Image Classification and Object Detection using CNN

A Comparative Study using Traffic Sign Imagery

Farhan Asif Chowdhury
Dept. of ECE
The University of New Mexico
Albuquerque, NM 87131, USA
fasifchowdhury@unm.edu

***Abstract*—After the revival of CNN by their impressive performance in ImageNet Classification task in 2012, they are now being widely used in Image Classification and Object Detection. There are numerous published method on image classification using CNN where they have achieved near perfect accuracy. In contrast object detection task in real life scenario is yet to produce satisfactory result. In this work we have used existing traffic sign image database, both low and high resolution, to perform the task of classification and detection. In our study we have also found highly satisfactory result on classification even with small dataset, where else detection accuracy was poor and highly dependent on training dataset size and number of training iteration. For classification task we have used a modified version of AlexNet and for the object detection task we have used Single Shot MultiBox Detector (SSD) and Faster R-CNN architecture.**

Keywords-Convolutional Neural Network (CNN); Image Classification; Objec Detection; AlexNet; SSD; Faster R-CNN

I. INTRODUCTION

Most of the Image Processing and Computer Vision related tasks are now being tackled using Convolutional Neural Networks (CNN). Due to their nature of learning required differentiating feature set on their own, they have relieved researchers from going through the laborious task of creating hand crafted feature set. Though CNNs are data hungry to learn those feature set and require high computational resource, still they are highly coveted in image classification and object detection due to their better accuracy. Though there are many a similarities in between the task of image classification and object detection, there are some finer boundaries and differences. In classification, the image has the object of interest as its main focus and the object mostly occupies the greater portion. For training it requires the image and the associated label. While in operation it takes an image and predicts the class or label of that image from a set of pre-defined classes/labels. Where else for object detection, the image might contain various object of interest and often times they occupy only a small portion of the total image and some part might be occluded or the image might have only some portion of the total object. For training, it needs the image along with bounding box co-ordinates of the object of interest in the image and their class labels. During testing, it gives the bounding box and class labels of the detected objects.

There are some fundamental differences in the approach and architecture of CNN regarding classification and detection. The general outline of classification architecture is some Convolutional layers followed by a few fully connected layer and a soft-max layer to produce the class label of the image. In object detection there have been many a different approaches, but in general there are some object proposal stage along with convolution layers to generate some region boundary or object of interest to classify. As evident from the complex nature of architecture, the task of detection is also intricate. Image classification has now achieved very promising accuracy and error reduction, but object detection is still going through lots of challenges and experiments.

One of the main challenge of object detection is there might be several objects to be detected and classified where they are very small in size as seen in real world. Researchers have applied CNN based object detection approach to tackle numerous types of real world computer vision problems. Traffic Sign classification and detection has become one such important and prospective research area in Computer vision task since the uprising of Self-Driving Car idea. Most of the work in this field has been focused on traffic sign classification where the image has the traffic sign in its major portion and good accuracy have been achieved in this case. There are some large and well curated dataset for this traffic sign classification task, one such is the German Traffic Sign Benchmark.

But in real world scenario the traffic sign is only a small part (often less than 1%) of the whole image. And the objective is to both detect and classify, where current methods fails to provide promising result. Recently a lot of focus are being given it detecting and classifying traffic sign in real world scenario as this has more real life implications. So, people have put effort to create dataset traffic signs as seen from the practical view point of car. Tshinghua-Tencent 100K is one such data set where they have collected images from Street View and annotated them. There is also another such database called LISA traffic sign dataset.

In this work we tried to tackle both classification and detection task using the aforementioned datasets. For the classification task we will be using the German Traffic Sign Benchmark and for the detection task we will be using Tshinghua-Tencent 100K and the LISA traffic sign dataset. At first we tried to do traffic sign classification using German Traffic Sign Benchmark. Though the task of traffic sign classification has been shown to achieve very good accuracy, we approached this task for some specific reason. To perform classification, the required dataset size is very small comparatively to the detection datasets. Also, the CNN architecture is less complicated and requires less iteration of training. While performing the classification task, we gained first hand practical experience of working with CNN architecture, and as the dataset was comparatively small we could perform the training using regular CPU. We achieved very good accuracy in this task which were in par with the existing methods. After the classification task with German Traffic Sign Benchmark dataset we then moved onto the detection task.

The rest of the paper is organized as follows: in section II detail methods of our classification task have been described, then in section III we gave description of our localization task (only detecting the traffic signs in the image, not classifying them in distinct category) and in section IV we described about the simultaneous detection and classification task. Each of these section are self-containing, as all of them has brief description about the dataset used, the employed CNN architecture, performance evaluation, experimental example outputs and challenges faced while implementing that particular task. In the section V, we gave an overall comparative analysis in between the three distinct tasks performed and in the section VI, we finish the paper by giving conclusive remarks and direction about future prospects of this work.

