Research on Traffic Sign Recognition Method Based on Generating Countermeasures Network

Guoxun Liang*

Chongqing University, Chongqing, China

*Corresponding author: leungkwokfan@cqu.edu.cn

Abstract. Nowadays, intelligent assisted driving technology is developing rapidly. In the field of intelligent assisted driving technology, traffic sign recognition is a crucial area which focuses on identifying input images and outputting corresponding traffic sign types in the images. However, due to the limitations of optical hardware in the image acquisition system or the poor environment in which the image acquisition system operates, the collected images may not be well recognized by classifiers used for traffic sign classification, despite containing traffic sign information. This could be due to low resolution of images, unclear images, and excessive noise. This article presents a modified generative adversarial network for reconstructing high-resolution traffic signs, specifically suited for the reconstruction of traffic sign images. The generated images and interpolated images are then sent into a convolutional neural network, which is a classifier, to complete the task of classifying traffic signs. The accuracy of the two sets of data is compared, and it is concluded that the classification accuracy of images reconstructed by the generative adversarial network is generally 15% higher than that of images processed by the interpolation method. This study not only proves that the low resolution of images containing traffic sign information can negatively affect subsequent traffic sign classification tasks, but also demonstrates that traffic sign recognition methods based on generating adversarial networks can indeed improve classification accuracy.

Keywords: Generative adversarial network; Traffic sign recognition; Convolutional neural network.

1. Introduction

With the continuous advancement of intelligent assisted driving technology, traffic sign recognition has become an important research topic. The goal of traffic sign recognition is to identify a traffic sign in a specific frame of an input video or image and provide its corresponding sign type by learning relevant data. By utilizing traffic sign recognition methods, drivers can be alerted to identified traffic sign information, ultimately reducing safety hazards where drivers may overlook important signs while focusing on the road.

Throughout the development process of traffic sign recognition methods, numerous constructive techniques have been proposed, and various classifiers have been applied to classify input data. Some researchers have also proposed efficient methods for processing traffic sign detection of images sent into classifiers. For example, S. Lafuente Arroyo et al. developed a support vector machines (SVM) approach for classifying the shape of traffic signs [1]. Fatin Zaklouta et al. proposed using K-d trees and Random Forests for achieving traffic sign classification [2]. Jianming Zhang et al. presented a cascaded R-CNN to complete the task of traffic sign detection [3]. However, most of the above methods focus on preprocessing images such as noise removal or improving and strengthening the classifier network structure. In real-life situations, due to technical limitations of optical hardware devices and environmental constraints of collecting images, some images with low resolution and missing image information may be obtained during the image collection process of traffic signs, which could negatively impact the traffic sign recognition task. Due to equipment and environmental constraints, low-resolution and unclear images collected still contain certain information. If all low-resolution images are discarded, valuable information might not be fully utilized. In actual traffic sign recognition systems, this could lead to drivers not receiving timely warnings, thereby increasing the risk of traffic accidents or illegal driving. Therefore, optimizing low-resolution images is necessary for reliable traffic sign recognition.The most thorough way to obtain clearer traffic sign images is to improve the technology of optical hardware systems. However, due to the difficulty in significantly improving the manufacturing process of optical hardware in a short period of time and the high cost of improvement, for some low-resolution images that affect the accuracy of traffic sign recognition tasks, using image super-resolution reconstruction technology to improve its resolution to improve the accuracy of traffic sign recognition tasks is a more suitable and cost-effective method. Therefore, this article mainly discusses the use of generative adversarial networks to preprocess some low-resolution images to improve the accuracy of traffic sign recognition.

2. Introduction to SRGAN Principle

Since the concept of generative adversarial neural networks (GAN) was proposed by Ian J. Goodfellow et al. in 2014 [4], GAN has been widely improved and applied. A typical GAN model includes a generator and a discriminator, fitting the corresponding loss function. Through the continuous dynamic game between the generator and the discriminator, i.e., the generator strives to generate data to deceive the discriminator, while the discriminator tries to distinguish the images generated by the generator. This approach can effectively complete many unsupervised learning tasks.In September 2016, Christian Ledig et al. proposed the Super-Resolution Generic Advisory Network (SRGAN) [5], which introduced residual blocks into GAN to improve the resolution of images. The effect was much better than the previous Super-Resolution Revolutionary Neural Network (SRCNN) [6]. Later, in 2018, Xintao Wang et al. proposed the Enhanced Super-Resolution Generated Adversarian Network (ESRGAN) based on SRGAN [7]. This model uses Residual in Residual Dense Block (RRDB) instead of Residual Block and removes all Batch Normalization layers. Wang et al. also improved the discriminator and perceptual loss function. Subsequently, Real ESRGAN was proposed by Xintao Wang et al. [8], which made some improvements to ESRGAN, including the use of sinc to improve the effect.

