Robotic Inference

Snigdha Dongre

Abstract— The paper aims at describing the implementations of the deep convolutional neural network for object detection application in NVIDIA's DIGIT workspace. Two different datasets are used here for the classification purpose, both having three categories, the first dataset is of pictures of candy boxes, bottles, and nothing (empty conveyor belt) for the purpose of real-time sorting and the second dataset is for transportation classification having pictures of cars, airplane, and motorbikes. The datasets were trained using GoogLeNet, evaluated and provided accuracy of 75.4% with an average runtime of 5 ms for the first dataset and accuracy of 98% for the second dataset.

Index Terms—Robot, Udacity, deep learning.

F

1 INTRODUCTION

In early days, the traditional image recognition model used a hand-designed feature extractor to separate the relevant data eliminating the irrelevant variables, followed by the trainable classifier, a standard neural network that classifies feature vectors into classes. The problem with this approach was the feature extraction cannot be tweaked according to classes and images, so if the features chosen to distinguish between the categories are lacking, then the accuracy of the model also suffered irrespective of the type of classification applied. Then came the CNN which acts as a feature extractor but is not hand-designed. The fully connected layer being used for classification are determined during the training process which resulted in saving a lot of memory, reduced the computational complexities and gave better results.

Deep learning is a set of automatic learning algorithms which uses an architecture composed of multiple non-linear transformations to learn multiple levels of representation that correspond to different levels of abstraction. Each layer uses the output of the previous layer as its input. Convolutional Neural network has a wide range of application such as in the field of speech recognition, computer vision, machine translation, social network filtering etc. One such application is object recognition or object classification which is being tested in this paper using two datasets.

2 BACKGROUND / FORMULATION

The objective of this project is to train a deep convolutional neural network to classify objects from the dataset into their respective categories. The first classification is called P1_classification which consist of three categories, pictures of candy boxes, bottles, and nothing (empty conveyor belt) and the second classification is called transportation_classification having pictures of cars, airplane, and motorbikes. NVIDIA's DIGIT workspace is used here to perform the classification. There are several architectures in the field of CNN like LeNet, AlexNet, ZF Net, VGGNet , GoogLeNet for different types of data. Both the dataset that we are using have colored (RGB) images, having a resolution of 256 x 256 pixels. AlexNet and GoogLeNet are most compatible architecture to deal with this dataset and amongst these two we will be using GoogLeNet to perform classification as it is the more recent architecture and have lots of advantages such as it has a so-called bottleneck layer which helps in massive reduction of the computation requirement, it also replaced the fully-connected layers at the end with a simple global average pooling which averages out the channel values across the 2D feature map, after the last convolutional layer which drastically reduces the total number of parameters, as the FC layers contains most of the parameters and the use of a large network width and depth allows it to remove the FC layers without affecting the accuracy. In the first classification, epoch were set to 5, as the dataset contained a lot of data to train the network and thus to save time epoch were set to 5. In case of the second classification, the epoch were expected to give a good results with epoch in the range of 20-30, so the epoch for this project were set to 30.

3 DATA ACQUISITION

The data for first dataset i.e. P1_classification is provided by udacity.This dataset is divided into three categories candy boxes, bottles, and nothing (empty conveyor belt).The dataset contains 10094 images out of which 7570 were used for training the neural network and the remaining 2524 images were used for validation. The images in this dataset were colored (RGB) and of size 256 x 256 pixels and the second dataset transportation _classification which also contain three categories, having pictures of cars, aeroplane and motorbikes was acquired from public dataset, so that a wide range of views and types of these things can be covered while training the model. This dataset contains 1738 images out of which 1303 images are used for training and the remaining 435 images are used for validation.The images in this dataset were also colored (RGB) and the size was changed to 256 x256 pixels to fit the GoogLeNet architecture.

4 RESULTS

The performance for the first dataset can be seen in the below images 4.1, where accuracy (Val) is near to 100% and the evaluation for the first dataset resulted in a model accuracy of 75.4 % and an average inference time of 5 ms, shown in fig 4.2 and fig 4.3.

Fig 4.1

```
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.5431 ms.
Average over 10 runs is 5.39267 ms.
Average over 10 runs is 5.19897 ms.
Average over 10 runs is 5.20651 ms.
Average over 10 runs is 5.07807 ms.

Calculating model accuacy...        Snigdha Dongre( 22 Mar 2018)
```

Fig 4.2

```
Average over 10 runs is 5.07807 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2316    0     0  100  2316      0     96  0:00:24  0:00:24 --:--:--     0
100 14676  100 12360  100  2316    207     38  0:01:00  0:00:59  0:00:01  2219

Your model accuacy is 75.4098360656 %
root@7d88ed3f0135:/home/workspace#  Snigdha Dongre ( 22 Mar 2018)
```

Fig 4.3

The performance of the second dataset can be seen in the below fig 4.4. This dataset gave an accuracy (Val) of nearly 98 %. The results for classification performed on some sample images of cars, airplane, and motorbikes from the internet of resolution 256 x256 are also shown in fig 4.5.