II. Traffic Sign Classification

A. *Description of the Dataset*

We have used German Traffic sign Benchmark for the classification part. The input image size is: 32 by 32 pixel. There were 43 different classes. For training purpose we used 39209 images and for testing we used 12630 images. 20% of the training images were used for validation. Class distribution of training data is shown in Fig 1.

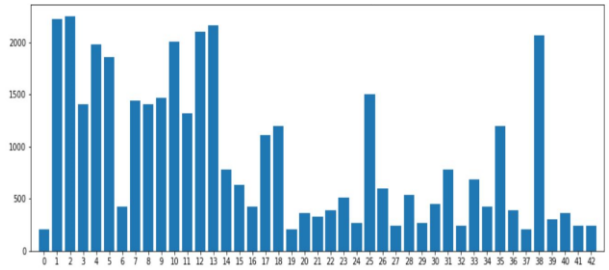

Figure 1a: Distribution of classes in Training Data

B. *Data Preprocessing*

As the dataset was small in size, it was augmented (ex: random rotate, random shift etc.) to create additional 100,000 training dataset. Also, the pixel values of the images were normalized in between -1 to 1. This standardization of pixel values is helpful for faster gradient convergence. The integer class label of images were converted into one hot encoding format. No conversion to gray scale was performed in the final setup as it didn't improve accuracy rather slowed down the process.

C. *CNN Archtecture*

Overall the architecture was a Convolutional Neural Network, we adopted the architecture of AlexNet with some modification. As the input image dimension was smaller, we reduced the dimension of all the layers. For regularization, dropout was used in the final fully connected layers. A tabular format of the full architecture along with dimension values is given below in Fig 2.

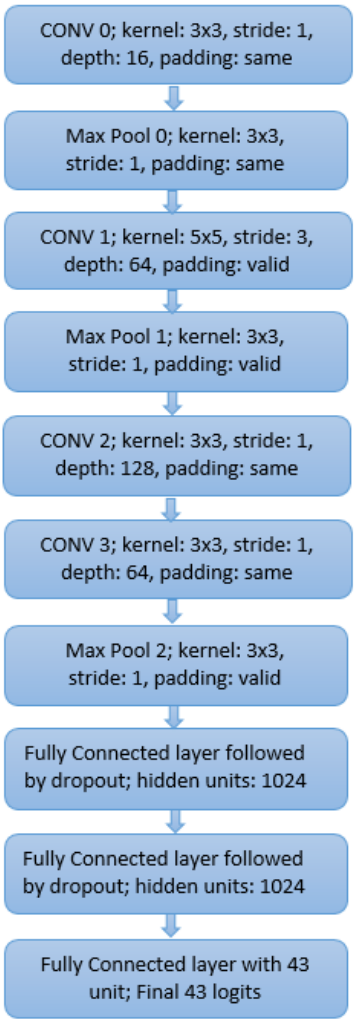

Figure 2: Description of the CNN Architecture

D. Train, Test, Result and Evaluation

The model was trained using Stochastic Gradient Descent. To select the batch size two things were considered mainly, if the batch size is too small then there will be high variance in initial gradient update which will result in slow convergence, and if the batch size is too big there are chances of getting out of memory along with longer time period for a single gradient update. That's why a mid-point value was chosen. The number of epochs was large enough to make the training and validation accuracy saturation visible. The final settings of hyper parameters were:

Optimizer: Stochastic Gradient Descent
Batch Size: 128
Epochs: 40
Learning Rate: 5e-3
Dropout keep-probability: 0.5

The accuracy on test set was: 96.21%, which is in per with other similar traffic sign classification methods.

Figure 4a: Actual Sign- Slippery Road
Prediction: Slippery Road 100%

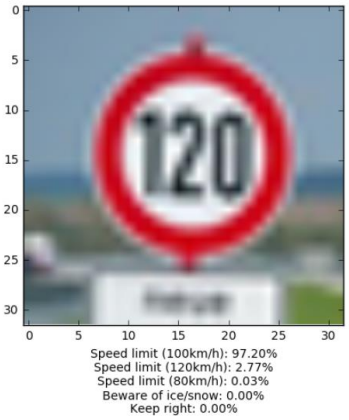

Figure 4b: Actual Sign- Speed limit 120 km/h
Prediction: (1) Speed limit 100km/h - 97.20%,
(2) Speed limit 120km/h – 2.77%,

III. TRAFFIC SIGN DETECTION AS A SINGLE CLASS

We wanted to detect and classify traffic sign in real life scenario, where the traffic sign only occupies a small portion of the image. Researchers from Tshinghua University, China have created one such traffic sign image database using Street View images. They have named this dataset as Tshinghua-Tencent 100k. For our detection task we primarily selected this database to start with as it mimicked the real world scenario very closely. The researcher from the same group has also published a CNN based traffic sign detection method using their dataset. We used their proposed method as a baseline model to start with.

