Traffic Sign Recognition with Light Convolutional Networks

Bo-Xun Wu*, Pin-Yu Wang, Yi-Ta Yang, and Jiun-In Guo†
Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan
Email: * brianwu0716.ee04@nctu.edu.tw, †jiguo@nctu.edu.tw

***Abstract*--In this work, we aim to design a light net that can be executed on the embedded system in real time. We modify VGG Net to a small net, called Safe Net, and utilize multi-scale features for traffic sign recognition. Moreover, we convert the dataset into grayscale, which has been proved that has a better performance on GTSRB dataset. In addition, we augment the training data by about 6.6 times more via spinning, distorting and flipping to boost the accuracy. On Nvidia Jetson TX1, Safe Net only takes 4.58ms per image including preprocessing at the testing and Safe Net can even achieve 99.34% accuracy.**

I. INTRODUCTION

Real-time traffic sign recognition is now an important topic for autonomous vehicles. GTSRB [1], a German road sign competition, provided dataset simulating different type of challenges on recognizing road signs. There are 43 class of signs, with 39210 images for training data, 12630 images for testing data. The size of images vary from 15x15 to 250x250 pixels. According to official information, the performance of human is 98.81% accuracy. We adopt GTSRB dataset for experiment to verify Safe Net can be perform well in real world.

Finding a recognition method with higher computation speed and accuracy is our primary goal. However, traffic sign recognition has a trade-off between the time and the accuracy. In general, the more parameters in the net, the more accurate on the result. After Alex Net [2] was introduced, the following advanced nets such as VGG Net [3] and Google Net [4] have been created in a massive size. We found that they took nearly a full second to classify an image on Intel Core i7-5930k processor, not to mention on the embedded system in a car. Therefore, we truncate VGG19, a prevalent net in the world, to only 7 convolutional layers to decrease the great amount of parameters and expect that it can classify an image in simply milliseconds. Pierre Sermanet and Yann LeCun [5] conclude that multi-scale convolutional networks and ignoring color information will benefit the accuracy on classification. As a result, we convert every RGB images into grayscale and add multi-scale features in our Safe Net. Furthermore, we spin, distort and flip the training dataset for a better performance on validation.

II. ARCHITECTURE

To reduce network parameters, we set input images dimension to 32x32, and use only one channel as input. Moreover, we apply two 5x5 convolutional layers and one 7x7 convolutional layer to construct the ConvNet. We add multi-scale features at the first and second pooling layers and concat them before fully connected layer. Fig. 1 illustrate the architecture of Safe Net

A. Convolutional layers

To reduce the parameters applied in the net, as VGG Net applied, we replace all 5x5 convolutional layers to two 3x3 convolutional layers and all 7x7 convolutional layers to three 3x3 convolutional layers. For a brief explanation, in the following article, we will just use "5x5 convolutional layer" and "7x7 convolutional layer" to represent our network.

We have tried 3 architectures: three consecutive 5x5 convolutional layers, two consecutive 5x5 convolutional layers then a 7x7 convolutional layer, and a 5x5 convolutional layers then two 7x7 convolutional layers. The performance of three consecutive 5x5 convolutional layer is the worst. However, the performance of the rest architectures are approximately equal.

Comparing to VGG Net, there are fewer filters on each layers in Safe Net. In VGG Net, there are 64 filters on the first layer and double numbers of filters on the second, and so on; in our Safe Net, we set only 32 filters on the first. With fewer parameters computed, not only the accuracy remains but also the computation speed is notably faster.

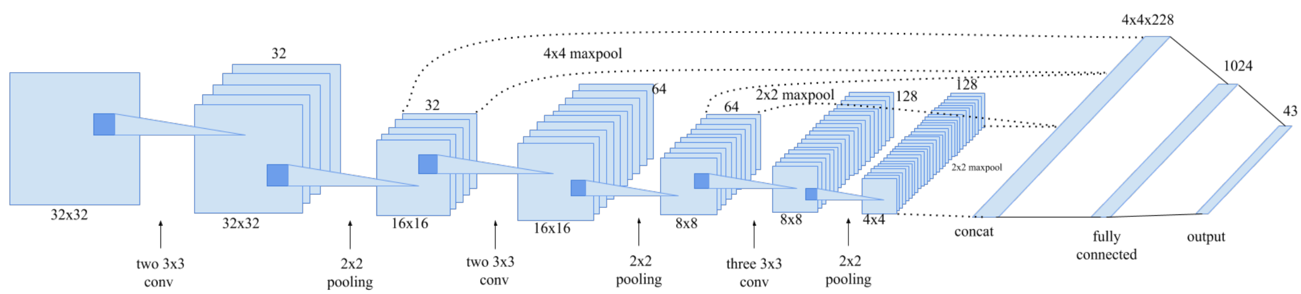

Fig. 1. The architecture of Safe Net.

