**Detection and Recognition of Traffic Signs**

**Through Deep Learning**

A Final Project Report for the

Final Project of the graduate Course

CS405: Machine Learning

January 14, 2019

Prepared by:

谭树杰11849060\*

钱凯11849333

卢博 11849159

李富恒11510835

\* Team Leader

**Abstract**

The traffic sign is a significant part for driving security, it provides many key information of road for drivers. In order to build a self-driving system, image recognition must be considered heavily, it is the fundamental system for further automatic driving decision make. We build the recognition system using DenseNet style CNN model and the detection system deploying Single Shot MultiBox Detector. We find that data augmentation can improve the performance of recognition but it makes the model need huge time to train. CondenseNet may be used to resolve this problem. We also proposed some possible improving methods.

.

***2.1 Research Question or Problem***

According to the experience from predecessors, there still exists some general problems for an accurate, completed recognition system. One traditional method for sign recognition is based on computer vision, to do the detection and classification. However, this solution needs much more time to handle the significant features in pictures, which is not acceptable for current requirement.

In the other hand, due to the driving environment, the image get from camera in the road is very hard to handle and process. Because this is a real world system, the shadow, the orientation, the mask and the size of the sign in an image are unpredictable and vary frequently. Even though, we still need a high accuracy for the result to ensure the security of driving. Therefore, we have to make the model to run as quick as possible while the accuracy should be kept high enough.

In the other hand, the data for training and testing is also a big problem needed to be solved. The GTSRB dataset is of highly class imbalance. We have to address this problem..

***2.2 Research Goals and Objectives***

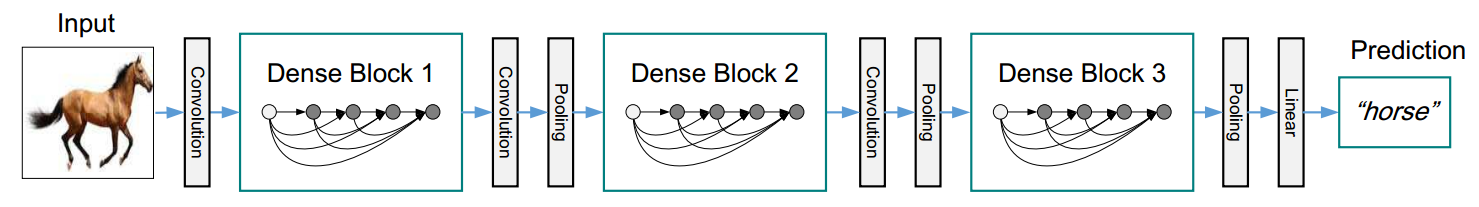
The research goal is to construct a traffic sign detection and recognition system by referring tools, methods and experiences from others. It will be an intelligent system to detect the traffic signs and classify them effiently to be used in self-driving system.

***2.3 Research Design and Methods***

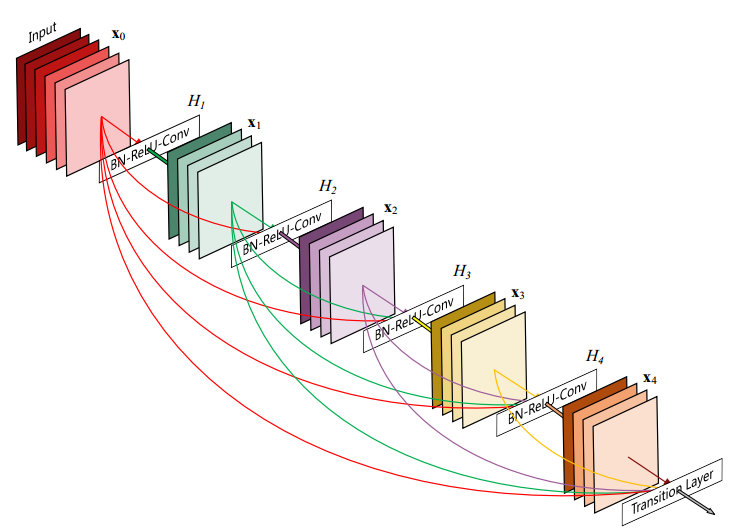
**2.3.1 DenseNet**

The traffic sign detection and recognition system can be deployed in mobile devices and consumer products. Therefore, not only are standard accuracy metrics important, such as the mean average precision (mAP), but other factors, such as memory consumption and running times, also play a critical role. Considering these factors, we will adapt DenseNet as our main model. Because DenseNets obtain significant improvements over the state-of-the-art on some on four highly competitive object recognition benchmark tasks, whilst requiring less memory and computation to achieve high performance.

