.00	0.00
food	0.00	0.00	0.00	0.00	0.00
front	0.00	0.00	0.00	0.00	0.00
menu	0.00	0.00	0.00	0.00	0.00
profile	100.00	100.00	100.00	100.00	100.00

รูปที่ 4.28 ผลการทดสอบด้วยชุดข้อมูลที่เกี่ยวข้องกับโปรไฟล์ที่ยังไม่ได้รับการฝึกสอน

จากที่รูปที่ 4.28 พบว่าการทำนายคลาสโปรไฟล์ให้ผลความถูกต้องที่แม่นยำด้วยความน่าจะเป็นร้อยละ 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ปัญหาพิเศษนี้ได้ทำการวิจัยในหัวข้อการจำแนกประเภทภาพจากธุรกิจร้านอาหารโดยใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งมีเป้าหมายในการนำภาพที่ถูกติดแท็กแล้วมาเป็นต้นแบบเพื่อสร้างโมเดลสำหรับการจำแนกรูปภาพที่ยังไม่ถูกติดแท็ก โดยมีการใช้ชุดข้อมูลจากคลังภาพของบริษัททวงใน มีเดีย จำกัด ซึ่งประกอบไปด้วยคลาสทั้งหมด 5 คลาส คือ คลาสอาหารหรือเครื่องดื่ม คลาสป้ายหน้าร้าน และสมุดเมนู คลาสบรรยากาศร้าน คลาสหน้าร้าน คลาสโปรไฟล์ และมีการใช้ชุดข้อมูล pretrained จาก Places365 และ ILSVRC-2012

จากขั้นตอนการทดลองในบทที่ 4 สามารถสรุปได้ว่าหลังทำการกระบวนการ cleansing ข้อมูล พบว่าหลังดำเนินการให้ค่าความถูกต้องเฉลี่ยสูงขึ้นถึงร้อยละ 2.93 จึงนำชุดข้อมูลที่ทำการ cleansing แล้วไปใช้ในการทดลองที่ 4.2-4.3 และจากทดลองที่ 4.2 พบว่า ADAM เป็น solver ที่ดีที่สุดบนทั้งสองสถาปัตยกรรมคือ AlexNet และ GoogLeNet โดยมีค่าความถูกต้องเฉลี่ยเท่ากับร้อยละ 78.66 และ 79.50 ตามลำดับ และพบว่า GoogLeNet ให้ค่าความถูกต้องเฉลี่ยดีกว่า AlexNet ถึงร้อยละ 0.84 และจากการทดลองที่ 4.3 พบว่า โมเดลที่ผ่านการ pretrained ให้ค่าความถูกต้องเฉลี่ยที่สูงกว่า non-pretrained ถึงร้อยละ 9.08 โดยใช้โมเดลที่ดีที่สุดของ pretrained และ non-pretrained และพบว่าบนสถาปัตยกรรม GoogLeNet ให้ค่าความถูกต้องเฉลี่ยที่สูงกว่า AlexNet ของโมเดลทั้งหมด ดังนี้ คือ pretrained จากชุดข้อมูล Places365 ให้ค่าความถูกต้องเฉลี่ยร้อยละ 88.3 จากชุดข้อมูล ILSVRC-2012 ให้ค่าความถูกต้องเฉลี่ยร้อยละ 88.58 และชุดข้อมูลของปัญหาพิเศษให้ค่าความถูกต้องเฉลี่ยร้อยละ 79.5

ดังนั้นจะพบว่าการใช้โครงข่าย GoogLeNet ที่ผ่านการ pretrained จาก ILSVRC-2012 และใช้ solver ของ ADAM ให้ผลลัพธ์ที่ดีที่สุด

5.2 ข้อจำกัด

- 1) บางโมเดลมีขนาดใหญ่เนื่องจากมีชั้นและพารามิเตอร์จำนวนมาก ทำให้ต้องใช้หน่วยความจำของการ์ดจอมากตามไปด้วยซึ่งไม่เพียงพอ เช่น restnet 152 ชั้น
- 2) การเลือกใช้ค่าของการแท็กมีการแบ่งแยกกันอย่างไม่ชัดเจน ทำให้มีความหมายที่ทับซ้อนกัน เช่น ควรจะเปลี่ยนจากบรรยากาศให้เป็นในร้านหรือนอกร้านแทน แล้วทำการยกเลิกคลาสหน้าร้าน

