การจำแนกพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึก
DOG SPECIES CATEGORIZATION USING DEEP LEARNING ALGORITHM

ธฤต ชัยขจรภัทร์
พิชญา เขียดสังข์
ภูริวัจน์ หนองใหญ่

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DOG SPECIES CATEGORIZATION USING DEEP LEARNING ALGORITHM

Tarid Chaikajonpat
Pichaya Kiadsung
Phuriwat Nongyai

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS 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 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การจำแนกพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึก Dog Species Categorization Using Deep Learning Algorithm
ชื่อนักศึกษา	นายธฤต ชัยขจรภัทร์ รหัสนักศึกษา 55050328 นางสาวพิชญา เขียดสังข์ รหัสนักศึกษา 55050398 นายภูริวัจน์ หนองใหญ่ รหัสนักศึกษา 55050418
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีระ ศิริธีรากุล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.) อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.อนันตพร หรรษคุณาฒัย ประธานกรรมการ	
อ.สันธนะ อู่อุดมยิ่ง กรรมการ	
ผศ.ธีระ ศิริธีรากุล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การจำแนกพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึก
ชื่อนักศึกษา	นายธฤต ชัยขจรภัทร์ รหัสนักศึกษา 55050328
	นางสาวพิชญา เขียดสังข์ รหัสนักศึกษา 55050398
	นายภูริวัจน์ หนองใหญ่ รหัสนักศึกษา 55050418
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีระ ศิริธีรากุล

บทคัดย่อ

เนื่องจากในปัจจุบันมีสุนัขสายพันธุ์ต่างๆได้ถือกำเนิดขึ้นมาอย่างมากมายหลากหลายสายพันธุ์ ทำให้อาจจะเกิดปัญหาในการจำแนกพันธุ์สุนัข ว่าสุนัขตัวนี้จะตรงกับสายพันธุ์ไหน ผู้พัฒนาจึงได้คิดที่จะทำปัญหาพิเศษนี้ขึ้นมา โดยที่ในปัญหาพิเศษนี้จะเป็นการเสนอวิธีในการจัดหมวดหมู่แยกประเภทของพันธุ์สุนัขและการศึกษาวิธีการการเรียนรู้เชิงลึก โดยการนำภาพสุนัขมาผ่านกระบวนการวิธีการทางคอมพิวเตอร์ อัลกอริทึมการเรียนรู้เชิงลึกในการแยกประเภทภาพของสุนัขจะมีหลักการทำงานโดยการนำภาพของสุนัขไปทำการเปรียบเทียบ กับ ฐานข้อมูล และ โมเดลที่ได้ทำการฝึกสอนไว้และระบบก็จะทำการแสดงพันธุ์สุนัขที่มีความใกล้เคียงกับรูปภาพที่ได้นำมาเปรียบเทียบในข้างต้น ซึ่งผลการทดสอบการแยกสายพันธุ์สามารถสรุปผลได้ว่า การใช้วิธีการการเรียนรู้เชิงลึกในการจำแนกพันธุ์สุนัขนั้นมีความแม่นยำ 31.24% โดยความน่าจะเป็นจากการแบ่งกลุ่มคิดเป็น 1/15 จากการทดสอบ 15 สายพันธุ์สุนัข การศึกษาวิธีการการเรียนรู้แบบเชิงลึกนี้จะช่วยพัฒนาในการจำแนกสายพันธุ์ของสุขันต่างๆให้มีความสะดวกง่ายดายขึ้นต่อผู้ที่ต้องการที่จะทราบสายพันธุ์ของสุนัข โดยที่ความสะดวกสบายนั้นมาจากการที่เทคโนโลยีใหม่ๆได้ถูกพัฒนาขึ้นมา และยังเป็นการนำส่วนหนึ่งของเทคโนโลยีใหม่ๆ เข้ามาเป็นตัวช่วยในการเรียนรู้ เพื่อเพิ่มประสิทธิภาพในเรื่องปัญหาการจำแนกข้อมูลอื่นๆอีกต่อไป

คำสำคัญ : การจำแนกประเภทพันธุ์สุนัข การประมวลผลภาพ โครงข่ายประสาทเทียม อัลกอริทึมการเรียนรู้เชิงลึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Dog Species Categorization Using Deep Learning Algorithm
Students	Mr. Tarid Chaikajonpat Student ID 55050328
	Miss Pichaya Kiadsung Student ID 55050398
	Mr.Phuriwat Nongyai Student ID 55050418
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang(KMITL)
Academic Year	2015
Advisor	Assistant Professor Teera Siriteerakul

ABSTRACT

With a number of dog breeds nowadays, people may find troubles determining a dog breed. It is, thus, an interesting area to develop a model to perform the task. In addition, the emerging of deep learning algorithm which enable human to obtain the learned model for classification problem is available. We propose to develop a dog species categorization using deep learning algorithm. The process starts with configuring the deep leraning model. Dog image are used for training the model. The deep leaning obtained is test using 4-fold cross validation. Our result shows the accuracy of 31.24% comparing to intuitively 1/15 chance of 15 breeds test. This new method shows that it is not hard for human to apply this technique to other classification problem with an significant accuracy improvement

Keywords : dog species classification, image processing, artificial neural network, deep learning algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

งานวิจัยในหัวข้อเรื่อง การจำแนกพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึก สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับการเกื้อหนุนจากบุคคลหลายฝ่ายที่ให้คำปรึกษาและชี้แนะจนสามารถดำเนินงานให้บรรลุเป้าหมายได้เป็นอย่างดี

ขอขอบคุณ ผศ.ธีระ ศิริธีรากุล เป็นอาจารย์ที่ปรึกษาปัญหาพิเศษทั้งให้คำปรึกษาและอุปกรณ์เครื่องมือในการทำปัญหาพิเศษ ผศ.ดร.อนันตพร หรรษคุณาฒัย ประธานกรรมการสอบปัญหาพิเศษ และ อ.สันธนะ อู่อุดมยิ่ง กรรมการสอบปัญหาพิเศษ สำหรับการแนะนำให้คำปรึกษาในการทำปัญหาพิเศษในครั้งนี้

ขอขอบคุณ โครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 18 จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ที่ได้สนับสนุนเงินทุนและคำแนะนำต่างๆในการทำปัญหาพิเศษในครั้งนี้

ขอมอบความกตัญญูกตเวทิตาคุณ แด่บิดา มารดา และผู้มีพระคุณทุกท่าน ที่คอยช่วยเหลือสนับสนุนและให้กำลังใจมาโดยตลอด

นายธฤต ชัยขจรภัทร์
นางสาวพิชญา เขียดสังข์
นายภูริวัจน์ หนองใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ 1
1.3 ขอบเขตการศึกษา 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 ขั้นตอนการดำเนินงาน 2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
2.1.1 โครงข่ายประสาทเทียม (Artificial neural network) 4
2.1.2 การเรียนรู้แบบเชิงลึก 5
2.2 ส่วนของโปรแกรมและภาษาที่นำมาใช้ 14
2.2.1 Caffe DeepLearning 14
2.2.2 NVIDIA DIGITS 21
2.3 งานวิจัยที่เกี่ยวข้อง 14
2.3.1 Dog Breed Classification Using Part Localization 24
2.3.2 การจำแนกสายพันธุ์ของนกโดยใช้วิธีการ Deep Convolutional Nets ... 25
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน 26
3.1 โครงสร้างการทำงานของวิธีการเรียนรู้เชิงลึกโดยใช้ Nvidia Digits 26
3.2 รายละเอียดวิธีการพัฒนา (Software Specification) 27
3.2.1 ฐานข้อมูลที่ที่นำมาใช้ในการฝึก (Train) 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.1 Input / Ouput Specification 28
3.3 ขั้นตอนการวิจัย 28
3.3.1 การเตรียมฐานข้อมูลไว้สำหรับการฝึกและทดสอบ 28
3.3.2 ขั้นตอนการฝึกโมเดล 35
3.3.3 ขั้นตอนการทดสอบ 46
บทที่ 4 ผลการดำเนินงาน 48
4.1 ภาพรวมส่วนของการสร้างโมเดลและ Train รูปภาพ โดยใช้ NVIDIA DIGTIS 49
4.1.1 การสร้างฐานข้อมูล โมเดล และการตั้งค่าเพื่อใช้ในการ Train 50
4.2 ส่วนของผลการดำเนินงาน 63
4.3 สรุปผลการทดสอบและการวัดประสิทธิภาพโมเดล 67
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 68
5.1 สรุปผลการดำเนินงาน 68
5.2 ข้อจำกัด 68
5.3 ข้อเสนอแนะ 69
5.4 ปัญหาและอุปสรรค 69
เอกสารอ้างอิง 70
ภาคผนวก 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงการใช้งาน framework ต่างๆในการพัฒนา Convolutional Neuron network.....15

3.1 ชื่อสายพันธุ์ของสุนัขและรหัสที่ใช้ในการทดสอบ.....27

4.1 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 1 2 3 และกลุ่มทดสอบที่ 4.....63

4.2 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 2 3 4 และกลุ่มทดสอบที่ 1.....64

4.3 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 3 4 1 และกลุ่มทดสอบที่ 2.....65

4.4 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 1 2 4 และกลุ่มทดสอบที่ 3.....66

4.5 ผลการทดสอบของโมเดลทั้ง 4 ครั้ง.....67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 Model ของ Neuron ในคอมพิวเตอร์ 4

2.2 โครงสร้างวงจร Neural Network 5

2.3 โครงสร้างการทำงานของ Deep learning 6

2.4 การทำงานทั้งหมดของระบบ 6

2.5 ชั้นของการทำ Convolutional 7

2.6 ชั้นของการทำ Convolutional 7

2.7 ชั้นของการทำ Subsampling Layers 8

2.8 ชั้นของการทำ pooling 8

2.9 ชั้นของการทำ Fully Conection 9

2.10 การทำงานทั้งหมดใน Convolutional Neuron network 9

2.11 แสดงการทำงานทั้งหมดใน Convolutional Neuron network 10

2.12 ประโยชน์การนำวิธีการการเรียนรู้เชิงลึกมาใช้ในปัจจุบัน 11

2.13 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกใบหน้ามนุษย์ 11

2.14 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกสิ่งที่ไม่มีชีวิต 12

2.15 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกเสียง 12

2.16 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกเครื่องแต่งกาย 13

2.17 ผู้พัฒนา Caffe Deep learnng 14

2.18 การทำงานของ Blob และการเขียนภาษาโปรแกรมมิ่งของ Blob 15

2.19 Forward และ Backward 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

2.20 การเขียนภาษาโปรแกรมมิ่งของการทำงานชั้นต่างๆ 16

2.21 ภาษาโปรแกรมมิ่งที่ใช้ในชั้นเครือข่ายและการทำงาน

ใช้ภาษา C++ และ Pythonขยาย layer 17

2.22 ภาษาโปรแกรมมิ่งที่ใช้ในการเรียนรู้และการทดสอบ 17

2.23 ตัวอย่างฐานข้อมูลที่ใช้ในการเรียนรู้และการทดสอบ 18

2.24 สัญลักษณ์ของ NviDia CUDA 18

2.25 การเปรียบเทียบประสิทธิภาพของ cuDNN 19

2.26 การติดตั้ง caffe ในระบบปฏิบัติการ Windows 20

2.27 การติดตั้ง caffe ในระบบปฏิบัติการ Ubantu 20

2.28 การติดตั้ง caffe ในระบบปฏิบัติการ OS X 21

2.29 Nvidia Digits 21

2.30 ภาพรวมของ Nvidia Digits 22

2.31 การสร้าง Dataset 22

2.32 ตัวอย่างการแสดงผลภาพ 23

2.33 การจำแนกสายพันธุ์สุนัขโดยวิธี Part Localization 24

2.34 การจำแนกสายพันธุนก โดยใช้วิธีการ Deep Convolutional Nets 25

3.1 ภาพรวมของ ระบบ Algorithm Deep Learning 26

3.2 การทำ Cross Validation 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.3 ตัวอย่างสุนัขพันธุ์ Affenpinscher, Alaskan malamute, American eskimo dog 29

3.4 ตัวอย่างสุนัขพันธุ์ Basenji, Bloodhound, Boston Terrier 29

3.5 ตัวอย่างสุนัขพันธุ์ Boxer, Briard, Bull Terrier 30

3.6 ตัวอย่างสุนัขพันธุ์ Cane corso, Cavalier king Charles spaniel, Dachshund 30

3.7 ตัวอย่างสุนัขพันธุ์ Dalmatian, Golden retriever, Irish Terrier 30

3.8 Dataset กลุ่มที่ 1 31

3.9 Dataset กลุ่มที่ 2 32

3.10 Dataset กลุ่มที่ 3 33

3.11 Dataset กลุ่มที่ 4 34

3.12 กระบวนการการทำงานของ DIGITS 35

3.13 การทำงานของขั้นตอน Process Data 36

3.14 การทำงานขั้นตอน Configure DNN 37

3.15 ผลของ Overfitting และ Underfitting 37

3.16 การแยกประเภทภาพ 38

3.17 เข้าพอร์ต Dataset ของ NVIDIA DIGITS 38

3.18 หน้าแรกของ Dataset 39

3.19 ส่วนต่างๆของหน้าแรกของ DIGITS 40

3.20 การนำเข้าฐานข้อมูล 40

3.21 การกำหนดค่าต่างๆ 41

3.22 หน้าการ Train และสถานะการทำงานต่างๆ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.23 หน้าแรกเมื่อทำการสร้างเสร็จพร้อมใช้งาน 43

3.24 สร้างโมเดล 44

3.25 หน้าการสร้างโมเดล 44

3.26 หน้าการสร้างโมเดล 45

3.27 อัพโหลดรูปภาพที่ต้องการทดสอบ 46

3.28 ผลการแยกสายพันธุ์ โดยบอกเปอร์เซ็นต์และค่าสถิติต่างๆ 46

3.29 ผลค่าสถิติต่างๆ 47

4.1 ภาพรวมการทำงานของ Caffe และ Digits 49

4.2 หน้าแรกการเข้า server ของ Nvidia Digits 50

4.3 Nvidia Digits แบ่งเป็น 2 ส่วนการทำงาน 50

4.4 หน้าของการตั้งค่าต่างๆในการสร้าง Datasets 51

4.5 การตั้งค่า Image Type 51

4.6 เลือก Image Encoding ของภาพ 51

4.7 การเลือกขนาดของรูปภาพ 52

4.8 เลือกขนาดรูปภาพที่ต้องการปรับ 52

4.9 ส่วนของฐานข้อมูลที่ถูกเก็บไว้ 52

4.10 ตัวอย่างการเรียก Path ของที่เก็บฐานข้อมูล 53

4.11 ตั้งค่าต่างๆในส่วนการ train 53

4.12 ตั้งชื่อให้ Datasets 53

4.13 หน้าการสร้าง Datasets และการแสดงสถานะการทำงาน 54

4.14 Datasets และโมเดล ที่เคยสร้างไว้ 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.15 เลือกปุ่ม New Image ในการสร้างโมเดล 55

4.16 หน้าการตั้งค่าต่างๆ 55

4.17 การเลือก Dataset เพื่อมาสร้างโมเดล 56

4.18 ส่วนการกำหนดค่า 56

4.19 วิธีการแก้ปัญหาโดยจะเลือกเป็น SGD 57

4.20 การตั้งค่าขงอัตราการเรียนรู้ 58

4.21 ตั้งค่าส่วนของการเปลี่ยนแปลงข้อมูล 58

4.22 การเลือกเครือข่ายการทดสอบ AlexNet 59

4.23 โมเดลที่ได้ทำการสร้างไว้ก่อนหน้านี้ 59

4.24 กำหนดค่าของเครือข่ายเองได้ 59

4.25 กราฟที่จะแสดงถึงการทำงานในส่วนของการTrain ทดสอบชุดข้อมูล 60

4.26 อัพโหลดรูปภาพที่ต้องการทดสอบ 61

4.27 แสดงผลการแยกสายพันธุ์ โดยบอกเปอร์เซ็นต์และค่าสถิติต่างๆและแสดงขั้นการ ทำงาน ของ Caffe ใน Trminal 61

4.28 ผลค่าสถิติต่างๆจนถึงขั้นตอนการทำงานในขั้นต่างๆ 62

1 Terminal ในระบบปฏิบัติการ Ubuntu 73

2 คำสั่งที่เปิดของไฟล์ Makefile.config 77

3 ผลการทำงานของการใช้คำสั่ง runtest 78

4 การเข้าเซิร์ฟเวอร์ของ Nvidia Digits 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันมีผู้คนสนใจที่จะเลี้ยงสุนัขเป็นจำนวนมาก โดยที่อาจจะเคยเห็นสุนัขที่สนใจแต่ไม่อาจทราบได้ทั้งหมดว่าสุนัขตัวนั้นสายพันธุ์อะไร และเนื่องจากมีสุนัขหลากหลายสายพันธุ์เพิ่มมากขึ้น ทำให้ยากต่อการจดจำ และทำให้เราไม่สามารถจำแนกสายพันธุ์สุนัขได้ถูกต้องทั้งหมด มันจึงอาจจะทำให้เป็นปัญหาสำหรับผู้ที่อยากทราบสายพันธุ์ของสุนัขที่สนใจนั้น ๆ

เพราะเหตุนี้จึงจำเป็นต้องใช้วิธีการทางเทคโนโลยีคอมพิวเตอร์ต่าง ๆ เข้ามาช่วยในการจำแนกสายพันธุ์สุนัข ด้วยความสามารถของคอมพิวเตอร์ในปัจจุบันที่มีมากมาย ซึ่งจะทำให้เป็นการง่ายที่จะทำให้ปัญหานั้นหมดไป ทางกลุ่มของผู้จัดทำได้เห็นถึงปัญหานี้ จึงได้ศึกษาวิธีการจำแนกพันธุ์สุนัขโดยนำวิธีการการเรียนรู้แบบเชิงลึกมาใช้ในการทำปัญหาพิเศษในครั้งนี้ โดยเป็นการถ่ายรูปหรืออัพโหลดรูปสุนัขหน้าตรงที่ต้องการทราบเข้าไปภายในซอฟแวร์ Nividia Digits ช่วยในการจำแนกพันธุ์สุนัข วิธีการการเรียนรู้เชิงลึกนี้ คือ ชุดคำสั่ง (algorithm) ที่ถูกสร้างขึ้นมาเพื่อการเรียนรู้ของเครื่องจักร โดยชุดคำสั่งนี้จะทำให้ตัวเครื่องจักรสามารถประมวลผลข้อมูลจำนวนมาก ๆ ด้วยการจำลองเครือข่ายประสาทแบบเดียวกับในสมองของมนุษย์ สามารถเข้าใจภาพได้เหมือนมนุษย์ และจะประมวลผลสิ่งที่ต้องการออกมา

การนำขั้นตอนวิธีการเรียนรู้เชิงลึกนี้มาประยุกต์ใช้ในการจำแนกพันธุ์สุนัขนั้น เพื่อเป็นการแก้ปัญหาของผู้ที่ไม่สามารถจำแนกพันธุ์สุนัขได้และเนื่องจากมีจำนวนสุนัขเพิ่มมากขึ้นในปัจจุบัน ยากต่อการจดจำ และเป็นการศึกษาการทำงานและการใช้ซอฟแวร์ Nvidia Digits และเพื่อทดสอบว่าโมเดลของวิธีการนี้มีความถูกต้องแม่นยำมากแค่ไหน และเพื่อนำไปพัฒนาต่อยอดในอนาคตต่อไป

1.2 วัตถุประสงค์

1) เพื่อศึกษาขั้นตอนและหลักการทำงานของวิธีการการเรียนรู้เชิงลึก
2) เพื่อทำให้ผู้ที่สนใจสุนัขได้รู้จักสายพันธุ์สุนัขมากขึ้นและทำให้ผู้ที่สนใจวิธีการการเรียนรู้เชิงลึกได้ศึกษาและนำไปต่อยอดหรือพัฒนาได้ในอนาคต
3) เพื่อทำให้สามารถจำแนกพันธุ์สุนัขได้ถูกต้องแม่นยำ
4) เพื่อทำการวิเคราะห์และทดสอบเพื่อตรวจสอบความแม่นยำของโมเดลที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตการศึกษา

1) ใช้วิธีการจำแนกสายพันธุ์สุนัขด้วยวิธีการการเรียนรู้เชิงลึก (Deep Learning) โดยจับภาพหน้าตรงของสุนัข
2) ฐานข้อมูลที่ใช้จะเป็นภาพจากฐานข้อมูลจากงานวิจัยก่อนหน้า โดยสุ่มสายพันธุ์สุนัขมา 15 สายพันธุ์ สายพันธุ์ละ 80 รูปภาพ
3) แสดงผลการทำนายสายพันธุ์โดยแสดงเป็นเปอร์เซ็นต์ความถูกต้อง 5 อันดับ
4) ทำการทดสอบโมเดลโดยวิธีการ Cross Validation เพื่อหาเปอร์เซ็นต์ความถูกต้องในการจำแนกสายพันธุ์สุนัขและเปรียบเทียบกับวิธีก่อนหน้า

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ทำให้ผู้ที่ต้องการทราบพันธุ์สุนัขสามารถจำแนกพันธุ์สุนัขได้
2) ศึกษาวิธีการทำงานและนำวิธีการการเรียนรู้เชิงลึกไปต่อยอดได้ในอนาคต
3) ทำให้ผู้ใช้งานรู้จักพันธุ์สุนัขได้มากขึ้น

1.5 ขั้นตอนการดำเนินงาน

1) ศึกษาข้อมูลและสภาพปัญหาต่าง ๆ
2) รวบรวมข้อมูล
3) พัฒนาโดยใช้วิธีการการเรียนรู้เชิงลึก
4) ทดสอบระบบ
5) จัดทำเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การใช้งานคอมพิวเตอร์ทุกวันนี้มนุษย์ยังคงใช้งานคอมพิวเตอร์ในรูปแบบที่ตรงไปตรงมา เช่น การคำนวณสมการคณิตศาสตร์ หรือการประมวลข้อความที่เป็นมีกฎตรงไปตรงมา ข้อดีของคอมพิวเตอร์ที่เราใช้คือความเร็วที่คอมพิวเตอร์สามารถทำงานซ้ำๆ ได้อย่างรวดเร็วเกินกว่าที่มนุษย์จะทำได้ แต่ในงานที่มนุษย์สามารถทำได้ง่ายๆ เช่น การทำความเข้าใจภาพถ่ายสักภาพมีอะไรอยู่ในภาพบ้างสำหรับคนทั่วไปสามารถทำได้โดยง่าย โดยสามารถตอบได้เลยว่าภาพใดมี บ้าน รถ หรือวัตถุอื่นๆ ในบ้านได้อย่างรวดเร็วและแม่นยำ แต่กระบวนการเหล่านี้กลับยากมากสำหรับคอมพิวเตอร์

ความพยายามในการพัฒนาให้คอมพิวเตอร์สามารถทำงานได้เหมือนมนุษย์เป็นสิ่งที่มีการวิจัยมาอย่างต่อเนื่อง คือหากต้องการให้คอมพิวเตอร์ทำความเข้าใจกับข้อมูลที่เราป้อนให้เข้าไป เราต้องสอนให้คอมพิวเตอร์เข้าใจข้อมูลในแบบเดียวกับมนุษย์ คณะผู้วิจัยจึงได้ศึกษาเอกสาร แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง เพื่อแสวงหาความรู้ที่จะนำมาใช้ในการทำปัญหาพิเศษ จึงได้รวบรวมทฤษฎีต่างๆและงานวิจัยโดยแบ่งเป็นหัวข้อใหญ่ 3 ส่วน ดังนี้

2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง
- 2.1.1 โครงข่ายประสาทเทียม (Artificial neural network)
- 2.1.2 การเรียนรู้แบบเชิงลึก (Deep Learning หรือ Convolutional Neuron network)

2.2 ส่วนของโปรแกรมและภาษาที่นำมาใช้
- 2.2.1 Caffe Deep learning และโปรแกรมที่เกี่ยวข้อง
- 2.2.2 NVIDIA DIGITS

2.3 งานวิจัยที่เกี่ยวข้อง
- 2.3.1 Dog Breed Classification Using Part Localization ของ Jiongxin Liu และคณะ
- 2.3.2 การจำแนกสายพันธุ์ของนกโดยใช้วิธีการ Deep Convolutional Nets ของ Steve Branson และคณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1.1 โครงข่ายประสาทเทียม (Artificial neural network)

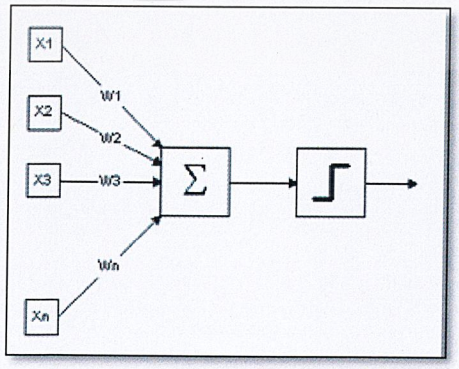
การทำให้คอมพิวเตอร์รู้จักคิดและจดจำในแนวเดียวกับโครงข่ายประสาทของมนุษย์นั้น เป็นการช่วยให้คอมพิวเตอร์ฟังภาษามนุษย์ได้เข้าใจ อ่านออก และรู้จำได้ โครงข่ายประสาทเทียม (Artificial Neural Networks) เป็นวิธีหนึ่งที่ช่วยให้คอมพิวเตอร์เข้าใจภาษา โดยมีผู้ให้คำนิยามโครงข่ายประสาทเทียมไว้มากมาย

โครงข่ายประสาทเทียม (Artificial neural network) หรือที่มักจะเรียกสั้น ๆ ว่า ข่ายงานประสาท คือโมเดลทางคณิตศาสตร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (connectionist) เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำแบบรูป(Pattern Recognition) และการอุปมานความรู้เช่นเดียวกับความสามารถที่มีในสมองมนุษย์

โครงข่ายประสาทเทียม คือ แบบจำลองทางคณิตศาสตร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์โดยที่โครงข่ายประสาทของมนุษย์ประกอบด้วยเซลล์ประสาท (Neuron) และจุดประสานประสาทหรือไซแนปส์(Synapses) โดยโครงสร้างของการส่งสัญญาณโครงข่ายประสาทเทียมมีคุณลักษณะคล้ายกับการส่งผ่านสัญญาณประสาทในสมองของมนุษย์ กล่าวคือ มีความสามารถในการรวบรวมความรู้(knowledge) โดยผ่านกระบวนการเรียนรู้ (learningprocess) และความรู้เหล่านี้จะจัดเก็บอยู่ในโครงข่ายในรูปแบบค่าน้ำหนัก (weight) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่ ๆ เข้าไปค่าน้ำหนักทำหน้าที่เปรียบเสมือนความรู้ที่รวบรวมไว้เพื่อใช้ในการแก้ปัญหาเฉพาะอย่างของมนุษย์ โดยมีหลักการและการทำงานดังนี้

1.หลักการ

สำหรับในคอมพิวเตอร์ Neurons จะประกอบด้วย input และ output โดยจำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะสามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทางานร่วมกันการทำงานนี้ในทางตรรกะแล้วก็จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง ดังรูปที่ 2.1

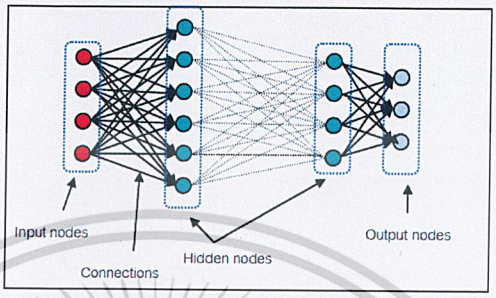


รูปที่ 2.1 Model ของ Neuron ในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การทำงาน

การทำงานของ Neural networks คือเมื่อมี input เข้ามายัง network ก็เอา input มาคูณกับ weight ของแต่ละขา ผลที่ได้จาก input ทุก ๆ ขาของ neuron จะเอามารวมกันแล้วก็เอามาเทียบกับ จุดเริ่มต้น (threshold) ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว neuron ก็จะส่ง output ออกไป output นี้ก็จะถูกส่งไปยัง input ของ neuron อื่น ๆ ที่เชื่อมกันใน network ถ้าค่าน้อยกว่า threshold ก็จะไม่เกิด output ดังรูปที่ 2.2



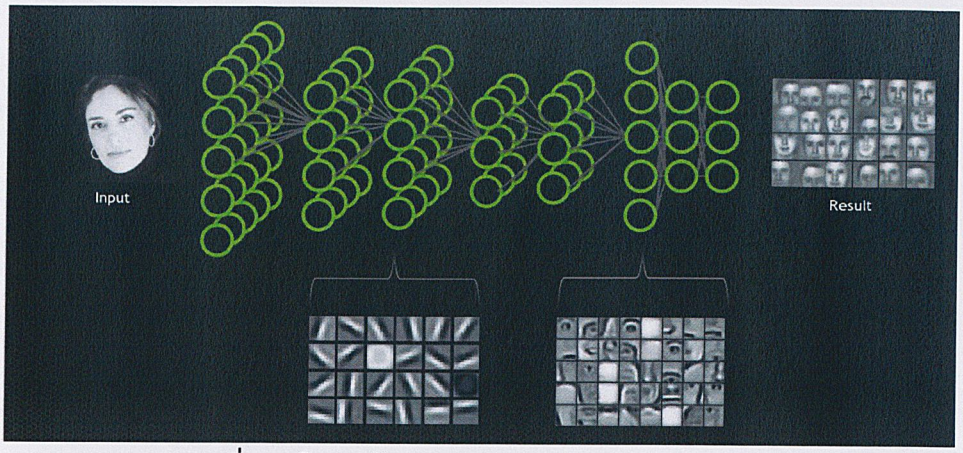
รูปที่ 2.2 โครงสร้างวงจร Neural Network

2.1.2 การเรียนรู้แบบเชิงลึก (Deep Learning หรือ Convolutional Neuron network)