The basic structure of DenseNet is shown below. DenseNet contains many Dense Blocks. We refer the layers between two adjacent blocks as transition layers, which do convolution and pooling and will reduce the feature-map size.



The Dense Block is shown in the right picture.



Every layer is connected to all other layers directly. This structure makes the reuse of features, yielding condensed models that are easy to train and highly parameter efficient. Therefore, the model would cost less memory and running times, which suits the application of traffic sign detection and recognition. In addition, because there is no need to relearn redundant feature-maps, the model requires fewer parameters than traditional convolutional networks.

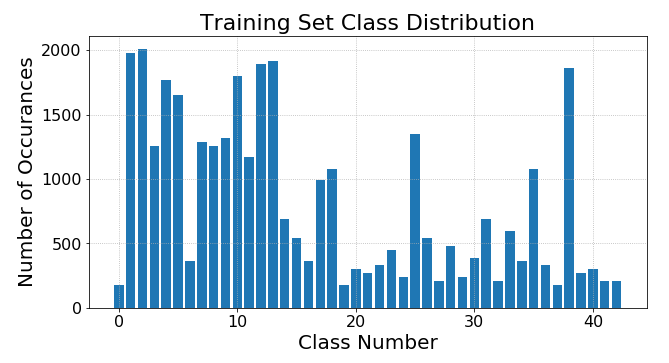
We will use the code of the paper *Densely Connected Convolutional Networks*. The code can be found at <https://github.com/liuzhuang13/DenseNet>. The code was implemented for four recognition benchmark tasks(CIFAR-10,CIFAR-100,SVHN, and ImageNet). Therefore, we modify and adapt the code for the traffic sign detection and recognition problem to get better performance.

Since the classes of GTSRB dataset is 43, which is between the classes of CIFAR-10(10 classes) and the classes of CIFAR-10(100 classes), we will design parameters and architecture according to that having best performance for CIFAR-10 and CIFAR-100. Hence, we will use DenseNet-BC and set growth rate k = 24, 28, 32, 36, 40, 48. The number of dense blocks and feature-map sizes are the same as in the paper *Densely Connected Convolutional Networks.* We may search for best number of layers in dense block for every growth rate k.

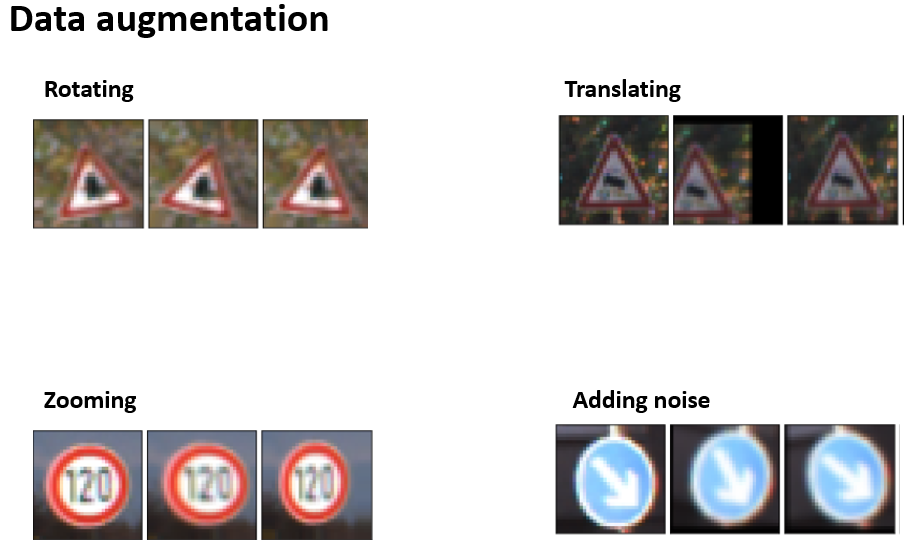
Furthermore, we will try to use some regularization technique such as cutout. As authors point out in *Improved Regularization of Convolutional Neural Networks with Cutout*. Cutout can improve the robustness and overall performance of CNN. It can also be used in conjunction with existing forms of data augmentation and other regularizers to further improve model performance.