5.3 ข้อเสนอแนะ

- 1) ปรับแต่งค่าไฮเปอร์พารามิเตอร์อื่นๆ ยกตัวอย่างเช่น ชนิดของฟังก์ชันการกระตุ้น
- 2) ปรับเปลี่ยนชนิดของชั้น Normalization

5.4 ปัญหาและอุปสรรค

1) จากการวิเคราะห์ชุดข้อมูลภาพพบว่า ภาพในแต่ละคลาสมีภาพที่ไม่ชัดเจนว่าควรจะอยู่กลุ่มไหน ดังนี้

- ภาพบรรยากาศมีความคล้ายกับภาพหน้าร้าน
- ภาพบรรยากาศมีความคล้ายกับภาพอาหาร
- ภาพบรรยากาศมีความคล้ายกับป้ายและเมนู
- ภาพหน้าร้านมีความคล้ายกับป้ายและเมนู

โดยจะทำการแก้ไขคือนำรูปที่ไม่ชัดเจนออกโดยใช้กระบวนการกำจัดสิ่งรบกวนออก (data cleansing)

2) การกำหนดพาร์ทิชันที่ใช้ลงระบบปฏิบัติการ ubuntu เล็กเกินไปทำให้ไม่เพียงพอสำหรับการฝึกโมเดลที่มีขนาดใหญ่

เอกสารอ้างอิง

- ชนาวุฒิ ประกอบผล. 2552. “โครงข่ายประสาทเทียม Artificial Neural Networks.” *มฉก. วิชาการ*. 12(24) : 73-87.
- วิกิพีเดีย. 2560. การเรียนรู้ของเครื่อง. [Online]. Available : <https://th.wikipedia.org/wiki/การเรียนรู้ของเครื่อง>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- วิกิพีเดีย. 2017. Convolutional neural network. [Online]. Available : https://en.wikipedia.org/wiki/Convolutional_neural_network. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- วิกิพีเดีย. 2017. Deep Learning. [Online]. Available : https://en.wikipedia.org/wiki/Deep_learning. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- Adit Deshpande. 2017. The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3).[Online]. Available : <https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- Hiroshi Kuwajima. 2017. Memo: Backpropagation in Convolutional Neural Network. [Online]. Available : <https://www.slideshare.net/kuwajima/cnnbp>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- NVIDIA Corporation. 2017. CUDA Toolkit. [Online]. Available : <https://developer.nvidia.com/cuda-toolkit>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- NVIDIA Corporation. 2017. NVIDIA cuDNN. [Online]. Available : <https://developer.nvidia.com/cudnn>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- NVIDIA Corporation. 2017. NVIDIA DIGITS. [Online]. Available : <https://developer.nvidia.com/digits>. เข้าถึงเมื่อ 9 กรกฎาคม 2560
- Rajarshi Roy. 2017. Yelp Restaurant Photo Classification. [Online]. Available : cs231n.stanford.edu/reports/2016/pdfs/014_Report.pdf. เข้าถึงเมื่อ 9 กรกฎาคม 2560

Stanford University. 2017. **Convolutional Neural Networks (CNNs / ConvNets)**.

[Online]. Available : <http://cs231n.github.io/convolutional-networks/>. เข้าถึงเมื่อ 9 กรกฎาคม 2560

Stanford University. 2017. **Parameter updates**. [Online]. Available :

<http://cs231n.github.io/convolutional-networks/>. เข้าถึงเมื่อ 9 กรกฎาคม 2560

The BAIR Caffe developers. 2017. **Solver**. [Online]. Available :

<http://caffe.berkeleyvision.org/tutorial/solver.html>. เข้าถึงเมื่อ 9 กรกฎาคม 2560

Yeephycho. 2017. **Normalizations in Neural Networks**. [Online]. Available :

<http://yeephycho.github.io/2016/08/03/Normalizations-in-neural-networks/>.

