Robotic Inference

Jiangdong Chen

Abstract—The implementations of deep convolutional neural network for image classification application in NVIDIA's digits are described. Two datasets are used, and both have three categories. The first dataset include photos taken from a Jetson mounted over a conveyorbelt for the purpose of real-time sorting. The second dataset is pictures of aircraft. CNN architecture used here include AlexNet and GoogLeNet are discussed and implemented.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, deep learning.

1 Introduction

The work belongs to the image classification task in the field of computer vision. The process of image classification is very clear: give the labeled image data, perform feature extraction, and train the model. In the field of image classification, famous data set includes MNIST, CIFAR-10, ImageNet and MS COCO.

In machine learning, strategies for classification, including K-means clustering and support vector machine, can be used to deal with this classification problem [1]. But in image classification, the neural network technology has more obvious advantages, especially the deep convolution neural network, which has been successfully applied.

In view of the advantages of deep convolution network in image classification, we will use such a classifier and gradually improve its performance. CNN learns to recognize basic lines, curves, then shapes, blocks, and finally more complex objects in the image. Finally, CNN classifier combines these large and complex objects to recognize the image. CNN learns to recognize objects by itself through forward and backward propagation, without requiring us to set specific features. CNN may have several layers of networks, each of which may capture different levels in the object abstraction hierarchy, as shown in Fig.1.

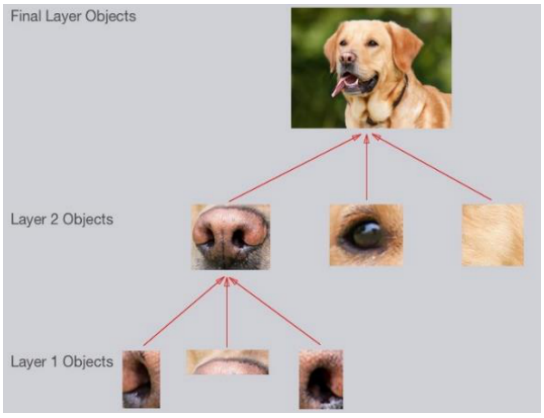

Fig. 1: A sketch of objects that CNN may recognize at each level on an image of dog.

2 Background / Formulation

The goal of this project is to utilize Nvidia Digits application to perform image classification tasks on two datasets. Since CNN has a big advantage over other machine learning approaches, deep convolutional neural network models able to classify objects from the dataset into their respective categories will be trained. There are several classical or popular CNN architectures to choose on Nvidia Digits like LeNet, AlexNet, and GoogLeNet.

2.1 AlexNet

The structure of AlexNet is shown in Fig.2. It consists of five convolution layers and three full connection layers.

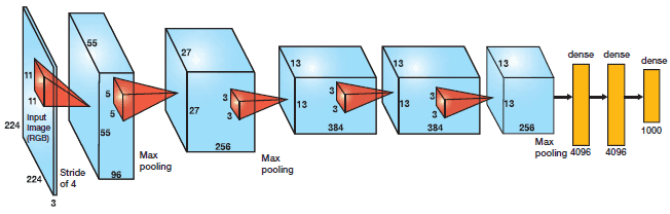

Fig. 2: AlexNet architecture.

2.2 GoogLeNet

"Inception" microarchitecture was first proposed by Szegedy in his paper "Going Deeper with Convolutions" in 2014 [2]. The Inception module and GoogLeNet architecture are shown in Fig.3 and Fig.4 respectively. The design philosophy is that the most efficient deep network architecture should be sparsely linked between activation values.

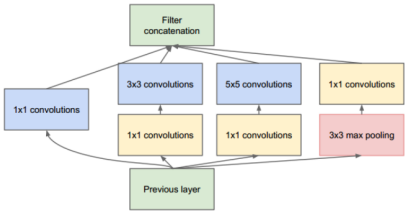

Fig. 3: Inception microarchitecture.