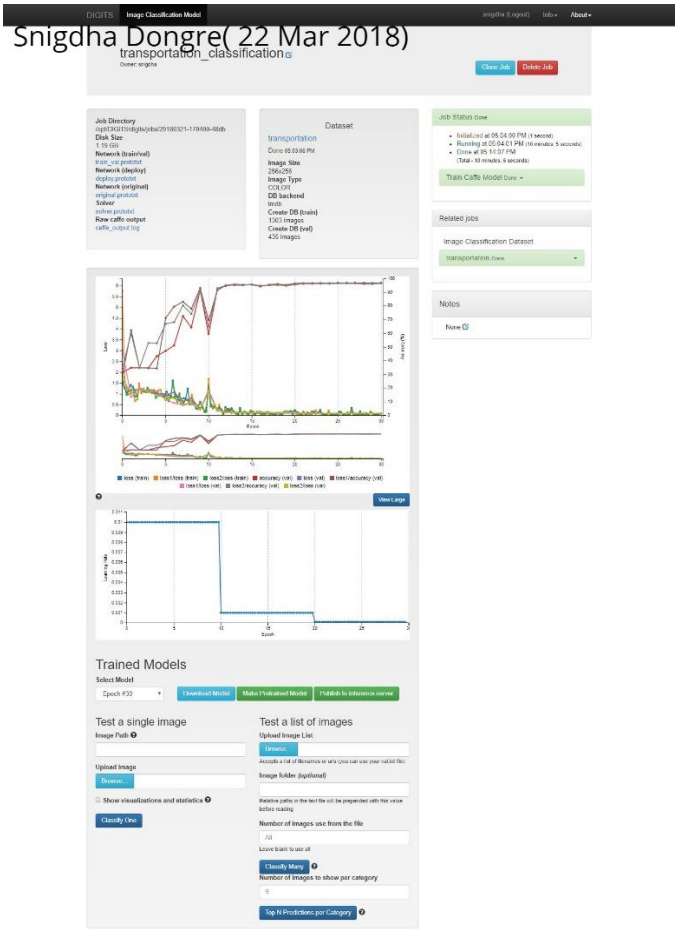

Fig 4.4

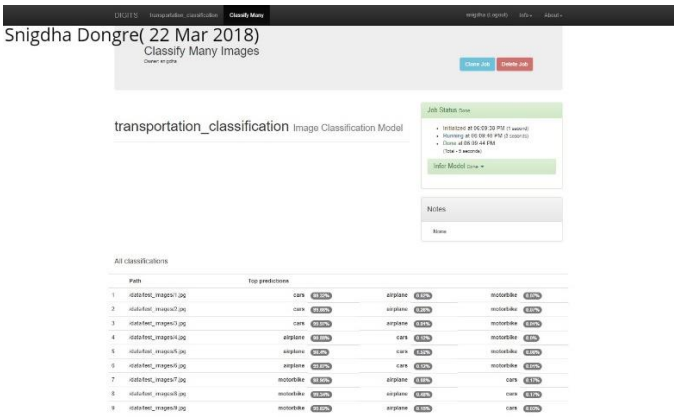

Fig 4.5

5 DISCUSSION

The classification performed using GoogLeNet architecture on both the dataset gave good results and met all the project requirements specified by Udacity. For the first dataset, the accuracy is close to 75.4% which is a good starting point but is far from state of the art. Also since the validation accuracy is close to 100%, I suspect possible overfitting scenario. However, with more data and some data augmentation, the accuracy of this model can be further improved. As for the second dataset, it is performing close to state of the art but can certainly benefit from more data. Transportation classification is used in many real time applications such as traffic management, journey planning, transportation planning, where it is important to make an inference rather than giving importance to accuracy. So in case of transportation classification inference is more important than accuracy.

6 CONCLUSION / FUTURE WORK

CNN's object classification application was successfully implemented using GoogLeNet architecture in NVIDIA's DIGIT workspace on both the datasets. The first classification P1_classification resulted in a model accuracy of 75.4 % and an inference time of 5 ms. While the second classification, the transportation_classification resulted in accuracy (Val) of 98%. For transportation_classification the model was tested using some random images from the internet of cars, airplane, and motorbikes and the results were fairly accurate.

The transportation_classification model can further be scaled to classify more types of vehicles and the accuracy of both the models can further be improved by using more data for training and some data augmentation. Mode of Transportation classification has a wide range of application such as for transportation planning, travel behavior research, traffic management, carbon di oxide emission estimation and it can also be used to improve location and positioning systems.

REFERENCES

[1] L. Lamport, *LATEX: a document preparation system: user's guide and reference manual*. Addison-Wesley, 1994.
[2] http://cs231n.github.io, Stanford CS class notes CS231n: Convolutional Neural Networks for Visual Recognition, 2017
[3] Wikipedia, https://en.wikipedia.org/wiki/Convolutional_neural_network
[4] UFLDL Tutorials, "http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork",
[5] Dhruv Parthasarathy, blog "https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4", 2017

[6] Samer Hijazi, Using Convolutional Neural Networks for Image Recognition, "pdfs.semanticscholar.org/bbf7/b5bdc39f9b8849c639c11f4726e36915a0da.pdf", 2015
[7] Xiaoyuan Liang, A CNN for transportation mode detection on smartphone platform