เข้าถึงเมื่อ 9 กรกฎาคม 2560



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งโปรแกรม

1. การเลือกการ์ดจอสำหรับงาน deep learning

-ตรวจสอบสเปค NVIDIA GPU

https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units

-ตรวจสอบความสามารถในการคำนวณ และ การรองรับ CUDA ในแต่ละเวอร์ชัน

<https://en.wikipedia.org/wiki/CUDA>

<https://developer.nvidia.com/cuda-gpus>

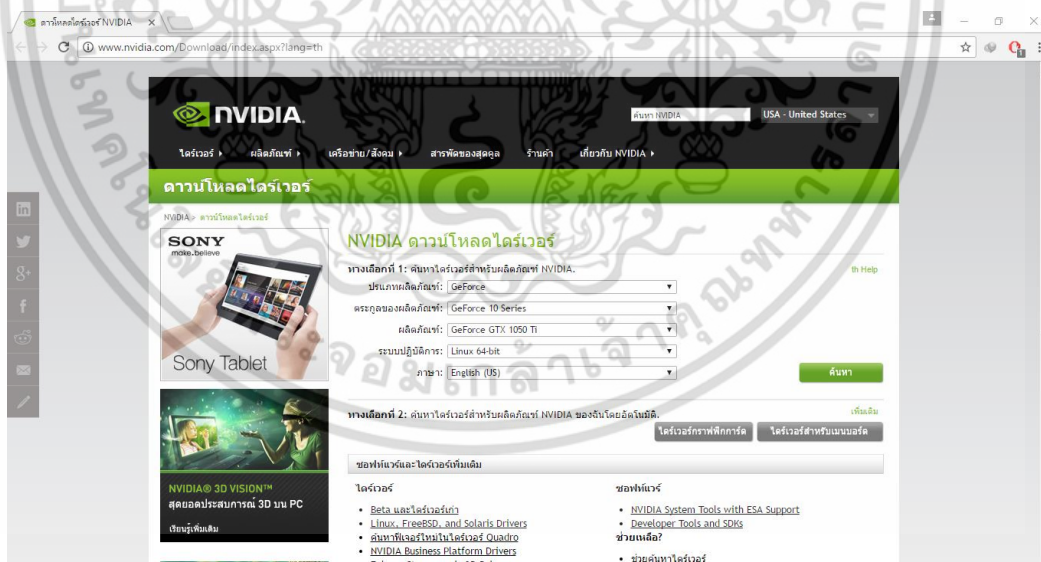
2. การติดตั้งระบบปฏิบัติการ Ubuntu เวอร์ชัน 16.04 LTS

