Traffic Sign Recognition: A Test of Small Neural Networks and Expanded Datasets

Chao Hui Zheng
University of Tennessee
Knoxville, United States
czheng4@vols.utk.edu

Kriss Gabourel
University of Tennessee
Knoxville, United States
kgaboure@utk.edu

***Abstract*—Traffic sign recognition systems use a popular corpus of training data to train their models. Models today are typically developed using deep convolutional neural networks, which take a relatively long time to predict and even longer to train. Our research focuses on whether small neural networks trained with a much larger an augmented dataset can classify traffic signs with an accuracy similar to deep convolutional neural networks in a less amount of time. Our results indicate that small convolutional networks can produce similar results as deep convolutional neural networks, and that fully connected neural networks can improve classification results when and expanded dataset is used.**

***Index Terms*—synthetic images, data augmentation, traffic sign, recognition, generative adversarial network**

I. Introduction

Traffic signs are used in every country with a motor population to quickly convey relevant, important and timely information to drivers at a given location. Traffic sign recognition plays a vital role in increasing the viability of the self-driving car as the primary choice for vehicle consumers of the future. Self-driving cars, like all vehicles on the road, must obey traffic laws and adhere to the rules and regulations posted on traffic signs. Traffic sign recognition can also play a significant role in improving road safety for traditional drivers. For example, traffic sign recognition software could warn drivers who fail to see a stop sign before the driver enters the intersection.

In the past, template matching algorithms have been used to classify traffic signs. Template matching predefines the features of each traffic sign. However, these algorithms are often unable to achieve high accuracy due to variances in the dataset [1]. Pictures taken at different angles and in different seasons and weather conditions are often difficult to classify based on a template. Currently driverless car manufacturers rely on deep convolutional neural networks (DCNNs) to classify traffic signals [2]. Although the DCNN approach is less time efficient than template matching, the difference in speed is essentially negligible with the use of GPU. GPUs are vital in self-driving technology due to their speed in processing image data. With the powerful GPU support, researchers have been trying different architectures of DCNN to achieve high accuracy. Though DCNNs regularly approach 98% accuracy using the popular German Traffic Sign Recognition Benchmark (GTSRB) dataset, there is still room for improvement. In the real world, classifications must be done in real time within a moving car. Additionally, there are hundreds of calculations and classifications the GPU must process at the same time. The fact that the slow prediction process for DCNNs is mitigated by the use of GPUs has slowed the research on this particular problem. However, if a fully connected network or a convolutional network with fewer layers could achieve the same results as a DCNN for traffic signal recognition, it would be a better model to use and would free up GPU processing for other important aspects of autonomous driving. Our objective with this research is to determine if a smaller neural network can achieve the same accuracy. Without the option of very deep CNN's, our proposal to train accurate smaller neural networks is dependent upon acquiring more training data. To this end, we employ the use of artificial datasets to train our models. Through the course of our research, we developed 3 different artificial datasets and 3 different small neural networks. We compare the speed and accuracy of the resulting 9 models with the models using the GTSRB.

II. Previous Work

Training a neural network with sufficient data is crucial to the development of accurate models. If the dataset is too small, the model will not generalize the data well and the model will likely overfit the data. This results in a model with lower accuracy. Unfortunately, adding images to datasets can be expensive and time consuming. Collecting and labeling a sufficient number of images can be cumbersome and error prone. Equipment and hardware can be expensive, and labeling may require professional knowledge.

A. Traffic Sign Recognition

The German Traffic Sign Recognition Benchmark (GTSRB) is a large multi-category classification dataset. It contains about 50,000 traffic sign images in 43 different categories. The dataset was first introduced for a competition at the 2011 International Joint Conference on Neural Networks (IJCNN) [3]. 2 teams were able to produce a total of 5 neural networks that correctly classified the images at the same level as, or better than, human performance. A second dataset, the German Traffic Sign Detection Benchmark (GTSDB), was introduced for a competition at the same conference two years later. [4]. Since then, researchers have worked to increase accuracy

in classifying these images. Since the consequences for a missclassification can be enormous [5], common standards for high accuracy is not enough. When it comes to autonomous vehicles, classification of signs and objects must be very close to 100%.