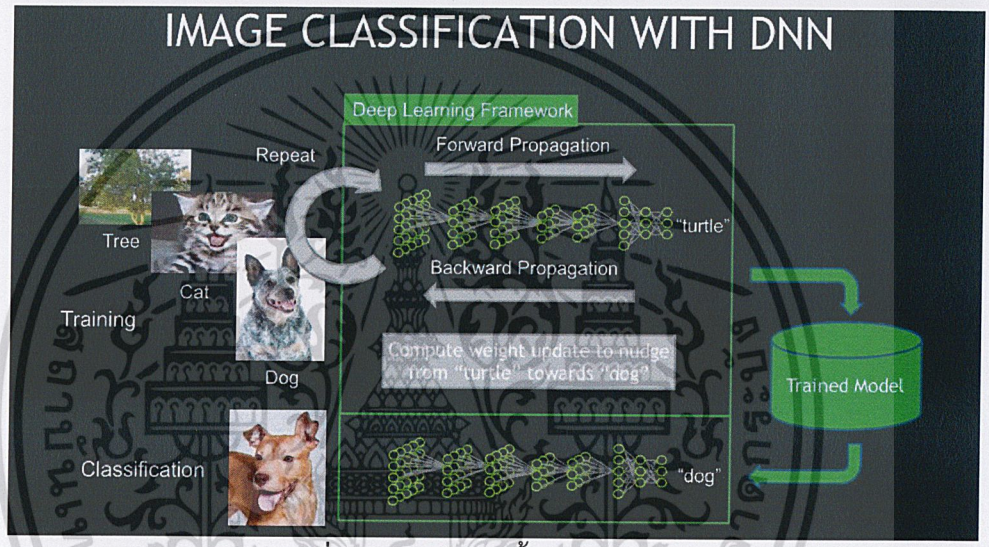
การเรียนรู้เชิงลึก หรือ Deep learning หรือ Convolutional Neuron network เป็นสาขาของการเรียนรู้ของเครื่องจักร พื้นฐานของการเรียนรู้เชิงลึก คือ อัลกอริทึมที่พยายามจะสร้างแบบจำลองเพื่อแทนความหมายของข้อมูลในระดับสูงโดยการสร้างสถาปัตยกรรมข้อมูลขึ้นมาที่ประกอบไปด้วยโครงสร้างย่อยๆหลายอัน และแต่ละอันนั้นได้มาจากการแปลงที่ไม่เป็นเชิงเส้น การเรียนรู้เชิงลึก อาจมองได้ว่าเป็นวิธีการหนึ่งของการเรียนรู้ของเครื่องจักรที่พยายามเรียนรู้วิธีการแทนข้อมูลอย่างมีประสิทธิภาพ ตัวอย่างเช่น รูปภาพภาพหนึ่ง สามารถแทนได้เป็นเวกเตอร์ของความสว่างต่อจุดพิกเซล หรือมองในระดับสูงขึ้นเป็นเซ็ตของขอบของวัตถุต่างๆ หรือมองว่าเป็นพื้นที่ของรูปร่างใดๆก็ได้ การแทนความหมายดังกล่าวจะทำให้การเรียนรู้ที่จะทำงานต่างๆทำได้ง่ายขึ้น ไม่ว่าจะเป็นการรู้จำใบหน้าหรือการรู้จำการแสดงออกทางสีหน้า การเรียนรู้เชิงลึกถือว่าเป็นวิธีการที่มีศักยภาพสูงในการจัดการกับฟีเจอร์สำหรับการเรียนรู้แบบไม่มีผู้สอนหรือการเรียนรู้แบบกึ่งมีผู้สอน

Deep Learning คือ ชุดคำสั่ง (algorithm) ที่ถูกสร้างขึ้นมาเพื่อการเรียนรู้ของเครื่องจักร โดยชุดคำสั่งนี้จะทำให้ตัวเครื่องจักรสามารถประมวลผลข้อมูลจำนวนมากๆด้วยการจำลองเครือข่ายประสาทแบบเดียวกับในสมองของมนุษย์เรา และด้วยวิธี Deep Learning นี้ก็ทำให้ซอฟท์แวร์ด้านการจดจำภาพและเสียงของเครื่องจักรพัฒนาขึ้นมาก จนสามารถนำไปต่อยอดทำเป็นแอพพลิเคชั่น อุปกรณ์ หรือบริการออนไลน์ ที่สามารถเข้าใจภาพ และเสียง ได้เหมือนกับมนุษย์นั่นเอง ดังรูปที่ 2.3 และแสดงภาพการทำงานของระบบดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างการทำงานของ Deep learning



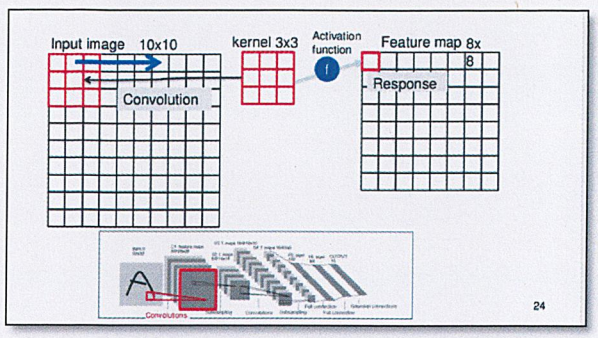
รูปที่ 2.4 การทำงานทั้งหมดของระบบ

1) โครงสร้างการทำงานของ Convolutional Neuron network

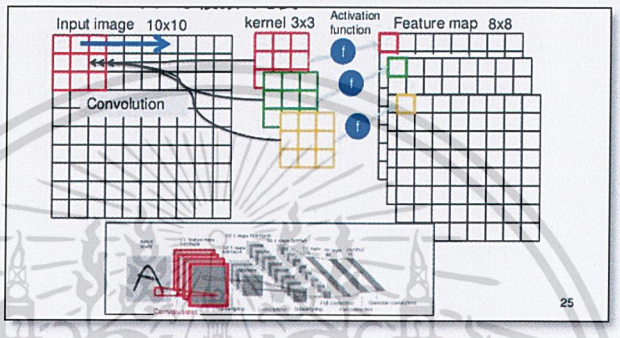
สถาปัตยกรรมของ Convolutional Neuron network เป็นการทำงานที่คล้ายกันกับโครงข่ายประสาทเทียมแต่จะมีชั้นในการทำงานเพิ่มขึ้น หรือเรียกว่า layer โดยมีโครงสร้างย่อยๆ หลายอัน และแต่ละอันนั้นได้มาจากการแปลงที่ไม่เป็นเชิงเส้น ซึ่งจะประกอบด้วย 3 ชั้นสำคัญ ดังนี้

1) **Convolutional Layers** จะประกอบไปด้วย ตารางสี่เหลี่ยม หรือ Filter/Kernel ใช้เป็นตัวกรองในการกรองข้อมูลที่ได้รับเข้ามา(Input) เพื่อกำหนดค่าน้ำหนัก(weights) ตัวกรองนี้จะนำไปใช้ทั่วทั้งภาพที่ได้รับเข้ามา โดยค่าที่ได้จะได้มาจากการนำตัวกรองไปคำนวณกับภาพ ผลลัพธ์ที่ได้จากตัวกรองจะเรียกว่า Feature Map ซึ่งแต่ละ Input ของ Feature Map จะเชื่อมต่อกับ ผลลัพธ์ออก(Output) ของแต่ละ Feature Map โดยมีวิธีการทำงาน ดังรูปที่ 2.5 และ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ชั้นของการทำ Convolutional



รูปที่ 2.6 ชั้นของการทำ Convolutional

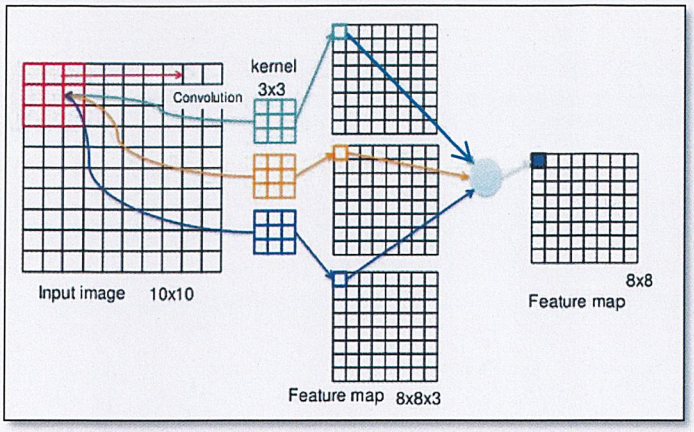
โดยมีสมการในการคำนวณ คือ $h_j^n = Max(0, \sum_{k=1}^{k} h_k^{n-1} * w_{kj}^n)$

h_j^n = ค่าของผลลัพธ์(Output) แต่ละ Feature map
h_k^{n-1} = ข้อมูลที่ได้รับมา(Input) แต่ละ Feature map
w_{kj}^n = Kernel/Filter

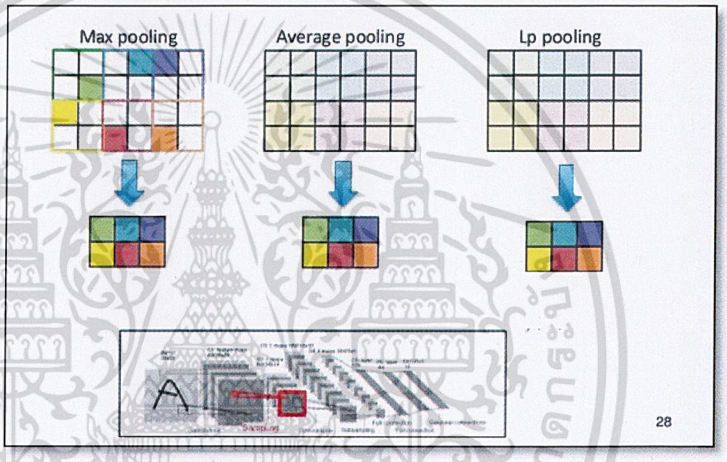
จะคำนวณตามสมการแล้วนำค่าที่มากที่สุดมาใช้ต่อไป

2) **Subsampling Layers** เป็นการลดขนาดของ Input แต่ละชั้นที่ได้รับเข้ามาจากวิธีข้างต้นมาคำนวณ ซึ่งมีหลายวิธีที่สามารถทำได้ คือ max pooling, average pooling, และ stochastic pooling. ดังรูปที่ 2.7 2.8 2.9 และ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ชั้นของการทำ Subsampling Layers



รูปที่ 2.8 ชั้นของการทำ pooling

โดยมีสมการในการคำนวณดังนี้

Max-pooling:

$$h_j^n(x,y)=max_{\bar{x}\in N(x),\,\bar{y}\in N(y)}\,h_j^{n-1}(\bar{x},\bar{y})$$

Average-pooling:

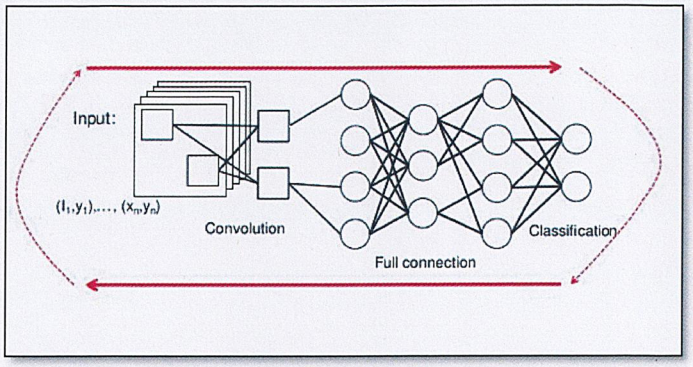
$$h_j^n(x,y)=1/K\sum_{\bar{x}\in N(x),\,\bar{y}\in N(y)}h_j^{n-1}(\bar{x},\bar{y})$$

L2-pooling:

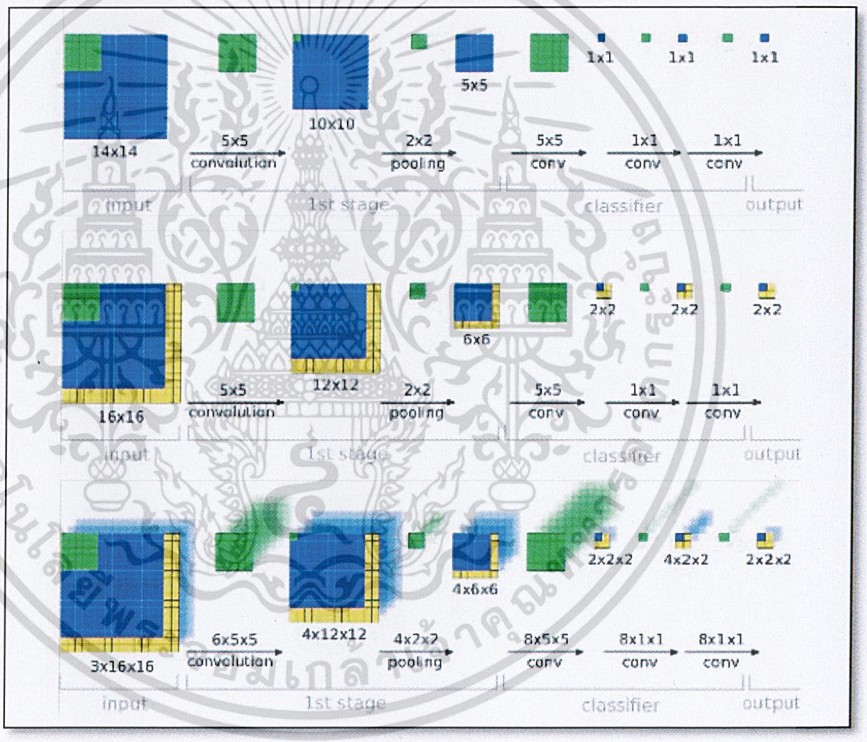
$$h_j^n(x,y)=\sqrt{\sum_{\bar{x}\in N(x),\,\bar{y}\in N(y)}h_j^{n-1}(\bar{x},\bar{y})^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) **Fully Conection** หลังจากการทำ pooling จะนำมาเชื่อมต่อกันทั้งหมดในเครือข่าย

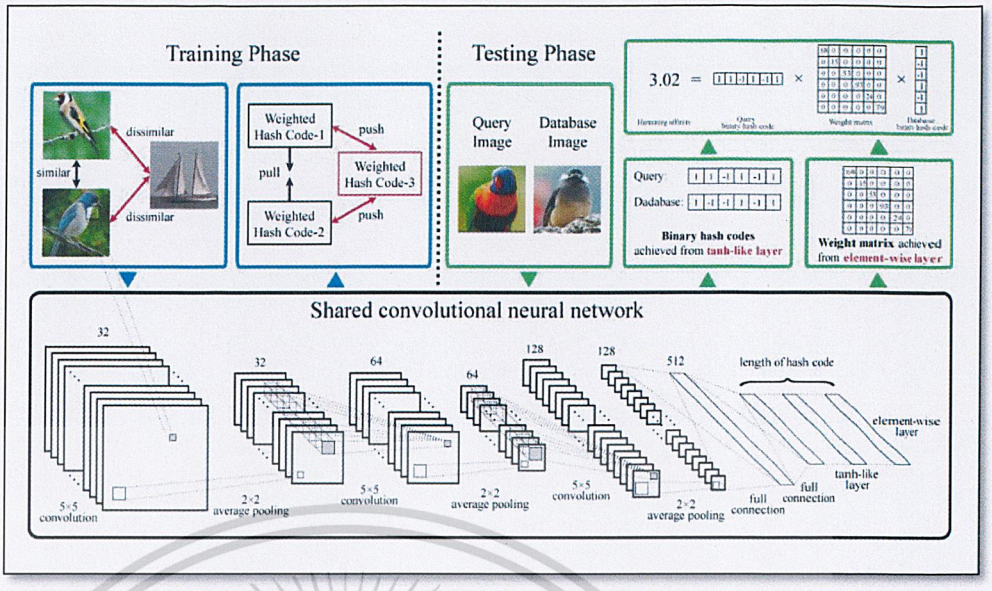


รูปที่ 2.9 ชั้นของการทำ Fully Conection



รูปที่ 2.10 การทำงานทั้งหมดใน Convolutional Neuron network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



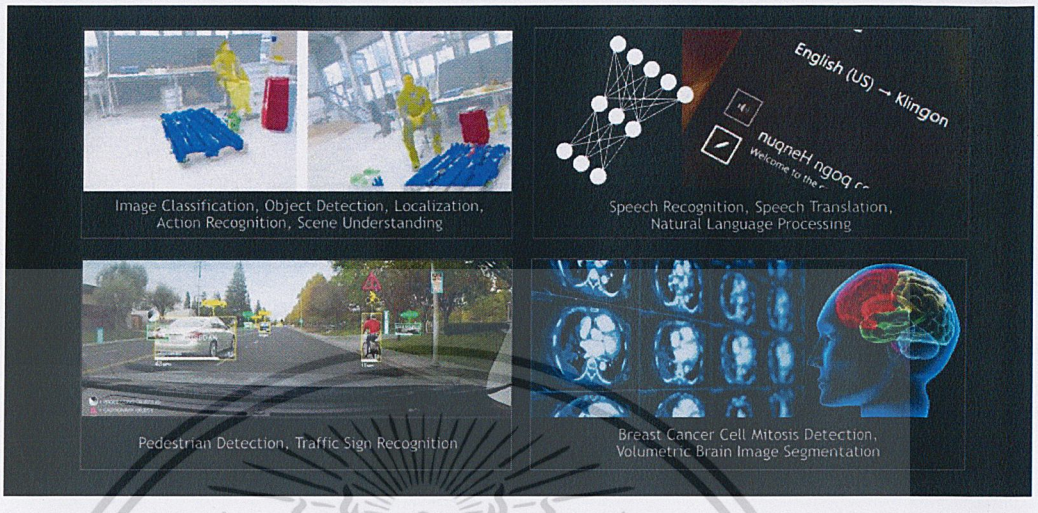
รูปที่ 2.11 การทำงานทั้งหมดใน Convolutional Neuron network

จากรูปที่ 2.11 แสดงภาพการทำงานทั้งหมดในชั้นต่างๆของการทำงาน Deep learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์จากการใช้งาน Deep learning

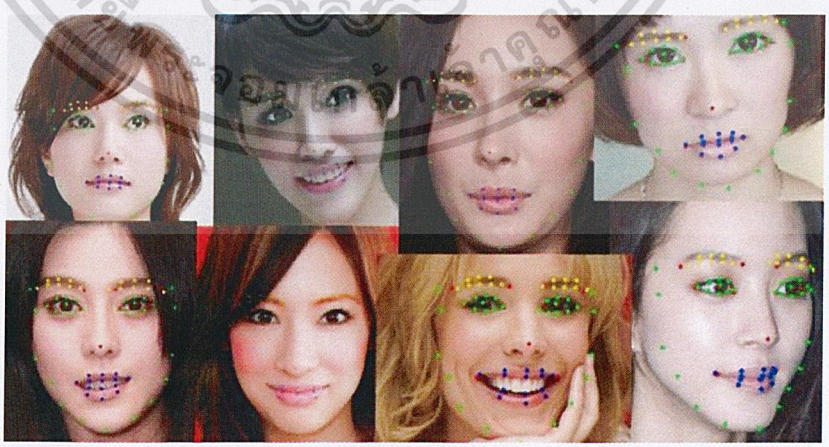
ในปัจจุบันมีการนำ Deep learning มาใช้ในชีวิตประจำวันมากมาย โดยจะยกตัวอย่าง 4 เทคโนโลยีที่ Deep Learning ที่นำมาใช้ในชีวิตประจำวัน ดังรูปที่ 2.12 และหัวข้อหลักๆดังต่อไปนี้



รูปที่ 2.12 ประโยชน์การนำวิธีการการเรียนรู้เชิงลึกมาใช้ในปัจจุบัน

1) **แยกใบหน้ามนุษย์**

สิ่งใกล้ตัวที่สุดที่ Deep Learning มามีบทบาทในชีวิตประจำวันแล้วก็คือ การแยกใบหน้าใน Google+ และ โดย Facebook สามารถแยกใบหน้าแต่ละคนได้อย่างแม่นยำในระดับหนึ่ง หลักการแยกอันแม่นยำนี้มาจากการที่คอมพิวเตอร์จำความสัมพันธ์ แล้วก็แยกตามคนที่เราอาจจะรู้จัก มันจึงให้ความแม่นยำมากเกิน 70% อีก 30% อาจจะมาจากการถ่ายในมุมที่ไม่ชัดมากๆ หรือหัวเล็กจนไม่สามารถแยกได้ ดังรูปที่ 2.13



รูปที่ 2.13 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกใบหน้ามนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) แยกวัตถุที่ไม่ใช่มนุษย์ได้

Deep Learning มีจุดมุ่งหมายอีกหนึ่งอย่างคือ สามารถแยกวัตถุทั้งมีชีวิตและไม่มีชีวิตได้ อย่างเช่น มีภาพของแมว โดยไม่ว่าจะเป็นแมวพันธุ์อะไร Deep Learning ก็ต้องสารมารถรู้ได้ว่าคือแมว (เหมือนที่มนุษย์มองก็รู้ว่ามันคือแมว) แยกแมวจากสุนัข ถ้ามนุษย์เราสามารถแยกสิ่งเหล่านี้ออกจากกันได้ Deep Learning ก็สามารถทำได้ ไม่เพียงแต่สิ่งมีชีวิตเท่านั้น สิ่งไม่มีชีวิตก็ต้องสามารถแยกได้เช่นกัน ดังรูปที่ 2.14

รูปที่ 2.14 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกสิ่งที่ไม่มีชีวิต

3) แยกเสียงและแปลภาษา

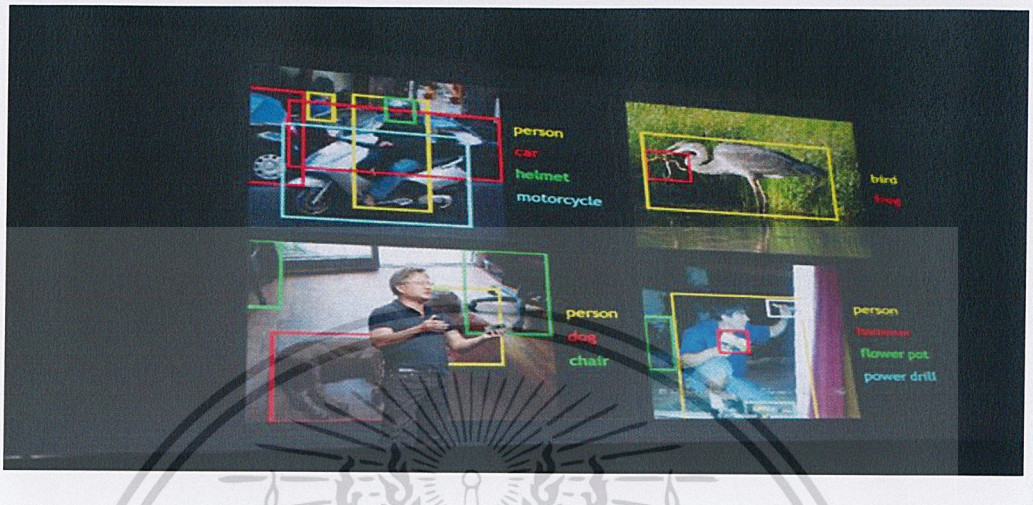
แนวคิดหลักๆ คือ ถ้ามนุษย์พูดสำเนียงปกติ ก็สามารถทำให้แปลเป็นภาษาเขียนได้ทันที โดยไม่ต้องออกเสียงให้ตรงสำเนียง เพราะ Deep Learning จะมีการเรียนรู้จากฐานข้อมูลทั่วโลก ซึ่งฐานข้อมูลที่ใช้ในการเรียนรู้ของ Deep Learning คือ วีดีโอต่างๆใน youtube นั่นจึงทำให้สามารถเรียนรู้ไปพร้อมๆ กันได้ทั้งเสียงและภาพ ดังรูปที่ 2.15

รูปที่ 2.15 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) แยกการแต่งกาย

การแยกการแต่งกายเป็นการค้นหาได้ว่า ใครแต่งตัวในสไตล์เดียวกับเราบ้าง ไสตล์ในที่นี้คือโทนเดียวกัน และสามารถบอกได้อีกว่า เสื้อผ้าแนวที่ใส่อยู่ มันมีอยู่ที่ไหนบ้าง ดังรูปที่ 2.16



รูปที่ 2.16 การใช้วิธีการการเรียนรู้เชิงลึกในการแยกเครื่องแต่งกาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ส่วนของโปรแกรมและภาษาที่นำมาใช้

2.2.1 Caffe Deeplearning และโปรแกรมที่เกี่ยวข้อง

2.2.2 NVIDIA DIGITS

2.2.3 ภาษา python

2.2.1 Caffe DeepLearning

Caffe DeepLearning คือ กรอบการทำงานของโค้ดภาษาทางโปรแกรมมิ่งของ Deep learning หรือเครือข่ายประสาทเทียม เป็นกรอบที่จะใช้ครอบรูปแบบการเขียนโปรแกรม เพื่อให้การพัฒนาโปรแกรมมีรูปแบบที่สวยงาม เป็นรูปแบบเดียวกัน ทำให้โค้ดมีขนาดสั้นเล็กลง เนื่องจากว่าได้มีการนำเอาชุดของโค้ดที่ต้องเขียนซ้ำๆกัน อาจจะเปลี่ยนแค่ตัวแปลบางตัว หรือไม่ได้เปลี่ยนเลย มาเขียนไว้รวมๆกัน เพื่อให้เรียกใช้ได้สะดวก รวมทั้งมีแบบจำลองและตัวอย่างการทำงานอีกด้วย และสามารถใช้งานได้หลายระบบปฏิบัติการ ซึ่งถูกพัฒนาขึ้นโดย Berkeley Vision and Learning Center (BVLC) และ Yangqing Jia ในระหว่างการทำโครงการปริญญาเอกที่มหาวิทยาลัยเบิร์กเลย์ ดังรูป 2.17

รูปที่ 2.17 ผู้พัฒนา Caffe Deep learnng

2.2.1.1 ข้อดีของ Caffe Deep Learning

นอกจาก Caffe Deep learning จะเรียกใช้งานได้สะดวก ยังมีแบบจำลองและลักษณะพิเศษต่างๆ ดังนี้

1) แสดงสถาปัตยกรรมที่สนับสนุนแอพพลิเคชั่นและนวัตกรรมต่างๆ แบบจำลอง (Models)ต่างๆมีประสิทธิภาพและเป็นการเข้ารหัสที่ไม่ยากต่อการใช้งานระหว่าง หน่วยประมวลผลกลางในการทำงานของคอมพิวเตอร์(CPU) และ หน่วยประมวลผลด้านกราฟฟิก 3 มิติ(GPU)

2) มีภาษาโปรแกรมมิ่งที่มีความยืดหยุ่น

3) จากความเร็วของ Caffe Deep learning ทำให้มีประสิทธิภาพมากในการพัฒนาทดสอบ

และการใช้งาน สามารถประมวลผลภาพมากกว่า 60M ต่อวันกับ NVIDIA GPU K40 เดียว นั่นคือ 1 มิลลิวินาที / ภาพ

4) มีการทำงานร่วมกันกับหลายๆส่วน คือ Caffe และ Github

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงการใช้งาน framework ต่างๆในการพัฒนา Convolutional Neuron network

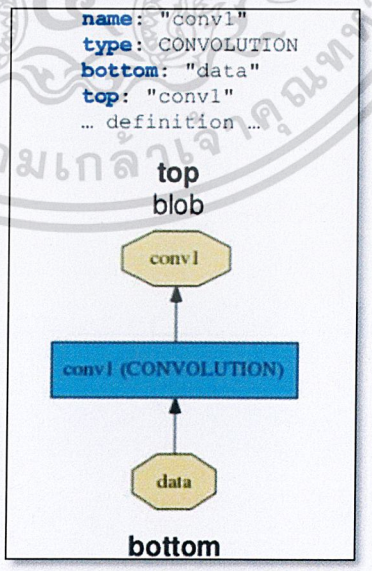
Framework	License	Core language	Binding(s)	CPU	GPU	Open source	Training	Pretrained models	Development
Caffe	BSD	C++	Python, MATLAB	✓	✓	✓	✓	✓	distributed
cuda-convnet [7]	unspecified	C++	Python		✓	✓	✓		discontinued
Decaf [2]	BSD	Python		✓		✓	✓	✓	discontinued
OverFeat [9]	unspecified	Lua	C++,Python	✓				✓	centralized
Theano/Pylearn2 [4]	BSD	Python		✓	✓	✓	✓		distributed
Torch7 [1]	BSD	Lua		✓	✓	✓	✓		distributed

จากตาราง 2.1 แสดงให้เห็นว่า caffe Deep learning มีประสิทธิภาพ ในการพัฒนา Convolutional Neuron network มากกว่า framework อื่นๆ

2.2.1.2 สถาปัตยกรรมของ Caffe DeepLearning

1.) การจัดเก็บข้อมูล (Data Storage)

การจัดเก็บข้อมูลและการสื่อสารข้อมูลจะเป็นแบบ arrays แบบ 4 มิติ ซึ่งเรียกว่า “blobs” หรือ Binary Large OBject คือ ประเภทต่างๆของภาพ เสียง object ของวัตุมัลติมีเดียอื่นๆ โดยเป็นการทำงานร่วมกันระหว่างหน่วยประมวลผลกลางในการทำงานของคอมพิวเตอร์(CPU) และ หน่วยประมวลผลด้านกราฟฟิก 3 มิติ(GPU) จะเป็นการโหลดข้อมูลจากการจัดเก็บข้อมูล(disk) ไปให้ Blob ในการแปลงรหัสในCPU เพื่อประมวลผล เรียกว่า CUDA โดยจะมีการเข้ามาช่วย CPU ประมวลผลชุดคำสั่งบางชนิด เพื่อแบ่งเบาภาระของ CPU ไม่ให้ทำงานหนักจนเกินไป แทนที่การ์ดจอจะประมวลผลภาพหรือแสดงผลภาพเพียงอย่างเดียว โดยสามารถสร้างชุดคำสั่งผ่านโปรแกรมต่างๆ เช่น Java, Fortran, Python, C, C++, C# ดังรูป 2.18



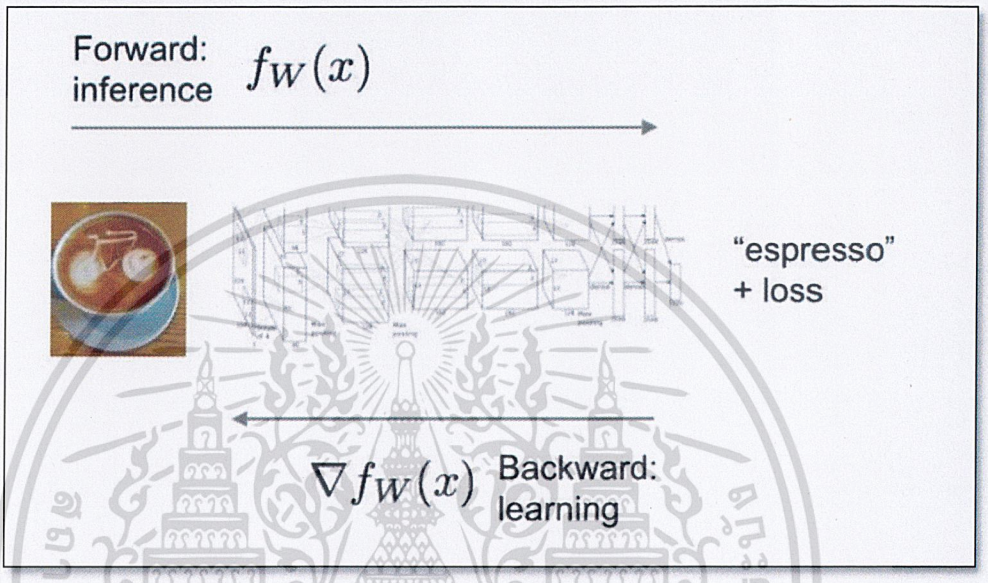
รูปที่ 2.18 การทำงานของ Blob และการเขียนภาษาโปรแกรมมิ่งของ Blob