<https://itseasys.blogspot.com/2016/06/install-ubuntu-1604-lts.html>

3. การติดตั้งไดรเวอร์การ์ดจอบนระบบปฏิบัติการ Ubuntu

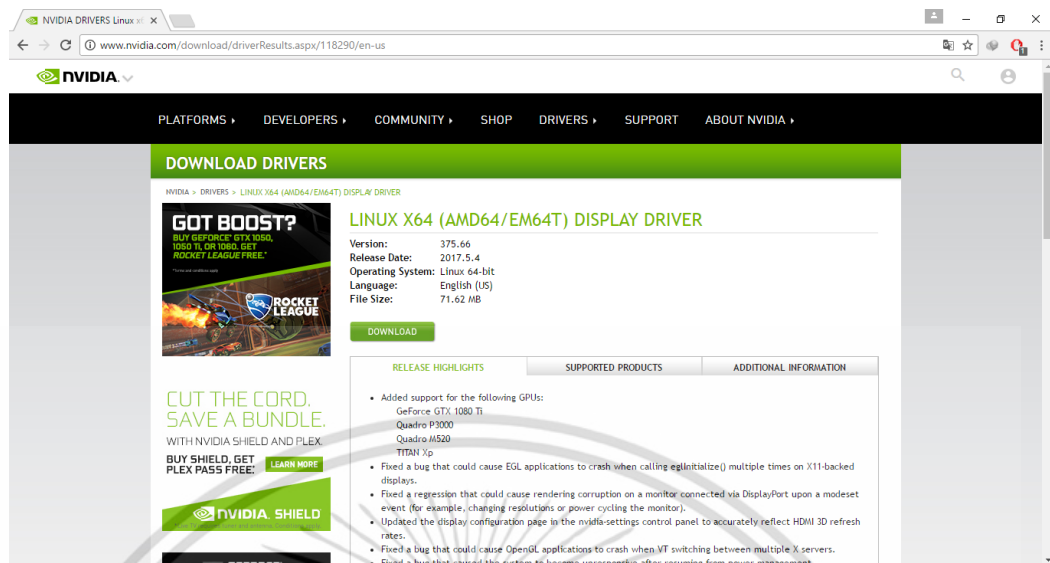
3.1 ดาวน์โหลดไดรเวอร์จากเว็บไซต์

<http://www.nvidia.com/Download/index.aspx?lang=th>



รูปที่ 1 หน้าต่างเว็บไซต์ดาวโหลดไดรเวอร์การ์ดจอ Nvidia

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

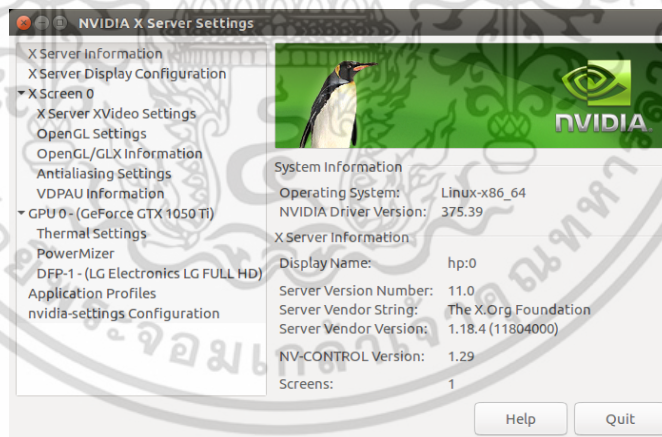


จะได้ไฟล์ NVIDIA-Linux-x86_64-375.26.run

3.2 ติดตั้งไดรเวอร์ ด้วยคำสั่ง

```
$ sudo sh ./NVIDIA-Linux-x86_64-375.26.run
```

3.3 ตรวจสอบรายละเอียดไดรเวอร์ในโปรแกรม NVIDIA X Server Settings



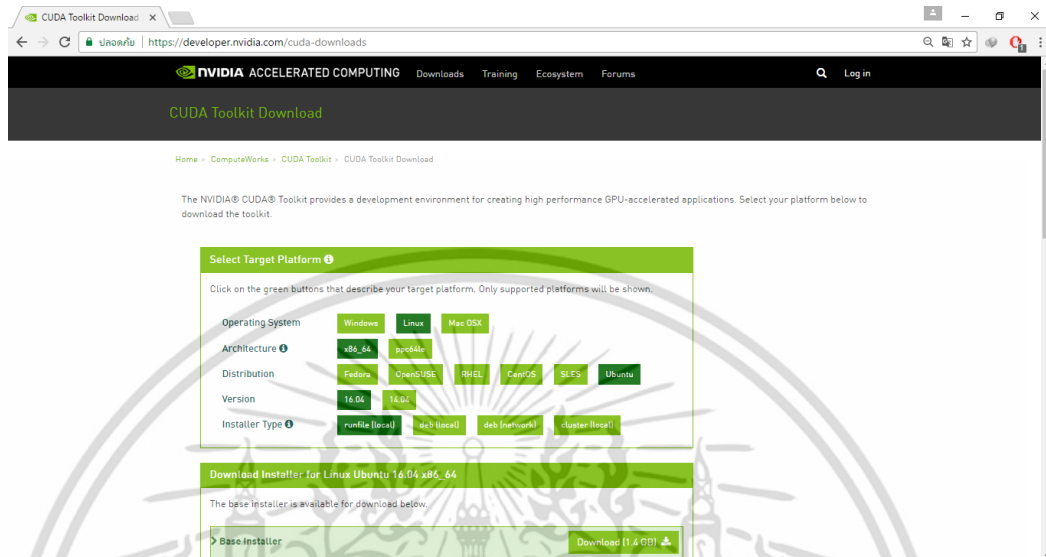
รูปที่ 3 หน้าต่างโปรแกรม NVIDIA X Server Settings

