Research on GAN-based Container Code Images Generation Method

Yan LIANG
School of Computer Science and Technology
Wuhan University of Technology
Wuhan, China
e-mail: 1357936009@qq.com

Hanbing YAO
School of Computer Science and Technology
Wuhan University of Technology
Wuhan, China
e-mail: 22876681@qq.com

***Abstract*—Recognizing images based on deep learning algorithms requires sufficient samples as a training dataset. In the port field, there is also a lack of container image datasets for deep learning research. This paper proposes a model based on GAN's container box character sample extended dataset (C-SAGAN), and addresses the problems of container box code character defaced and corrupt caused by the port environment, the generative adversarial network is trained with a small amount of real images to generate container character samples. The C-SAGAN model introduces class tags and self-attention in the generator and discriminator. The class tags can control the image generation process. The self-attention mechanism can extract image features based on global information and generate image samples with clear details. The experimental results show that the quality of the samples generated by the generative adversarial network model proposed in this paper is excellent. The samples are used in the CRNN model as the training dataset and the real images are used as the test sets, won the high recognition rate.**

Keywords-image generation; generative adversarial networks; self-attention mechanism; container code identification

I. INTRODUCTION

As the unique identification of a container, the container code is the basic information of the container and plays an extremely important role in container transportation and management. Port container handling efficiency has been increasing year by year, and container automatic management technology is also rapidly developing. With the rise of deep learning, people have tried to identify box code with deep learning methods. For example, Yang Mei[1] improved the structure on the basis of LeNet5 network, and designed two types of convolutional neural networks, one network is used for letter recognition, another is used for digital recognition; Huang et al[2], further combined the convolutional neural network algorithm with the template matching algorithm for character recognition.

In the existing container number recognition research based on deep learning, the container image dataset used by the recognition model is small, and the trained recognition model cannot guarantee that the recognition rate obtained for all test sets can be achieved.

In recent years, there have been many developments in the field of image generation. Among the more influential models in generation are Variational Auto Encoder (VAE)[3], Pixel Convolutional Neural Networks (PixelCNN)[4], Generative Adversarial Network (GAN)[5].

VAE can directly compare the difference between the reconstructed picture and the original picture through encoding and decoding, but because it does not use an adversarial network, it tends to produce blurred images; PixelCNN uses likelihood modeling for images, and the training of this method will be more stable, because the image is generated pixel by pixel only based on the above information, the entire training is slow and the quality of image sampling is not as good as GAN. GAN can generate clear and similar images based on a small number of data sets, and the problems of GANs do not limit the development of GANs, the research on continuous improvement of GANs is endless.

This paper combines Conditional Generative Adversarial Network and Self-Attention Generative Adversarial Network to build a Conditional Self-Attention Generative Adversarial Network (C-SAGAN) for container box character image after generating the dataset, it can be used as a training set for identifying the container number model.

II. RELATED TECHNOLOGIES

A. Conditional Generative Adversarial Network

The original GAN has problems such as unstable training, model collapse, and excessive training. Conditional generative adversarial network[6] adds some additional information to the generator and discriminator of the GAN model as the condition c, and use c to guide the direction of data generation. This condition c can be the class of the image, the attributes of the object or the text description of the image to be embedded, or even a picture. Figure 1 is the CGAN structure. The objective function of CGAN is shown in formula 1.

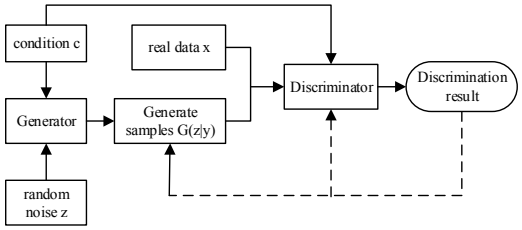


Figure 1. The structure of CGAN

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x|c)] + E_{z\sim P_z(z)}\left[\log\left(1 - D\left(G(z|c)\right)\right)\right] \quad (1)$$

B. Self-Attention Generative Adversarial Network

Self-Attention GAN introduces self-Attention Mechanism[7], which is widely used in Natural Language Processing, into the generation adversarial network to make the generator and discriminator can effectively model the relationship between widely separated spatial regions.