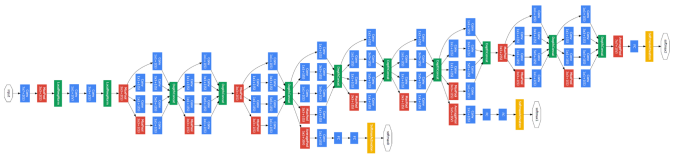

Fig. 4: GoogLeNet architecture.

3 Data Acquisition

3.1 Dataset 1: Bottle vs Candy-Box

The dataset include photos taken from a jetson mounted over a conveyorbelt, and is devided into three classes, containing candy boxes, bottles, and nothing. Some examples of the data is shown in Fig.5. The dataset has a problem of varying image sizes, which can be solved by resizing manually. However, Nvidia Digits can automatically performed resizing operation during loading the data. The dataset contains 10094 images.

Fig. 5: Sample images of the first dataset.

3.2 Dataset 2: Aircraft

The data set contains images of three types of aircraft, namely, airplane, quadcopter and helicopter. Each class. Each category contains 600 images, which are obtained through Baidu Search and Google Search. Some examples of the data is shown in Fig.6. The size of the image contained in this data set also varies.

Fig. 6: Sample images of the second dataset.

4 Results

4.1 Task 1

The training parameters of ALexNet model for Task 1 are given on Table 1.

TABLE 1: Training parameters of AlexNet

Parameter	Value
Optimizer	SGD
Learning rate	0.005
Epoch	15

Fig.7 shows the training process of AlexNet model on the first dataset.

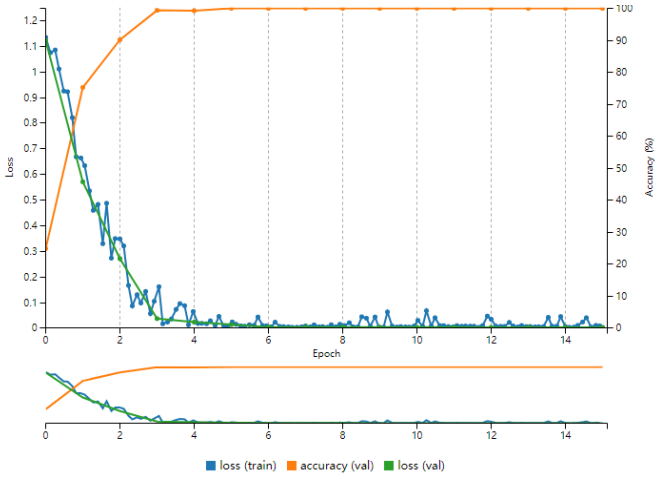

Fig. 7: Training graph of AlexNet on first dataset.

Fig.8 is the screenshot of the Digits console when evaluating the AlexNet model for Task1.

```
root@51efb28a4567:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20181224-130619-86c8

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20181224-130619-86c8/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20181224-130619-86c8/snapshot_iter_900.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.63186 ms.
Average over 10 runs is 4.62824 ms.
Average over 10 runs is 4.64658 ms.
Average over 10 runs is 4.6315 ms.
Average over 10 runs is 4.14882 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14636  100 12320  100  2316    976    183  0:00:12  0:00:12 --:--:--  2169

Your model accuacy is 75.4098360656 %
```

Fig. 8: Digits console output.

4.2 Task 2

The training parameters of GoogLeNet model for Task 2 are given on Table 2.

Fig.9 shows the training process of the GoogLeNet model on the second dataset.

Fig.10, Fig.11 and Fig.12 shows the predictions result on the test images of aircraft, which are not used on the training phase.