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การติดตั้ง CUDA เวอร์ชัน 8

4.1 ดาวโหลด CUDA8 จากเว็บไซต์ <https://developer.nvidia.com/cuda-downloads>

เลือก Platform ดังรูป



รูปที่ 4 หน้าต่างเว็บไซต์ดาวโหลด CUDA เวอร์ชัน 8

จะได้ไฟล์ `cuda_8.0.61_375.26_linux.run`

4.2 รันคำสั่ง

```
$ sudo sh cuda_8.0.61_375.26_linux.run
```

4.3 ตรวจสอบว่าการติดตั้งสมบูรณ์

4.3.1 ตรวจสอบเวอร์ชัน ด้วยคำสั่ง

```
$ dpkg -l cuda
```

```
neonlight@hp: ~
neonlight@hp:~$ dpkg -l cuda
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-inst/trig-await/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version      Architecture Description
+++-----+
ii cuda           8.0.61-1    amd64        CUDA meta-package
neonlight@hp:~$
```

รูปที่ 5 ผลการรันชุดคำสั่งบนเทอร์มินอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ทำการรีบูตเครื่อง ด้วยคำสั่ง

```
$ sudo reboot
```

4.3.3 การตั้งค่า environment

- เพิ่มค่าให้ตัวแปร PATH ด้วยคำสั่ง

```
$ export PATH = /usr/local/cuda-8.0/bin$ { PATH:+:${PATH} }
```

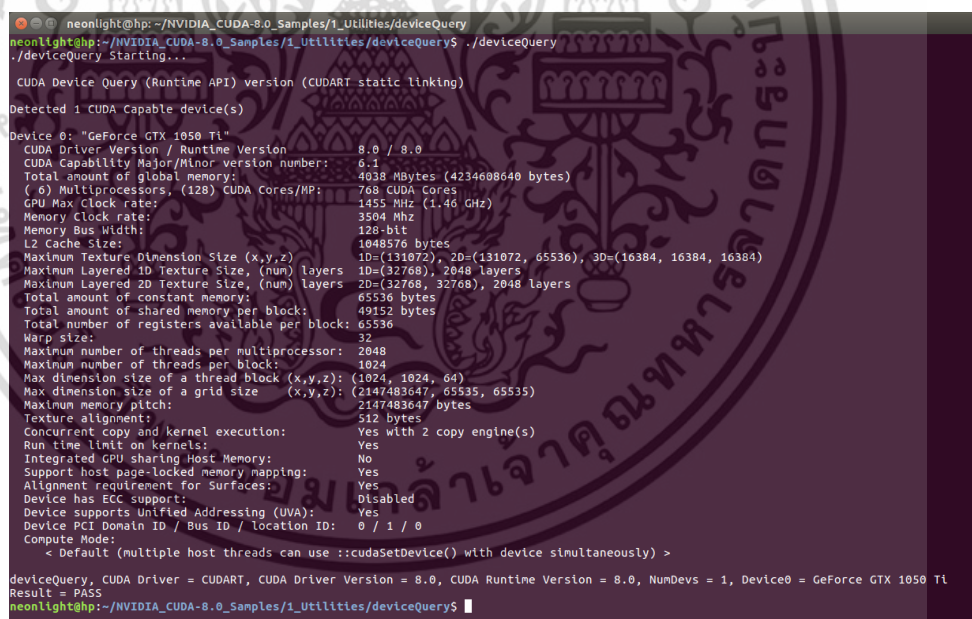
- เพิ่มตัวแปร LD_LIBRARY_PATH ให้เก็บ /usr/local/cuda-8.0/lib64

ในระบบ 64 บิตด้วยคำสั่ง

```
$ export LD_LIBRARY_PATH = /usr/local/cuda-8.0/lib64\
${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH} }
```