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

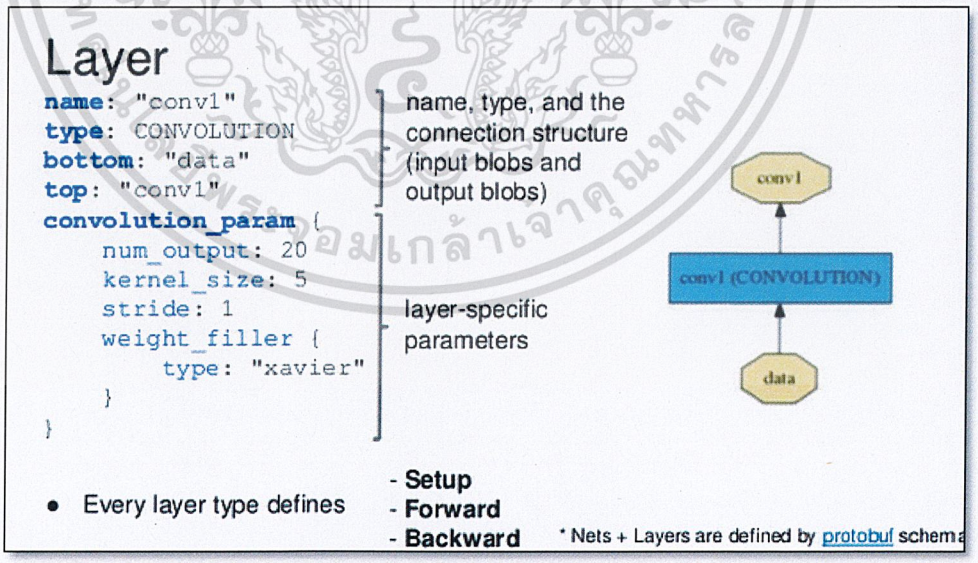
2.) ชั้นการทำงาน (Layers)

ชั้นการทำงานของ Caffe จะเป็นส่วนสำคัญในการทำ Deep learning ซึ่งจะมี 2 ชั้นที่สำคัญสำหรับการดำเนินงานทั้งหมด คือ ดังรูปที่ 2.19 และ 2.20

1) การทำงานแบบ Forward pass คือการทำงานไปข้างหน้าเพื่อให้ได้ output ออกมา
2) การทำงานแบบ Backward pass คือการใช้เป้าหมายเพื่อหา Output



รูปที่ 2.19 Forward และ Backward

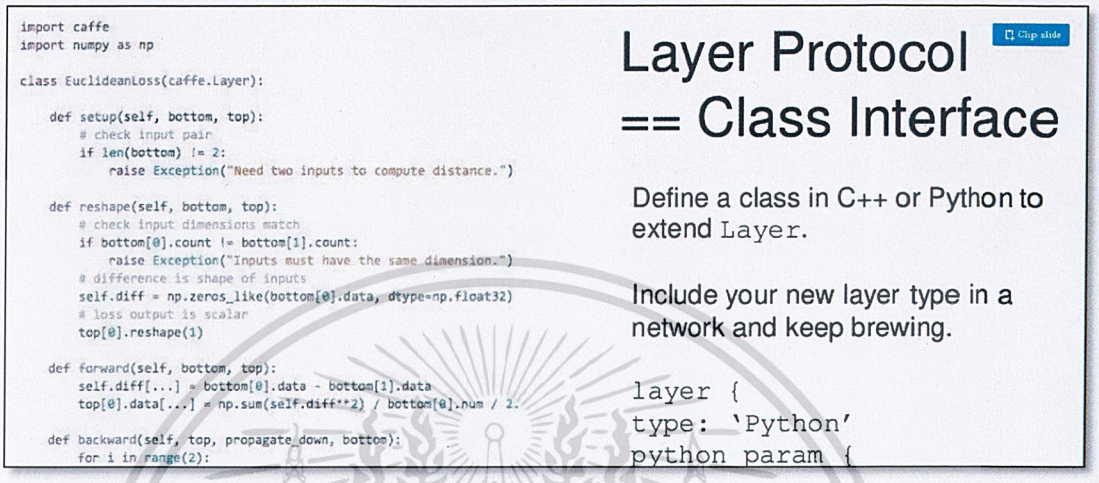


รูปที่ 2.20 การเขียนภาษาโปรแกรมมิ่งของการทำงานชั้นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) ชั้นเครือข่ายและการทำงาน (Networks and Run Mode)

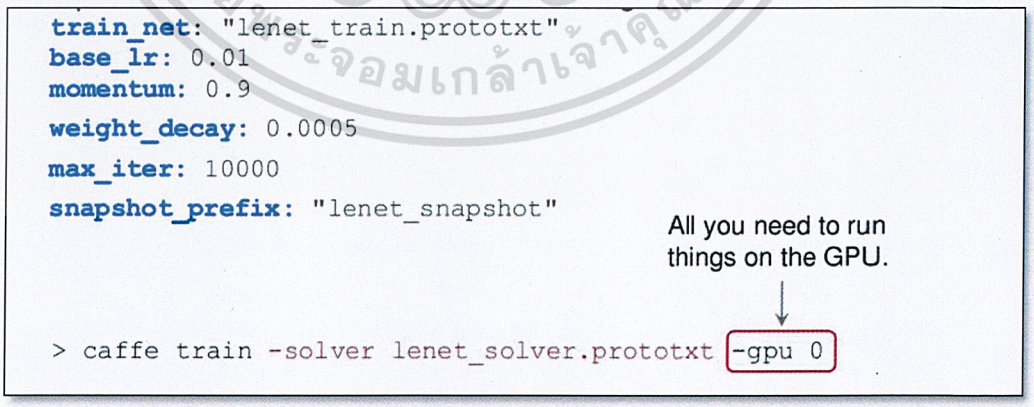
รูปแบบการทำงาน จะเป็นการทำงานของเครื่องจักรแบบเสร็จสมบูรณ์ เป็นประเภทของเครือข่ายที่เริ่มต้นด้วยชั้นข้อมูลที่โหลดจากที่เก็บข้อมูล(disk)และสิ้นสุดในชั้นการคำนวณ จะเป็นการทำงานบนหน่วยประมวลผลกลางในการทำงานของคอมพิวเตอร์(CPU) และ หน่วยประมวลผลด้านกราฟฟิก 3 มิติ(GPU) แบบ single switch. ดังรูปที่ 2.21



รูปที่ 2.21 ภาษาโปรแกรมมิ่งที่ใช้ในชั้นเครือข่ายและการทำงาน ใช้ภาษา C++ และ Python ขยาย layer

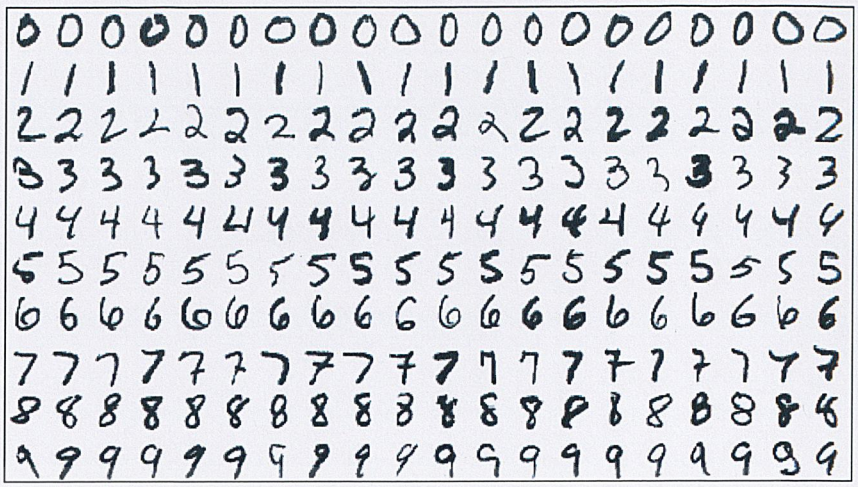
4.)การทดสอบเครือข่าย(Training A Network)

รูปแบบของ Caffe จะมีความเร็วในการทำงานและมีมาตรฐาน จะมี MNIST เป็นตัวทดสอบอย่างนึง MNIST คือ ฐานข้อมูลขนาดใหญ่ที่ใช้ในระบบการประมวลผลของภาพ นอกจากนี้ยังมีฐานข้อมูลที่ใช้กันอย่างแพร่หลายสำหรับการทดสอบในด้านการเรียนรู้ของเครื่องจักรด้วย ดังรูปที่ 2.22 และ รูปที่ 2.23



รูปที่ 2.22 ภาษาโปรแกรมมิ่งที่ใช้ในการเรียนรู้และการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



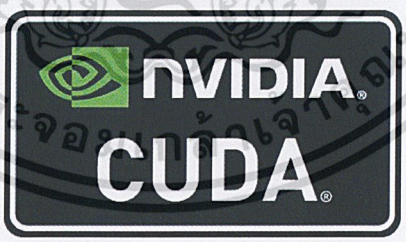
รูปที่ 2.23 ตัวอย่างฐานข้อมูลที่ใช้ในการเรียนรู้และการทดสอบ

2.2.1.3 การใช้งานในระบบปฏิบัติการต่างๆ

Caffe Deep learning ได้ออกแบบให้สามารถใช้ได้หลากหลายในระบบปฏิบัติการต่างๆ ตามความสะดวกของผู้พัฒนา โดยรองรับระบบปฏิบัติการ Ubuntu guide, OS X guide, and RHEL / CentOS / Fedora guide, Windows

ส่วนที่เพิ่มการทำงานให้มีประสิทธิภาพมากขึ้น ดังนี้

1) **CUDA** หรือ สถาปัตยกรรมการผลิตชนิดหนึ่งที่ถูกพัฒนาโดย NVIDIA ซึ่งจะอยู่บนการ์ดจอ วัตถุประสงค์หลักของ CUDA คือ การเข้ามาช่วย CPU ประมวลผลชุดคำสั่งบางชนิด เพื่อแบ่งเบาภาระของ CPU ไม่ให้ทำงานหนักจนเกินไป แทนที่การ์ดจอจะประมวลผลภาพหรือแสดงผลภาพเพียงอย่างเดียว โดยนักเขียนโปรแกรมสามารถสร้างชุดคำสั่งผ่านโปรแกรมยอดนิยมต่างๆ เช่น Java, Fortran, Python, C, C++, C# เป็นต้น โดยใช้ library version 7.0 ดังรูปที่ 2.24



รูปที่ 2.24 สัญลักษณ์ของ NviDia CUDA

2) **BLAS** หรือ Basic Linear Algebra Subroutines คือ การจัดเตรียมสมการพีชคณิตเชิงเส้น ในการดำเนินการแบบ matrix-matrix, matrix-vector BLAS และ ESSL สามารถช่วยประหยัดเวลา ในการปรับการคำนวณทางคณิตศาสตร์ และยังคงจัดเตรียมผลการทำงานที่ดีกว่า หรือการทำให้เหมาะสมที่สุดของ การคำนวณทางคณิตศาสต์ที่เขียนโค้ดขึ้น สามารถเรียกฟังก์ชันได้จากทั้งไลบรารีจาก FORTRAN, C และโปรแกรม C++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

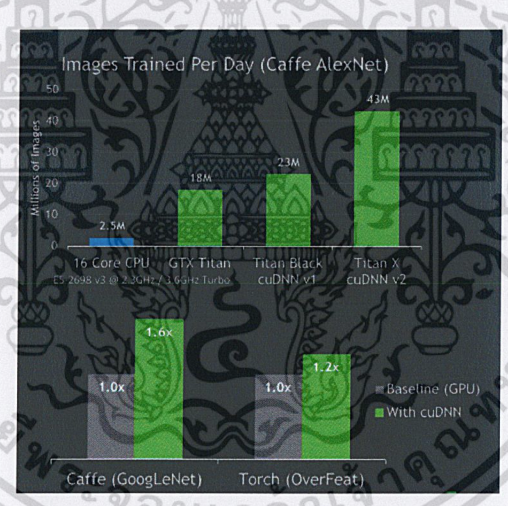
3) **Boost** คือ การโปรแกรมเร่งความเร็วโปรเซสเซอร์และเพิ่มประสิทธิภาพระบบกราฟิกเพื่อรองรับภาระงานสูงสุด ทำให้แกนโปรเซสเซอร์สามารถเร่งความเร็วให้เร็วกว่าอัตราความถี่ปฏิบัติการโดยอัตโนมัติ

4) **OpenCV** เป็น Library ในภาษา C++ และ Phyton สำหรับการพัฒนา โปรแกรมที่เกี่ยวข้องกับ Image Processing และ Computer Vision โดยสามารถพัฒนาได้ทั้ง ในระบบปฏิบัติการวินโดว์ และ ระบบปฏิบัติการ Linux

5) **protobuf, glog, gflags** คือสิ่งที่ช่วยในการแก้จุดบกพร่องและการใช้งานขั้นสูง

6) **IO libraries hdf5, leveldb, snappy, lmdb** เป็นรูปแบบข้อมูลและ Library ที่เป็นรูปแบบไฟล์สำหรับการจัดเก็บและการจัดการข้อมูล

7) **CUDA Deep Neural Network library (cuDNN)** คือ ไลบรารี่พื้นฐานของการฝึกโครงค่ายประสาทเทียมแบบเชิงลึก หรือ Deep learning ดังรูปที่ 2.25



รูปที่ 2.25 การเปรียบเทียบประสิทธิภาพของ cuDNN

ส่วนที่ใช้งานใน Pythoncaffe และ Matlabcaffe

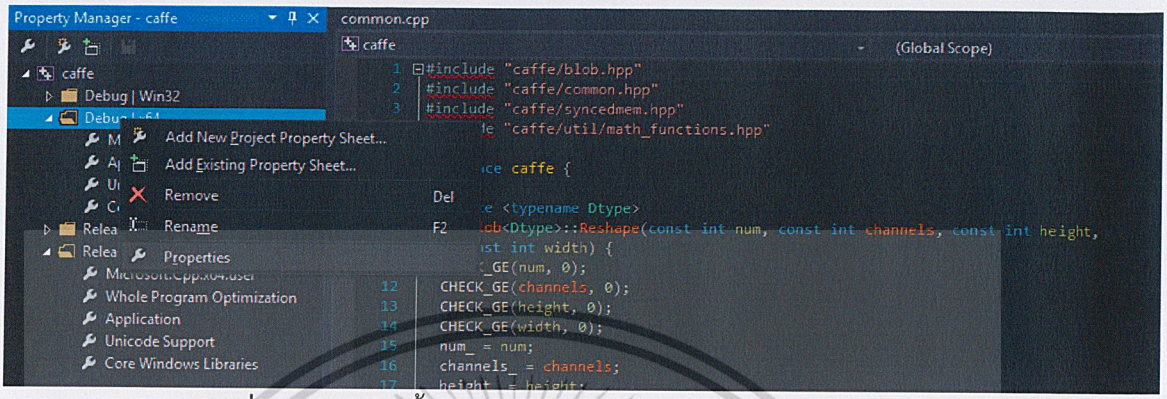
1.) สำหรับ Python Caffe คือ Python 2.7 หรือ Python 3.3+

2.) สำหรับ MATLAB Caffe คือ MATLAB with the mex compiler.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานในระบบปฏิบัติการ Windows

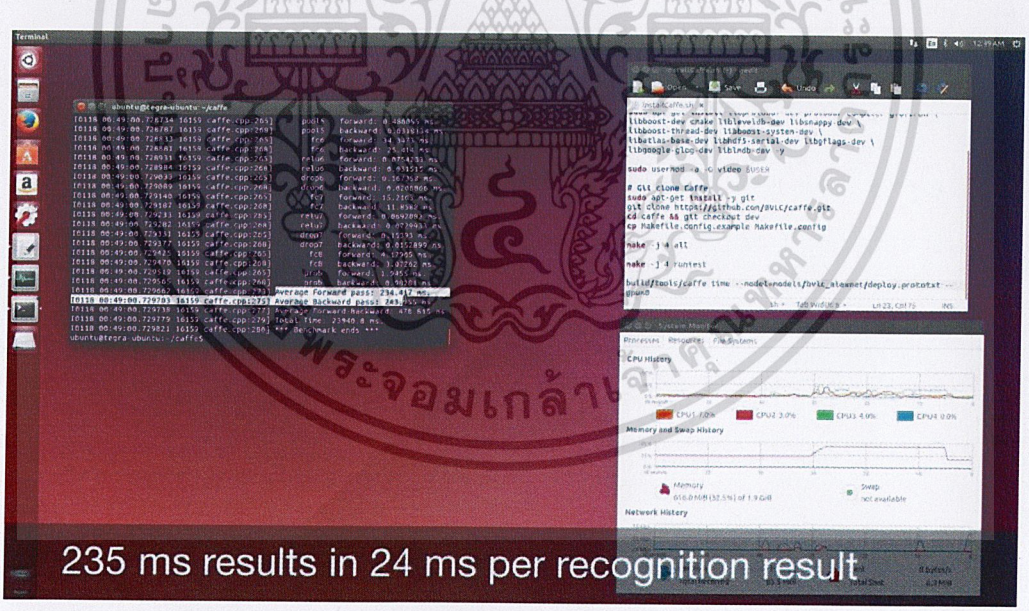
ในระบบปฏิบัติการ Windows จะใช้ Caffe Deeplearning ร่วมกันกับ โปรแกรม Visual studio 2013 , OpenCV และ CUDA ภาษาโปรแกรมมิ่งที่ใช้ คือ ภาษา C++ ดังรูปที่ 2.26



รูปที่ 2.26 การติดตั้ง caffe ในระบบปฏิบัติการ Windows

การใช้งานในระบบปฏิบัติการ Ubantu

ในระบบปฏิบัติการ Ubantu จะใช้ Caffe Deeplearning ร่วมกันกับ CUDA,BLAS,cuDNN และ ภาษาโปรแกรมมิ่งที่ใช้ คือ ภาษา Python ดังรูปที่ 2.27

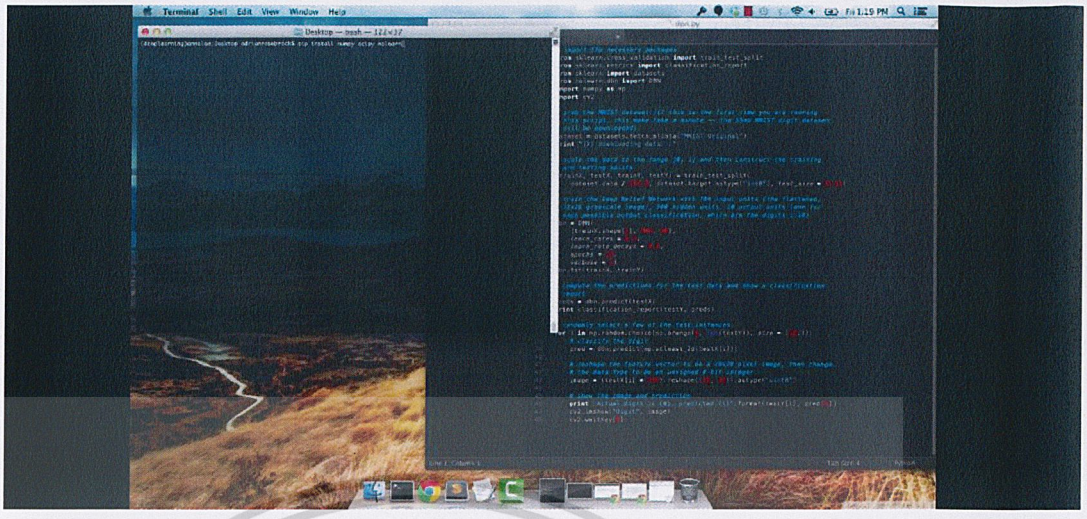


รูปที่ 2.27 การติดตั้ง caffe ในระบบปฏิบัติการ Ubantu

การใช้งานในระบบปฏิบัติการ OS X

ในระบบปฏิบัติการ OS X จะใช้ Caffe Deeplearning ร่วมกันกับ CUDA โดยใช้ Libraly ของภาษาโปรแกรมมิ่ง C++ ภาษาโปรแกรมมิ่งที่ใช้ คือ ภาษา Python ดังรูปที่ 2.28

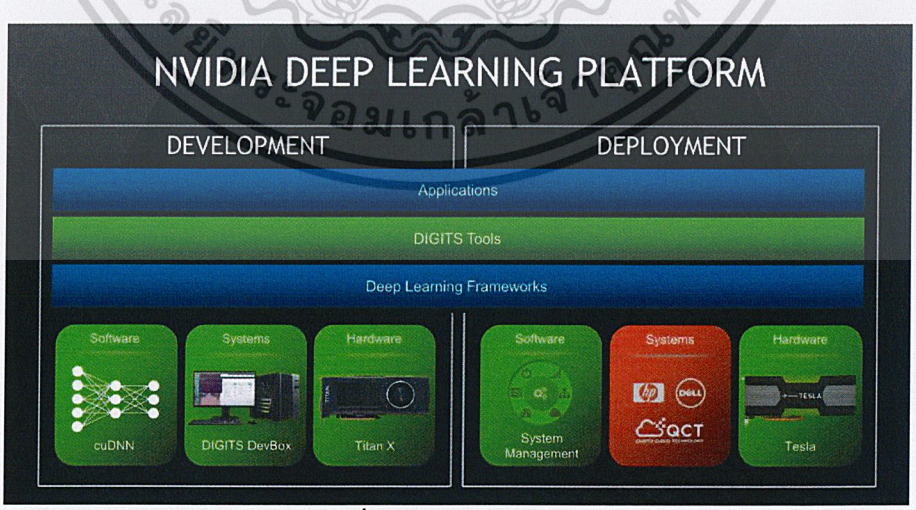
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 การติดตั้ง caffe ในระบบปฏิบัติการ OS X

2.2.2 NVIDIA DIGITS

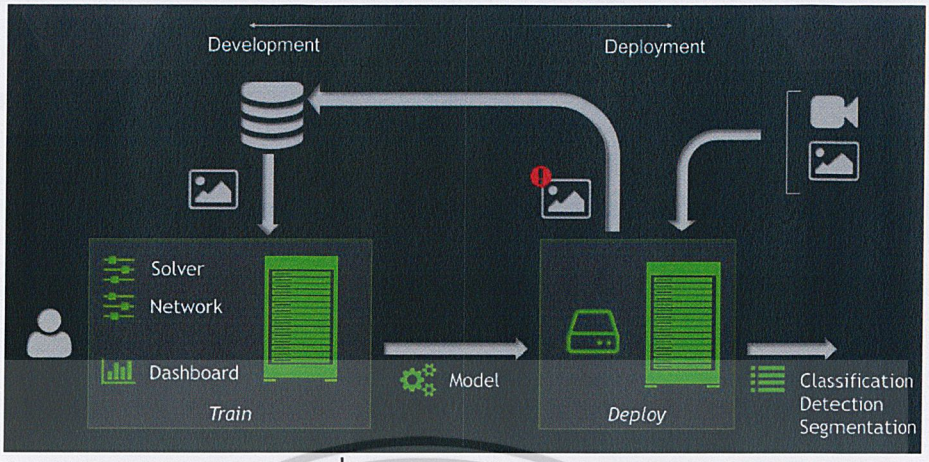
NVIDIA DIGITS คือ ซอฟต์แวร์สำหรับฝึกโครงข่าย Deep Learning ชื่อว่า Deep Learning GPU Training System (DIGITS) เป็นซอฟต์แวร์เพื่อการสร้างโมเดลโครงข่ายประสาทเทียมแบบลึก (deep neural network - DNN) DIGITS ทำให้สามารถออกแบบโครงข่ายประสาทเทียมด้วยตัวเอง ด้วยการคอนฟิกเครือข่ายว่าต้องการจำรูปร่างแบบใด แล้วใส่ชุดข้อมูลเพื่อฝึกโครงข่ายประสาทเพื่อฝึกให้ได้โมเดลโครงข่ายประสาทเทียมไปใช้งาน ตัวซอฟต์แวร์ DIGITS เป็นโอเพนซอร์สสามารถดาวน์โหลดแบบคอมไพล์จากเว็บไซต์ NVIDIA
ดังรูปที่ 2.29



รูปที่ 2.29 Nvidia Digits

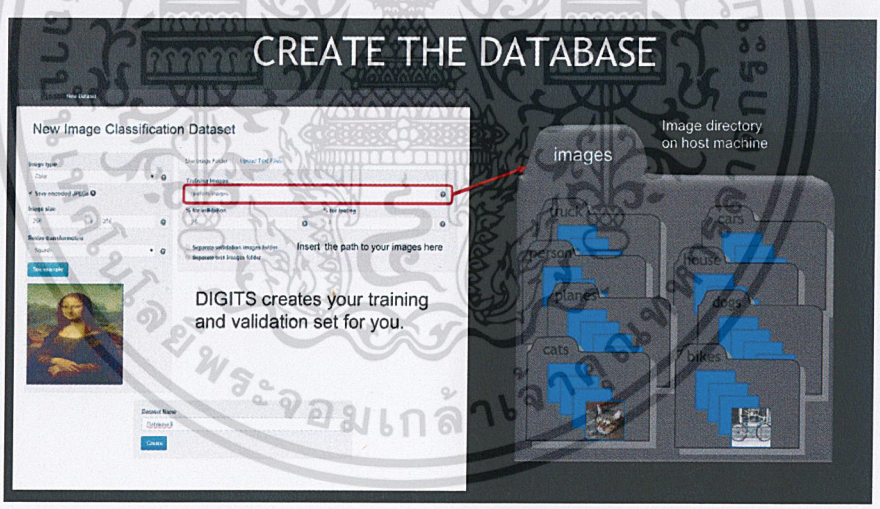
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) โครงสร้างการทำงานของ Caffe และ NVIDIA DIGITS



รูปที่ 2.30 ภาพรวมของ Nvidia Digits

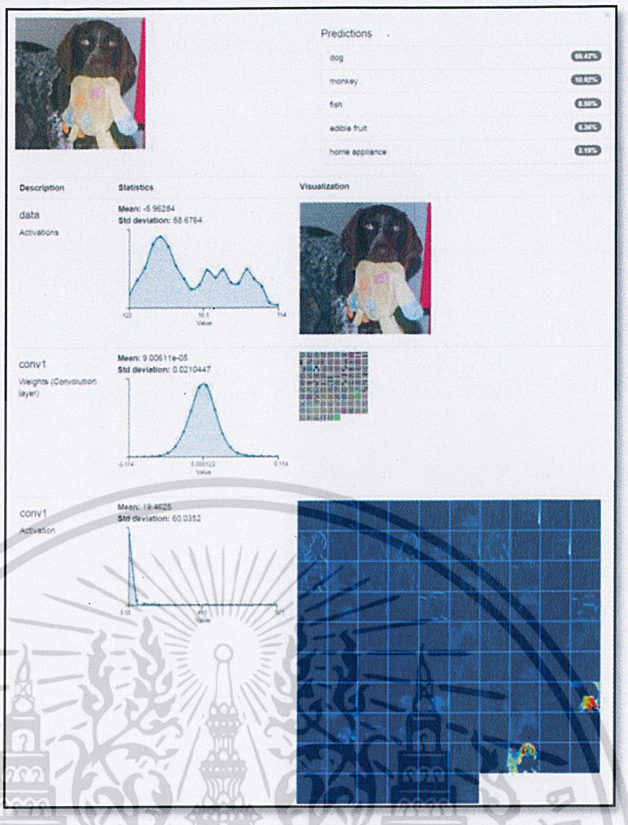
จากรูปที่ 2.30 แบ่งออกเป็น 2 ขั้นตอน คือขั้นตอนการพัฒนาและการใช้งาน โดยจะมี Interface สำหรับผู้ใช้งาน ซึ่งผู้ใช้จะมีการ Dataset เอาไว้เพื่อในการนำไป train ภาพ จากนั้นจะมีการสร้างโมเดลในการ train ภาพ โดย Digits จะทำการ train ภาพจาก Dataset และโมเดลที่ผู้ใช้อัพโหลดไว้ แล้วแสดงผลออกมา ดังตัวอย่างของรูปที่ 2.31 และ 2.32



รูปที่ 2.31 การสร้าง Dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการแสดงผลภาพ



รูปที่ 2.32 ตัวอย่างการแสดงผลภาพ

2) **ข้อดีของ NVIDIA DIGITS**

1) เป็นการ train ภาพ ของเครือข่ายประสาทเทียมแบบเชิงลึก หรือ deep learning โดยมีการสร้างฐานข้อมูลได้โดย AlexNet และ GoogleNet ซึ่งง่ายต่อการใช้งานแบบเดิม
2) เป็นการสร้างโมเดลที่ใช้งานง่ายและรวดเร็ว โดยการติดตั้งในคอมพิวเตอร์และเรียกผ่านบราวเซอร์
3) มีการจัดการการ train ภาพที่หลากหลายเพื่อเพิ่มประสิทธิภาพในการใช้งาน
4) Digits เป็นการใช้งานการ Train ภาพที่ง่าย เนื่องจากมีการออกแบบ Interface ที่ใช้งานได้ง่าย
5) แสดงให้เห็นกระบวนการการทำงานในส่วนต่างๆของ Digits