**2.3.2 Data Augmentation**

We chose the dataset from the German Traffic Sign Recognition Benchmark for our training and test. The dataset, including 39,209 images for training and 12,630 images for test, features 43 different signs under various sizes and lighting conditions that are very similar to real-life situations. However, there are still some problems with the dataset: (1) it is large but not large enough for real environment; (2) overfitting may occur due to the limited size of training dataset; (3) the dataset is distributed unevenly in different categories. Therefore, we need to do data augmentation based on the original dataset to solve these problems.



Some typical data augmentation methods are applied: rotating, translating, zooming and adding random noise. A notable thing about data augmentation here is that due to the characteristics of traffic signs, there are some limitations. For example, for “turning left” and “turning right” images, it is not allowed to do the inverting actions since the inverted images obviously belong to the other category.



**2.3.3 Detection Design**

In our system, sliding window is used for extracting a candidate part from one original frame of our video, and the extracted images will be processed by our classifier. For real time system, it is really a time-consuming job to examine all the candidate areas and thus we have to optimize such a process.

(1) Limited window size

In real environment, the size of windows including traffic signs in each frame should be limited. Too large windows occupying a large part of the frame is impossible and too small windows means the traffic sign is too far and should not be taken into account immediately until it comes closer. Therefore, we should carefully set the window size limitation. By doing this, we only need to examine a small number of candidate windows.

(2) Center prediction

For one same traffic sign, the window of it will move smoothly in our continuous frames. To make full use of this characteristic, we can predict the possible window in the next frame. In most cases, the center of next window including traffic signs should be close to the current one, so we can only examine some of neighbors of the current window. If one of its neighbor is positively identified as one traffic sign, other windows are not ignored. This will help a lot for real time processing.

***2.4 Results***

The result of the experiments are shown below. As we can see, DenseNet without augmentation achieved 93.46% accuracy in test set. We did not tune the parameters much times, therefore the result has much room to improve. After using data augmentation, we find that the validation accuracy improved significantly, but the test accuracy did not change heavily. We may need more experiments to solve this puzzle. On the other hand, the time we need to train the model become seven times of the time required by the model without data augmentation. Hence, it may worth to design architecture to train more efficiently.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Validation Accuracy** | **Test**  **Accuracy** | **Training Time** |
| **DenseNet** | **0.9458** | **0.9346** | **20min** |
| **DenseNet +**  **Data Augmentatjion** | **0.9982** | **0.9313** | **2h20min** |

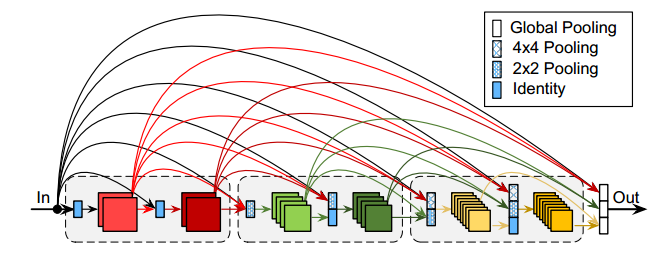
***2.5 Demo and Future work***

In our demo 1, there are two kind of traffic signs, which are indoor signs and outdoor signs. As we can see, the traffic signs in this demo1 are all detected and recognized correctly. First part in this demo1 shows that three different traffic signs, the speed limit sign, this way sign and turn right sign are all marked with boxes and texts as shown in the demo1. The second part shows the real situation in our life contains the one way sign, turn right sign and this way sign which are also marked correctly.