- รันโปรแกรม deviceQuery ได้ผลลัพธ์ดังรูป แสดงว่าการติดตั้งสมบูรณ์

```
$ ./deviceQuery
```



```

neonlight@hp:~/NVIDIA_CUDA-8.0_Samples/1_Utillities/deviceQuery$ ./deviceQuery
./deviceQuery Starting...

CUDA Device Query (Runtime API) version (CUDA static linking)
Detected 1 CUDA Capable device(s)
Device 0: "GeForce GTX 1050 Ti"
  CUDA Driver Version / Runtime Version      8.0 / 8.0
  CUDA Capability Major/Minor version number:  6.1
  Total amount of global memory:              4938 Mbytes (4234608640 bytes)
  ( 0) Multiprocessors, (128) CUDA Cores/MP:   768 CUDA Cores
  GPU Max Clock rate:                         1455 MHz (1.46 GHz)
  Memory Clock rate:                          3504 Mhz
  Memory Bus Width:                          128-bit
  L2 Cache Size:                              1048576 bytes
  Maximum Texture Dimension Size (x,y,z)      1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers 1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
  Total amount of constant memory:            65536 bytes
  Total amount of shared memory per block:    49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                  32
  Maximum number of threads per multiprocessor: 2048
  Maximum number of threads per block:        1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size (x,y,z):  (2147483647, 65535, 65535)
  Maximum memory pitch:                      2147483647 bytes
  Texture alignment:                          512 bytes
  Concurrent copy and kernel execution:       Yes with 2 copy engine(s)
  Run time limit on kernels:                  Yes
  Integrated GPU sharing Host Memory:         No
  Support host page-locked memory mapping:    Yes
  Alignment requirement for Surfaces:         Yes
  Device has ECC support:                     Disabled
  Device supports Unified Addressing (UVA):   Yes
  Device PCI Domain ID / Bus ID / location ID:  0 / 1 / 0
  Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery CUDA Driver = CUDART, CUDA Driver Version = 8.0, CUDA Runtime Version = 8.0, NumDevs = 1, Device0 = GeForce GTX 1050 Ti
Result = PASS
neonlight@hp:~/NVIDIA_CUDA-8.0_Samples/1_Utillities/deviceQuery$

```

รูปที่ 6 ผลการรันชุดคำสั่งบนเทอร์มินอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การใส่ cuDNN เวอร์ชัน 5.1

5.1 ดาวโหลด cuDNN จากเว็บไซต์ <https://developer.nvidia.com/cudnn> โดยเลือก cuDNN v5.1 สำหรับ CUDA 8.0 และระบบปฏิบัติการ Linux

cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

I Agree To the Terms of the [cuDNN Software License Agreement](#)

Please check your framework documentation to determine the recommended version of cuDNN. If you are using cuDNN with a Pascal (GTX 1080, GTX 1070), version 5 or later is required.

Download cuDNN v5.1 [Jan 20, 2017], for CUDA 8.0

- cuDNN User Guide
- cuDNN Install Guide
- [cuDNN v5.1 Library for Linux](#)
- cuDNN v5.1 Library for Power8
- cuDNN v5.1 Library for Windows 7
- cuDNN v5.1 Library for Windows 10
- cuDNN v5.1 Library for OSX
- cuDNN v5.1 Release Notes
- cuDNN v5.1 Runtime Library for Ubuntu14.04 [Deb]
- cuDNN v5.1 Developer Library for Ubuntu14.04 [Deb]
- cuDNN v5.1 Code Samples and User Guide [Deb]
- cuDNN v5.1 Runtime Library for Ubuntu16.04 Power8 [Deb]
- cuDNN v5.1 Developer Library for Ubuntu16.04 Power8 [Deb]
- cuDNN v5.1 Code Samples and User Guide Power8 [Deb]

Download cuDNN v5.1 [Jan 20, 2017], for CUDA 7.5

QUICKLINKS

- Accelerated Computing - Training
- CUDA GPUs
- Tools & Ecosystem
- OpenACC: More Science Less Programming
- CUDA FAQ

GPU Computing [Follow](#)

CUDA, GPU Computing @GPUComputing

Snap a photo of your room and this #AI-based platform lets you easily 'try-on' items from a retailer's catalogue. [nvidia.ws/2leCpOG](#)