2.1. Generator

The GAN generator used in this article is an improvement on the original SRGAN generator. In his paper proposing ESRGAN, Xintao Wang et al. pointed out that even though the purpose of adding Batch Normalization to SRGAN is to train deeper networks to improve the ability of network fitting mapping, Batch Normalization can lead to a waste of more computing resources and generate some unpleasant artifacts, limiting the generalization ability of the model. Therefore, the Residual Block in the Generator in this article removes Batch Normalization.

The model used in this article uses a neural network library that can run on TensorFlow, namely Keras. Using this open-source library, the generator first includes an input layer, and then is connected to a convolutional layer representing the previous residual block. Its activation function is ReLU, or Rectified Linear Unit. After this, the generator includes 20 residual blocks, among which each residual block contains a convolution layer with the activation function of Leaky_ReLU and a normal convolutional layer of ReLU. To prevent network degradation, a shortcut has been added. After 20 residual blocks, there is a convolution layer representing the residual blocks, whose activation function is also ReLU, and connected to a batch layer. After that, there are two down sampling blocks, each implemented using the UpSampling2D function and a convolutional layer with an activation function of ReLU. Finally, it is connected to a convolutional layer with an activation function of Tanh and output.

2.2. Discriminator

The discriminator mentioned in this article is based on SRGAN. Similarly, Discriminator is built from components and functions of the open source library Keras. The discriminator first receives the shape of the input image and inputs it to a convolution layer with the activation function of Leaky_ReLU. After this, the data will go through seven consecutive sets of "convolutional layer-batch normalization layer" combinations, where the activation function of each convolutional layer is Leaky_ ReLU, the parameter alaph is set to 0.2. The data will then be transmitted to the fully connected layer with the activation function of sigmoid, and finally will be output.

2.3. Loss Function

Unlike SRCNN, which used Mean Square Error (MSE) as a loss function to solve supersession problems previously, SRGAN uses a perceptual loss function as its loss function during training. Christian Ledig et al. pointed out that using MSE as a loss function can lead to difficulties in recovering lost high-frequency details and reconstructing the texture of original high-resolution images. Therefore, the perceptual loss function is used in SRGAN.

The perceptual loss function includes two parts, namely, content loss and adversarial loss.

First, the calculation of content loss is based on the pre training model of Visual Geometry Group Network 19 (VGG 19), which includes 19 hidden layers (16 convolutional layers and 3 fully connected layers). Since training VGG 19 requires a lot of computing resources and memory, in order to save resources and time, the pre trained model can also be referenced through the open source library Keras. The content loss is obtained by using the features of the active layer ReLu to obtain high-resolution images and generated images through the VGG19 specific layer feature map, and calculating the Euclidean distance.

Secondly, as in the classic generative adversarial network, adversarial loss is used to calculate the probability that the discriminator will determine that the image generated by the generator is a true high-resolution image.

By weighting these two components, the loss function used in SRGAN is obtained.

$$l^{SR} = l^{SR}_{VGG} + 10^{-3} l^{SR}_{Gen} \tag{1}$$

2.4. Effect Analysis

The images processed by above SRGAN can achieve relatively good results. The following Fig.1 show a comparison of images reconstructed from 16 * 16 pixels to 64 * 64 pixels. The left is a low resolution image, the middle is an original high resolution image, and the right is an image generated by a trained SRGAN.

It can be seen that the images processed by SRGAN are closer to the original high-resolution images, achieving good results. This is of great significance for improving the resolution of low-resolution images containing traffic sign information described in this article to improve recognition and classification effects.

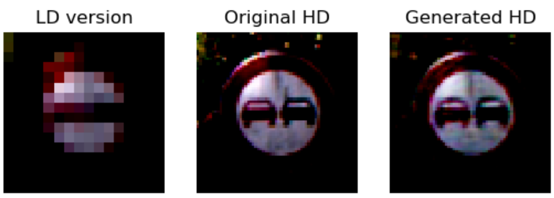

Fig. 1 Two or more references

3. Experiment and Analysis

3.1. Data Sset

This article uses the German Traffic Sign Recognition Benchmark (GTSRB) as a dataset for training and testing [9]. This dataset consists of three parts. The first part is Meta, which includes the best image data for 43 classification results. The second part is Train, which includes 43 types of data with different resolutions that have been classified based on label results. The third part is the Test, which includes 12,630 pieces of unclassified data with different resolutions. In addition, the dataset also includes three csv files for labeling images. This dataset is from the Kaggle website.