A. Description of the Dataset

There are in total one hundred thousand images in this dataset, among them ten thousand had traffic sign in them and the rest of the ninety thousands were only background images. The images with traffic sign were annotated manually, a bounding box was drawn around the traffic signs. In total there were more than 150 classes of traffic signs in those images, but we only considered those traffic sign classes that had at least 100 sample images and finally there were 45 traffic sign classes. The image resolution of these images were 2048 by 2048. One sample image is shown in figure 5. It can be seen that the traffic sign are really small in size compared to the total image size. Actually there are four traffic signs in this image which are very hard to notice. In figure 6 we show the distribution of the 45 traffic sign classes.

Figure 5: A sample image from the Tshinghua-Tencent 100K Traffic Sign Database

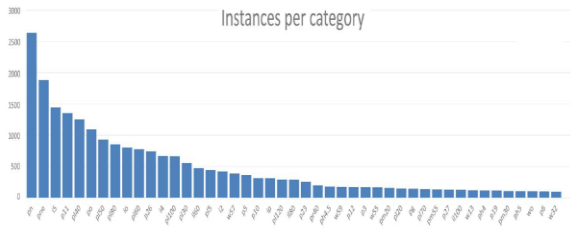

Figure 6: Distribution of images of 45 traffic sign classes

B. Detection Procedure

The research group of Tshinghua-Tencent 100K [1] has used Overfeat object detection framework [2] as their basic CNN architecture for traffic sign detection and classification. They have modified the original approach to simultaneously detect and classify the traffic signs. Their used modified architecture is shown in figure 7a and their architecture description is given in figure 7b. At first we tried to implement their architecture from scratch using TensorFlow framework, but there were many an ambiguity in their architecture description and we weren't able to finish the intricate task of creating a CNN and object proposal based object detection method as it not only required defining the CNN architecture in the first few feature generation stage but also the sliding window based multi-scale object proposal scheme.

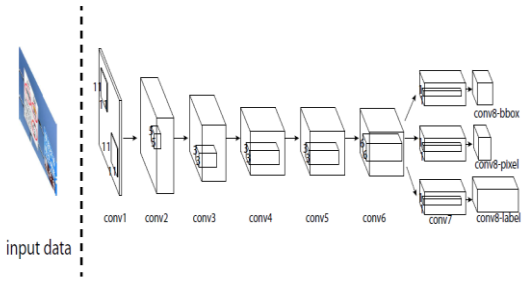

Figure 7a: CNN Architecture by Tshinghua group [1]

layer	data	conv1	conv2	con3	conv4	conv5
output size (chan×h×w)	3,480, 640	96,118, 158	256,59, 79	384,29, 39	384,29, 39	384,29, 39
input size		3,480, 640	96,59, 79	256,29, 39	384,29, 39	384,29, 39
kernel size, stride, pad		11,4,0	5,1,2	3,1,1	3,1,1	3,1,1
pooling size, stride		3,2	3,2			3,2
addition		lrn layer	lrn layer			

layer	conv6	conv7	conv8-bbox	conv8-pixel	conv8-label
output size (chan×h×w)	4096,15, 20	4096,15, 20	256,15, 20	128,15, 20	1000,15 20
input size	384,29, 39	4096,15, 20	4096,15, 20	4096,15, 20	4096,15, 20
kernel size, stride, pad	6,1,3	1,1,0	1,1,0	1,1,0	1,1,0
pooling size, stride					
addition	dropout 0.5	dropout 0.5			

Figure 7b: Description of the CNN by [1]

So, later we switched on using Single Shot MultiBox Detector (SSD) [3] and Faster R-CNN [4] method as they are the current state of the art object detection methods. Another reason of using these to scheme was that they are available in TensorFlow Object Detection API as built in architecture to use. Also in TensorFlow Object Detection API these models are pertained with some existing dataset and can be used from those saved model checkpoints to achieve faster convergence with new training dataset.

At first we choose SSD framework as it has higher mAP and faster computational capacity then Faster R-CNN. Also, it has shown promising performance in detecting smaller sized object. The first few feature extraction Convolutional layers of SSD is similar to VGG-16.The architecture of SSD is shown in figure 8.