B. Augmented Datasets

Data augmentation is the practice of transforming real images by performing one or more alterations. The change may seem minimal to the human eye, but to the model, it is a completely new image with new features to detect. Traditional transformations include shifting, rotation, flipping, zooming in or out, distortion, and color changes. Current work on augmentation is focused on adding sufficient noise to datasets. Neural networks can pre-train on augmented datasets to improve training efficiency [6]. Another augmentation practice is to add random white noise to images. Training on noisy data help to make CNNs more robust, and therefore less likely to misclassify images due to slight alterations in data that humans would consider inconsequential.

C. Synthetic Datasets

A synthetic dataset completely manufactured data, generated based on a features extracted from a real dataset. We created our synthetic images by building a Generative Adversarial Network (GAN) to generated the images. GANs provide an additional way to retrieve further information from a dataset by generating synthetic samples.

III. Technical Approach

We used the GTSDB dataset as loaded on kaggle.com [3]. The dataset contains 39,209 images of various sizes and colors (Fig. 1). The dataset is classified into 43 different categories. We used 80% of data for training and 20% for validation and testing.

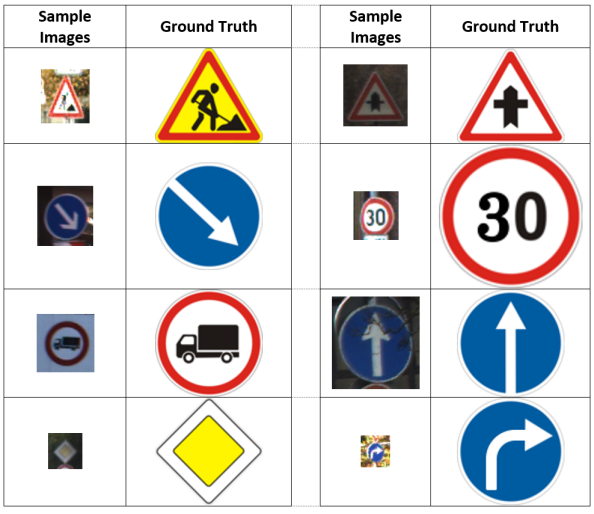

Fig. 1. Sample images in GTSRB

We present 3 different neural network architectures (Table I). Model 1 is a fully connected neural network trained using grayscale images. Model 2 and Model 3 are both 2-dimensional convolutional neural networks with a single hidden layer. They are trained using RGB images.

TABLE I
Neural Network Models

Model 1
Fully-connected-128, tanh
Dropout, rate = 0.25
Fully-connected-64, tanh
Fully-connected-43,softmax
Model 2
Conv2d, kernel 8 * 3 * 3, relu
Conv2d, kernel 4 * 3 * 3, relu
Flatten
Fully-connected-43, softmax
Model 3
Conv2d, kernel 16 * 3 * 3, relu
Conv2d, kernel 8 * 3 * 3, relu
Flatten
Fully-connected-43, softmax

For each model, We use the defaults for the adam optimizer to train our neural networks. Learning rate = 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We used categorical cross entropy as our loss function. We trained each model for 50 epochs. For the baseline training data we used a batch size of 64. For the other datasets we used a batch size of 32. We used 4vCPU, 15 GB memory, and no GPU for VM instance.

A. Image Dataset Augmentation

Supplementing datasets with augmented images is widely used today to expand datasets. Making subtle changes to real images produces a more robust dataset on which to train. Augmented datasets can help reduce overfitting and allow models to classify a more diverse collection of images after implementation.

Since the images in the GTRSB dataset are centered, several data augmentation methods were not appropriate for our use because they would change the location of the traffic sign. We excluded cropping, padding, horizontal flipping, and shifting. We employed 2 different data augmentation methods – brightness and rotation. We used the keras ImageDataGenerator function to modify these image properties. The brightness augmentation randomly changes the brightness of original images in range (0.85, 1.15). A brightness value less than 1 will darken the image and a brightness value greater than 1 will brighten the image. The rotation parameter randomly rotates the traffic sign clockwise anywhere from 0 degrees to 15 degrees.