3.2. Classifier

Similar to Generator and Discriminator, the classifier used in this article is built using the open-source neural network library Keras. In this paper, a convolutional neural network is used as a classifier to obtain image labels.

Since its inception, convolutional neural networks have been used in many machine learning tasks, including image classification tasks, due to their excellent ability to correct bias terms and weights and significantly reduce the computational resources consumed compared to traditional neural networks.

In fact, many classifiers have been proposed for traffic sign recognition tasks, and they are not limited to neural convolutional networks. However, this article mainly discusses the improvement of classification performance by using SRGAN to reconstruct low resolution images containing traffic sign information. Therefore, selecting neural convolutional networks is suitable. The convolutional neural network mentioned in this article mainly refers to the CNN model proposed by Aashrith Vennelakanti et al. [10] (Table 1).

The classifier first contains two convolutional layers with 32 filters of size 3x3, and then connects a pooling layer to speed up processing. Then, in order to prevent overfitting, a dropout layer is required. After that, connect two convolutional layers with 64 filters of size 3x3 and a pooling layer, and similarly, connect a dropout layer again. The results will be sent to the full connected layer using a flatten layer to flatten data. After passing through the dropout layer again, the data will be sent to the fully connected layer with the activation function of sigmoid, and the final output will be the result. In this convolutional neural network, except for the last convolutional layer, the activation function of all other convolutional layers is ReLU.

Table 1. Layers of cnn

Layers
Convolutional Layer (32 filters of size 3x3, ReLU)
Convolutional Layer (32 filters of size 3x3, ReLU)
MaxPool2D (2x2) + Dropout (0.2)
Convolutional Layer (64 filters of size 3x3, ReLU)
Convolutional Layer (64 filters of size 3x3, ReLU)
MaxPool2D (2x2) + Dropout (0.2)
Flatten ()
Fully connected (256 nodes, ReLU) + Dropout (0.5)
Fully connected (43 nodes, softmax)

In addition, the loss function of this convolutional neural network is Categorical CrossEntropy Loss and the optimizer is Adam.

3.3. Comparison and Analysis

In the partitioning process of the GTSRB dataset, the data in Train will be divided into a training set and a verification set at a 9:1 ratio, that is, the training set contains 35,288 samples and the verification set contains 3,921 samples. All 12,630 samples in the Test will be used for testing.

In general, for the size of the collected image containing traffic sign information, 30 * 30 pixels should provide sufficient image information for the classifier to perform more accurate and reasonable classification. However, in some cases, due to hardware and environmental constraints, the size of the collected image containing traffic sign information may be smaller than the above size. This is the case discussed in this article and the case on which the experiment is based.

In order to simulate the low resolution, unclear, and traffic sign information images that may be collected when collecting traffic signs, all samples in the test set will first be resized to a size of 16 * 16 pixels. After adjustment, the test set will be copied into two identical copies, one for comparison, and one will be input to SRGAN to improve resolution. The comparison group will use the functions in the open-source library Scipy to adjust the size. The adjusted size is 64 * 64 pixels. The other group will be fed into the SRGAN mentioned above, and will generate 64 * 64 pixel generated images using the trained model. Finally, two sets of data sets containing 12,630 images will be fed into the aforementioned neural convolutional network classifier trained with the data contained in Train in GTSRB, and the prediction accuracy will be calculated and compared based on the label information in Test.csv.

For example, the results of using the functions in the open-source library Scipy to resize and using SRGAN to generate images are compared as follows (Figure 2).

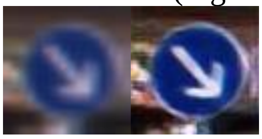

Fig. 2 Image processed by Scipy and Image processed by SRGAN

Use the classifier mentioned in 3.2 and Train in GTSRB for training. After Epoch is greater than 15, the accuracy of this convolutional neural network in the training set can reach more than 90%, and the accuracy in the verification set can reach 97%, indicating that this model has a certain classification ability. Therefore, the accuracy of the model in the training set is greater than 90%, and the accuracy of the model in the verification set is greater than 95%. As a result, the threshold for training to obtain a model that can be used to predict two sets of test data is used to avoid the impact of errors generated during training on the accuracy of subsequent analysis and evaluation of two sets of test data.