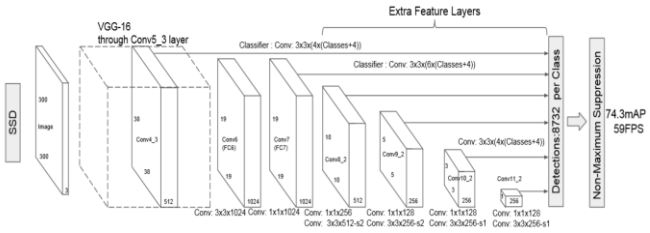

Figure 8: SSD CNN Architecture

The total training images used by [1], was one hundred thousand. They had high amount of computational resources at their disposal to train their model with that huge amount of dataset (around 100 Gigabyte). For our training purpose we chose 5072 images (6.2 GB) as training set and 1232 (1.32 GB) images as test set. There were 45 traffic signs to classify. We ran the ssd_inception_v2_coco model from the TensorFlow Object Detection API which was pertained on the COCO dataset, and we started our training using our 5072 images from that pre-trained model checkpoint for faster convergence. We had to convert our images and the corresponding annotation into tfrecord format to give as input into the API input pipeline. The evaluation image set was also converted into tfrecord format. We ran the training in a high configuration computer with 16 GB of GPU memory for more than 8 hours and later stopped the training as the loss value wasn't decreasing. Then we used our saved weight from the trained model to test its performance. But it was evident from the high training loss value that the model didn't converge. While testing, it couldn't even detect a single traffic sign. We tried to detect the underlying reason of our failure, for which we ran the model again with different hyper parameters, but nothing changed. We came in conclusion that, as our image size was really big and the object of interest, the traffic signs were very small (often less than 0.1% of the total image), it does require lots of training images along with high number of iteration. Also, the variation of classes of traffic signs was another main reason. Later we tried to train the faster_rcnn_nas model

from the TensorFlow Object Detection API, but it also failed to detect the traffic signs.

From the failure and debugging we could realize that considering the size of traffic signs in the images, the small number of training images and lower number of batch size and iteration it was not viable to detect and classify all the 45 traffic sign classes. So, rather than doing simultaneously detection and classification we resorted our effort on only detecting all the traffic sign as a single class. In our training input tfrecord we labeled all the traffic signs as a single class and used the Faster R-CNN model for this purpose. Then we ran our model training using previous high configuration computer. We used the saved model after fifteen hour of training to detect traffic signs in the test images. This time the model could detect the traffic signs with high variance and lower positional accuracy. We give some example of our traffic sign detection as a single class in figure 9a, 9b, and 9c.

Figure 9a: Detected traffic sign in the upper right corner

Figure 9b: Detected traffic sign in the upper middle portion

Figure 9c: Detected traffic sign in the upper left corner

As the image size is very high in resolution, when it has been made smaller to fit in a single page column, the bounding box become very thin to notice. From this new approach we could come to conclusion that our used model was in fact working but it couldn't perform up to the mark with very high resolution images with very small object of interest with lower number of training example and iteration. To perform both detection and classification it does require large number of training example and longer training period.

IV. TRAFFIC SIGN DETECTION AND CLASSIFICATION

To perform the task of simultaneous detection and classification we chose another dataset, LISA traffic sign dataset, which has images with small resolution which is suitable for training with SSD models.

A. Dataset Description

In the LISA traffic sign dataset created by the University of California San Diego (UCSD), there are in total 47 different classes of traffic sign. The image size vary from 640x480 to 1024x522 pixels. There in total 6610 image frames with 7855 annotations. For our detection and classification task we only considered two different traffic sign classes to test our model accuracy. The traffic sign classes were the Stop Sign and Pedestrian Crossing Sign. We only used those images and annotations that have these two traffic signs for the training and evaluation purpose.

B. Detection and Classification Procedure

We have used the previously mentioned SSD architecture which failed for Tshinghua-Tencent 100K dataset. The model was first trained on VOC 2102 dataset and then transfer learning was performed to train on the LISA traffic

sign dataset. Input image resolution was selected as 400x260 and images were resized accordingly. Some example of the detected and classified images are shown in figure 10a and 10b.