B. Generative Adversarial Network

Supplementing a dataset with synthetic images is less widely used, but may also help to train networks better, resulting in more accurate classification. Synthetic datasets are particularly useful in the supplementation of smaller datasets. We built a conditional generative adversarial network (GAN) based on the code supplied by Jason Brownlee [7]. We optimized using adam with learning rate = 0.002, $\beta_1 = 0.5$ and

$\beta_2 = 0.999$. We used the binary cross entropy loss function. The GAN model was trained for 100 epochs and roughly 40 hours.

The conditional aspect of this GAN uses the training labels to generate specifically classified data. One advantage of GAN is that it is an unsupervised algorithm (or in the case of cGAN, self supervised) which essentially attempts to learn how to create 'augmented' images on its own [8]. Our conditional GAN model trained using the GTRSB dataset. The generator then generated images for each given label.

C. Expanded Training Sets

We applied each of these supplemental datasets our training set to create 3 separate expanded training datasets. The 'brightness' and 'rotated' expanded training sets contain twice as many images as the baseline training set. The GAN expanded training dataset added 500 synthetic images per category for a total of 21,500 additional images. We trained our 3 different models the baseline data and each of the expanded datasets. The testing data was also used as validation data, and the parameters of each model that had the highest accuracy of validation data during training were saved. In keeping with our research objective, our expectation was that using the expanded training sets would increase the accuracy of our neural network models when the baseline data was used.

IV. Experiments and Results Analysis

The classification accuracy and training time of each expanded dataset and model combination are shown in Table II and Fig 2. The results show that the brightness and rotation augmentations help increase the accuracy of the testing data. This is shown most strongly in Model 1. The addition of the rotated dataset improved Model 1 accuracy by a full percentage point without increasing the training time. For Model 3, data augmentation methods did little to increase accuracy. Model 3 is the most complex of our models, and it seems to be complex enough to learn any features that might be highlighted by a more robust training set.

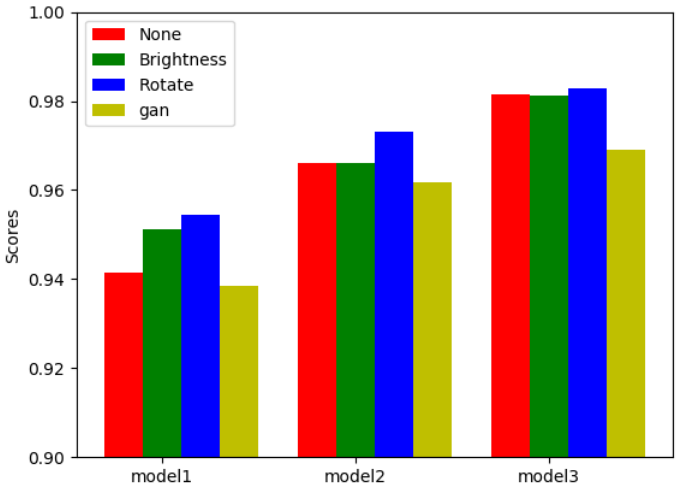

Fig. 2. Model Accuracy Graph

The GAN augmentation decreased the accuracy of the testing dataset in all 3 models. In general, generated images are of lower quality than real images. It appears that in the case of this dataset, this is indeed true (Fig. 3). Our generated images are extremely similar to the baseline training dataset; however, a visual inspection confirms they are of lower quality. Using the GAN extended dataset, our modes are hampered in their ability to learn features when the lower quality synthetic images are present.