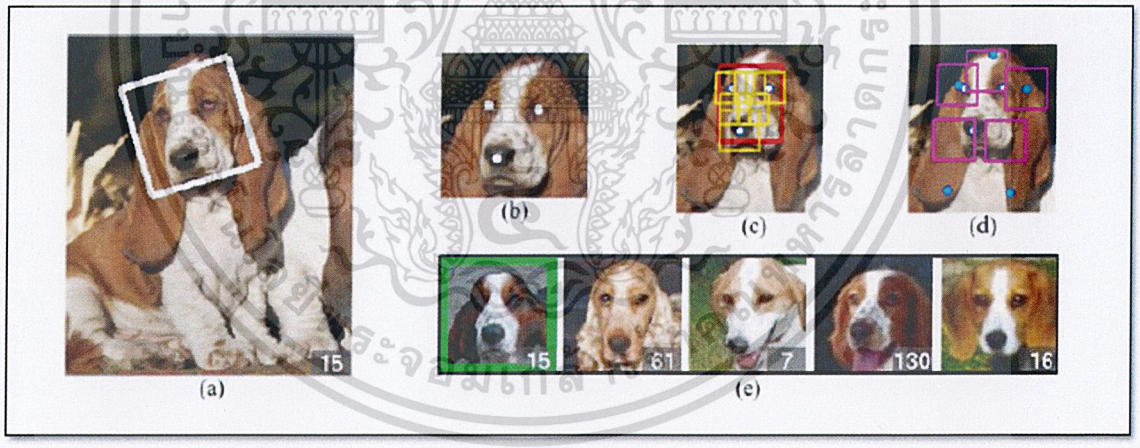
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 งานวิจัยที่เกี่ยวข้อง

จากทำปัญหาพิเศษการจำแนกสายพันธุ์สุนัขโดยวิธีการที่กล่าวมาข้างต้น ผู้วิจัยได้ทำการศึกษาผลงานวิจัยต่างๆและปัญหาพิเศษที่เกี่ยวข้องซึ่งสามารถสรุปสาระสำคัญจากเอกสารที่เกี่ยวข้องได้ดังนี้

2.3.1 Dog Breed Classification Using Part Localization ของ Jiongxin Liu และคณะ

งานวิจัยนี้เป็นการเสนอวิธีที่ใช้ในการจัดหมวดหมู่ของภาพโดยมีการใช้ความหลากหลายของรูปทรงและลักษณะที่มีการปรากฏออกมาของภาพในการจัดหมวดหมู่ของภาพนั้นๆ โดยในงานวิจัยนี้ผู้วิจัยได้ทำการทดสอบโดยการใช้สายพันธุ์ของสุนัขเป็นกรณีทดสอบ ซึ่งจะแสดงให้เห็นว่า หากได้ทำการแยกชิ้นส่วนของภาพที่มีความสอดคล้องกันจะสามารถช่วยเพิ่มประสิทธิภาพในการจัดหมวดหมู่ได้ เนื่องจากลักษณะของชิ้นส่วนของภาพแต่ละภาพที่มีความเหมือนกัน แต่อาจจะมีความแตกต่างกันได้ ผู้วิจัยจึงได้สร้างรูปทรงเรขาคณิตขึ้นมา โดยจะมีการใช้รูปแบบลักษณะของสายพันธุ์สุนัข และการใช้ชิ้นส่วนใบหน้าของสุนัข เป็นตัวช่วยในการถอดและเปรียบเทียบภาพ เพื่อที่จะใช้ในการอธิบายลักษณะของภาพได้ โดยวิธีการนี้มีการแบ่งลำดับชั้นของชิ้นส่วนของภาพ เช่น อาจจะทำในส่วนของใบหน้าก่อน แล้วค่อยมาทำในส่วนตาต่อ เป็นต้น และ วิธีนี้ยังต้องมีการใช้ในสายพันธุ์ที่เฉพาะเจาะจงเพื่อที่เป็นตัวช่วยในการแปลภาพส่วนหนึ่งอีกด้วย การวิจัยนี้ได้ส่งผลประสบความสำเร็จ ในการจัดหมวดหมู่ภาพ 67 % จาก ชุดของข้อมูล ที่มี 133 สายพันธุ์สุนัข ซึ่งจะแบ่งออกเป็นภาพได้ 8351 ภาพด้วยกัน ดังรูปที่ 2.33

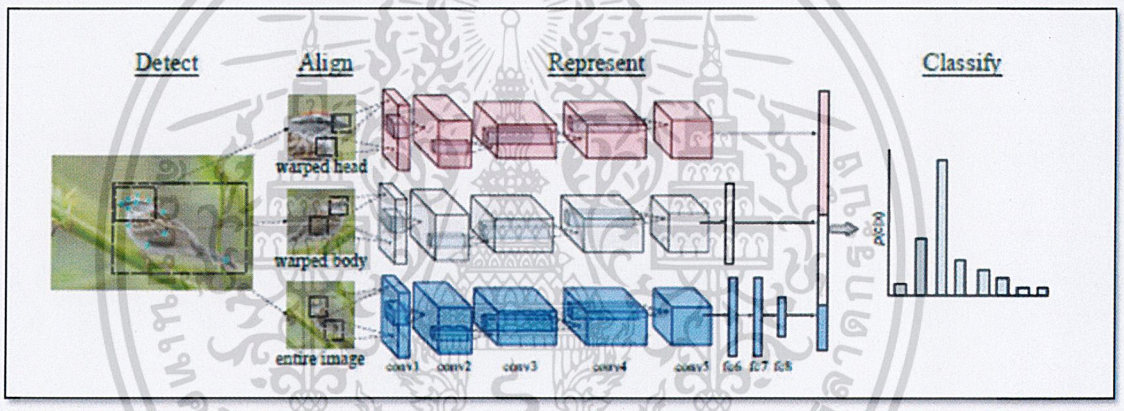


รูปที่ 2.33 การจำแนกสายพันธุ์สุนัขโดยวิธี Part Localization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 Bird Species Categorization Using PoseNormalized Deep Convolutional Nets (การจำแนกสายพันธุ์ของนกโดยใช้วิธีการ Deep Convolutional Nets) ของ Steve Branson และคณะ

งานวิจัยนี้จะเป็นการนำเสนอสถาปัตยกรรมในการจัดหมวดหมู่ของภาพที่มีรายละเอียดที่สูง โดยที่ผู้วิจัยจะมีการใช้สถาปัตยกรรมในการจำแนกประเภทและสายพันธุ์ของนก จากกลุ่มตัวอย่าง นก 200 สายพันธุ์ ซึ่งประกอบไปด้วย 788 ภาพ โดยสถาปัตยกรรมที่ถูกนำมาใช้ในการวิจัยนี้จะเป็นสถาปัตยกรรมที่มีคุณสมบัติในการจัดหมวดหมู่ โดยสถาปัตยกรรมที่ผู้วิจัยได้นำมาใช้คือ Convolutional neural network โดยที่จะเป็นการเรียนรู้ของเครื่องจักรผ่านทางเครือข่ายประสาทเทียม ผู้วิจัยได้ทำการสังเกตว่ารูปแบบที่ผ่านการวิเคราะห์ในระดับที่ต่ำ พร้อมทั้งกับมีขั้นตอนการ pose-normalized จะแสดงให้เห็นถึง คุณลักษณะและชั้นระดับสูงที่มีคุณสมบัติที่ภาพมีความไม่สอดคล้องจะสามารถทำงานได้ดีที่สุด ผลของการทดลองของผู้วิจัย จะแสดงให้เห็นว่า การปรับปรุงขนาดใหญ่ของอัตราการจัดประเภทมีความถูกต้องมากกว่าวิธีการก่อนหน้า (75% เทียบกับ 55-65%) ดังรูปที่ 2.34



รูปที่ 2.34 การจำแนกสายพันธุนก โดยใช้วิธีการ Deep Convolutional Nets

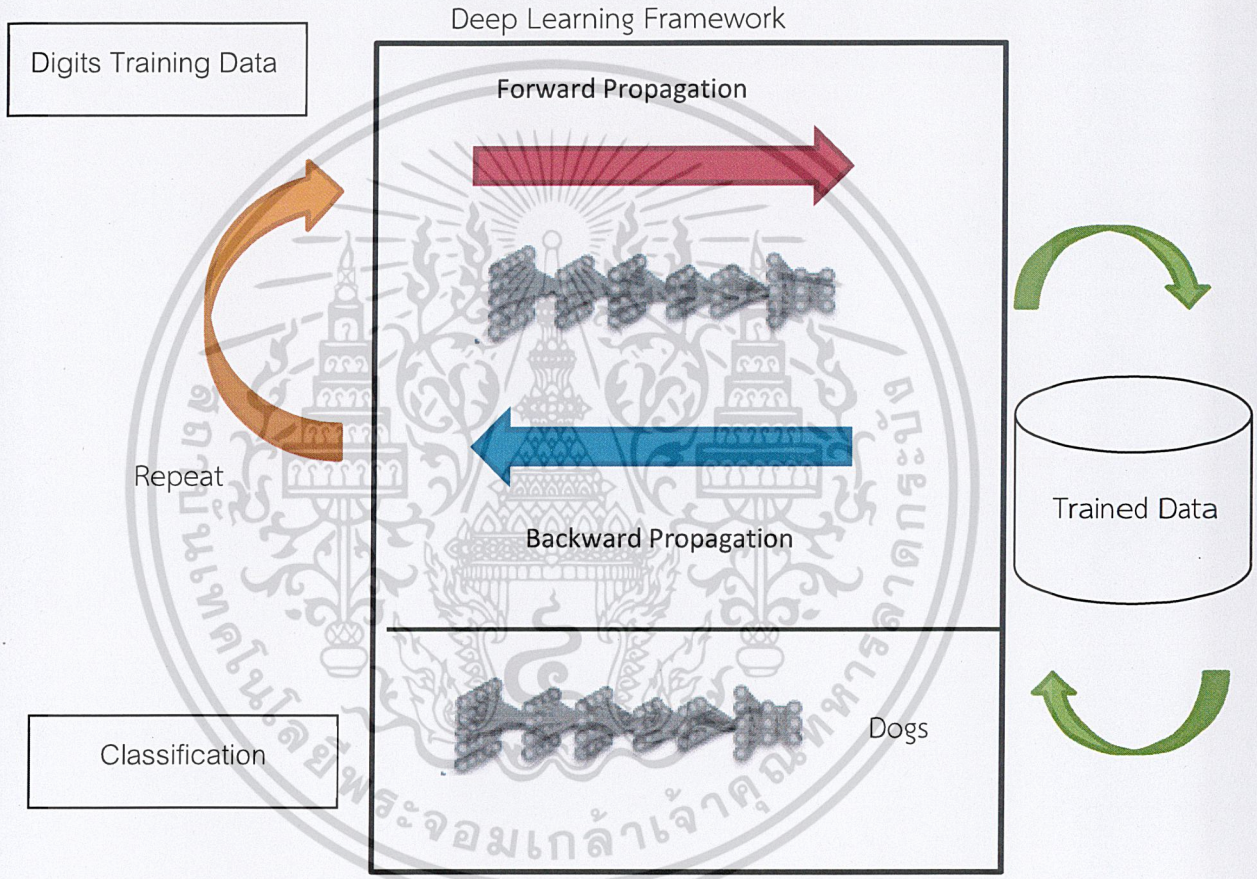
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

3.1 โครงสร้างการทำงานของวิธีการเรียนรู้เชิงลึกโดยใช้ Nvidia Digits

ปัญหาพิเศษนี้เป็นการนำการเรียนรู้เชิงลึกมาใช้ในการจำแนกพันธุ์สุนัขและเป็นการศึกษาซอฟแวร์ Nvidia Digits โดยที่จะมีภาพรวมการทำงานของระบบ Deep Learning ขั้นตอนการจำแนกพันธุ์สุนัขโดยใช้ซอฟแวร์ Nvidia Digits มีโครงสร้างการทำงาน ดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของ ระบบ Algorithm Deep Learning

จากรูปที่ 3.1 จะแสดงภาพรวมของระบบในขั้นตอนวิธีการแยกภาพ ด้วยวิธีของ Algorithm Deep Learning โดยจะมีขั้นตอนเริ่มต้นคือ จะต้องทำการเชื่อมต่อ กับ Server ของ ทาง NVIDIA เมื่อได้ทำการเชื่อมต่อ กับ Server เป็นที่เรียบร้อยแล้ว ก็จะทำการ Train ข้อมูล ในส่วนของ Digits เป็นส่วนต่อมา ในส่วนนี้จะมีการส่งรูปภาพ ที่เราต้องการ Train เข้าไปประมวลผลกับ Algorithm ของ Deep Learning แล้วระบบก็จะนำรูปภาพนั้น ไปทำการ Train กับ ข้อมูลชุด Dataset ที่ได้ทำการเตรียมไว้ เพื่อที่จะทำการเปรียบเทียบ ว่ารูปภาพที่ได้นำมา Train ตรงกับ Category ไหนใน Dataset แล้ว ระบบก็จะทำการประมวลผล และจะ แสดภาพที่มีความใกล้เคียงกับภาพที่เรานำไป Train ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 รายละเอียดวิธีการพัฒนา (Software Specification)

ในส่วนของระบบการจำแนกพันธุ์สุนัขจะแบ่งรายละเอียดออกเป็นสองส่วนด้วยกัน ดังหัวข้อดังต่อไปนี้

3.2.1 ฐานข้อมูลที่ที่นำมาใช้ในการฝึก (Train)

ในการจำแนกสายพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึกนั้น จะใช้ Nvidia Digits ในการจำแนกสายพันธุ์ และนำฐานข้อมูลรูปภาพพันธุ์สุนัข 133 สายพันธุ์สุนัขจากงานวิจัย Dog Breed Classification Using Part Localization ซึ่งทางผู้จัดทำได้สุ่มตัวอย่างพันธุ์สุนัขจาก 133 สายพันธุ์ เหลือเพียง 15 สายพันธุ์ในการทดสอบ สายพันธุ์ละ 80 รูปภาพ และกำหนดรหัสในการทดสอบดังตารางที่ 3.1

ตารางที่ 3.1 ชื่อสายพันธุ์ของสุนัขและรหัสที่ใช้ในการทดสอบ

รหัสสายพันธุ์	ชื่อสายพันธุ์	จำนวนรูปภาพในการ Train
001	Affenpinscher	80
005	Alaskan malamute	80
006	American eskimo dog	80
014	Basenji	80
027	Bloodhound	80
032	Boston terrier	80
034	Boxer	80
036	Briard	80
039	Bull terrier	80
044	Cane corso	80
046	Cavalier king Charles spaniel	80
056	Dachshund	80
057	Dalmatian	80
076	Golden retriever	80
087	Irish terrier	80

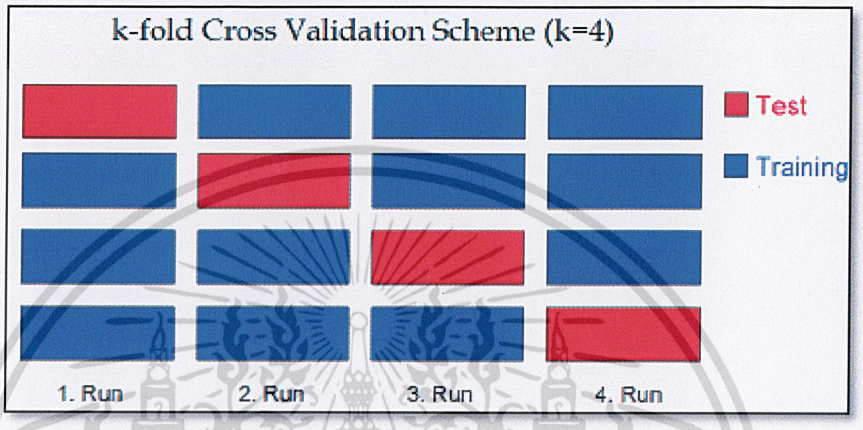
3.2.2 Input / Ouput Specification

1) **Input** : ภาพพันธุ์สุนัขที่ต้องการทราบจากรูปภาพในกลุ่มที่นำมาทดสอบ 1 กลุ่ม และ ฐานข้อมูล 3 กลุ่มในการฝึก
2) **Output** : แสดงเป็นรูปภาพและเปอร์เซ็นต์ที่แม่นยำ 5 อันดับ พร้อมผลทางสถิติต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการวิจัย

เมื่อทำการเตรียมฐานข้อมูลเป็นที่เรียบร้อยแล้วทางผู้จัดทำได้ทำการทดสอบเพื่อที่จะหาวิธีการที่ให้ผลที่แม่นยำที่สุดโดยนำมาเปรียบเทียบกับวิธีก่อนหน้า เพื่อเป็นแนวทางในการนำไปพัฒนาในด้านอื่นๆต่อไป โดยการนำภาพสุนัขมาทดสอบความแม่นยำของโมเดลนั้นจะใช้วิธี Cross Validation ซึ่งเป็นการนำข้อมูล N-1 ชุดไว้ใช้สำหรับการสร้างโมเดล และข้อมูลส่วนที่เหลือไว้สำหรับการทดสอบ ทำวนจนครบจำนวน N ดังตัวอย่างรูปที่ 3.2



รูปที่ 3.2 การทำ Cross Validation

3.3.1 การเตรียมฐานข้อมูลไว้สำหรับการฝึกและทดสอบ

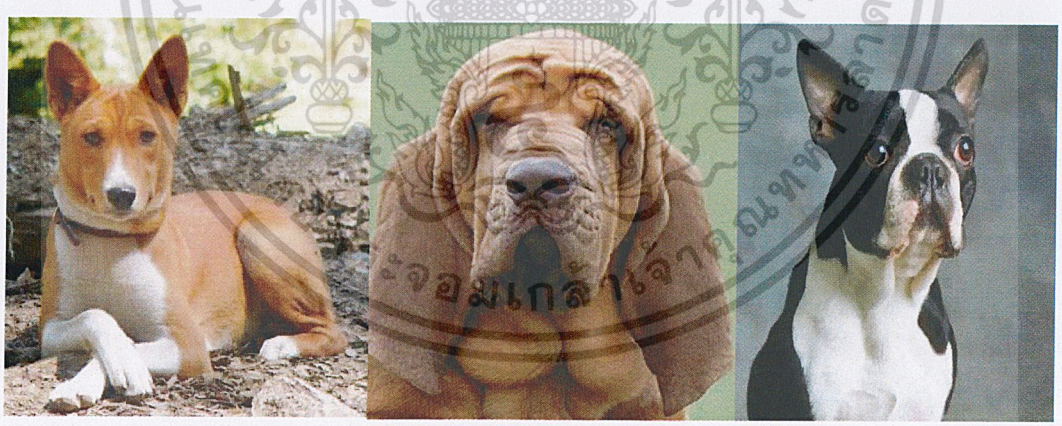
ในการทดสอบนี้จะมีวิธีการทำได้โดยนำฐานข้อมูลที่เตรียมไว้ 15 สายพันธุ์สุนัข จำนวน 80 รูปภาพมาแบ่งเป็นออกเป็น 4 กลุ่ม โดยแต่ละกลุ่มจะประกอบด้วย 15 สายพันธุ์สุนัข และสายพันธุ์ละ 20 รูป แสดงตัวอย่างรูปภาพดังตารางที่ 3.1 และนำมาทำการฝึกฝน (Train) โดยแต่ละครั้งในการทดสอบจะแบ่งชุดข้อมูลเป็นข้อมูลที่ใช้ในการทดสอบความถูกต้อง และชุดข้อมูลที่ใช้ในการเรียนรู้ของเครื่อง โดยนำมาฝึกฝน 3 กลุ่ม และทดสอบ 1 กลุ่ม ทำวนจนครบ 4 กลุ่ม ดังภาพฐานข้อมูลที่ได้จัดกลุ่มดังนี้ 3.8 -3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อสายพันธุ์ของสุนัขและตัวอย่างรูปภาพที่ใช้ในการฝึกและทดสอบ

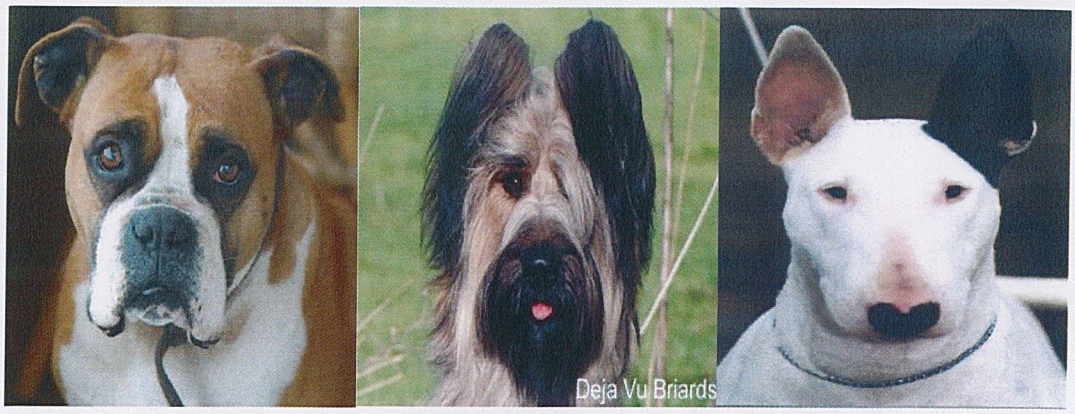
การคัดเลือกพันธุ์สุนัขที่นำมาใช้ในการ Train และการทดสอบนั้น ได้ทำการเลือกจากลักษณะทางกายภาพของสุนัขต่างๆเพื่อให้มีความแตกต่างกัน คือ สีขน (สีขาว สีดำ สีน้ำตาล สีผสม) ความยาวขน ขนาดตัว ลักษณะใบหู (หูตก หูตั้ง) และลักษณะของปาก ตา จมูก เป็นต้น และแสดงตัวอย่างรูปภาพสุนัขที่นำมาใช้ ดังรูปที่ 3.3-3.7

รูปที่ 3.3 ตัวอย่างสุนัขพันธุ์ Affenpinscher, Alaskan malamute, American eskimo dog



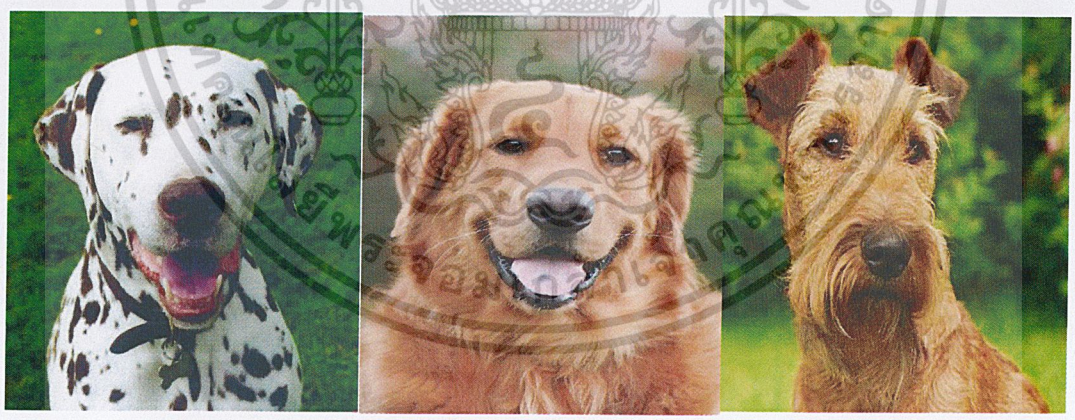
รูปที่ 3.4 ตัวอย่างสุนัขพันธุ์ Basenji, Bloodhound, Boston Terrier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ตัวอย่างสุนัขพันธุ์ Boxer, Briard, Bull Terrier

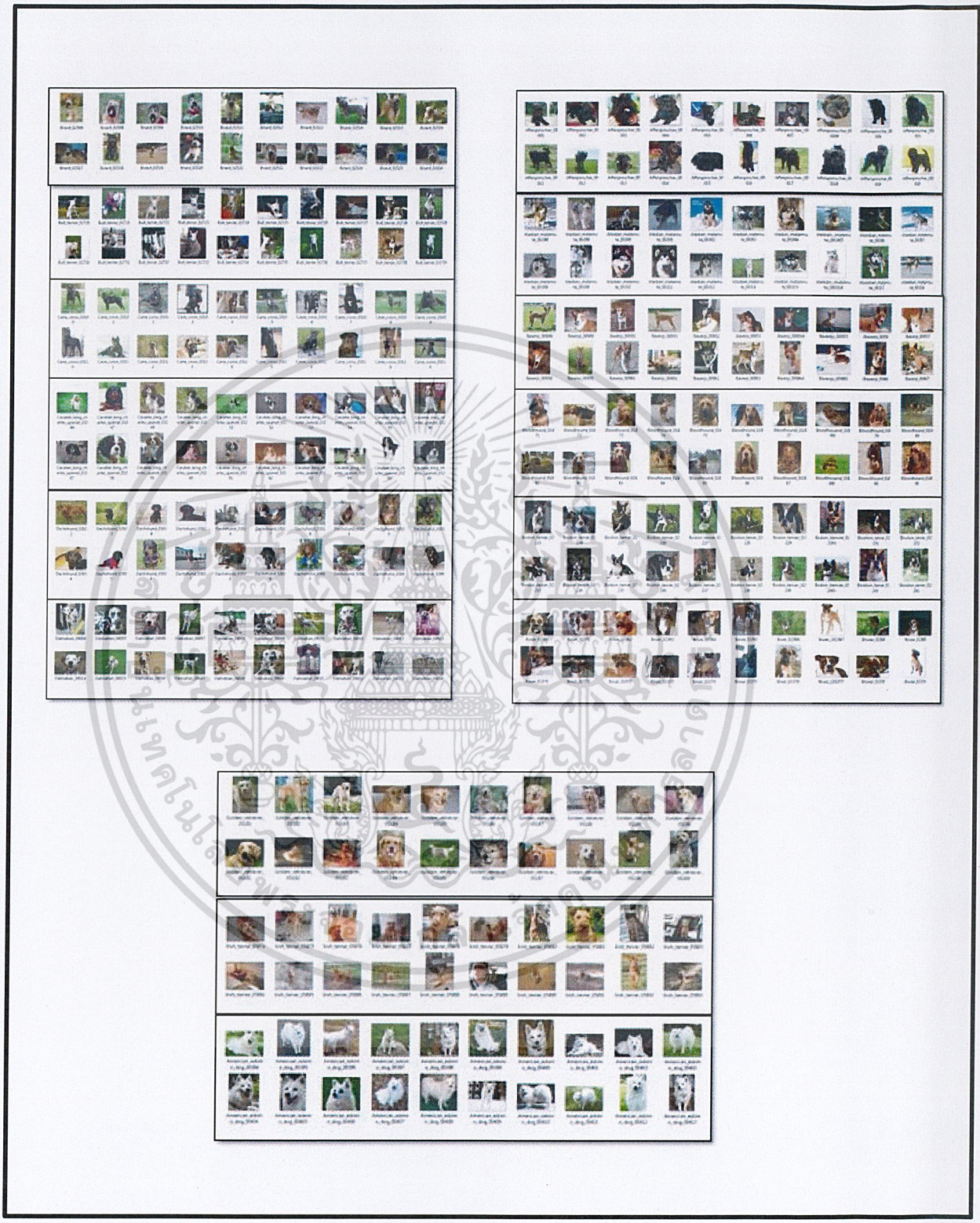
รูปที่ 3.6 ตัวอย่างสุนัขพันธุ์ Cane corso, Cavalier king Charles spaniel, Dachshund



รูปที่ 3.7 ตัวอย่างสุนัขพันธุ์ Dalmatian, Golden retriever, Irish Terrier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) กลุ่มที่ 1 มีจำนวน 15 สายพันธุ์สุนัข และ สายพันธุ์ละ 20 รูป ดังรูปที่ 3.8



รูปที่ 3.8 Dataset กลุ่มที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) กลุ่มที่ 2 มีจำนวน 15 สายพันธุ์สุนัข และ สายพันธุ์ละ 20 รูป ดังรูปที่ 3.9

รูปที่ 3.9 Dataset กลุ่มที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) กลุ่มที่ 3 มีจำนวน 15 สายพันธุ์สุนัข และ สายพันธุ์ละ 20 รูป ดังรูปที่ 3.10

รูปที่ 3.10 Dataset กลุ่มที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) กลุ่มที่ 4 มีจำนวน 15 สายพันธุ์สุนัข และ สายพันธุ์ละ 20 รูป ดังรูปที่ 3.11

รูปที่ 3.11 Dataset กลุ่มที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

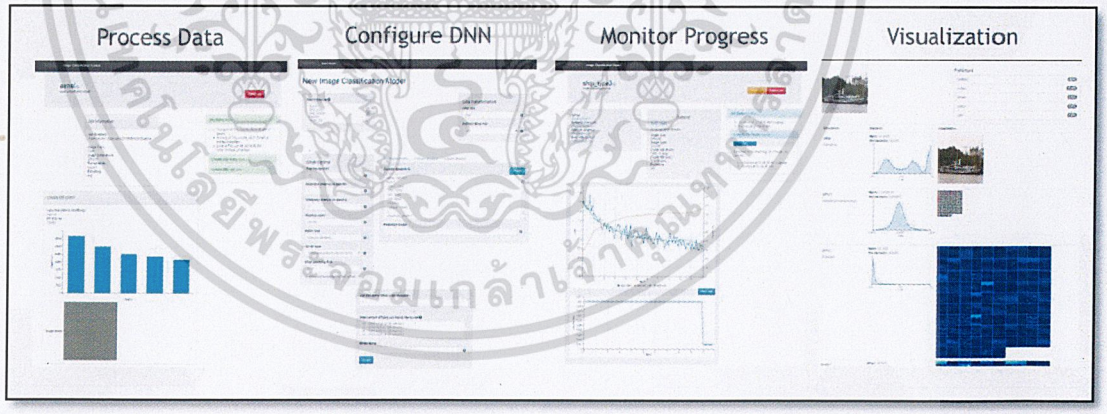
3.3.2 ขั้นตอนการฝึกโมเดล

ขั้นตอนการฝึกจะทำการฝึกและทดสอบโดยใช้ซอฟต์แวร์ของ Nvidia Digits ในการจำแนกพันธุ์สุนัข และในขั้นตอนการทดสอบความแม่นยำของโมเดลนี้จะเป็นการนำผลความถูกต้องจากการทดลองมาเปรียบเทียบกับวิธีก่อนหน้า โดยใช้การทำ Cross validation คือการแบ่งข้อมูลออกเป็น 2 ส่วน โดยส่วนแรกนั้นจะเป็นข้อมูลที่แบ่งไว้สำหรับการฝึกข้อมูลและส่วนที่2 นั้นไว้สำหรับทดสอบ ดังนี้

1) กลุ่มที่ 1 กลุ่มที่ 2 และกลุ่มที่ 3 ไว้ทำการเรียนรู้ของเครื่อง และ กลุ่มที่ 4 ไว้ทำการทดสอบ
2) กลุ่มที่ 1 กลุ่มที่ 2 และกลุ่มที่ 4 ไว้ทำการเรียนรู้ของเครื่อง และ กลุ่มที่ 3 ไว้ทำการทดสอบ
3) กลุ่มที่ 1 กลุ่มที่ 3 และกลุ่มที่ 4 ไว้ทำการเรียนรู้ของเครื่อง และ กลุ่มที่ 2 ไว้ทำการทดสอบ
4) กลุ่มที่ 2 กลุ่มที่ 3 และกลุ่มที่ 4 ไว้ทำการเรียนรู้ของเครื่อง และ กลุ่มที่ 1 ไว้ทำการทดสอบ