Figure 10a: Detected and classified traffic sign (Pedestrian crossing)

Figure 10b: Detected and classified traffic sign (Stop sign)

V. COMPARISON BETWEEN CLASSIFICATION AND DETECTION

In this work, we have tried to tackle two very fundamental problems of Image Processing and Computer Vision using Convolutional Neural Network. We have achieved reasonable accuracy in the classification task with very limited resource, where else for detection the scenario was different. It was difficult and challenging and the accuracy was very low and often time highly erroneous. Our detected bounding boxes of the traffic signs were not accurate and mostly covered larger region in comparison with the actual traffic sign which is an indication that our model needs more training to converge further to increase detection and localization accuracy. We performed quantitative analysis on the classification task and the overall accuracy on testing dataset was more than 96%. But for detection we didn't do any quantitative evaluation as the performance was not in that level to compare with ground truth and would yield very low accuracy. From our experience of working on this project, we can say that the task of detection and simultaneous classification is still an open field to work on.

To perform object detection in real life scenario, the models need to go through lots of fine tuning to set the hyper-parameters to address that particular type of task. These parameters are highly dependent on image size, size of the object to be detected, dataset size etc. One glaring example is that the existing SSD models in the TensorFlow Object Detection API couldn't handle our high resolution Tshinghua-Tencent 100K images. Also, one model trained on some particular type of dataset may not be used on a different types of dataset directly simply via transfer learning. Moreover, it requires large amount of training example to converge to a reasonable inference model and large amount of training example requires high computational resources. So, to develop object detection methods with accuracy as per with that of image classification, one needs to address a diverse set of challenges.

VI. CONCLUSION AND FUTURE WORK

One of the main reason of selecting this particular task and these datasets were to get a practical exposure to working with large scale dataset that resembles real world scenario. While working on this project we had to tackle the challenges of high computational resources due to the large size of our dataset. Our primary goal was to run our training on High Performance Computing System of CARC-UNM, but as it was trouble shooting we couldn't use that resource. Later, we got hold of a Computer with 16 GB of GPU memory which helped a lot during the training phase of our models.

To solve the high stake problem of detection, we took a modular approach. First, we acquired enough experience on how to deal with CNN architecture, training steps, tuning hyper-parameters by working on the less complicated classification task. When we couldn't get any positive output on detecting and classifying high number of classes, we made our problem at hand simpler by unifying all classes as a single one and then moved onto doing multi-class detection and classification.

In future, we want to work on human activity recognition in real world scenario. One of the crucial step of human activity recognition is to detect and identify different body parts and their relative position in the wild which can be viewed as multi-class object detection and classification. We have gained a first-hand experience in that regard, we also became highly acquainted with TensorFlow framework

and TensorFlow Object Detection API which will be very useful in using off the shelf architecture and pertained models.

Our obtained accuracy in the object detection is still very low and has lots of scope of improvement via trial and error and fine-tuning. Also, it will require high performance computing system to play with this large amount of dataset. In future, we want to concentrate on using a greater portion of the dataset and to let our model train for a longer period of time using high performance computing system. If we can obtain reasonable accuracy on detecting the small sized traffic signs in the real world Street View images, it will also be effective in many other similar small sized object detection tasks in practical scenario.

ACKNOWLEDGMENT

We are highly grateful to Dr. Marios Pattichis for the practical and up-to-date course materials and class lectures which came in very handy while working on this project. His guidelines and outlines regarding this project were very fruitful. Also, we want to give a special thanks to Mr. Manish Bhattarai, a fellow course mate, for letting us use his High Configuration Computer with 16 GB GPU for our training purpose.

REFERENCES

1. Traffic-Sign Detection and Classification in the Wild - by Z Zhu (2016)
2. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks - by P Sermanet (2013)
3. SSD: Single Shot MultiBox Detector - by Wei Liu (2016)
4. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks - by S Ren (2015)
5. An Empirical Evaluation of Deep Learning on Highway Driving – by B Huval (2015)
6. Traffic Sign Recognition with Multi-Scale Convolutional Networks - by P Sermanet (2011)
7. ImageNet Classification with Deep Convolutional Neural Networks - by A Krizhevsky (2012)
8. Rich feature hierarchies for accurate object detection and semantic segmentation Tech report - R Girshick (2013)
9. Scalable High Quality Object Detection - by C Szegedy (2014)
10. Selective Search for Object Recognition - by JRR Uijlings (2103)
11. German Traffic Sign Benchmark Website:http://benchmark.ini.rub.de/?section=gtsrb &subsection=news
12. Tshinghua-Tencent 100K Traffic Sign Dataset Website: http://cg.cs.tsinghua.edu.cn/traffic-sign/
13. LISA Traffic Sign Dataset Website:http://cvrr.ucsd.edu/LISA/lisa-traffic-sign-dataset.html
14. TensorFlow Object Detection API Website:https://github.com/tensorflow/models/tree/master/research/object_detection