TABLE II
Model Accuracy

Model	Supplement	Accuracy	Train Time
Model 1	None	0.9413340091705322	75us/sample
Model 1	Brightness	0.951281726360321	72us/sample
Model 1	Rotate	0.9544700980186462	74us/sample
Model 1	GAN	0.9385282397270203	75us/sample
Model 2	None	0.9662032723426819	312us/sample
Model 2	Brightness	0.9662032723426819	305us/sample
Model 2	Rotate	0.9730901718139648	310us/sample
Model 2	GAN	0.9617395997047424	320us/sample
Model 3	None	0.9813799262046814	330us/sample
Model 3	Brightness	0.9812523722648621	340us/sample
Model 3	Rotate	0.9829103350639343	335us/sample
Model 3	GAN	0.9691365957260132	337us/sample

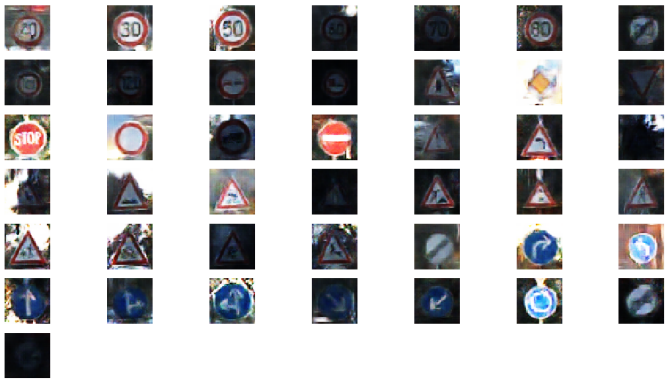

Fig. 3. Images generated by GAN

V. Conclusion

Smaller neural networks definitely benefit from the addition of augmented data added to the training set. Our fully connected model (Model 1) had significant gains in accuracy when the augmented data was added to the training data. Our synthetic images generated by our GAN did nothing to improve our accuracy. In fact, the addition of the images decreased the learning ability of all three models.

A. Future Work

There are two main areas we believe could benefit from further research. First, more research on fully connected models would be beneficial. Our fully connected model (Model 1) was the least accurate of all 3 models; however, the training time was significantly less than any other model. We are confident

that with a larger and more varied augmented dataset, the accuracy of Model 1 could match if not exceed Model 2, and do so in less training time. Different fully connected architectures, including the combination of probabilities of different models, might possibly produce even better accuracy without sacrificing efficiency. Since prediction speed was a major factor prompting our research, fully connected neural networks should be more thoroughly investigated to determine if the accuracy could reach the near 100% accuracy of the DCNNs. Secondly, a direct comparison is needed to DCNNs to determine if a fully connected model or a small convolutional neural network is an adequate replacement. Our research took the preliminary step of evaluating whether smaller neural networks were feasible, but no conclusions can be made regarding actual replacement of the DCNNs until direct comparisons are researched.

REFERENCES

[1] Fleyeh, H., & Dougherty, M. (2005). Road and traffic sign detection and recognition. *Proceedings of the 16th Mini-EURO Conference and 10th Meeting of EWGT* (pp. 644-653).

[2] Tiron, G. Z.,& Poboroniuc, M. S. (2019). Neural Network Based Traffic Sign Recognition for Autonomous Driving. *2019 International Conference on Electromechanical and Energy Systems* (SIELMEN) (pp. 1-5). IEEE.

[3] Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* (pp. 323-332). Data retrieved from: https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign on 4/25/2020.

[4] Houben, S, Stallkamp, J., Salmen, J., Schlipsing, M. & Igel, C. (2013). Detection of traffic signs in real-world images: The German traffic sign detection benchmark (GTSDB). *International Joint Conference on Neural Networks*.

[5] Wakabayashi, D. (2018). Self-driving Uber car kills pedestrian in Arizona, where robots roam. *The New York Times*, 3, 19. https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html

[6] Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. Paper presented at the *International Interdisciplinary PhD Workshop (IIPhDW)*. Swinoujście, Poland. https://ieeexplore.ieee.org/abstract/document/8388338

[7] Brownlee, J. (2019). How to develop a conditional GAN (cGAN) from scratch. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch on 5/2/2020.