This experiment was conducted 10 times, and 10 models that meet the above threshold were obtained. Input the control group data directly using the Scipy library for size adjustment, that is, using the interpolation method for size adjustment, and the data using the trained SRGAN to generate target resolution into the obtained classifier model, respectively. The results are shown in the following table 2.

Table 2. Results

Round	Interpolated Images	Generated Images
1	0.729770387	0.878226445
2	0.729770388	0.875059382
3	0.642121932	0.866429137
4	0.715439430	0.870863025
5	0.666587490	0.880601742
6	0.682343626	0.873555028
7	0.747505938	0.870704671
8	0.767300071	0.902692003
9	0.615914489	0.873396675
10	0.578543151	0.895091053

From the results shown in the above table, it can be concluded that in the task of traffic sign recognition, images with low resolution, unclear images, but containing certain traffic icon information, may indeed not be well recognized and classified into appropriate labels due to low resolution. A model with an accuracy of over 90% in both the training set and the verification set has a recognition accuracy of only about 57% to 73% for low resolution images. This indicates that the information in these low-resolution images is partially recognized, but does not achieve good results. In reality, unclear images obtained due to equipment limitations or excessive distance may lead to the neglect of these traffic signs, increasing the risk of driving accidents and violations.

On the other hand, the performance of the test set data processed by SRGAN is commendable, with recognition accuracy rates exceeding 85%. This indicates that through SRGAN processing, these images with low resolution, containing noise, but also containing traffic sign information can make the information contained for classification of traffic signs more prominent, thereby enabling the classifier, namely convolutional neural network, to better recognize and classify these data.

4. Conclusion

This article explores the problems of low resolution and unclear image data collected in the task of traffic sign recognition in intelligent assisted driving and the implementation and analysis of solutions to process these images with traffic sign information using SRGAN. By establishing an improved SRGAN and using it to process blurred images with traffic sign information, compared with images processed by image interpolation to improve resolution, it is shown that using SRGAN to generate high-resolution images can improve the classification accuracy of classification methods using convolutional neural network models as classifiers. It has been preliminarily proved that the use of generative adversarial networks can indeed improve the effectiveness of traffic sign recognition tasks.

On the other hand, in intelligent assisted driving systems, in addition to traffic sign recognition tasks, traffic sign detection tasks are also very important. For example, in the detection of traffic sign images in dark environments, a good traffic sign detection method can effectively improve the accuracy of subsequent recognition.

The integration of various methods of traffic sign detection and recognition can better improve the accuracy and smoothness of the task of detecting and recognizing traffic signs in intelligent assisted driving. The traffic sign recognition method based on SRGAN will also make a certain contribution to this goal.

References

[1] Lafuente-Arroyo S, Gil-Jimenez P, Maldonado-Bascon R, et al. Traffic sign shape classification evaluation I: SVM using distance to borders[C]//IEEE Proceedings. Intelligent Vehicles Symposium, 2005. IEEE, 2005: 557-562.

[2] Zaklouta F, Stanciulescu B, Hamdoun O. Traffic sign classification using kd trees and random forests[C]//The 2011 international joint conference on neural networks. IEEE, 2011: 2151-2155.

[3] Zhang J, Xie Z, Sun J, et al. A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection[J]. IEEE access, 2020, 8: 29742-29754.

[4] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[5] Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4681-4690.

[6] Dong C, Loy C C, He K, et al. Image super-resolution using deep convolutional networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 38(2): 295-307.

[7] Wang X, Yu K, Wu S, et al. Esrgan: Enhanced super-resolution generative adversarial networks[C]//Proceedings of the European conference on computer vision (ECCV) workshops. 2018: 0-0.

[8] Wang X, Xie L, Dong C, et al. Real-esrgan: Training real-world blind super-resolution with pure synthetic data[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 1905-1914.

[9] Stallkamp J, Schlipsing M, Salmen J, et al. The German traffic sign recognition benchmark: a multi-class classification competition[C]//The 2011 international joint conference on neural networks. IEEE, 2011: 1453-1460.

[10] Vennelakanti A, Shreya S, Rajendran R, et al. Traffic sign detection and recognition using a CNN ensemble[C]//2019 IEEE international conference on consumer electronics (ICCE). IEEE, 2019: 1-4.