1) สร้างโมเดลไว้สำหรับฝึกรูปภาพ โดยใช้ NVIDIA DIGITS

นำฐานข้อมูลที่ได้เตรียมไว้มาทำการฝึกในซอฟต์แวร์โดยใช้ Nvidia Digits ซึ่ง Digits คือ Dataset ที่ใช้ในการสร้างโมเดลในการ Train รูปภาพ โดยทำงานรวมกับ Framework Caffe เมื่อมีสร้างฐานข้อมูลรูปภาพเข้าไปใน Dataset ของ DIGTIS จะมีการนำไปประมวลผลใน Caffe และแสดงผลการจำแนกประเภทออกมาผ่านทาง Interface ของ DIGITS ซึ่งมีกระบวนการการทำงาน 4 ขั้นตอน ดังรูปที่ 3.12



รูปที่ 3.12 กระบวนการการทำงานของ DIGITS

กระบวนการการทำงานของ DIGITS แบ่งออกเป็น 4 ขั้นตอน ดังรูปที่ 3.12 ดังนี้

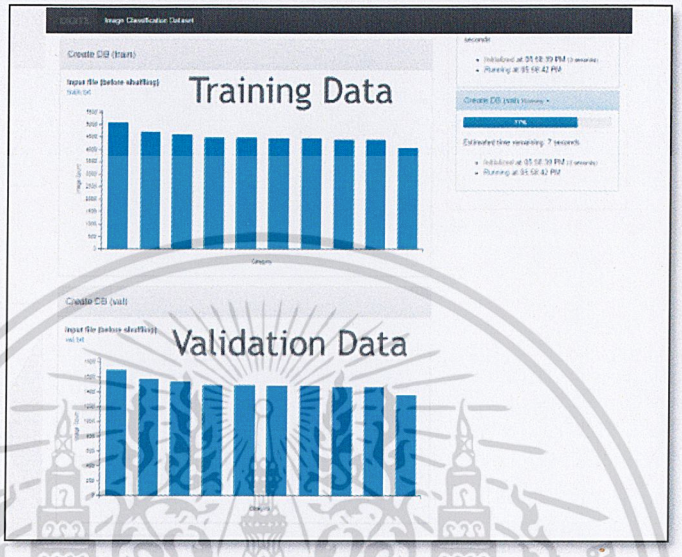
1) **Process Data** คือ ขั้นตอนการประมวลผลข้อมูลที่ได้ทำการสร้าง Dataset รูปภาพเอาไว้แบ่งเป็น 2 ส่วน ดังนี้

1.1 **Training Data** คือ การสอนให้เครือข่ายในการจำแนกประเภทของวัตถุ ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 **Validation Data** คือ การตรวจสอบประสิทธิภาพของข้อมูลและเครือข่าย ดังรูปที่ 3.13

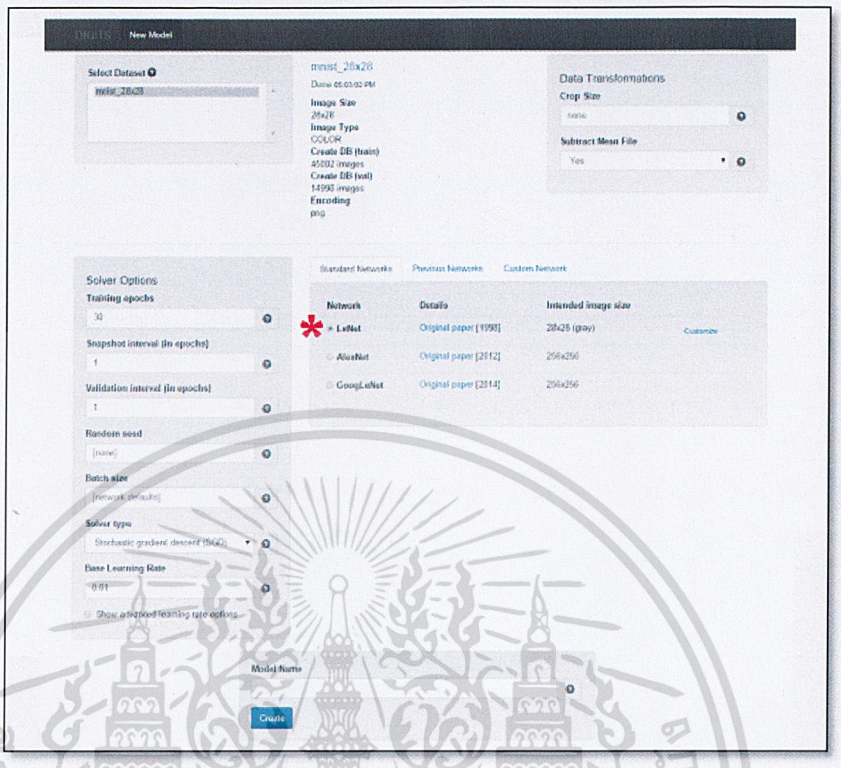
1. ข้อมูลนี้เป็นเพียงใช้สำหรับการทดสอบความสามารถทั่วไปของเครือข่าย
2. ไม่สามารถใช้เป็นการ Train ได้



รูปที่ 3.13 การทำงานของขั้นตอน Process Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) **Configure DNN** คือ การกำหนดค่าต่างๆในเครือข่าย โดยใช้ Dataset ที่ได้สร้างไว้ ดังรูปที่ 3.14

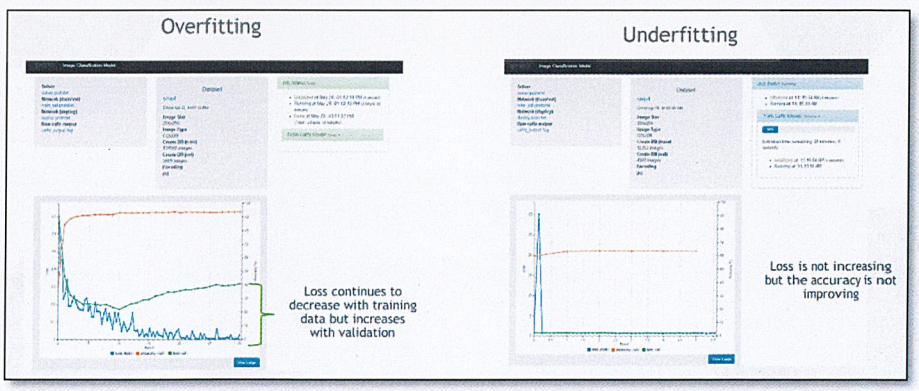


รูปที่ 3.14 การทำงานขั้นตอน Configure DNN

1) **Monitor Progress** คือ การติดตามความคืบหน้า เมื่อสร้าง Dataset และ Model เสร็จ

1) **Overfitting** คือ การที่ Model ที่ได้จากการใช้ Training Data set มีค่าความถูกต้องในการบ่งบอกคลาสเป้าหมายสูง แต่เมื่อนำไปใช้กับข้อมูล Test Data set ได้ค่าความถูกต้องต่ำ ดังรูปที่ 3.15

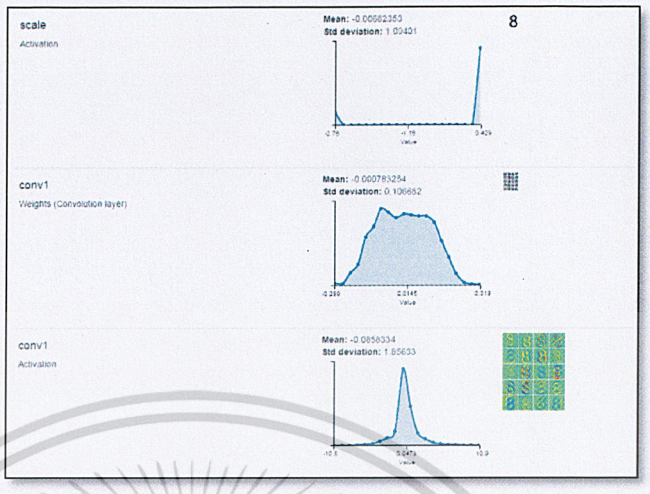
2) **Underfitting** คือ การที่ Model ที่ได้จากการ Training Data set มีความถูกต้องน้อยหรือต่ำ ดังรูปที่ 3.15



รูปที่ 3.15 ผลของ Overfitting และ Underfitting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) **Visualization** คือ การสร้างภาพเพื่อคัดแยกประเภท ตัวอย่างดังรูปที่ 3.16



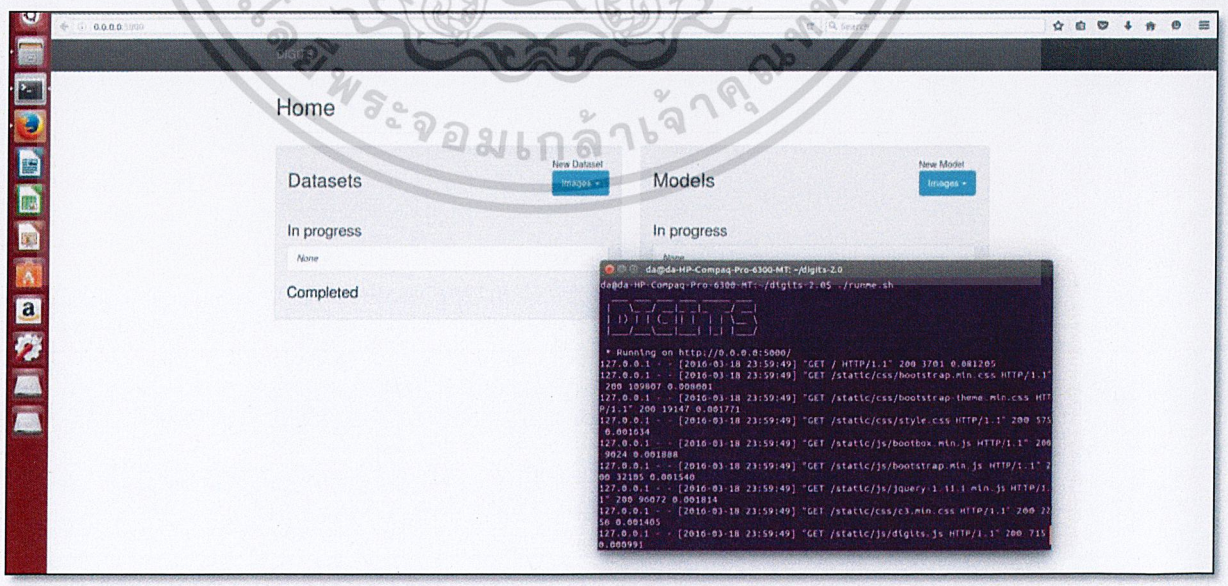
รูปที่ 3.16 การแยกประเภทภาพ

เริ่มต้นการ train ภาพและ การสร้างโมเดลจากฐานข้อมูล

การใช้งาน NVIDIA DIGITS แบ่งการทำงานออกเป็น 2 ส่วน ดังนี้

1) **การสร้าง Dataset**

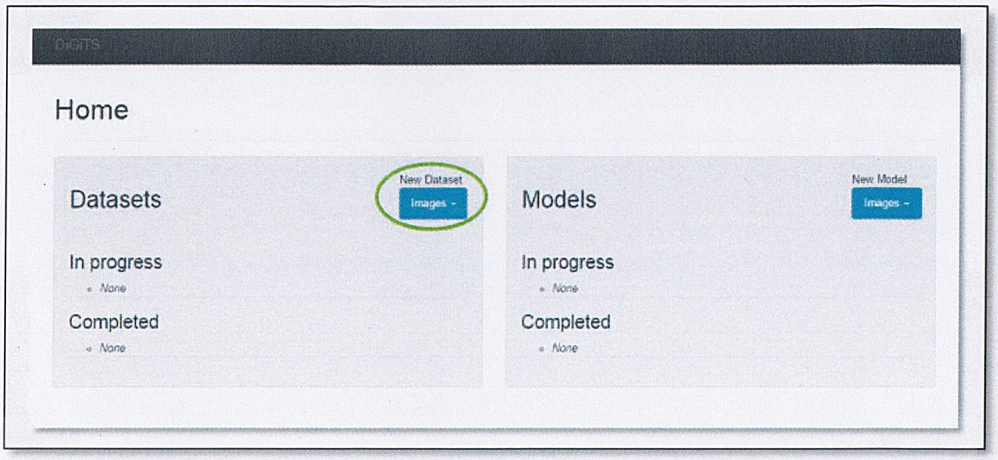
เป็นการนำฐานข้อมูลมาสร้าง Dataset ที่ได้จัดไว้มาทำการฝึก เมื่อได้ทำการติดตั้ง Caffe และ Nvidia digits แล้ว จะเรียกใช้ชุดฐานข้อมูลในการ train ภาพ โดยใช้พอร์ต 5000 ดังรูปที่ 3.17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.17 เข้าพอร์ต Dataset ของ NVIDIA DIGITS

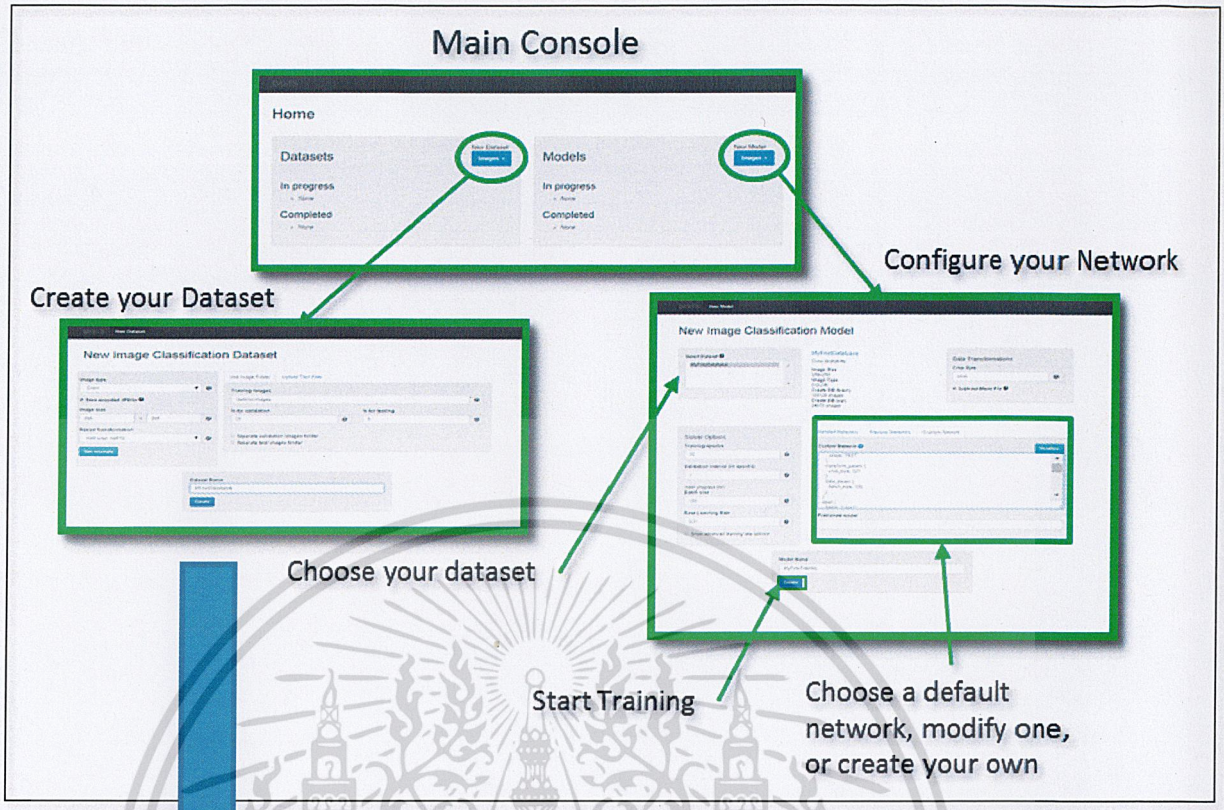
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



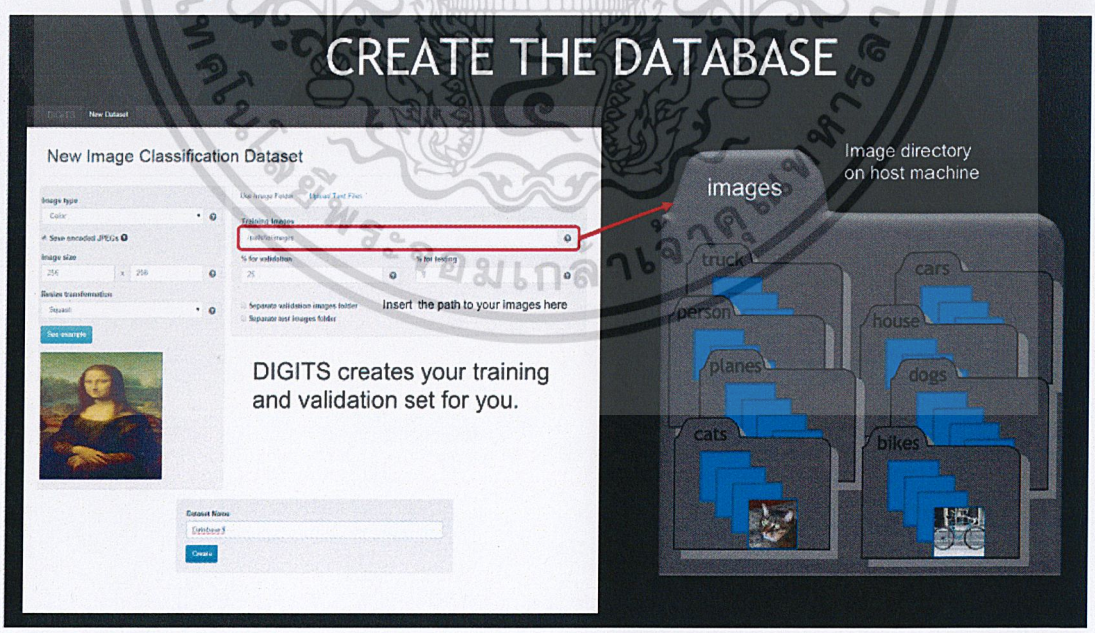
รูปที่ 3.18 หน้าแรกของ Dataset

จากรูปที่ 3.18 แสดงหน้าแรกของ NVIDIA DIGITS และมีส่วนประกอบ ดังรูปที่ 3.19 และ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 ส่วนต่างๆของหน้าแรกของ DIGITS

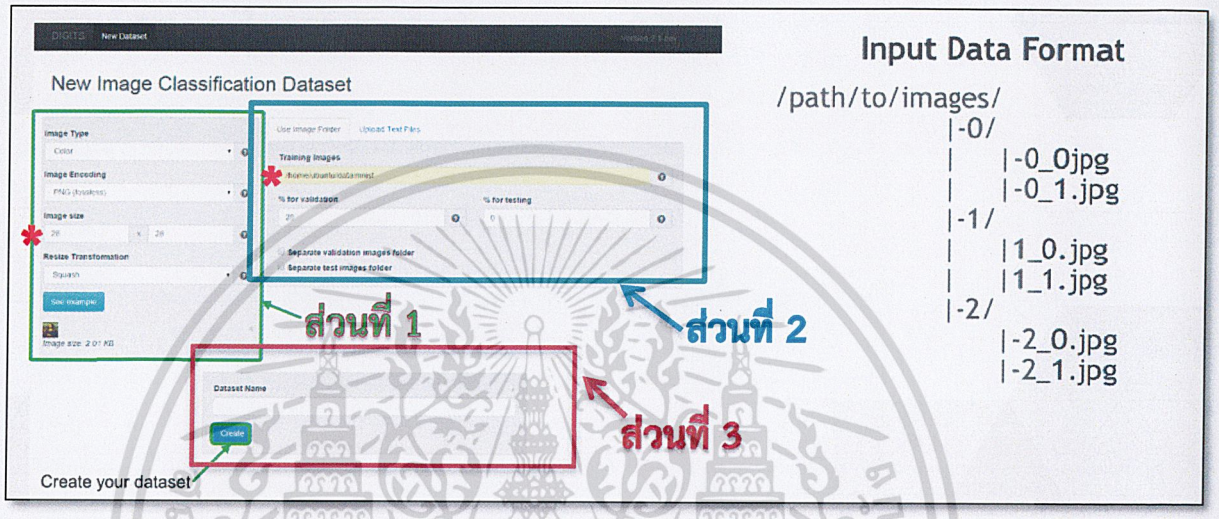


รูปที่ 3.20 การนำเข้าฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.20 แสดงหน้าแรกของ dataset จะแบ่งออกเป็น 2 ส่วน คือ ส่วน Create Dataset คือส่วนที่จะนำฐานข้อมูลมาสร้างโมเดลโดยการเข้าถึง Path ของฐานข้อมูล ดังรูปที่ 3.20 และส่วนที่ 2 คือการสร้างโมเดลในการ train โดยนำ Dataset จากส่วนที่ 1 มาใช้

การกำหนดค่าต่างๆเพื่อใช้ในการสร้าง Dataset



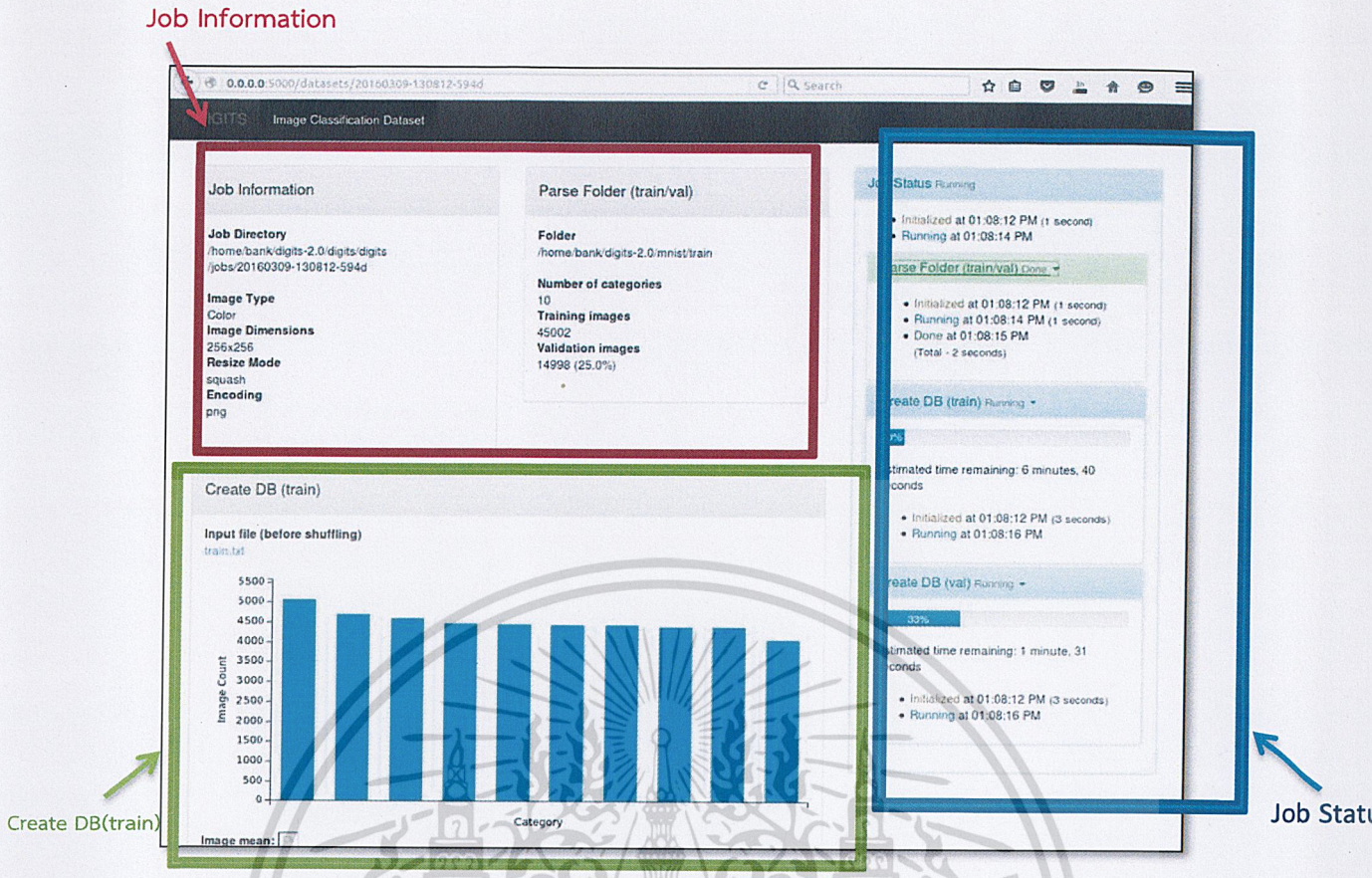
รูปที่ 3.21 การกำหนดค่าต่างๆ

จากรูปที่ 3.21 แสดงการกำหนดค่าต่างๆในการสร้าง Dataset มี 3 ส่วนหลักๆ ดังนี้

1. **ส่วนที่ 1** คือ พารามิเตอร์ที่ต้องการนำเข้ามา Train ใน Dataset
    1. **Image Type** (ประเภทรูปภาพ เช่น สี หรือ ขาวดำ)
    2. **Image Encoding** (ไฟล์ภาพ)
    3. **Image Size** (ขนาดรูปภาพของ Dataset)
    4. **Resize Transformation** (ขนาดที่ต้องการปรับ)
2. **ส่วนที่ 2** คือ ส่วนที่รูปภาพหรือฐานข้อมูลถูกเก็บไว้
    1. **Training Images** (โฟล์เดอร์ที่เก็บรูปภาพ)
3. **ส่วนที่ 3** คือ การตั้งชื่อให้ Dataset และการ Create Dataset เพื่อทำการ train

เมื่อทำการกำหนดค่าต่างๆเรียบร้อยแล้วจะแสดงหน้าการ Train รูปภาพ ดังรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



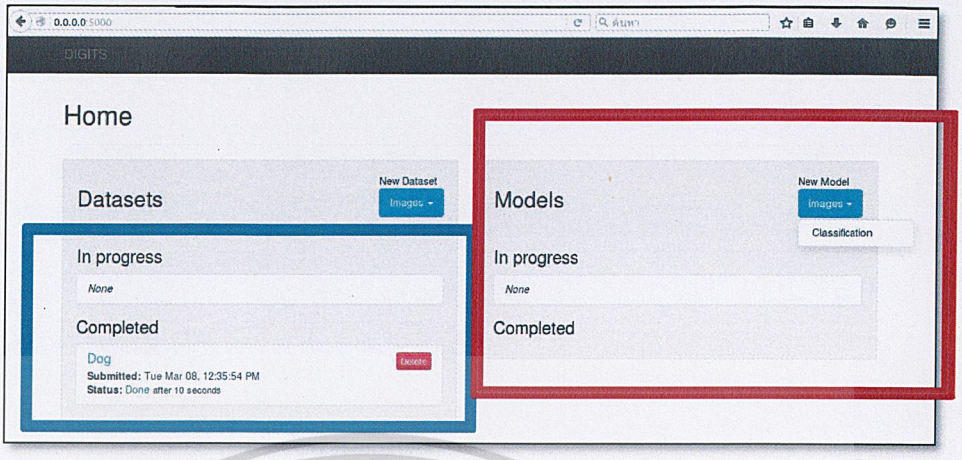
รูปที่ 3.22 หน้าการ Train และสถานะการทำงานต่างๆ

จากรูปที่ 3.22 แสดงหน้าการ Train ภาพ มีส่วนต่างๆดังนี้

1) **Job Information** คือ อธิบายข้อมูลต่างๆของฐานข้อมูลที่ได้ทำการสร้าง Dataset ไว้
2) **Job Status** คือ แสดงสภานะการทำงานและหารประมวลผลของ Dataset
3) **Create DB (Train)** คือ ขั้นตอนการสร้าง Database หรือการ train จะแสดงผลออกมาเป็นกราฟแท่งแสดงจำนวน Dataset ที่ได้ทำการอัพโหลดเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อประมวลผลเสร็จแล้วจะได้ Dataset ที่พร้อมใช้งาน ดังรูปที่ 3.23



รูปที่ 3.23 หน้าแรกเมื่อทำการสร้างเสร็จพร้อมใช้งาน

จากรูปที่ 3.23 เมื่อกลับมาหน้าแรกจะแสดงหน้าแรกของ Dataset ซึ่งแบ่งเป็น 2 ส่วน ดังนี้

ส่วนที่ 1 คือ ส่วน Dataset จะแสดงสถานะของ Dataset ต่างๆ

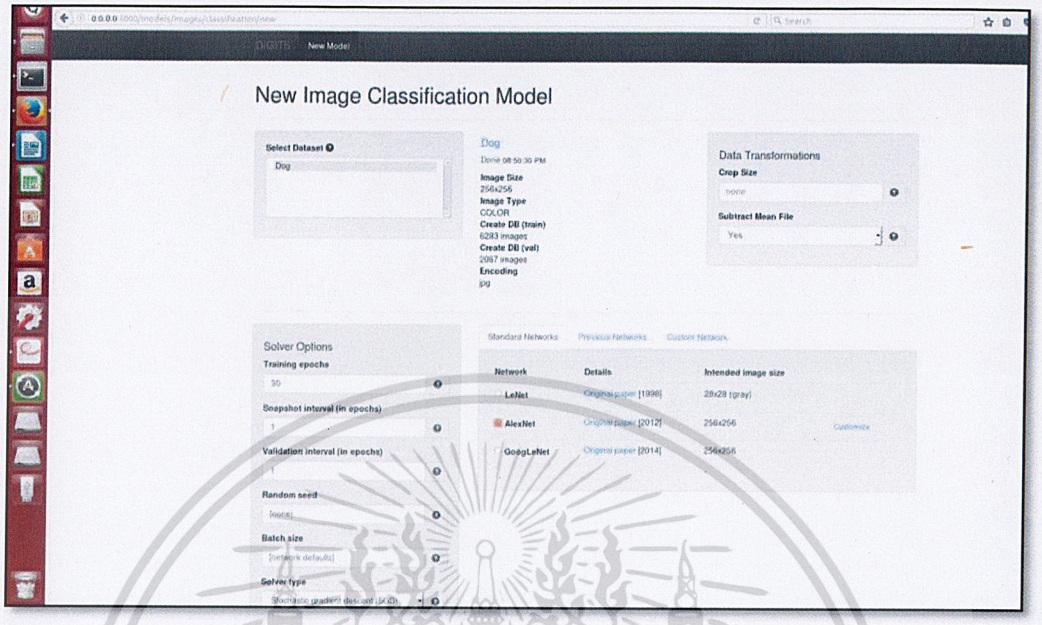
1) **In progress** คือ ส่วนที่กำลังประมวลผล จะแสดงสถานะว่า Running หรือ None
2) **Completed** คือ ส่วนที่ทำการประมวลผลเสร็จแล้วและพร้อมใช้งาน จะแสดงสถานะว่า Done