News

- Algorithm Identifies Skin Cancer as Accurately as a Dermatologist January 25, 2017
- AI Technology Helps Drones Sense and Avoid Obstacles January 25, 2017

รูปที่ 7 หน้าต่างเว็บไซต์ดาวโหลด cuDNN เวอร์ชัน 5.1

จะได้ไฟล์ `cuda-8.0-linux-x64-v5.1.tgz`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 แยกไฟล์ด้วยคำสั่ง

```
$ tar -xzvf cuda-8.0-linux-x64-v5.1.tgz
```

จะได้ไฟล์ตามพาดังนี้

```
cuda/include/cudnn.h
```

```
cuda/lib64/libcudnn.so.5
```

```
cuda/lib64/libcudnn.so.5.1.10
```

```
cuda/lib64/libcudnn.static.a
```

5.3 คัดลอกไฟล์ libcudnn.so.5 libcudnn.so.5.1.10 และ libcudnn.static.a ไปยังพาร
/usr/local/cuda-8.0/lib64 และคัดลอกไฟล์ cudnn.h ไปยังพาร /usr/local/cuda-
8.0/include ด้วยคำสั่งดังนี้

```
$ sudo cp cuda/lib64/lib* /usr/local/cuda-8.0/lib64
```

```
$ sudo cp cuda/include/cudnn.h /usr/local/cuda-8.0/include
```

6. การติดตั้ง Caffe

6.1 ดาวน์โหลด Caffe จากเว็บ <https://github.com/NVIDIA/caffe>

6.2 แยกไฟล์ แล้วย้ายไฟล์เดอร์ไปยัง \$HOME เพื่อความสะดวก

6.3 ติดตั้ง dependencies ดังต่อไปนี้

```
$ sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev  
libopencv-dev libhdf5-serial-dev protobuf-compiler
```

```
$ sudo apt-get install --no-install-recommends libboost-all-dev
```

```
$ sudo apt-get install libatlas-base-dev
```

```
$ sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

6.4 ปรับค่าในไฟล์ Makefile.config.example ให้มีคำสั่งดังต่อไปนี้

```
USE_CUDNN := 1
CUDA_DIR := /usr/local/cuda
CUDA_ARCH := -gencode arch=compute_30,code=sm_30 \
-gencode arch=compute_35,code=sm_35 \
-gencode arch=compute_50,code=sm_50 \
-gencode arch=compute_52,code=sm_52 \
-gencode arch=compute_60,code=sm_60 \
-gencode arch=compute_61,code=sm_61 \
-gencode arch=compute_61,code=compute_61

BLAS := atlas
PYTHON_INCLUDE := /usr/include/python2.7 \
/usr/lib/python2.7/dist-packages/numpy/core/include
PYTHON_LIB := /usr/lib/x86_64-linux-gnu/
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
/usr/include/hdf5/serial/
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-
linux-gnu/hdf5/serial/
BUILD_DIR := build
DISTRIBUTE_DIR := distribute
TEST_GPUID := 0
Q ?= @
```

คัดลอกไฟล์นี้ แล้วเปลี่ยนชื่อเป็น Makefile.config

6.5 การ compile caffe โดยใช้ CMake

```
$ mkdir build
$ cd build
$ cmake
$ make all
$ make install
$ make runtest
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การติดตั้ง NVIDIA DIGITS เวอร์ชัน 4.0.0

7.1 ดาวน์โหลด DIGITS จากเว็บไซต์ <https://github.com/NVIDIA/DIGITS>

7.2 ติดตั้ง dependencies ต่างๆ ด้วยคำสั่ง

```
$ sudo apt-get install --no-install-recommends git graphviz python-dev
python-flask python-flaskext.wtf python-gevent python-h5 py python-
numpy python-pil python-pip python-protobuf python-scipy
```

7.3 ติดตั้ง PyPI Packages ด้วยคำสั่งดังนี้

```
$ sudo pip install -r $DIGITS_ROOT/requirements.txt
```

7.4 ทดสอบการเรียกใช้ DIGITS ด้วยคำสั่ง

```
$ ./digits-devserver
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้