[8] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A.,...Rueckert, D. (2018). GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks.*ArXiv preprint*. https://arxiv.org/abs/1810.10863

[9] Li, J., Tao, J., Ding, L., Gao, H., Deng, Z., Luo, Y., & Li, Z. (2018). A new iterative synthetic data generation method for CNN based stroke gesture recognition. *Multimedia Tools and Applications*, 77(13), 25.

[10] Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *ArXiv preprint*. https://arxiv.org/pdf/1511.06434.pdf%C3

[11] Renjith, R., Reshma, R., & Arun, K. V. (2017). Design and implementation of traffic sign and obstacle detection in a self-driving car using SURF detector and Brute force matcher. Paper presented at *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India.

[12] Rosebrock, A. (2019). Traffic sign classification with keras and deep learning. Retrieved from: https://www.pyimagesearch.com/2019/11/04/traffic-sign-classification-with-keras-and-deep-learning

[13] Sermanet, P., & LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *2011 International Joint Conference on Neural Networks* (pp. 2809-2813). Retrieved from: https://ieeexplore.ieee.org/abstract/document/6033589

[14] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. *Procedia engineering*, 201, 718-725. https://doi.org/10.1016/j.proeng.2017.09.594

[15] Sirbu, M.A., Baiasu, A., Bogdan, R., & Crisan-Vida, M. (2018). Smart traffic sign detection on autonomous car. Paper presented at the *International Symposium on Electronics and Telecommunications (ISETC)*. https://doi.org/10.1109/ISETC.2018.8583948

[16] Takács, Á., Rudas, I., Bösl, D., & Haidegger, T. (2018, December, 2018). Highly Automated Vehicles and Self-Driving Cars. *IEEE Robotics & Automation Magazine*, 25, 7.

[17] Taylor, L., & Nitschke, G. (2017). Improving Deep Learning using Generic Data Augmentation.

[18] Wang, J., & Perez, L. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. In *ArXiv preprint*, (pp. 8). https://arxiv.org/abs/1712.04621

APPENDIX A
CODE DESIGN

- The code for conditional GAN model (condition.py and gan_plot.py) is from the Machine Learning Mastery blog [7].
  - **condition.py** Trains the model. To run, type
    ```
    python3.7 condition.py
    ```
  - **gan_plot.py** Loads pretrained model and shows the generated images. To run, type
    ```
    python3.7 gan_plot.py
    ```
- **main.py** Functions are described in Table III. Run examples:
  ```
  python3.7 main.py model1 ""
  python3.7 main.py model2 "gan"
  ```
- **prediction.py** Takes a list of pretrained models as command line arguments and outputs the accuracy of testing data. Classification decisions are made based on the sum of probabilities of output layer from all models. Example:
  ```
  python3.7 prediction.py model1.h5
    model1brightness.h5
  ```

TABLE III
MAIN.PY FUNCTIONS

Function Name	Purpose
load_data()	Loads and splits the data to training set and testing set.
brightness_augmentation (x_train, y_train)	Produces a new set of images whose brightness ranges from 0.85 to 1.15 compared to the original images. This new set of images will be added to the original training set.
rotate_augmentation (x_train,y_train)	Rotates the original images clockwise from 0 degree to 15 degree and adds it to the original training set.
gan_augmentation (x_train, y_train, n_samples)	GAN generator will produce 43 * n_samples images and add it to the original training set.
model1(augmentation) model2(augmentation) model3(augmentation)	Trains the model with a specific augmentation method. If the augmentation is empty string, no augmentation method will be applied. The augmentation choices are "", "brightness", "rotate", "gan"
save_data (history, filename)	saves the loss and accuracy for each epoch into filename (a .npy file)
plot(filename)	Reads a file(.npy) and plot the graph.

APPENDIX B
WORKLOAD DISTRIBUTION

We divided our workload evenly based on the strengths of each team member. Each member was responsible for ensuring that a certain portion of the project was completed; however, all members collaborated on all sections as needed.

TABLE IV
WORKLOAD DISTRIBUTION

Name	Responsibility
Gabourel, Kriss	Writing final report, Creating Powerpoint presentation
Zheng, Chao Hui	Coding