ส่วนที่ 2 คือ ส่วนการสร้างโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

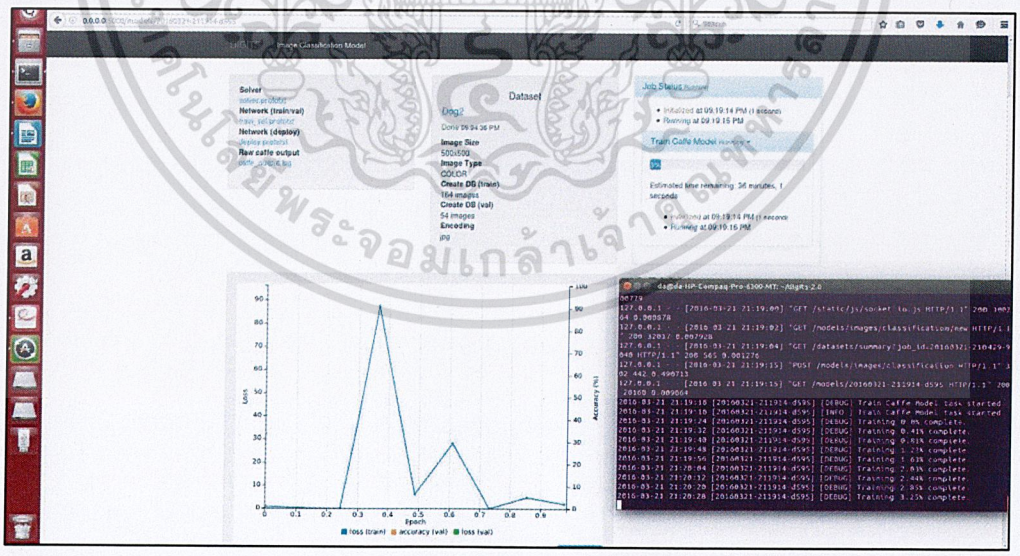
2) การสร้างโมเดลในการฝึก (Train) รูปภาพ

คือการนำรูปภาพจาก Dataset ที่สร้างไว้มาทำการสร้างโมเดล ดังรูปที่ 3.24



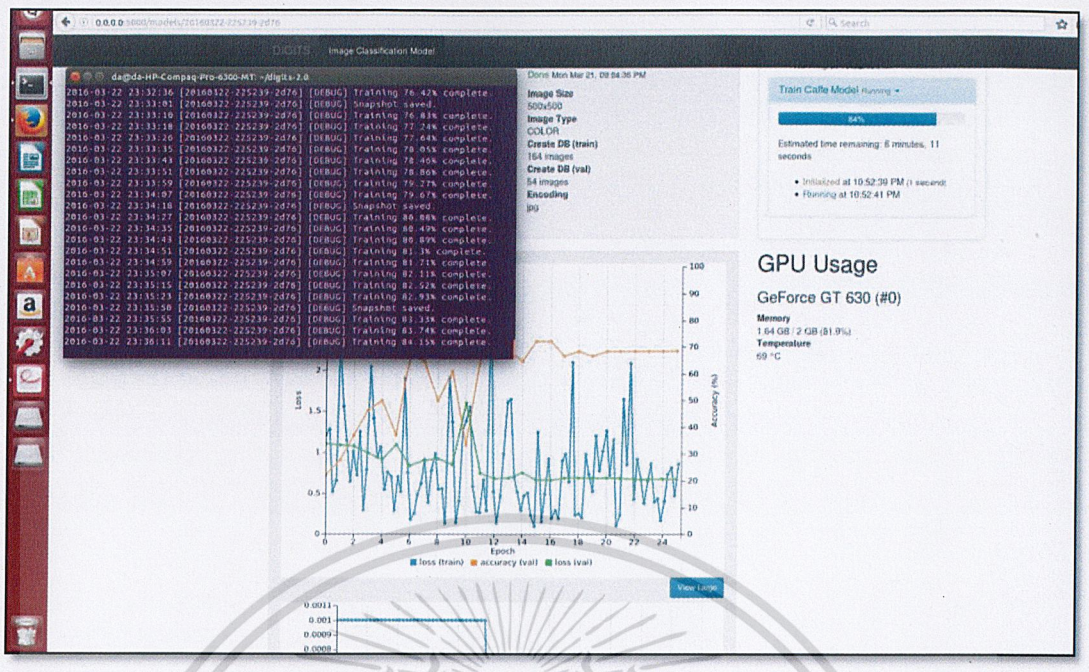
รูปที่ 3.24 สร้างโมเดล

ในขั้นตอนการสร้างโมเดล จะเลือกในส่วน Models โดยกดปุ่ม Images จะแสดงหน้าต่างดังรูป 3.25 และ 3.26



รูปที่ 3.25 หน้าการสร้างโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

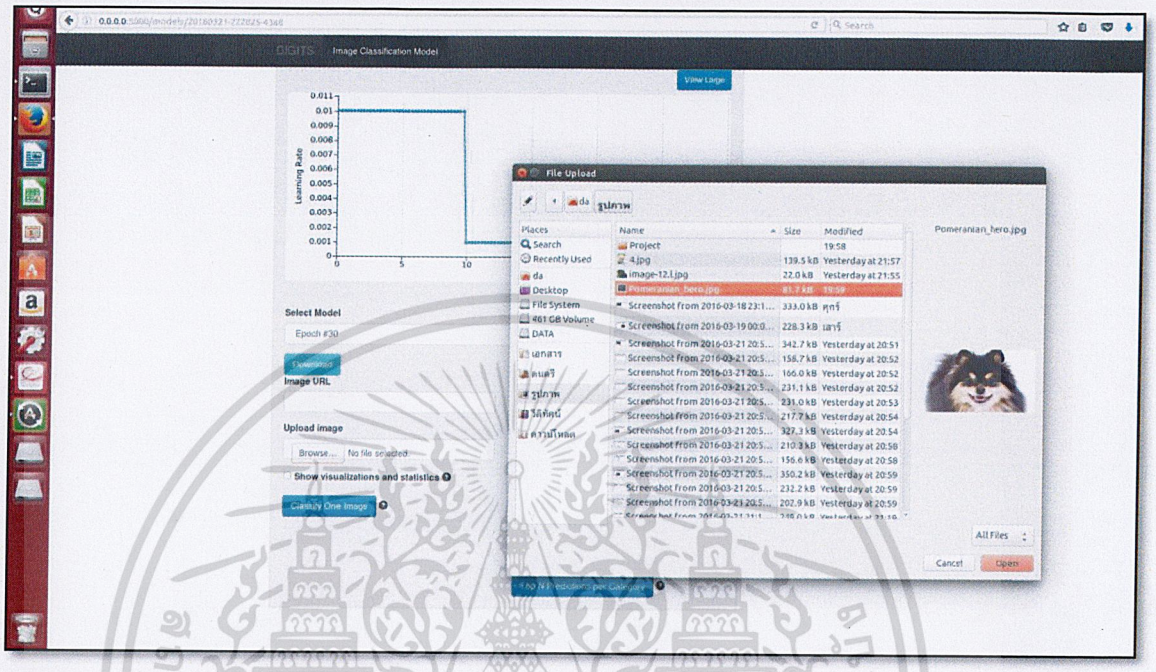


รูปที่ 3.26 หน้าการสร้างโมเดล

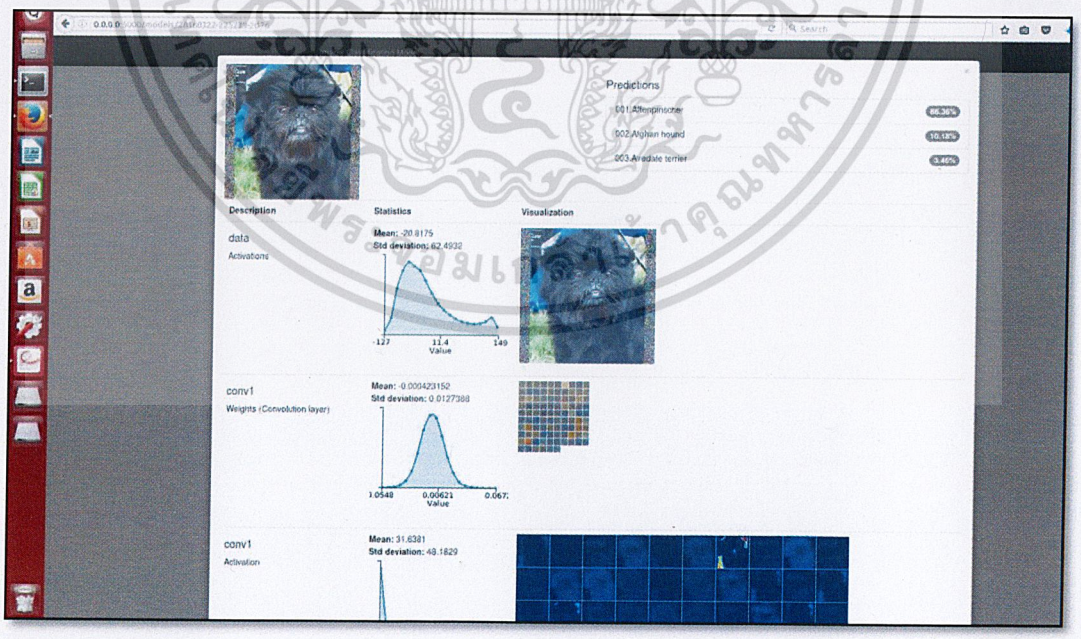
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ขั้นตอนการทดสอบ

หลังจากได้ทำการสร้างโมเดลเสร็จก็ทำการทดสอบรูปภาพจากฐานข้อมูลและโมเดลที่ได้สร้างไว้ ดังรูปที่ 3.27 และแสดงผลลัพธ์ดังรูปที่ 3.28 และ 3.29

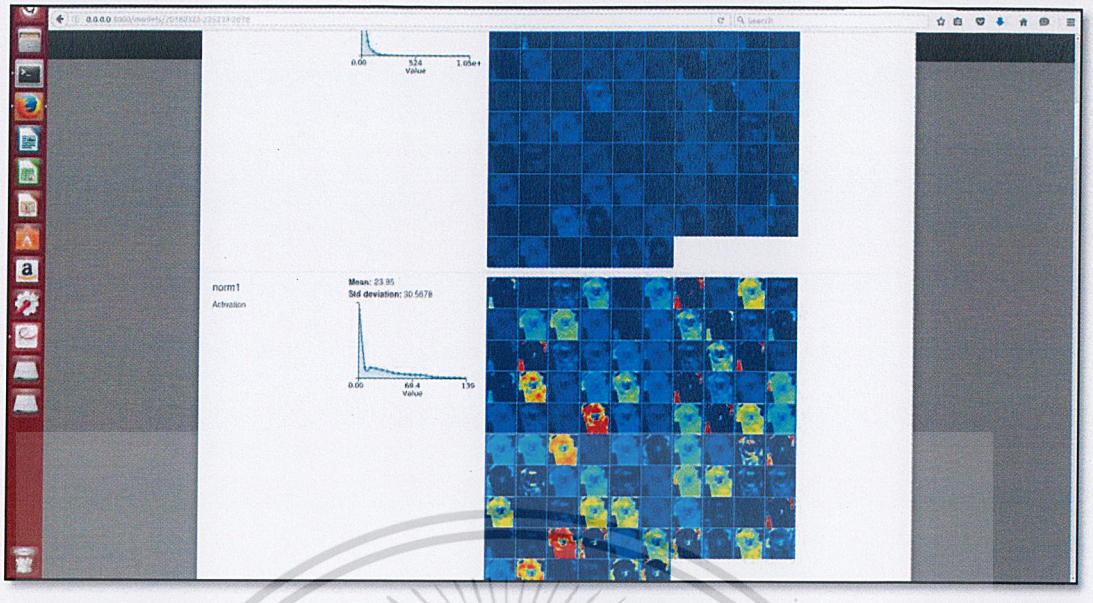


รูปที่ 3.27 อัพโหลดรูปภาพที่ต้องการทดสอบ



รูปที่ 3.28 ผลการแยกสายพันธุ์ โดยบอกเปอร์เซ็นต์และค่าสถิติต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 ผลค่าสถิติต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

การจำแนกพันธุ์สุนัขโดยใช้วิธีการ Deep learning จะใช้ฐานข้อมูลภาพพันธุ์สุนัขจากงานวิจัย Dog Breed Classification โดยวิธี Part Localization ของ Jiongxin Liu และคณะ โดยนำฐานข้อมูลนั้นมาทำการสร้างโมเดลและ train รูปภาพโดยใช้ซอฟแวร์ของ NVIDIA DIGITS และ Framework ของ Caffe ซึ่งผลศึกษาและทำการทดสอบเป็นประโยชน์ของผู้ที่อยากทราบพันธุ์สุนัขและยังเป็นการศึกษาการใช้ซอฟต์แวร์ของ Nvidia Digits อีกด้วย นอกจากนี้ยังสามารถนำวิธีการ Deep learning ไปใช้ประโยชน์ในด้านต่างๆได้อีกด้วย โดยการดำเนินการแบ่งได้เป็น 3 ส่วน คือ

4.1 ภาพรวมส่วนของการสร้างโมเดลและ Train รูปภาพ โดยใช้ NVIDIA DIGITS

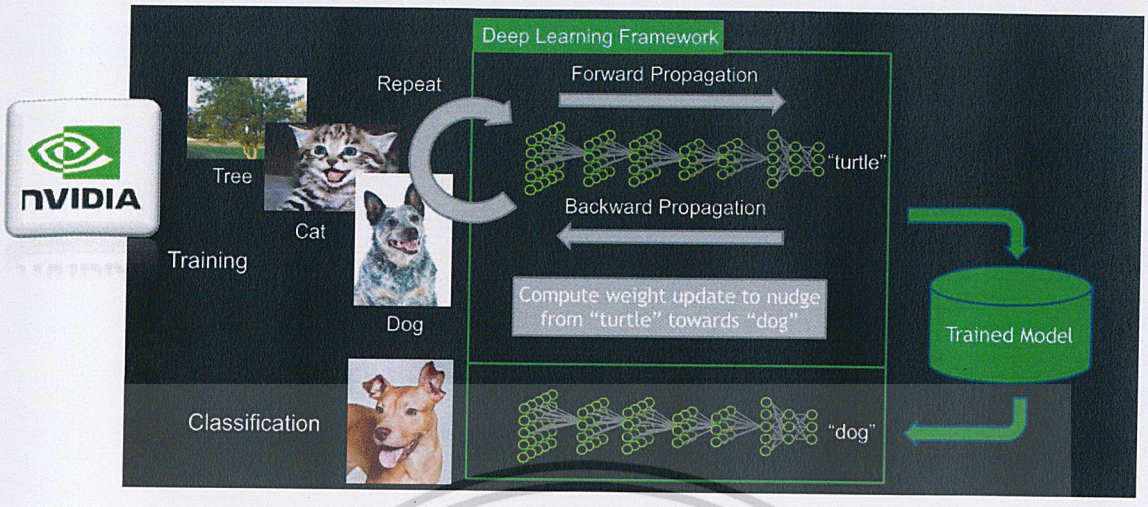
4.1.1 การสร้างฐานข้อมูล โมเดล และการตั้งค่าเพื่อใช้ในการ Train

4.2 ส่วนของผลการดำเนินงาน

4.3 สรุปผลการทดสอบและการวัดประสิทธิภาพโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ภาพรวมส่วนของการสร้างโมเดลและ Train รูปภาพ โดยใช้ NVIDIA DIGTIS



รูปที่ 4.1 ภาพรวมการทำงานของ Caffe และ Digits

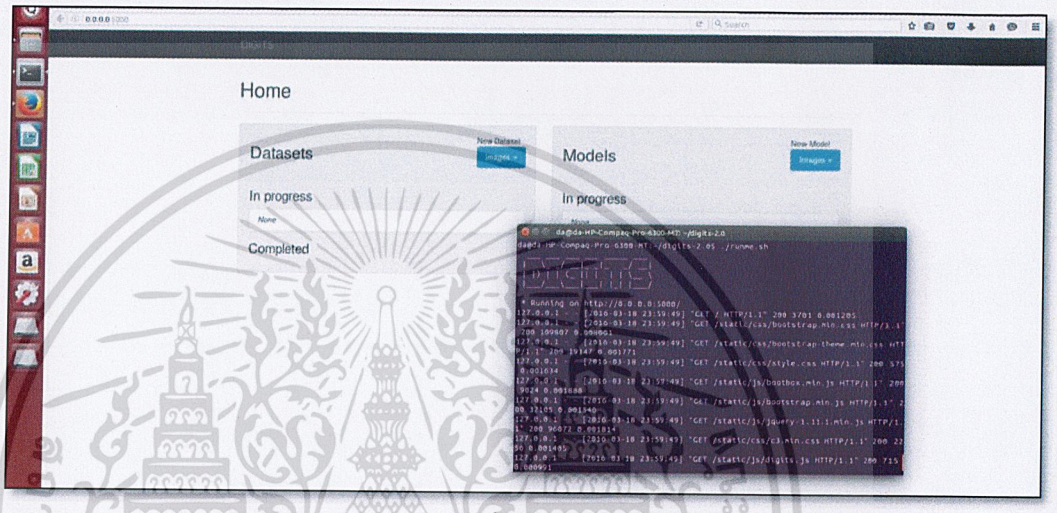
จากรูปที่ 4.1 แสดงให้เห็นการทำงานในภาพรวมทั้งหมด ซึ่ง Digits คือ Dataset ที่ใช้ในการสร้างโมเดลในการ Train รูปภาพ โดยทำงานรวมกับ Framework Caffe เมื่อมีสร้างฐานข้อมูลรูปภาพเข้าไปใน Dataset ของ DIGTIS จะมีการนำไปประมวลผลใน Caffe และแสดงผลการจำแนกประเภทออกมาผ่านทาง Interface ของ DIGITS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

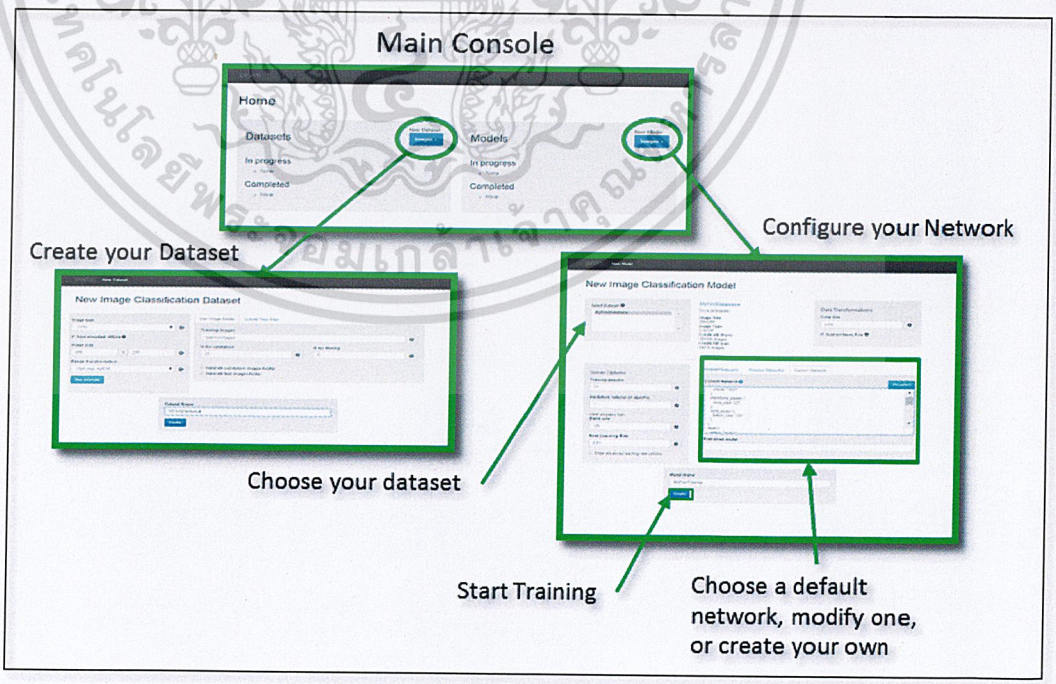
4.1.1 การสร้างฐานข้อมูล โมเดล และการตั้งค่าเพื่อใช้ในการ Train

1) เริ่มต้นสร้าง Dataset

เมื่อทำการเตรียมฐานข้อมูลเป็นที่เรียบร้อยแล้ว จากรูปที่ 3.3 -3.6 ก็จะเข้าสู่หน้าแรกในการสร้าง Dataset ใน Nvidia Digits เลือกปุ่ม New Datasets แล้วเลือก Image เลือก Create datasets ดังรูปที่ 4.2-4.3



รูปที่ 4.2 หน้าแรกการเข้า server ของ Nvidia Digits

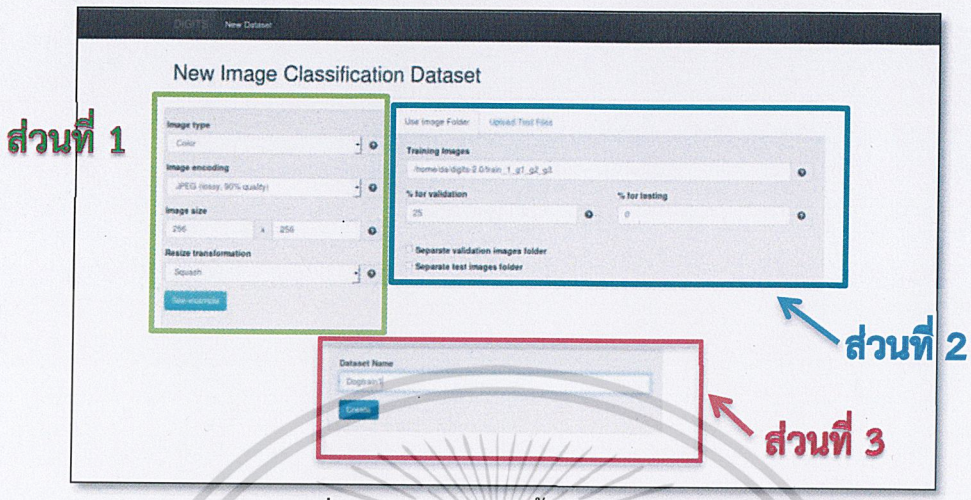


รูปที่ 4.3 Nvidia Digits แบ่งเป็น 2 ส่วนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าต่างๆเพื่อใช้ในการสร้าง Dataset

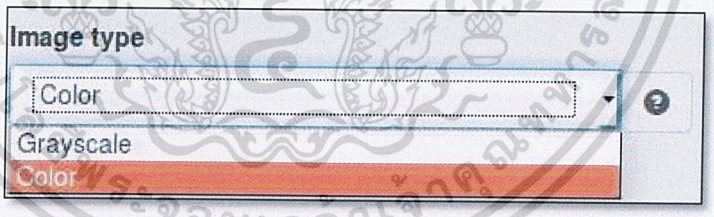
จากรูปที่ 4.4 เป็นหน้าการสร้าง Datasets นี้ จะมีด้วยกัน 3 ส่วน ดังนี้



รูปที่ 4.4 หน้าของการตั้งค่าต่างๆในการสร้าง Datasets

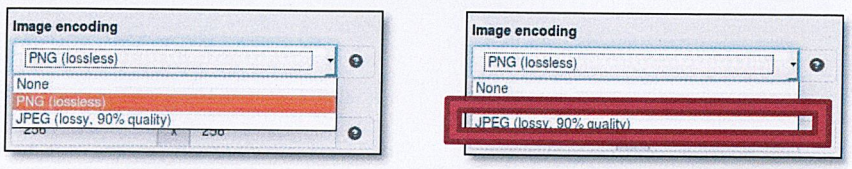
1. **ส่วนที่** 1 คือ ส่วนประเภทของรูปภาพของ Datasets ที่เราจะทำการสร้างและกำหนดค่าพารามิเตอร์ต่างๆ ดังนี้

   1.1 Image Type (ประเภทรูปภาพ เช่น สี หรือ ขาวดำ) จะกำหนดเป็นประเภทสี ดังรูปที่ 4.5



รูปที่ 4.5 การตั้งค่า Image Type

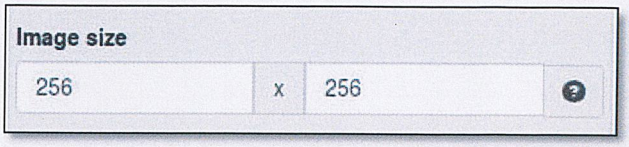
1.2 Image Encoding (ไฟล์ภาพ) ประเภทไฟล์ภาพจะมีให้เลือก ดังนี้ คือ None PNG และ JPEG จะกำหนดเป็น JPEG ดังรูปที่ 4.6



รูปที่ 4.6 เลือก Image Encoding ของภาพ

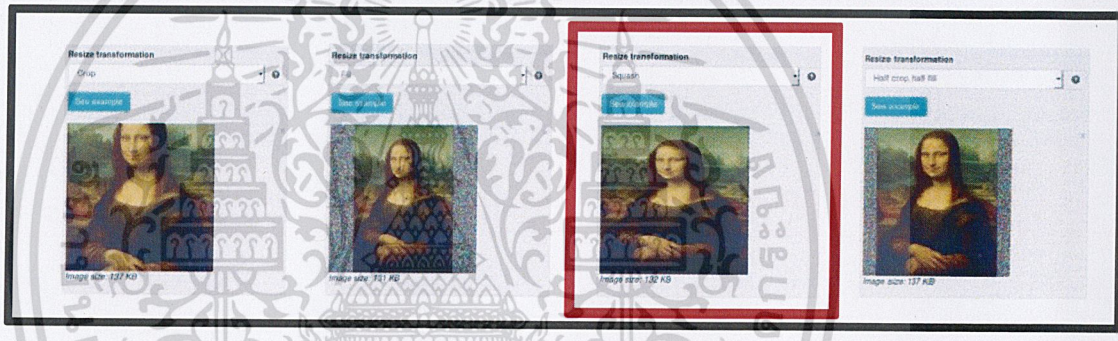
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 **Image Size** (ขนาดรูปภาพของ Dataset) ในส่วนนี้จะกำหนดตามขนาดของรูปภาพใน Datasets แต่ในที่นี้จะกำหนดขนาดมาตรฐาน คือ 256 * 256 ดังรูปที่ 4.7



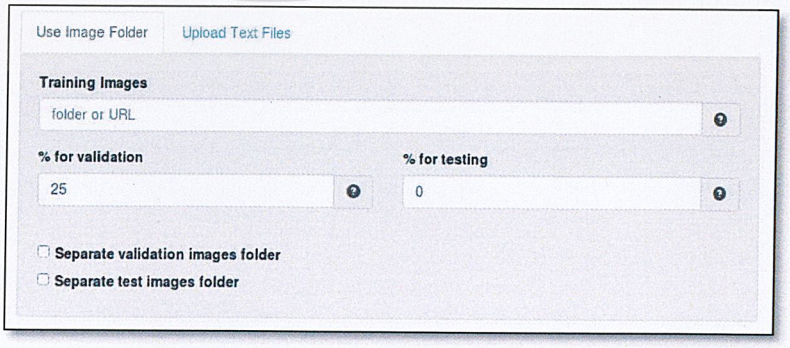
รูปที่ 4.7 การเลือกขนาดของรูปภาพ

1.4 **Resize Transformation** (ขนาดที่ต้องการปรับ) จะมีทั้งหมด 4 แบบด้วยกัน คือ Crop Fill Squash Hill crop,Half fill ดังตัวอย่างรูปที่ 4.8 และเลือกขนาดที่ต้องการปรับเป็น Squash



รูปที่ 4.8 เลือกขนาดรูปภาพที่ต้องการปรับ

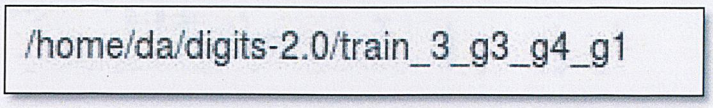
2. **ส่วนที่ 2** คือ ส่วนที่ฐานข้อมูลถูกเก็บไว้ ดังรูปที่ 4.9 กำหนดค่าต่างๆ ดังรูปที่ 4.10-4.11



รูปที่ 4.9 ส่วนของฐานข้อมูลที่ถูกเก็บไว้

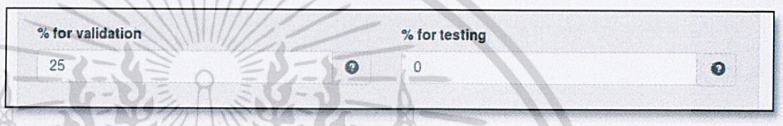
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 Training Image คือ ส่วนที่อยู่ของไฟล์ฐานข้อมูลที่ได้เก็บไว้ หรือ URL ของไฟล์ฐานข้อมูล ในที่นี้จะกำหนดดังตัวอย่างรูปที่ 4.10



รูปที่ 4.10 ตัวอย่างการเรียก Path ของที่เก็บฐานข้อมูล

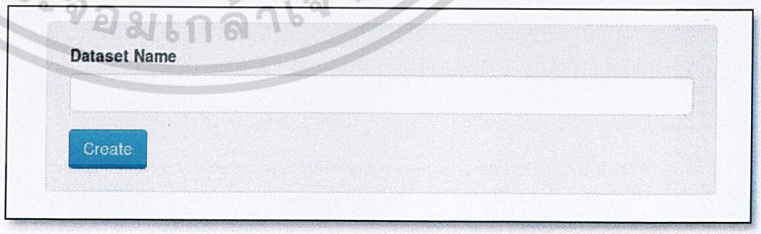
2.2 % for Validation คือ ค่าเปอร์เซ็นต์ที่จะนำไปเทียบกับ dataset เพื่อที่จะนำบางส่วนของรูปภาพในชุดข้อมูล dataset มาทำการหาความถูกต้อง และจะทำการแสดงในส่วนชุดข้อมูล varlidation โดยจะแสดงผลออกมาผ่านทางกราฟ ในกราฟจะเป็นการระบุค่าเป็น loss(var) (ค่าสูญเสียความถูกต้อง) จะกำหนดเป็นค่า Default คือ 25 ดังรูปที่ 4.11



