Deep Learning for Fruit Classification

Vijayasri Iyer

Abstract—This paper presents a novel use case for GoogLeNet, a popular convolutional network architecture, for classifying fruits. This application forms the basis for a kitchen service robot to develop relevant scene understanding capabilities using deep learning.

Index Terms—NVIDIA DIGITS, GoogLeNet, Deep learning.

1 Introduction

The three main qualities of a robot are perception, decision-making and actuation. For a kitchen service robot, it has to recognize the relevant objects in the scene, understand the action to be taken, and then interact with the real-world objects to perform the action. This paper aims to solve the first of three problems in implementing this robot, which had become quite easy with the rise of deep learning algorithms. Here GoogLeNet, a commercially available deep convolutional net architecture is used to classify fruit. Fruit classification is used a basis to create a model that ultimately classifies all types of kitchen items and foods with ease.

2 Background / Formulation

The NVIDIA DIGITS interface was used to train the GoogLe Net model on both, the custom as well as the supplied data set. DIGITS simplifies the common deep learning tasks such as managing data, training neural nets on Multi-GPU systems, monitoring performance in real-time and selecting the best performing model for deployment. GoogLeNet, is a 22-layer convolutional network, which won the IMAGENET 2014 challenge with an extremely low error margin of 6.66 percent, thus outperforming humans in image recognition tasks [1]. The notable part about this network is that it uses inception modules instead of regular convolutional layers. An inception module computes multiple different transformations over the same input map in parallel, concatenating their results into a single output. A schematic of the inception layer is shown in fig 1.

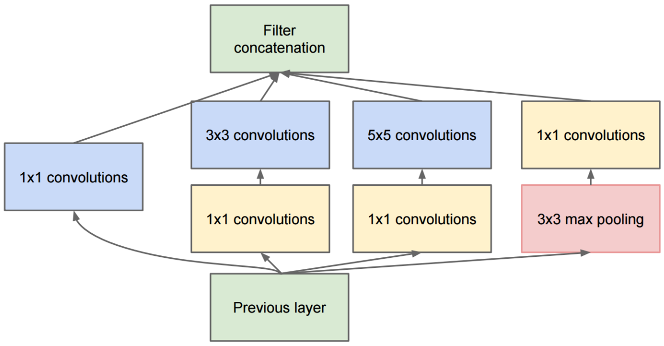


Fig. 1. Inception Module.

GoogLeNet inception modules consist of convolutional, maxpooling as well as average pooling layers. The GoogLeNet model has 4 million parameters in total. Hence this model is easier to maintain and gives a better overall result than other popular networks such as AlexNet, LeNet. However, the training time is comparatively longer, which makes it better suited to smaller data sets, especially if time is a constraint. It accepts images of type RGB (3-channels) and dimensions 256x256 pixels. The architecture of GoogLeNet is shown in fig 2.

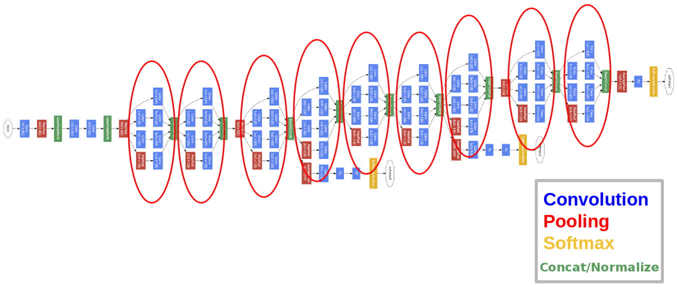


Fig. 2. Architecture of GoogLeNet.

3 Data Acquisition

The data set used in this paper is a subset of the Fruits360 data set available on Kaggle. While the original data set consists of 60 fruits and 38409 images, the classification task used 4 classes with around 440-500 images each for training and validation and 4 images per class for testing. The training data was further split into separate training and validation sets by the DIGITS interface. The categories in the dataset are Banana, Plum, Strawberry and Papaya. A few sample images from the data set are shown in fig 3.

The NVIDIA digits was evaluated with a sample data set, before training with the above data. This sample data set consists of 3 categories : bottles, candy-boxes and nothing. The total number of images in the data set is 7750, which was split into training, validation and testing sets by the DIGITS interface. Sample images from the supplied data set are shown below in fig 4.

4 Results

The results of the training and evaluation are discussed in the following two sections.

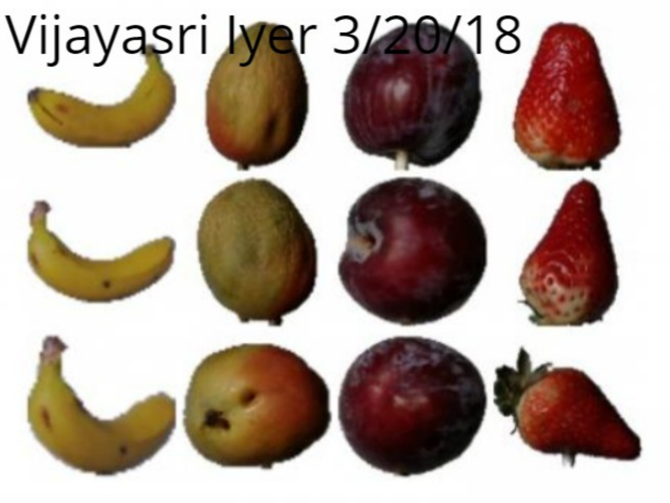


Fig. 3. Fruit data set images.