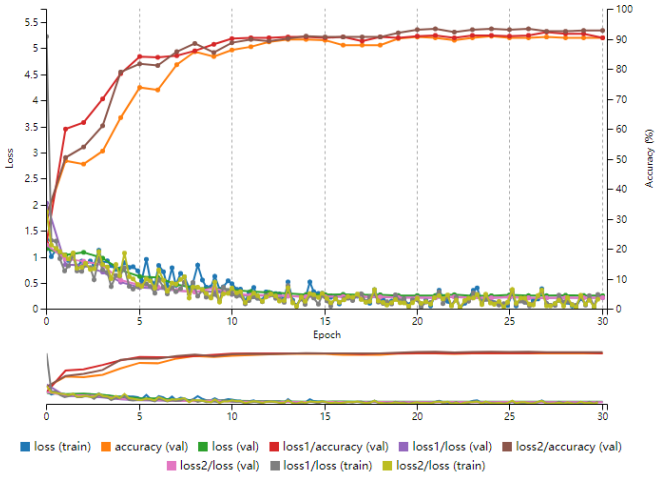

Fig. 9: Training graph of GoogLeNet on aircraft dataset.

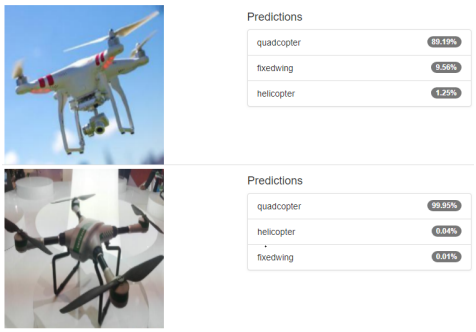

Fig. 10: Predictions on the test images of fixed-wing.

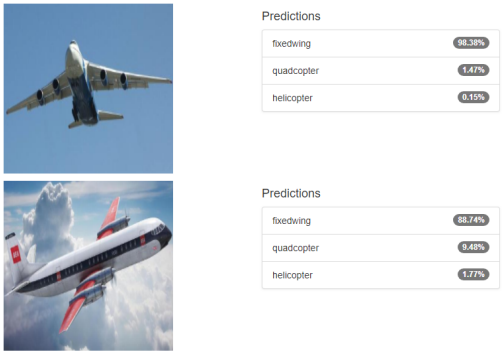

Fig. 11: Predictions on the test images of quadcopter.

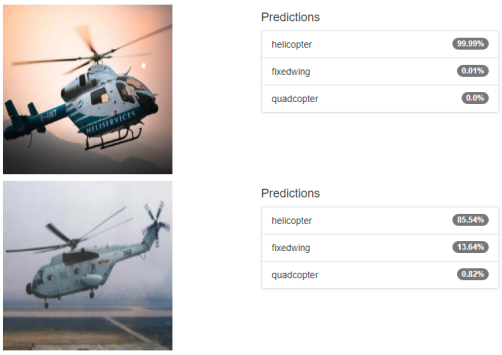

Fig. 12: Predictions on the test images of helicopter.

TABLE 2: Training parameters of GoogLeNet

Parameter	Value
Optimizer	Adam
Learning rate	0.0002
Epoch	30

5 Discussion

The AlexNet achieves a very high validation accuracy near 100% on the first dataset which contains 10094 images. The GoogLeNet achieves a validation accuracy over 90% on the second dataset which contains 1800 images. The AlexNet perform reletively better than GoogLeNet, which can be mainly due to sufficient amount of images, and high quality images – the background of images taken from a Jetson mounted over a conveyorbelt is very clean, while the images of aircraft are relatively complex. However, the accuracy of the second model can be improved by feeding more training data.

Accuracy and inference time are both important in industrial field. Accuracy ensures the normal operation, while inference time decides the real-time performance of the system, which indirectly affects corporate earnings. However, in industrial applications, the accuracy is more important than inference time. For example, a sorting system with a high classification accuracy can be robust enough to make products in an orderly manner.

6 Conclusion / Future work

The AlexNet fulfils the requirements and achieves an accuracy of 75.4 percent and an average inference time less than 5 ms, which is efficient enough for hardware deployment. The trained GoogLeNet also produces meaningful results on aircraft image classification.

For future works, collecting more data is a necessary step. Doing researches on making dataset better for network training can be helpful as well. In the future, air traffic will become more and more popular, such as flying cars and drones. Therefore, the real-time detection and recognition of aircraft are of great significance.

References

[1] D. A. Forsyth and J. Ponce, "A modern approach," *Computer vision: a modern approach*, pp. 88–101, 2003.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.