รูปที่ 4.11 ตั้งค่าต่างๆในส่วนการ train

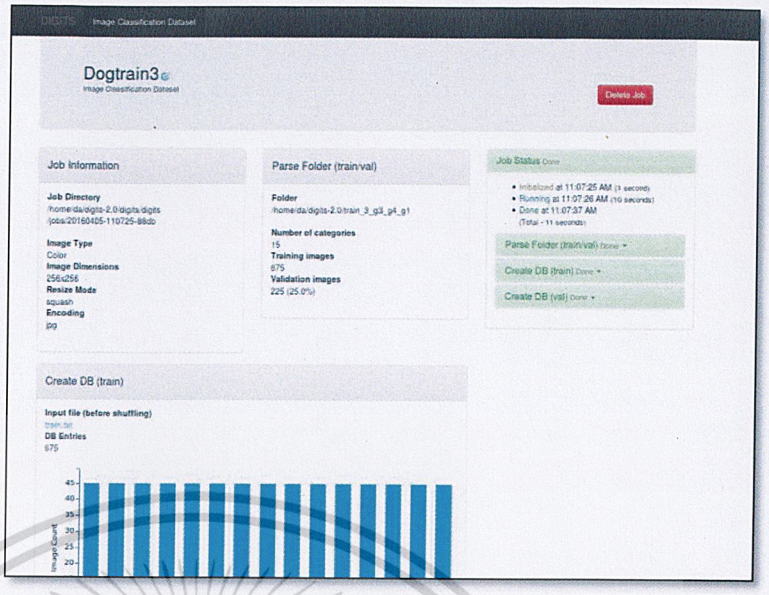
2.3 % for testing คือ ค่าเปอร์เซ็นต์การตรวจสอบของการทดสอบรูปภาพ จะกำหนดเป็นค่า Default คือ 0 ดังรูปที่ 4.11

3. **ส่วนที่ 3** คือ การตั้งชื่อให้ Dataset ดังรูปที่ 4.12 และการ Create Dataset เพื่อทำการ train เมื่อทำการกำหนดค่าต่างๆเรียบร้อยแล้ว จะแสดงหน้าการ train ดังตัวอย่างรูปที่ 4.13



รูปที่ 4.12 ตั้งชื่อให้ Datasets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

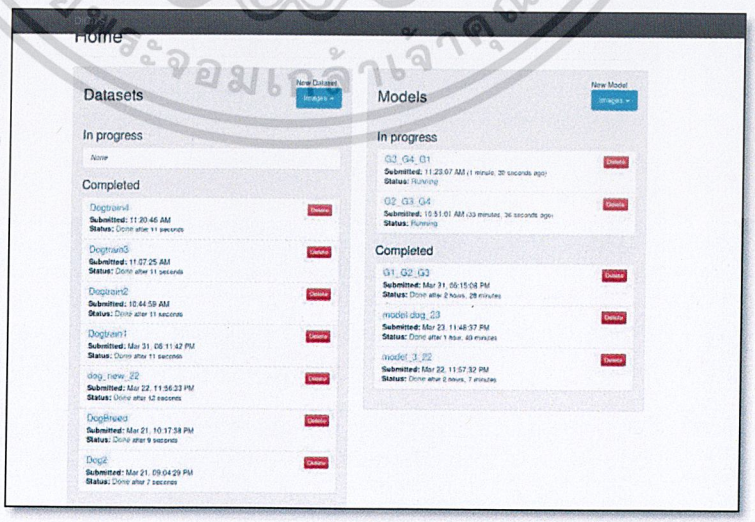


รูปที่ 4.13 หน้าการสร้าง Datasets และการแสดงสถานะการทำงาน

จากรูปที่ 4.13 แสดงหน้าการสร้าง Datasets มีส่วนต่างๆดังนี้

1) **Job Information** คือ อธิบายข้อมูลต่างๆของฐานข้อมูลที่ได้ทำการสร้าง Dataset ไว้
2) **Job Status** คือ แสดงสภานะการทำงานและหารประมวลผลของ Dataset
3) **Create DB (Train)** คือ ขั้นตอนการสร้าง Database หรือการ train จะแสดงผลออกมาเป็นกราฟแท่งแสดงจำนวน Dataset ที่ได้ทำการอัพโหลดเข้าไป

และเมื่อทำการสร้าง Dataset เสร็จแล้ว ก็จะแสดงผล ดังรูปที่ 4.14 จะแสดง Dataset ที่เคยสร้างไว้ทั้งหมด

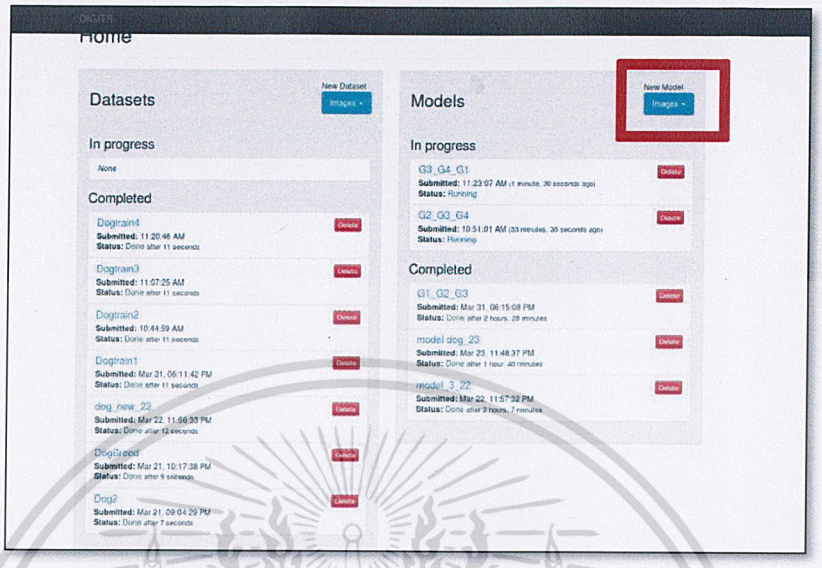


รูปที่ 4.14 Datasets และโมเดล ที่เคยสร้างไว้

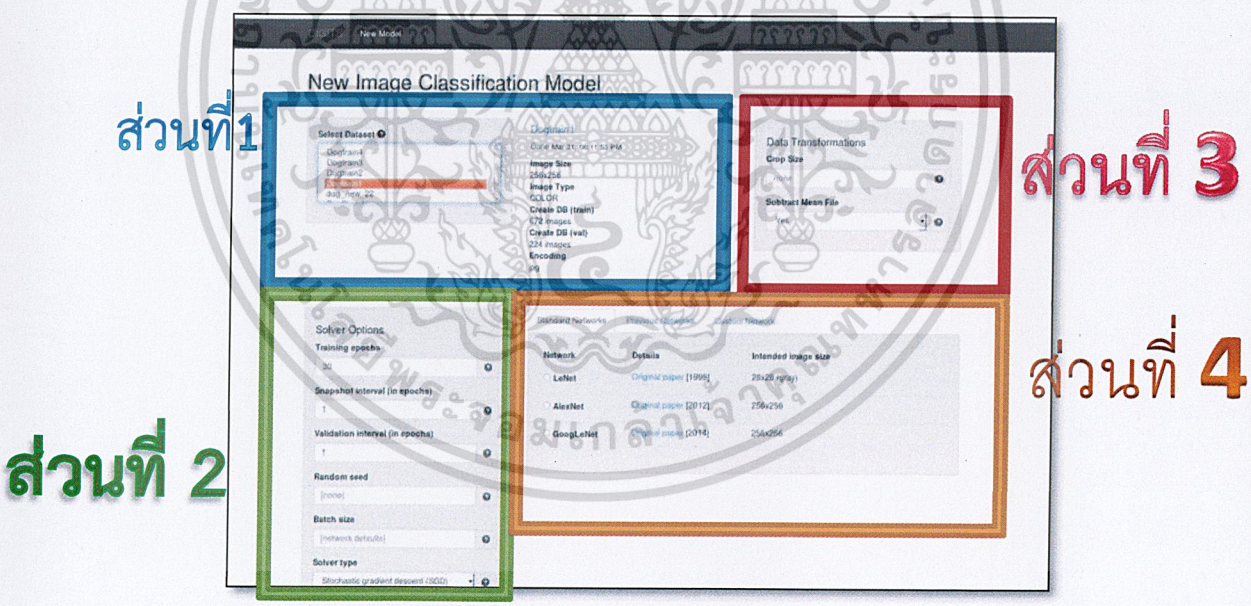
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การสร้างโมเดลในการ Train

นำ Dataset ที่ได้สร้างไว้มาทำการสร้างโมเดล ดังรูปที่ 4.15 โดยเลือกปุ่ม New Image และจะแสดงหน้าการสร้างโมเดลดังรูปที่ 4.16 และมีการกำหนดค่าต่างๆ ดังนี้



รูปที่ 4.15 เลือกปุ่ม New Image ในการสร้างโมเดล

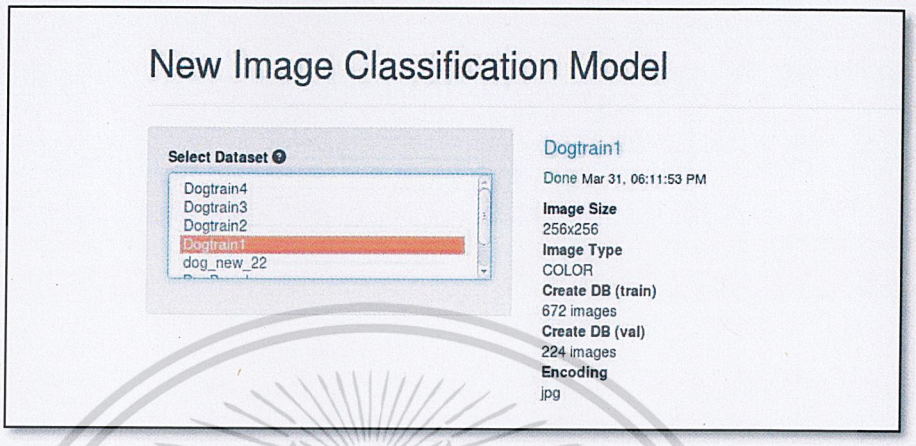


รูปที่ 4.16 หน้าการตั้งค่าต่างๆ

จากรูปที่ 4.16 แสดงหน้าแรกการสร้างโมเดลในการ train รูปภาพ โดยมีส่วนประกอบต่างๆ และจะมีการกำหนดค่าต่างๆ ดังนี้

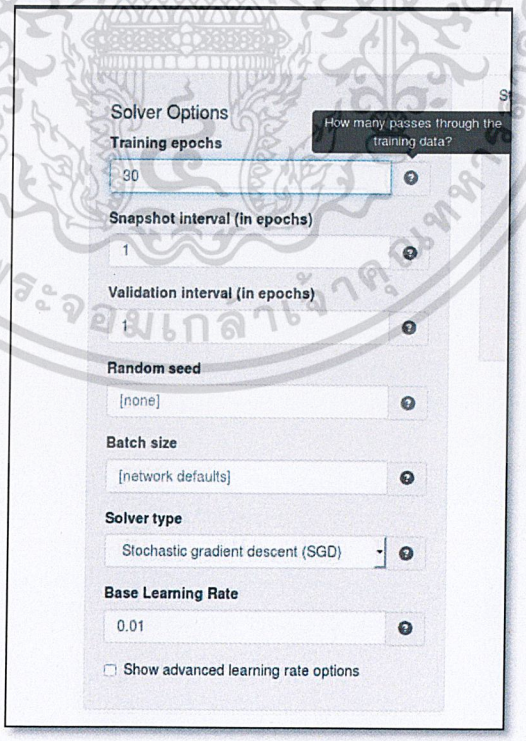
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **ส่วนที่1** คือ เป็นการเลือกชุดข้อมูลที่เราได้ทำการสร้างไว้เพื่อที่จะนำชุดข้อมูลมาทำการทดสอบต่อไป โดยส่วนที่ 1 ยังบอกถึงรายละเอียดของชุดข้อมูลที่เราได้ตั้งค่าไว้ก่อนหน้านี้ ดังตัวอย่างรูปที่ 4.17 และจะบอกถึงการตั้งค่า Datasets ที่ได้สร้างไว้



รูปที่ 4.17 การเลือก Dataset เพื่อมาสร้างโมเดล

2. **ส่วนที่ 2** เป็นการแสดงค่าพารามิเตอร์ต่างๆที่จะใช้ในการทดสอบโดยที่ผู้ใช้จะต้องเป็นคนกำหนดค่าต่างๆเหล่านั้นเองโดยบางค่าจะเป็นค่าที่ระบบกำหนดมาไว้ให้เรียบร้อยแล้ว ดังรูปที่ 4.18 และมีรายละเอียด ดังนี้



รูปที่ 4.18 ส่วนการกำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 Training Epoch คือ ค่าที่บอกถึงช่วงของการเรียนรู้และการ Training ค่า Epoch ซึ่งช่วง Epoch ที่ใช้จะมีค่า เท่ากับ 30 Epoch ยังขึ้นอยู่กับขนาด Batch Size อีกด้วย

2.2 Snapshot Interval คือ เป็นการบันทึกในส่วนของการฝึกฝน จะตั้งตาม Default คือ 1

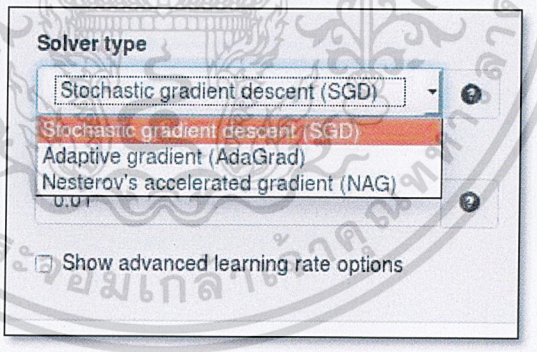
2.3 Validation Interval(In Epochs) คือ ค่าที่ไว้ตรวจสอบความถูกต้องในแต่ละรอบการเรียนรู้ จะตั้งตาม Default คือ 1

2.4 **Random Seed** คือ ตัวเลขสุ่มสถานะเริ่มต้น จะตั้งตาม Default คือ None

2.5 **Batch Size** คือ จะการเป็นค่าที่กำหนด iteration ซึ่งค่า Batch Size จะขึ้นอยู่กับขนาดของ Memory ของ GPU กับ RAM โดยที่ทั้งสองกระบวนการจะมีการทำงานร่วมกัน โดยค่าของ Batch Size ที่ได้ทำการใช้จะมีค่าเท่ากับ 1 เนื่องจาก CPU ไม่สามารถประมวลผลค่า Batch Size ที่มีจำนวนมากได้

2.6 Solver Type คือ วิธีแก้ปัญหาที่จะให้เลือกใช้ในการทดสอบ โดยวิธีแก้ปัญหาที่นิยมมากสุดจะเป็นวิธีของ SGD แต่ในตัวเลือกจะมี 3 ตัวด้วยกัน ดังรูปที่ 4.19

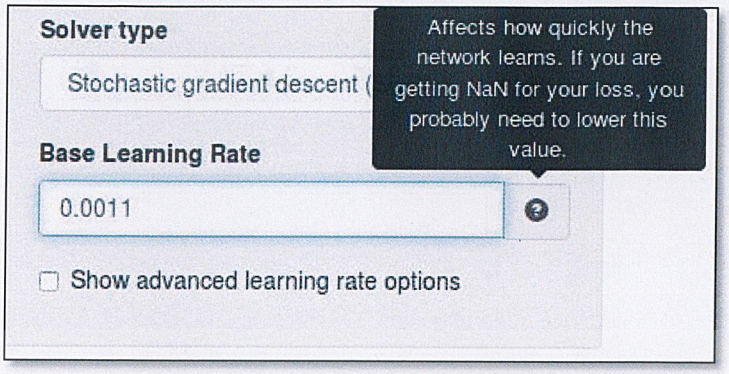
สำหรับ **Stochastic gradient descent** คือ ในแต่ละการคำนวณ gradient (เป็นการหาค่าที่เหมาะสมที่สุดในการแก้ปัญหา) แต่ละครั้งนั้น ต้องใช้ data ทั้งหมดเพื่อมาอัพเดทพารามิเตอร์ใหม่ซึ่งใช้เวลานาน ดังนั้นวิธีนี้จะสุ่ม data เพียงบางส่วนเพื่อใช้อัพเดทเท่านั้น ไม่ได้ใช้ data ทั้งหมด



รูปที่ 4.19 วิธีการแก้ปัญหาโดยจะเลือกเป็น SGD

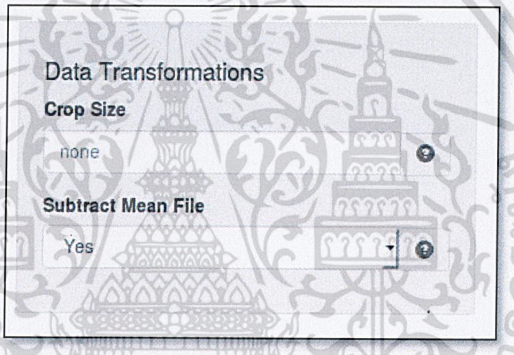
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 **Base Learning Rate** เป็นค่าของอัตราการเรียนรู้ ที่จะใช้ในการทดสอบ ในที่นี้จะใช้อัตราการเรียนรู้เป็น 0.0011 ดังรูปที่ 4.20



รูปที่ 4.20 การตั้งค่าของอัตราการเรียนรู้

3. **ส่วนที่** 3 คือ ส่วนของการเปลี่ยนแปลงข้อมูล ดังรูปที่ 4.21 มี 2 ส่วนดังนี้



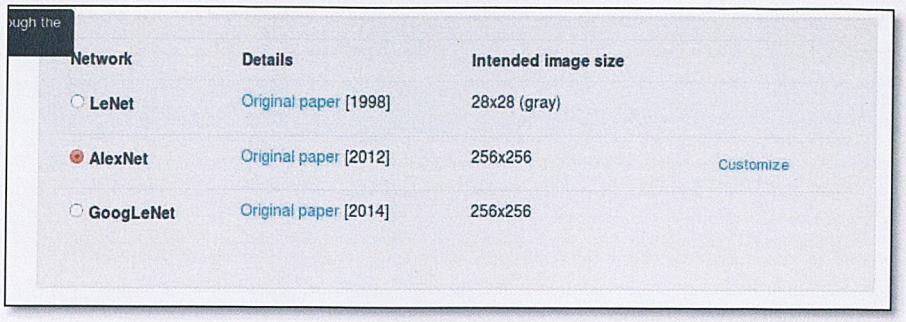
รูปที่ 4.21 ตั้งค่าส่วนของการเปลี่ยนแปลงข้อมูล

3.1 **Crop Size** เป็นการตัดขนาดของข้อมูลในช่วงการทำการทดสอบ จะตั้งตาม Default คือ None

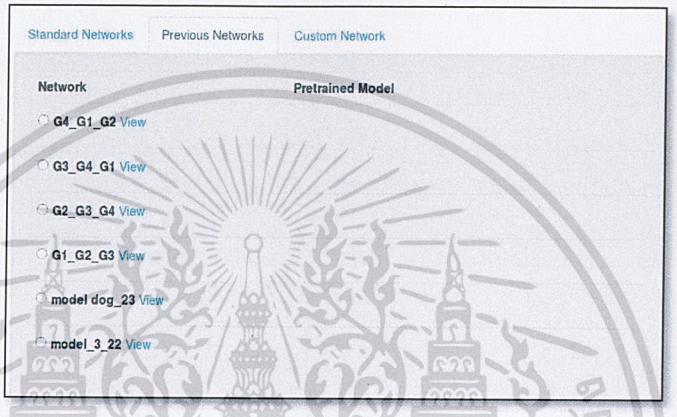
3.2 **Subtract Mean File** เป็นการลบไฟล์ของชุดข้อมูลในแต่ละภาพจะตั้งตาม Default คือ None

4. **ส่วนที่** 4 เป็นการเลือกเครือข่ายที่จะใช้ในการทดสอบ โดยเครือข่ายที่ได้ทำการเลือกใช้จะเป็นเครือข่ายของ Alexnet ซึ่งเป็นเครือข่ายที่มีการใช้กันมากที่สุดและเหมาะสำหรับการใช้งานของ Caffe เพราะถูก Train โดย Caffe ในส่วนที่ 4 เรายังสามารถดาวน์โหลด หรือสร้างเครือข่ายใหม่ขึ้นมาเพื่อใช้ในการทดสอบ และในส่วนนี้ยังสามารถดูการทดสอบที่เราได้ทำสำเร็จไปก่อนหน้านี้ได้อีกด้วย ดังรูปที่ 4.22-4.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 การเลือกเครือข่ายการทดสอบ AlexNet



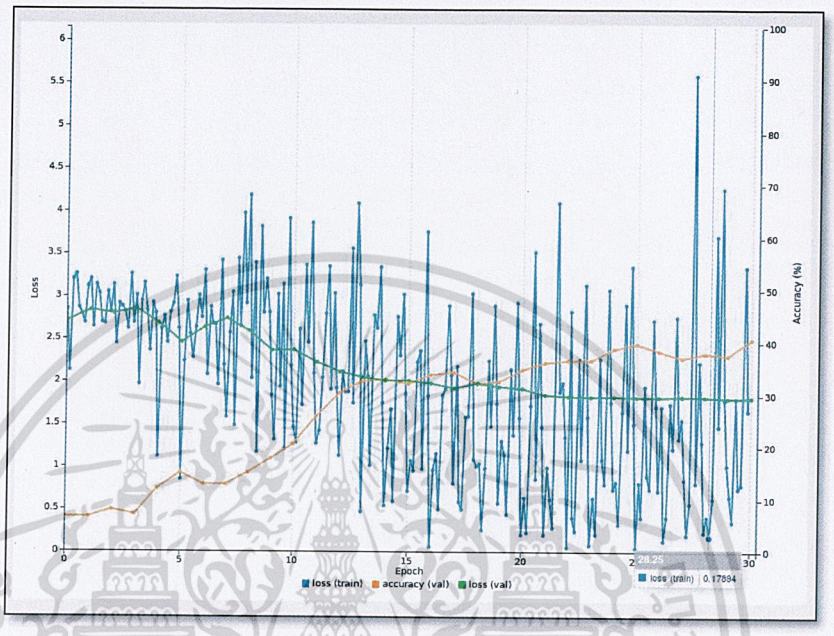
รูปที่ 4.23 โมเดลที่ได้ทำการสร้างไว้ก่อนหน้านี้

รูปที่ 4.24 กำหนดค่าของเครือข่ายเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของกราฟ

เมื่อทำการสร้างโมเดลเสร็จแล้วจะเป็นขั้นตอนการแสดงผลเป็นกราฟที่จะแสดงถึงการทำงานในส่วนของการทดสอบชุดข้อมูล กราฟจะมีการแบ่งออกเป็นสามเส้นกราฟด้วยกัน ในส่วนของกราฟนี้จะแสดงให้เห็นภาพรวมของการ Train ถึงความถูกต้องและความผิดพลาดที่เกิด ดังรูปที่ 4.25



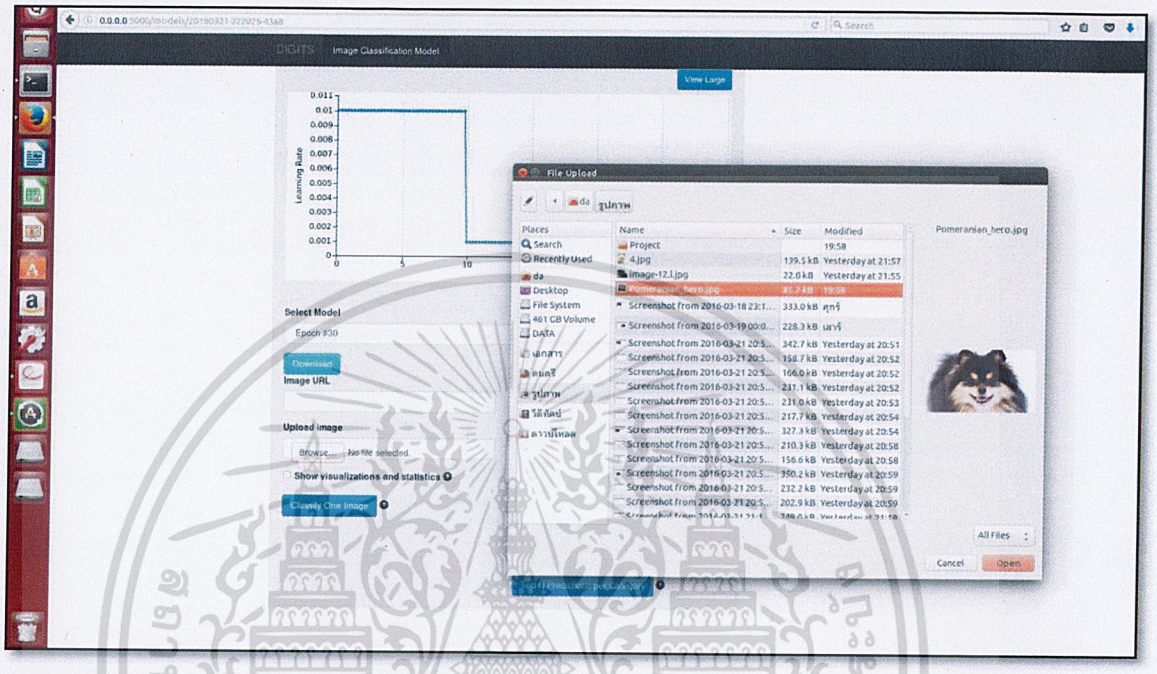
รูปที่ 4.25 กราฟการทำงานในส่วนของการTrain ทดสอบชุดข้อมูล

1. Accuracy คือ ค่าที่แสดงถึงความแม่นยำของการทดสอบชุดข้อมูล
2. Loss(train) คือ ค่าที่มีการสูญเสียในการ train ข้อมูล
3. Loss(Val) คือ ค่าความถูกต้องที่เสียไปในการทดสอบข้อมูล โดยที่ค่านี้จะไม่เกี่ยวข้องกับชุดข้อมูลหลักแต่ค่านี้มีเพื่อ ทดสอบความถูกต้องในส่วนชุดข้อมูล validation

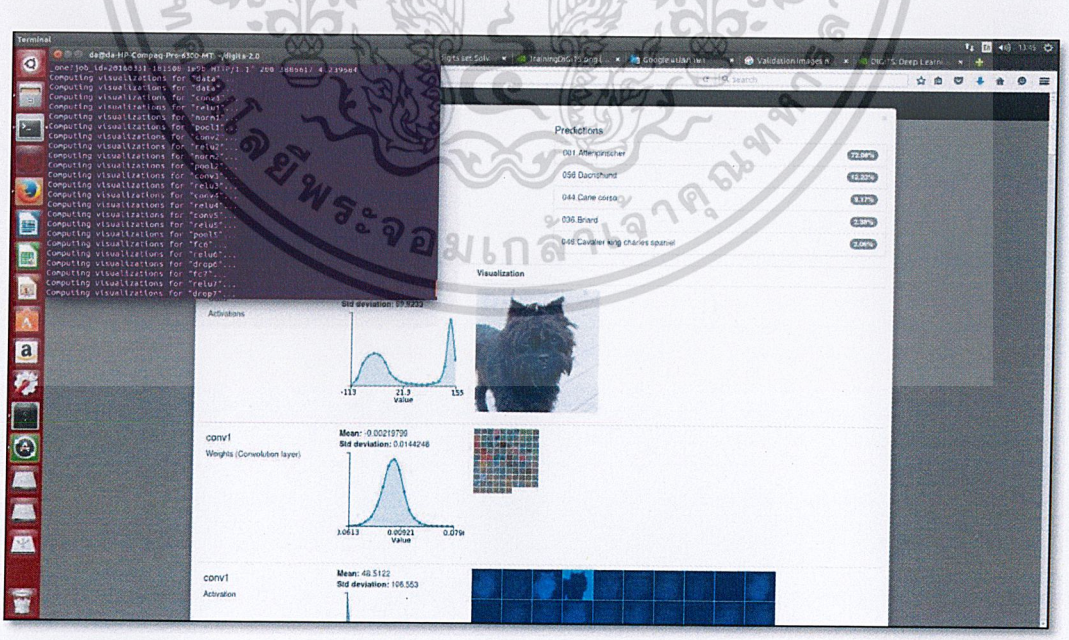
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดสอบ

หลังจากได้ทำการสร้างโมเดลเสร็จก็ทำการทดสอบรูปภาพจากฐานข้อมูลและโมเดลที่ได้สร้างไว้ ดังรูปที่ 4.26 และแสดงผลลัพธ์การทำงานในชั้นต่างๆดังรูปทาง Terminal ที่ 4.27 และ และการบอกขั้นตอนการทำงานในชั้นต่างๆ ดังรูปที่ 4.28



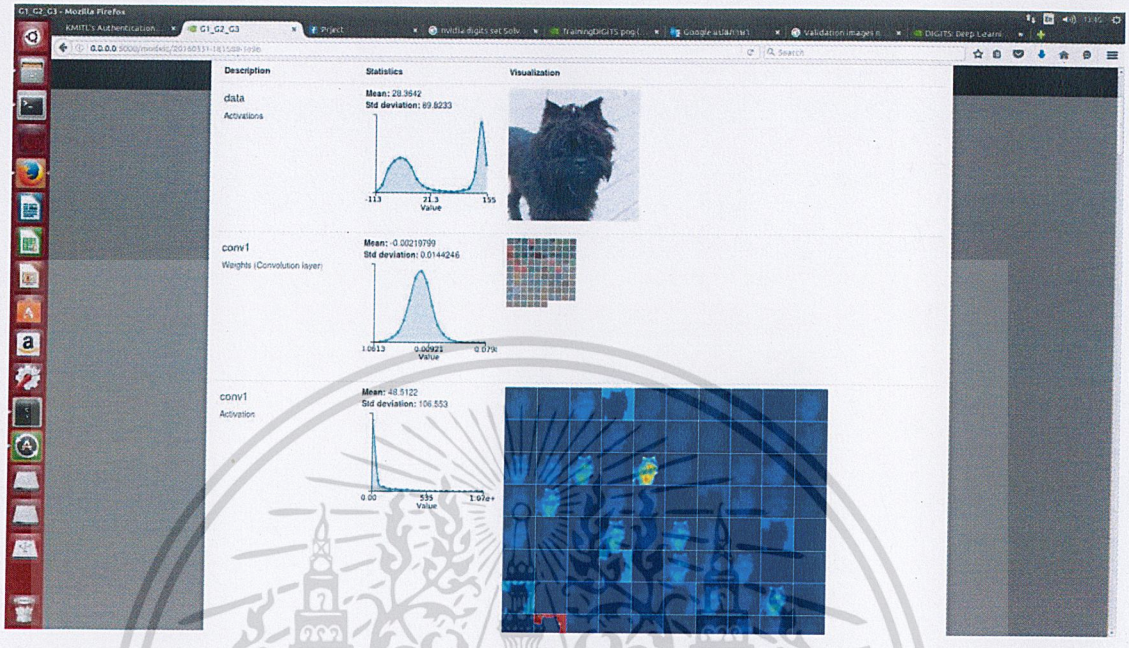
รูปที่ 4.26 อัพโหลดรูปภาพที่ต้องการทดสอบ