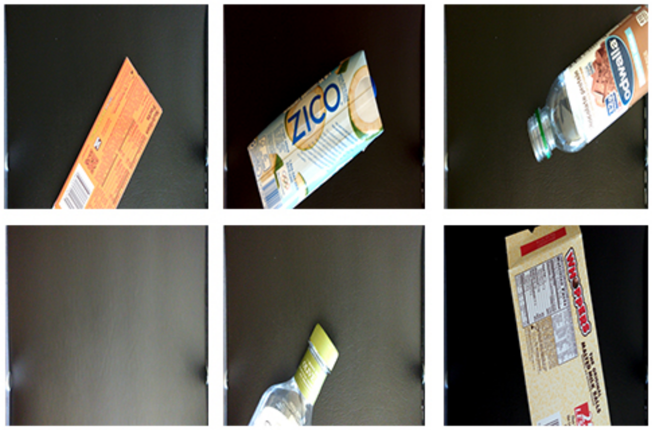
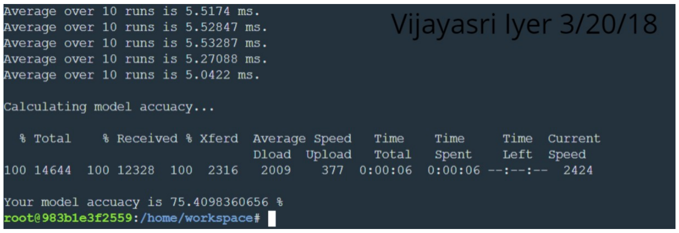


Fig. 4. Sample data set images.

4.1 DIGITS Sample Data

This data was trained on GoogLeNet for 15 epochs, with a batch size of 64 and learning rate of 0.01. The model achieved a final accuracy of 75.4 percent in the testing phase. The results of running the evaluate in the DIGITS workspace terminal can be seen below in fig 5.



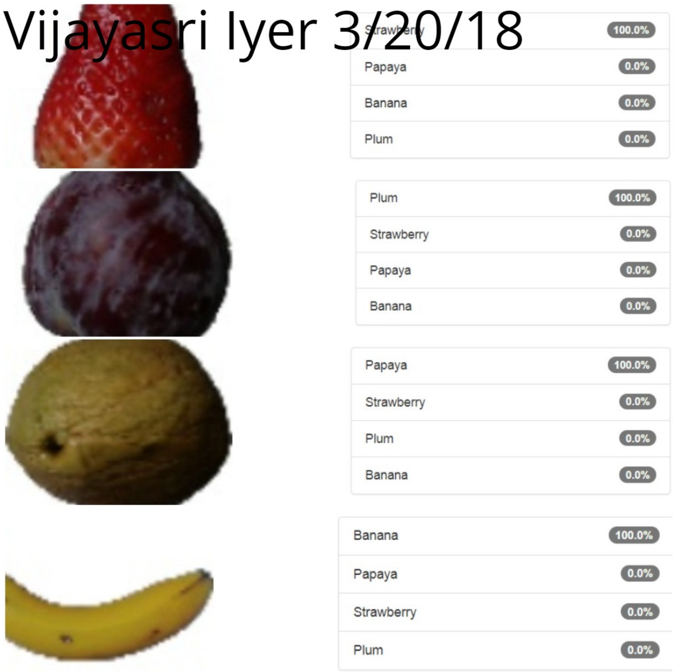
```
Average over 10 runs is 5.5174 ms.
Average over 10 runs is 5.52847 ms.
Average over 10 runs is 5.53287 ms.
Average over 10 runs is 5.27088 ms.
Average over 10 runs is 5.0422 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14644  100 12328  100  2316   2009    377  0:00:06  0:00:06 --:--:--  2424

Your model accuacy is 75.4098360656 %
root@983b1e3f2559:/home/workspace# 
```

Fig. 5. Sample data set images.

The average inference time of the model is 5ms which is sufficiently high to be deployed on hardware for real-time testing.

4.2 Fruit Classification Data

This data was trained for 30 epochs, with a batch size of 8 and learning rate 0.01. The model achieved a training accuracy of over 99 percent, and classified all the 16 test images correctly with a 100 percent confidence score. Results of the DIGITS classify command for the testing data can be seen below in fig 6.



Strawberry	100.0%
Papaya	0.0%
Banana	0.0%
Plum	0.0%

Plum	100.0%
Strawberry	0.0%
Papaya	0.0%
Banana	0.0%

Papaya	100.0%
Strawberry	0.0%
Plum	0.0%
Banana	0.0%

Banana	100.0%
Papaya	0.0%
Strawberry	0.0%
Plum	0.0%

Fig. 6. Sample data set images.

5 Discussion

As one can infer from the above sections, the Fruit classification model has quite a high accuracy while training as well as in the testing phase. However, there is reason to believe that the model may be overfitting. The reasons for this could be the nature of data set, where all the fruit images consist of a white background and that the training dataset is augmented by rotating the object by a certain angle to create more data. One of the problems with using the GoogLeNet for such a task is that, the nature of this task maybe to simple for such a complex architecture.

6 Conclusion / Future work

This Fruit classifier system, was an attempt to making an end-to-end system that can recognize all kinds of different kitchen and food items. This could then be implemented on a kitchen service robot, which gives it full-understanding of the environment around. To extend this project, various aspects can be improved. This may include increasing the number of categories, or getting a different dataset that covers more items to be classified Another option may be to choose a different network architecture for the following architecture which might be better suited to the purpose.

References

[1] Y. J. P. S. S. R. D. A. D. E. V. V. A. R. Christian Szegedy, Wei Liu, "Going deeper with convolutions," 2014.