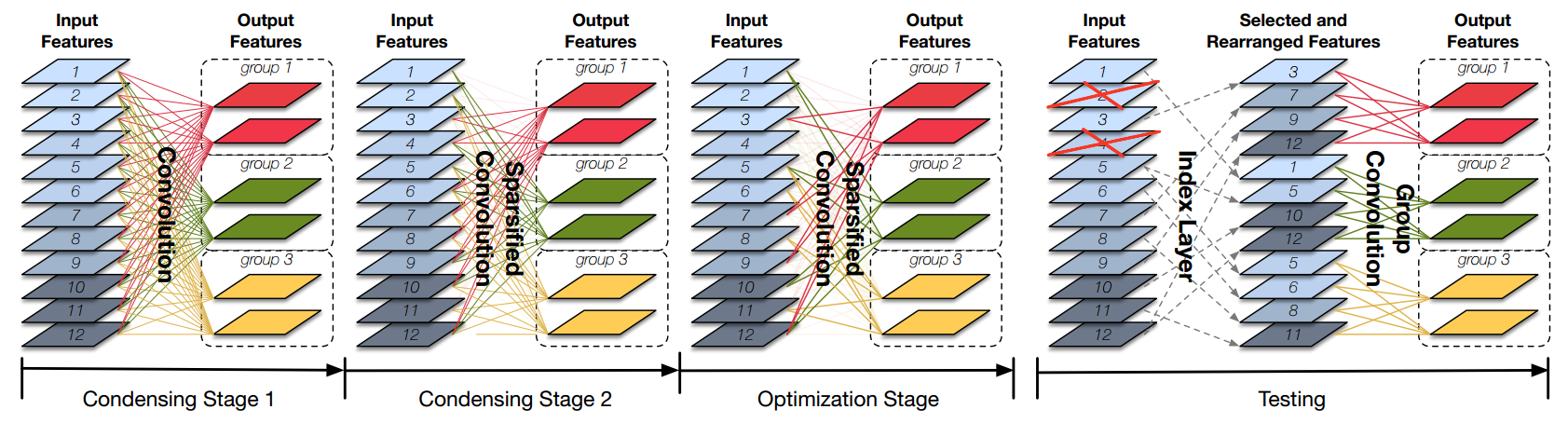
As shown in demo1, our marked box is not very continuous, and some of it is intermittent. The traffic signs in real life will not be easy to find and have no external interference like our experiments, so this experiment will continue to study on detection and recognition. For detection, we have already used the real time object detection such as Faster R-CNN, SSD and YOLO. Here we use the SSD method, the result is shown in demo 2(output.AVI). The marked box is stable and continuous, and we can find the model is faster and lighter in terms of memory consumption, making it an optimal choice for deployment in mobile and embedded devices.

For recognition, we will use more data augmentation to get the higher accuracy and there must contains hard negative mining for the real time object detection, and we will modify the network structure when necessary.

Although DenseNet has many advantages, but we should do more. First, DenseNet cannot propagate low level features to following stage. As shown in picture below, we can propagate the features to the following stages by down-sampling them first.



Second, the densely connections in dense block is needed on training. But after training, most of the weights are near zero and can be dropped, therefore we may use learned group convolution to compress the neural network. After pruning, it may be possible to deploy DenseNet in self-driving cars which have limited computational resources.



***2.6 Staffing***

Shujie Tan (CS postgraduate; 11849060): Traffic Classification by DenseNet CNN

Kai Qian (CS postgraduate; 11849333): Neural Network Architecture and Training

Bo Lu (EEE postgraduate; 11849159): Demo and Model improvement

Fuhuan Li (CS Senior; 11510835): Data collection and model analysis

***2.7 Timeline***

PROPOSAL: 10/17

Detection and recognition of traffic signs through deep learning. Describe our goal.

FINAL PRESENTATION: 01/13

Evaluate the relations between different positions of the transportation system Report our final method and explain how we improve the accuracy.

***2.8 Conclusion***

In this report, we use the sliding window to detect the traffic sign, while using CNN with DenseNet to build a model with high performance of anti-jamming for the recognition. The sliding window is good but not do well for our experiment, so we choose the real time object detection called SSD to get a more stable and continuous result. In additional, SSD is faster and lighter in terms of memory consumption, making it an optimal choice for deployment in mobile and embedded devices. Also, The DenseNet choosed as our final solution is compared to the traditional CNN model and CNN with data augmentation. The accuracy, generality and convergence time are all significantly improved. In our future work, we will use more data augmentation and hard negative mining for the real time object detection with higher recognition accuracy. And the network structure will be changed if it’s necessary.

**Reference**

[1] Eddie Forson. Recognising Traffic Signs With 98% Accuracy Using Deep Learning, https://towardsdatascience.com/recognizing-traffic-signs-with-over-98-accuracy-using-deep-learning-86737aedc2ab?tdsourcetag=s\_pctim\_aiomsg, Aug 24, 2017

[2] Á. Arcos-García, J.A. Álvarez-García, L.M. Soria-Morillo Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methodsNeural Netw., 99 (2018), pp. 158-165

[3] Sermanet P, Lecun Y. Traffic sign recognition with multi-scale Convolutional Networks[C]// International Joint Conference on Neural Networks. IEEE, 2011:2809-2813.

[4]J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 1453–1460. 2011.

[5]<https://github.com/chsasank/Traffic-Sign-Classification.keras>

[6]Huang G, Liu S, Laurens V D M, et al. CondenseNet: An Efficient DenseNet using Learned Group Convolutions[J]. 2017.