B. *Multi-Scale Feature*

In usual ConvNets, each layer forward its parameters only to the next. Instead, with the multi-scale feature, parameters branch out and feed to a concatenate layer.

An additional 4x4 pooling layer is added between the first convolutional layer and the concatenate layer. An additional 2x2 pooling layer is added between the second convolutional layer and the concatenate layer. With multi-scale feature, the classifier gets different scales of ConvNet output instead of just the last stage output. As a result, both the outline and the detail of images will be considered.

III. EXPERIMENT

A. *Dataset*

We do some process on dataset to improve training and testing performance.

1) *Preprocessing*

We convert entire RGB images into grayscale and resize them to 32x32, as shown in Fig. 2(a). Then apply local histogram equalization to each image, as shown in Fig. 2(b). With data preprocessing, it will gain some accuracy on testing.

2) *Augmentation*

We flip symmetric road sign images. Moreover, we apply randomly distortion to each image, as shown in Fig. 3. Each image adds five transform version, almost can be seen as new data. The data augmentation can significantly increase the accuracy on testing.

(a) Original with grayscale (b) Histogram equalization

Fig. 2. An example image for data preprocess

Fig. 3. Three example images for random distortion

B. *Training*

We use dropout to avoid overfitting. We drop 30% parameters of the three convolutional layers forwarding to concatenate layer, and 50% for fully connected layer.

We use Adam [6] as optimizer, it can stably forward to the minimum. In addition, we decrease the learning rate before the end of training 100 epoch, it can decrease the training loss, and make the accuracy stable. The Networks in the experiment are trained with deep learning model Caffe [7].

C. *Result*

We train Safe Net with preprocessed data and original data, and train VGG Net with preprocessed data, which input size is 32x32. Then we do the testing part of the experiment on NVIDIA Jetson TX1, with NVIDIA Maxwell™ architecture and 256 CUDA cores.

Comparison of our method and state-of-the-art methods with both recognition accuracy and recognition speed is demonstrated in Table I. Safe Net brings about the best computational efficiency with high accuracy.

TABLE I
NETWORKS COMPARISON WITH ACCURACY AND SPEED

Method	Accuracy	Computation Time
Safe Net (preprocessed)	99.34%	4.58ms (Jetson TX1)
Safe Net (original)	99.14%	4.02ms (Jetson TX1)
VGG Net	98.73%	20.21ms (Jetson TX1)
Hunan average	98.81%	N/A
Hunan best	99.22%	N/A
Design [5]	99.17%	N/A
Design [8]	99.46%	11.4ms (GPU: GTX 580)
Design [9]	99.52%	40ms (CPU core i3)
Design [10]	97.75%	3ms (CPU 4-core 3.7 GHz)

Detail testing environment of our experiment: Input image format: PPM. Image size: 32x32. Preprocessing time is included.

IV. CONCLUSION

Safe Net reached 99.34% accuracy using grayscale dataset. Safe Net is a better solution than VGG Net on traffic sign recognition due to rapid testing speed. Additionally, we suggest that the testing image should be converted to grayscale. In actuality, converting takes merely 0.5ms on Nvidia TX1 and attains 0.2% accuracy.

REFERENCE

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *Proc. IJCNN,* pages 1453–1460. 2011. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism,* 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp. 68-73.

[2] A. Krizhevsky, I. Sutskever and G.E. Hinton, “ImageNet classification with deep Convolutional neural networks, ” in *Proc. NIPS*, 2012, pp. 1106-1114.

[3] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with Convolutions, ” *CoRR*, abs/1409.4842, 2014.

[5] P. Sermanet and Y. LeCun, “Traffic sign recognition with multi-scale Convolutional networks,” in *Proc. IJCNN*, 2011, pp. 2809-2813.

[6] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, “Project Adam: Building an efficient and scalable deep learning training system, ” in *Proc. OSDI*, 2014, pp. 571-582.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell “Caffe: Convolutional architecture for fast feature embedding, ” in *Proc. 22nd ACM international conference on Multimedia,* 2014.

[8] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” *Neural Netw.*, vol. 32, Aug. 2012, pp. 333-338.

[9] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, “A hierarchical method for traffic sign classification with support vector machines, ” in *Proc. IEEE IJCNN*, 2013, pp. 1-6.

[10] Y. Yang, H. Luo, H. Xu, and F. Wu, “Towards Real-Time Traffic Sign Detection and Classification, ” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, Jul. 2016.