รูปที่ 4.27 ผลการแยกสายพันธุ์ โดยบอกเปอร์เซ็นต์และค่าสถิติต่างๆและแสดงชั้นการทำงาน ของ Caffe ใน Trminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.27 ได้ทดสอบเลือกสุนัขมา 1 สายพันธุ์ คือ สายพันธุ์รหัสที่ 001 เมื่อผลการทำงานออกมามีการทำนายเป็นสายพันธุ์ที่ 001 เป็นอันดับที่ 1 มีความแม่นยำคิดเป็น 72.08% ซึ่งมีความแม่นยำเป็นที่น่าพอใจ



รูปที่ 4.28 ผลค่าสถิติต่างๆจนถึงขั้นตอนการทำงานในชั้นต่างๆ

จากรูปที่ 4.28 ในส่วน Nvidia Digits จะแสดงชั้นการทำงานและค่าสถิติต่างๆ ดังนี้

1. **Description** คือ การอธิบายถึงชั้นการทำงาน
2. **Mean** คือ ค่าเฉลี่ยของข้อมูล
3. **std deviation** คือ ค่าเบี่ยงเบนมาตรฐานเป็นการหาค่าการกระจายตัวของข้อมูลที่ออกหากจากค่าเฉลี่ยกลางของข้อมูล ค่ายิ่งมากแสดงว่ามีการแปรปรวนหรือการกระจายของข้อมูลสูง
4. **Visualization** คือ แสดงภาพในการทำงานในชั้นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการดำเนินงาน

หลังจากได้ทำการสร้างฐานข้อมูลและสร้างโมเดลเป็นที่เรียบร้อย ได้มีการทดสอบความแม่นยำโดยวิธีการทำ Cross validation ในงานวิจัยนี้ใช้การแบ่งข้อมูลสำหรับการทดสอบเป็น 4 ชุดย่อย (fold) ฝึกสอนด้วยชุดข้อมูล 3 ชุด ส่วนที่เหลืออีก 1 ชุดเก็บไว้สำหรับการทดสอบ ทำการทดลองซ้ำ 4 ครั้งแต่เปลี่ยนชุดข้อมูลสำหรับการฝึกสอนและทดสอบใหม่ โดยการวัดประสิทธิภาพของความถูกต้องของข้อมูลในงานวิจัยนี้ วัดได้จากความถูกต้องของการจัดกลุ่มของข้อมูลที่ได้เป็นอันดับ1 สรุปได้ดังตารางที่ 4.1- 4.4

ตารางที่ 4.1 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 1 2 3 และกลุ่มทดสอบที่ 4

รหัสสายพันธุ์	Correct	Incorrect
001	12	8
005	9	11
006	10	10
014	8	12
027	8	12
032	5	15
034	1	19
036	2	18
039	5	15
044	4	16
046	4	16
056	2	18
057	8	12
076	4	16
087	6	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 2 3 4 และกลุ่มทดสอบที่ 1

รหัสสายพันธุ์	Correct	Incorrect
001	9	11
005	7	13
006	14	6
014	11	9
027	7	13
032	6	14
034	2	18
036	6	14
039	2	18
044	8	12
046	3	17
056	4	16
057	7	13
076	2	18
087	1	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 3 4 1 และกลุ่มทดสอบที่ 2

รหัสสายพันธุ์	Correct	Incorrect
001	16	4
005	7	13
006	11	9
014	8	12
027	8	12
032	4	16
034	5	15
036	6	14
039	11	9
044	6	14
046	5	15
056	3	17
057	3	17
076	4	16
087	4	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลการทดสอบของการฝึกโมเดลกลุ่มที่ 4 1 2 และกลุ่มทดสอบที่ 3

รหัสสายพันธุ์	Correct	Incorrect
001	15	5
005	3	17
006	6	14
014	10	10
027	12	8
032	8	12
034	9	11
036	1	19
039	4	16
044	5	15
046	2	18
056	6	14
057	8	12
076	1	19
087	7	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปผลการทดสอบ

สามารถสรุปผลการทดสอบทั้งหมดได้ดังตารางที่ 4.5

ตารางที่ 4.5 ผลการทดสอบของโมเดลทั้ง 4 ครั้ง

4-folds cross	Correctly (%)
1	29.66
2	33.66
3	32.33
4	29.33
Average	31.24

จากผลการทดลองจะเห็นได้ว่าวิธีการ Deep Learning ที่ใช้ Nvidia Digits นั้น ให้ความแม่นยำในการทำนายพันธุ์สุนัขได้ถูกต้องคิดเป็น 31.24% ซึ่งเป็นที่น่าพอใจ เนื่องจากการเลือกสายพันธุ์สุนัขมาเพียง 15 สายพันธุ์ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ปัญหาพิเศษนี้มีจุดมุ่งหมาย คือ การศึกษาการแยกพันธุ์ของสุนัขโดยใช้อัลกอริทึมเชิงลึกโดยใช้ซอฟแวร์ Nvidia Digits มาเป็นตัวช่วยในการจำแนกพันธุ์ของสุนัข และใช้วิธีการทดสอบโมเดล คือ การทำ Cross Validation ในการทดสอบนี้จะมีวิธีการทำได้โดยนำฐานข้อมูลที่เตรียมไว้ 15 สายพันธุ์สุนัข จำนวนสายพันธ์ละ 80 รูปภาพมาแบ่งเป็นออกเป็น 4 กลุ่ม โดยแต่ละกลุ่มจะประกอบด้วย 15 สายพันธุ์สุนัข และสายพันธุ์ละ 20 รูป และนำมาทำการฝึกฝน (Train) โดยแต่ละครั้งในการทดสอบจะแบ่งชุดข้อมูลเป็นข้อมูลที่ใช้ในการทดสอบความถูกต้อง และชุดข้อมูลที่ใช้ในการเรียนรู้ของเครื่อง โดยนำมาฝึกฝน 3 กลุ่ม และทดสอบ 1 กลุ่ม ทำวนจนครบ 4 กลุ่ม ซึ่งการทำงานของระบบแยกพันธุ์สุนัขโดยวิธีการการเรียนรู้เชิงลึกและซอฟแวร์ Nvidia Digits จะมีการทำงานดังนี้ เริ่มจากการให้ผู้ใช้ทำการสร้างคลังข้อมูลและทำการเลือกโมเดลที่จะใช้ในการฝึกฝนข้อมูล หลังจากนั้นผู้ใช้ก็จะทำการอัพโหลดรูปภาพสุนัขที่ต้องการทราบสายพันธุ์ลงไป แล้วระบบก็จะนำรูปภาพไปทำการเปรียบเทียบกับ ข้อมูลในคลังข้อมูลที่ผู้ใช้ได้ทำการสร้างไว้ และระบบก็จะแสดงผลลัพธ์ออกมาให้กับผู้ใช้ ซึ่งความสามารถหลักของระบบนี้ คือ การจำแนกพันธุ์ของสุนัขโดยการใช้ภาพและองค์ประกอบต่างๆของสุนัขเป็นตัวแยกสายพันธุ์ ระบบการจำแนกพันธุ์ของสุนัขยังสามารถที่จะบอกสายพันธุ์ที่มีความใกล้เคียงกับภาพที่นำมาอัพโหลดได้อีกด้วย ซึ่งการระบุจำนวนของสายพันธุ์ใกล้เคียง รวมไปถึงค่าต่างๆที่อาจจะส่งผลต่อผลลัพธ์ที่จะออกมาให้กับผู้ใช้ ผู้ใช้จะต้องเป็นคนที่กำหนดถึงค่าต่างๆนั้นเอง และระบบยังมีการคำนวณถึงค่าทางสถิติต่างๆ ให้ออกมาในรูปแบบของกราฟเพื่อง่ายต่อการศึกษา และ เพื่อให้ง่ายต่อความเข้าใจของผู้ใช้งาน ซึ่งผลการทดสอบของโมเดลโดยรวมนั้นจะมีความแม่นยำในการจำแนกพันธุ์สุนัขคิดเป็น 31.24%

5.2 ข้อจำกัด

จากความสามารถในการทำงานของระบบแยกพันธุ์ของสุนัขที่ได้กล่าวมานั้นยังมีความสามารถในบางส่วนที่ทางผู้พัฒนาเห็นว่าเป็นข้อจำกัดอยู่ดังนี้

1) ระบบจำแนกพันธุ์ของสุนัข ยังไม่สามารถแสดงผลลัพธ์ออกมา ได้อย่างถูกต้องเท่าที่ต้องการ เนื่องจาก เครื่องคอมพิวเตอร์ที่จะใช้ในการทดสอบระบบ จะต้องมีการ์ดจอของทาง NVDIA และ มีคุณสมบัติของเครื่องคอมพิวเตอร์ที่มีความสูงพอสมควร จึงจะได้ผลลัพธ์ที่แม่นยำมากขึ้น
2) เนื่องจากระบบที่ใช้ในการแยกพันธุ์สุนัข จะต้องมีการฝึกฝนรูปภาพในจำนวนที่มากเพื่อให้เครื่องเกิดการฝึกฝนเรียนรู้และจดจำ ก่อนที่จะนำไปทดสอบแยกพันธุ์สุนัข แต่ในฐานข้อมูลที่ทางผู้จัดทำได้ทำการจัดเตรียมไว้ มีสุนัขอยู่ทั้งหมด 15 สายพันธุ์ ซึ่งจะประกอบไปด้วยสายพันธุ์ละ 80 รูป ซึ่งข้อมูลในฐานข้อมูลอาจจะยังไม่เพียงพอที่จะให้เครื่องได้ทำการจดจำ จึงแสดงผลลัพธ์ออกมาได้ไม่เต็มประสิทธิภาพ

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆ... ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สุนัขจำนวน 15 สายพันธุ์ที่ได้นำมาทำการแยกสายพันธุ์นั้น จะมีบางสายพันธุ์ที่มีลักษณะทางกายภาพที่มีความคล้ายคลึงกัน จึงอาจจะส่งผลให้ระบบไม่สามารถทำนายพันธ์ของสุนัขได้ออกมาอย่างถูกต้อง
4) การประมวลผลในส่วนของการทดสอบระบบแยกพันธุ์สุนัขจะขึ้นอยู่กับค่าพารามิเตอร์ในส่วนของค่า Batch Size ที่เราได้ทำการตั้งขึ้น ถ้าเราตั้งค่า Batch Size ให้สูงก็จะส่งผลที่ดีในการทดสอบระบบแยกพันธุ์สุนัข แต่เนื่องด้วยข้อจำกัดของเครื่องที่ใช้ในการทดสอบจึงส่งผลให้ไม่สามารถตั้งค่า Batch Size ให้สูงได้

5.3 ข้อเสนอแนะ

เนื่องด้วยข้อจำกัดต่างๆของระบบ ทางผู้พัฒนามีความคิดเห็นว่า ความสามารถบางอย่างของระบบ สามารถที่จะพัฒนาเพื่อให้ผลลัพธ์ดียิ่งขึ้นกว่าเดิม ดังนี้

1) พัฒนาในส่วนของฐานข้อมูลที่ใช้ในการเก็บภาพของสายพันธุ์สุนัข ให้มีจำนวนสายพันธุ์ที่มากขึ้นกว่าเดิม เพื่อความถูกต้องและความสมบูรณ์ที่มากขึ้น
2) พัฒนาให้สามารถแยกสัตว์สายพันธุ์อื่นๆได้ นอกจากสายพันธุ์ของสุนัข
3) พัฒนาในส่วนของโมเดลที่จะใช้ในการฝึกฝนข้อมูล เพื่อให้ผลลัพธ์ที่ออกมามีความถูกต้องแม่นยำมากกว่าเดิม
4) หากทำการปรับค่าพารามิเตอร์มิเตอร์บางค่าในระบบให้มีความสมบูรณ์มากขึ้น จะส่งผลให้ผลลัพธ์ที่ออกมามีประสิทธิภาพที่มากขึ้น

5.4 ปัญหาและอุปสรรค

1) การลงเฟรมเวิร์ค Caffe deep Learning เพื่อจะใช้เป็นตัวเชื่อมต่อกับเซิร์ฟเวอร์ของ Nvidia Digits จะต้องใช้คอมพิวเตอร์ที่มีการ์ดจอของ NVDIA และต้องมีคุณสมบัติของเครื่องที่สูง แต่ทางผู้พัฒนามีคอมพิวเตอร์ที่มีคุณสมบัติที่ไม่เพียงพอต่อการลง และการ์ดจอที่ใช้นั้นสามารถประมวลผลรูปภาพได้ในปริมาณที่จำกัดทำให้เกิดปัญหาในส่วนของขั้นตอนของการเรียก Server และ ส่วนของการทดสอบข้อมูล
2) ในส่วนของการ ฝึกฝนการเรียนรู้ของโมเดลนั้นจะต้องใช้หน่วยความจำ ของ GPU ที่มากในการประมวลผลเพื่อเพิ่มความถูกต้องในขั้นตอนการประมวลผล ซึ่งคอมพิวเตอร์ที่ใช้ในการประมวลผลของทางผู้จัดทำนั้นมีหน่วยความจำของ GPU ที่ต่ำจึงต้องมีการใช้หน่วยประมวลผลของ CPU เข้ามาช่วยในการประมวลผลผลลัพธ์ที่ได้ออกมาจึงไม่เต็มประสิทธิภาพ
3) การจะทำนายว่าสุนัขที่ได้ทำการอัพโหลดลงไปนั้นเป็นสายพันธุ์อะไรเป็นเรื่องค่อนข้างยากในระดับนึง เนื่องจากลักษณะทางกายภาพของสุนัขนั้นมีลักษณะที่คล้ายคลึงกันมาก รวมไปถึง เครื่องคอมพิวเตอร์ จะต้องมีคุณสมบัติของเครื่องที่เหมาะต่อการนำมาใช้เพื่อให้สามารถปรับค่าพารามิเตอร์ได้ตามที่ระบบต้องการ และระบบจะต้องมีการเรียนรู้และจดจำรูปภาพในจำนวนที่มหาศาล เพื่อให้ผลลัพธ์ที่ออกมามีความแม่นยำมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ธนาวุฒิ ประกอบผล. 2552. **โครงข่ายประสาทเทียม Artificial Neural Networks.** มฉก. วิชาการ, ปีที่ 73, ฉบับที่ 12, 24 มกราคม, หน้า 73-87.

วิกิพีเดีย. 2559. **การเรียนรู้เชิงลึก.** [Online]. Available : http://th.wikipedia.org/wiki/การเรียนเชิงลึก) เข้าถึงเมื่อ 4 กรกฎาคม 2559

Andrew Gibiansky. 2015. **Convolutional Neural Networks.** [Online]. Available : http://andrew.gibiansky.com/blog/machine-learning/convolutional-neuralnetworks/. เข้าถึงเมื่อ 4 กรกฎาคม 2559

Chubu University. 2015. **Deep Learning.** [Online]. Available : http://www.slideshare.net/Takayosi/miru2014-tutorial-deeplearning-37219713. เข้าถึงเมื่อ 4 กรกฎาคม 2559

erogol. 201. **make pycaffe gives following error at Ubuntu 14.04 even I installed numpy.** [Online]. Available : https://github.com/BVLC/caffe/issues/1284.

Gimme. 2557. **Deep Learning เทคโนโลยีเบื้องหลังความฉลาดของ Smartphone.** [Online]. Available : http://droidsans.com/deep-learning-behind-smartphone. เข้าถึงเมื่อ 4 กรกฎาคม 2559

Jiongxin Liu, et al. 2015. **Dog Breed Classification Using PartLocalization.** [Online]. Available : https://www.researchgate.net/publication/262347046_Dog_Breed_Classification_Using_Part_Localization. เข้าถึงเมื่อ 6 กรกฎาคม 2559

lukeyeager. 2015. **Nvidia/digits GettingStarted.** [Online]. Available :. https://github.com/NVIDIA/DIGITS/blob/digits-2.0/docs/GettingStarted.md.เข้าถึงเมื่อ 12 กรกฎาคม 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NVIDIADeveloper. 2015. **NVIDIA Deep Learning Course: Class #3 - Getting started with Caffe [Video file]**. [Online]. Available : https://www.youtube.com/watch?v=rvMVqPsXL10. เข้าถึงเมื่อ 6 กรกฎาคม 2559

SATYA MALLICK. 2016. **NVIDIA DIGITS 3 on EC2**. [Online]. Available : http://www.learnopencv.com/nvidia-digits-3-on-ec2/. เข้าถึงเมื่อ 6 กรกฎาคม 2559

Shouno Hayaru. 2015. **Deep Learning**. [Online]. Available : http://www.slideshare.net/HAL9801/20130925deeplearning-26525042. เข้าถึงเมื่อ 6 กรกฎาคม 2559

Steve Branson et al. 2015. **Bird Species Categorization Using PoseNormalized Deep Convolutional Nets**. [Online]. Available : http://arxiv.org/ เข้าถึงเมื่อ 6 กรกฎาคม 2559

Yangqing Jia, et al.2015. **Caffe: Convolutional Architecture forFast Feature Embedding.**

(n.d.).**2015.Crossvalidation**.[Online].Available: http://202.44.34.144/.../adm.../nccit_files/NCCIT-2011040251.pdf.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

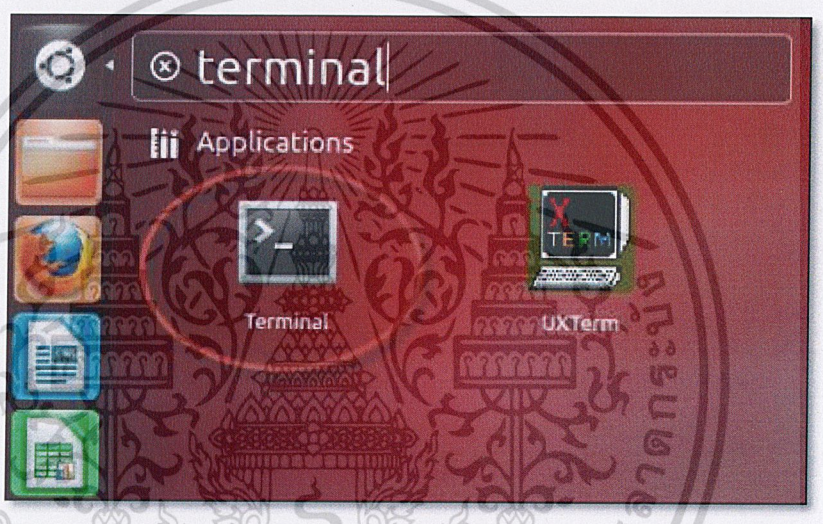
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

วิธีการติดตั้ง Caffe และ Nvidia Digits

ขั้นตอนการลง Caffe และติดตั้งเซิร์ฟเวอร์ของ Nvidia Digits ผู้ใช้งานต้องทำการลงระบบปฏิบัติการ Ubuntu เสียก่อน จากนั้นจะทำการลงส่วนต่างๆได้โดยใช้ Terminal ดังรูปที่ 1 และมีสิ่งที่จำเป็นดังนี้

1) ระบบปฏิบัติการ Ubuntu 14.04 LTS
2) การ์ดจอ Nvidia Gefroce



รูปที่ 1 Terminal ในระบบปฏิบัติการ Ubuntu

1. ทำการอัพเดทระบบ โดยใช้คำสั่ง ดังนี้

```
-sudo apt-get update
-sudo apt-get dist-upgrade
```

2. ติดตั้งเครื่องมืออื่จำเป็น เช่น gcc หรือชุดแปลภาษาโปรแกรม โดยใช้คำสั่งดังนี้

```
-sudo apt-get install -y gcc g++ gfortran build-essential \
  git wget linux-image-generic libopenblas-dev python-dev \
  python-pip python-nose python-numpy python-scipy
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ติดตั้ง CUDA โดยทำการดาวน์โหลดก่อนติดตั้ง ใช้คำสั่งดังนี้

```
-wget
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1404
/x86_64/cuda-repo-ubuntu1404_7.0-28_amd64.deb
```

```
-sudo dpkg -i cuda-repo-ubuntu1404_7.0-28_amd64.deb
-sudo apt-get update
```

3.1 เมื่อดาวน์โหลดเสร็จแล้วทำการติดตั้ง CUDA โดยใช้คำสั่งดังนี้

```
-sudo apt-get -y install cuda
```

4. ตั้งค่าตัวแปลต่างๆ โดยใช้คำสั่งดังนี้

```
-echo -e "\nexport PATH=/usr/local/cuda/bin:$PATH" >> .bashrc
-echo -e "\nexport LD_LIBRARY_PATH=/usr/local/cuda/lib64" >> .bashrc
```

4.1 ทำการ reboot ระบบ

```
-sudo reboot
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการติดตั้งส่วนตัวอย่างและตรวจสอบ GPU โดยใช้คำสั่งดังนี้

```
-cuda-install-samples-7.0.sh ~/
-cd NVIDIA\_CUDA-7.0\_Samples/1\_Utilities/deviceQuery
-make
-./deviceQuery
```

6. ดาวน์โหลด CuDNN และติดตั้ง โดยใช้คำสั่งดังนี้

```
-wget https://s3-eu-west-1.amazonaws.com/christopherbourez/public/cudnn-6.5-linux-x64-v2.tgz
-gzip -d cudnn-6.5-linux-x64-v2.tgz
-tar xf cudnn-6.5-linux-x64-v2.tar
-sudo cp cudnn-6.5-linux-x64-v2/cudnn.h /usr/local/cuda-7.5/include/
-sudo cp cudnn-6.5-linux-x64-v2/libcudnn* /usr/local/cuda-7.5/lib64/
```

7. ติดตั้ง git โดยใช้คำสั่งดังนี้

```
-sudo apt-get -y install git
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ดาวน์โหลดและติดตั้ง Caffe โดยใช้คำสั่งดังนี้

```
-git clone https://github.com/BVLC/caffe.git
-cd caffe
-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 libgflags-dev libgoogle-glog-dev liblmdb-dev
-sudo apt-get install libatlas-base-dev
-sudo apt-get install python-dev
-sudo apt-get install awscli
-cp Makefile.config.example Makefile.config
-vi Makefile.config
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1 จะได้ผลลัพธ์ดังรูปที่ 2 ผู้ใช้งานต้องตรวจสอบว่ามีการเปิดใช้งาน CUDNN หรือไม่ (USE_CUDNN =1) ถ้ายังไม่มีการเปิดใช้ให้เอาเครื่องหมาย '#' ออก แล้วบันทึก

```
USE_CUDNN := 1
CUDA_DIR := /usr/local/cuda
CUDA_ARCH := -gencode arch=compute_20,code=sm_20 \
             -gencode arch=compute_20,code=sm_21 \
             -gencode arch=compute_30,code=sm_30 \
             -gencode arch=compute_35,code=sm_35 \
             -gencode arch=compute_50,code=sm_50 \
             -gencode arch=compute_50,code=compute_50
BLAS := atlas
PYTHON_INCLUDE := /usr/include/python2.7 \
             /usr/lib/python2.7/dist-packages/numpy/core/include \
             /usr/local/lib/python2.7/dist-packages/numpy/core/include
PYTHON_LIB := /usr/lib
WITH_PYTHON_LAYER := 1
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib
BUILD_DIR := build
DISTRIBUTE_DIR := distribute
TEST_GPUID := 0
Q ?= @
```

รูปที่ 2 แสดงคำสั่งที่เปิดของไฟล์ Makefile.config

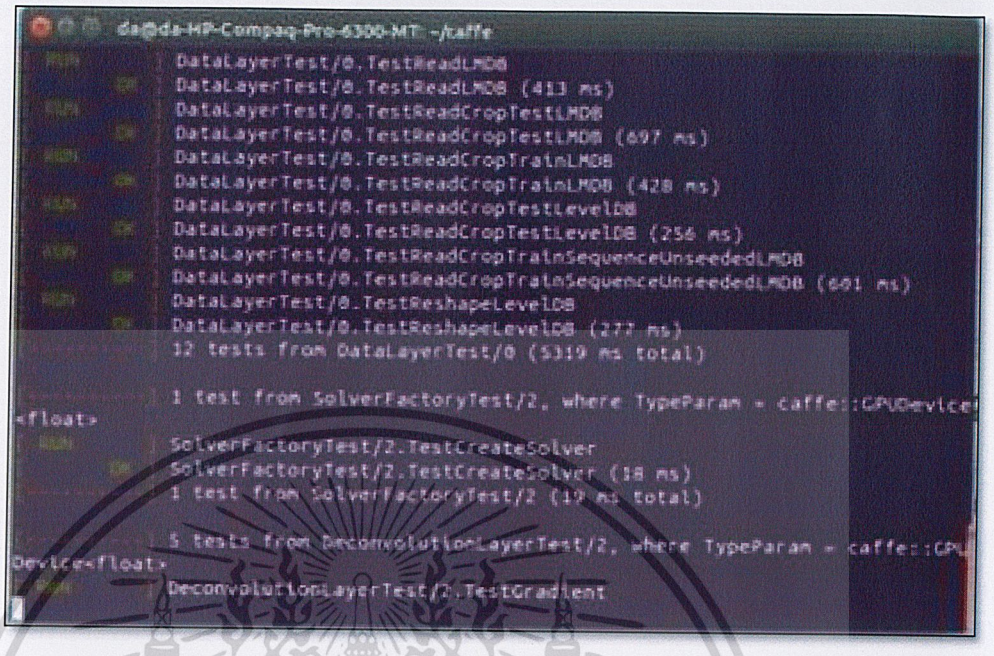
-พิมพ์ ':q' (การออก)

9. เรียกใช้และสร้าง caffe โดยใช้คำสั่งดังนี้

```
-make all -j8
-make test
-make runtest
-sudo ln /dev/null /dev/raw1394
-echo "export
LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH" >>
~/.bashrc
-sudo apt-get install linux-image-extra-$(uname -r)
-sudo reboot
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.1 จะได้ผลลัพธ์ดังรูปที่ 3 ถือว่าถูกต้อง



รูปที่ 3 ผลการทำงานของการใช้คำสั่ง runtest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ดาวน์โหลด เซิร์ฟเวอร์ และติดตั้ง Digits เวอร์ชั่น 2.0 โดยใช้คำสั่งดังนี้

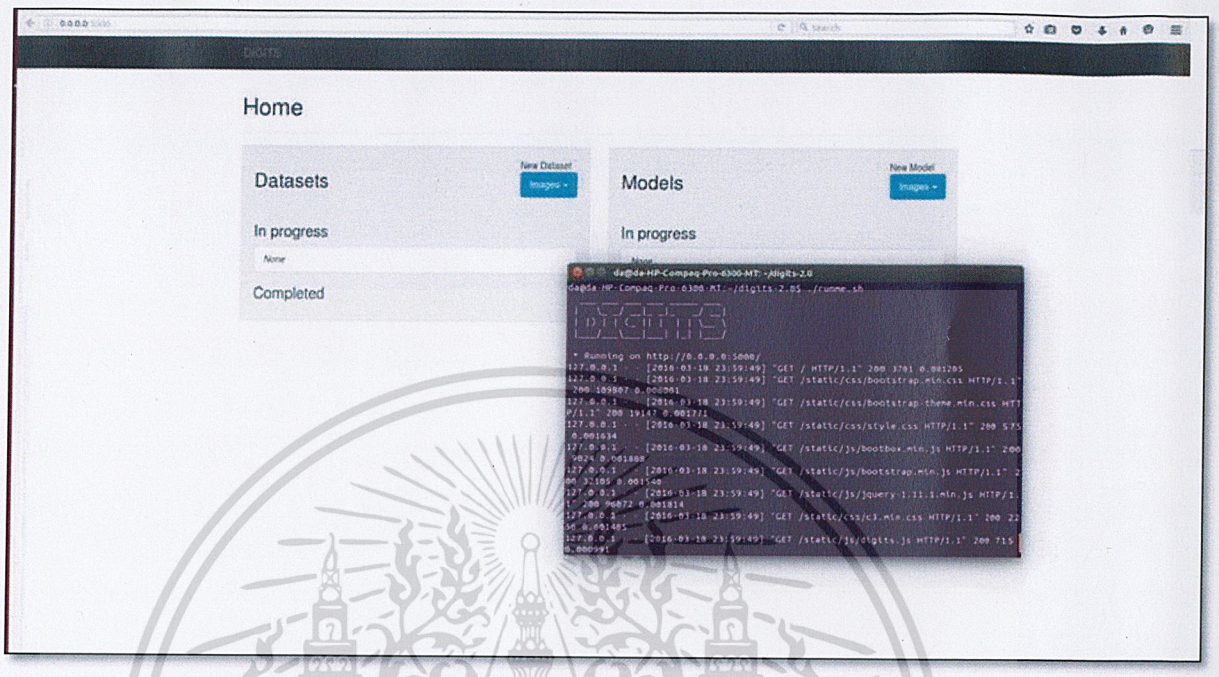
```
-wget https://s3-eu-west-1.amazonaws.com/christopherbourez/public/digits-2.0.0-preview.gz
-tar xvzf digits-2.0.0-preview.gz
-cd digits-2.0/
-./install.sh
-cd caffe
-sudo apt-get -y install --no-install-recommends libboost-all-dev
-make all --jobs=8
-cd ../digits/
-sudo ln /dev/null /dev/raw1394
-sudo pip install -r requirements.txt
-mkdir /home/ubuntu/data/mnist -p
-python tools/download_data/main.py mnist ~/data/mnist
-apt-get install linux-image-extra-$(uname -r)
-echo '[DIGITS]' >> digits/digits.cfg
-echo 'caffe_root = /digits/digits-2.0/caffe' >> digits/digits.cfg
-LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
./digits-server –D
```

11. เริ่มใช้งานเซิร์ฟเวอร์ โดยใช้คำสั่งดังนี้

```
-cd digits-2.0
-./runme.sh
-เข้า port http://0.0.0.0:5000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ผลลัพธ์ดังรูปที่ 4 และพร้อมใช้งาน



รูปที่ 4 การเข้าเซิร์ฟเวอร์ของ Nvidia Digits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้