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00059

Attention mechanism[8] can be described as a mapping of a query to a series of key-value pairs. When calculating the attention value, the query and each key are similarly calculated to obtain weights, then a softmax function is used to normalize these weights, finally, the weights and corresponding key are weighted and summed to obtain the final attention value. The self-attention mechanism is a special case of the attention mechanism, that is, key = value = query. In image generation, it can handle long-range, multi-level dependencies in the image. The details are well coordinated, and the discriminator can more accurately implement complex geometric constraints on the global image structure. The self-attention mechanism is shown in Figure 2:

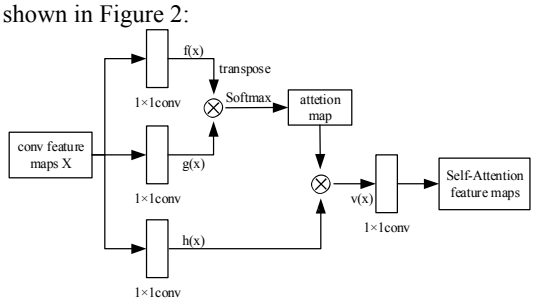


Figure 2. The structure of The self-attention mechanism

Conventional convolutional GAN networks rely on convolutional networks to simulate the dependencies between regions of different images. Convolution operations are based on local receptive fields and can only process local information. Only through multiple convolutional layers can process the long distance dependencies. The self-attention mechanism considers global information at each layer of the network. Compared with the fully connected layer, the global information in SAGAN does not have such a large amount of parameters, and a better balance is obtained between increasing the receptive field and reducing the amount of parameters.

Aiming at the problems of GAN mode collapse and training non-convergence, in addition to adding batch normalization(BN) [9] layer to the generator, SAGAN also applies spectral normalization(SN)[10] to image generation process. SN allows the network parameters of each layer of network to be divided by the spectral norm of the parameter matrix of this layer to satisfy Lipschitz constraints. Lipschitz constraints limit the severity of the function change, the gradient of the function, which can reduce the amount of training calculation and make the training more stable.

III. CONTAINER BOX CHARACTER IMAGE'S GENERATING METHOD

After an on-site inspection of a port in Chongqing, it was found that the background of the container box is more complex, and the position and order of characters are not fixed. In addition, factors such as dust, humid air, rain and fog in the port environment will cause certain damage to the container, the box code characters will be stained and corroded. When collecting container box character image samples, images with large defacement that make characters difficult to distinguish cannot be used as data samples, while small defaced parts do not affect clear recognition of clear character images and can be included in the training dataset. But in the end, there are not many images that can be used as a training dataset in the port, and if the dataset is used to recognize the model, it cannot train a high-precision recognition effect.

For the training data set used for deep learning research, the character part of the container number should be complete and clear. The self-attention mechanism can learn the global characteristics of the image, and it can perform image generation for the entire box number character image. The label can specify the image category generated in image generation.

A. C-SAGAN model structure

The C-SAGAN model adds the label c and the self-attention mechanism to the generator and discriminator, controls the image generation direction during the generation process, and generates an image sample with clear overall box characters.

The generator model consists of transposed convolution, BN, spectral normalization and ReLU activation function[11]. The generator structure of the C-SAGAN model is shown in Figure 3. In deep learning, batch normalization solves to some extent the gradient dispersion phenomenon caused by the random gradient decline of deep neural networks during training, which improves the training speed and accuracy of the model.

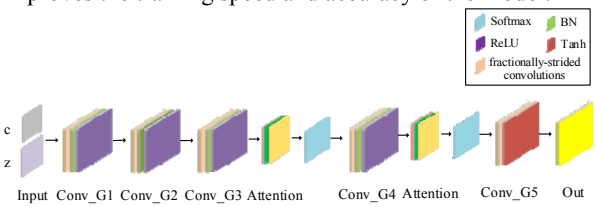


Figure 3. Generator model

The 92-dimensional random noise z is connected with the 36-dimensional label c as input, the first three layers in the generator are similar, including a deconvolution layer, a batch normalization layer, a spectral normalization layer, and a ReLU activation function. Outputs a tensor of (64, 128, 16, 16) after three layers, then enter the self-attention layer, the input channel C is 128, and the process structure is shown in Figure 2. A correlation matrix (64, 256, 256) is obtained, the correlation between the pixels of the feature map, and then softmax regression after normalization, an Attention matrix is obtained, and then multiplied by a 1 × 1 convolution layer to obtain 256 new pixels as the output O. Take a parameter γ, γ gradually increases from 0, multiply it with the output O and then add the feature map obtained before to form the final output of self-attention. After that, a layer4 and a self-attention layer similar to the first three layers are passed, the input channel C is 64, and the correlation matrix structure is (64, 1024, 1024). The tensor of (64, 3, 64, 64)

is output after the last layer of convolution.

TABLE I. GENERATOR CONVOLUTION LAYER PARAMETER

convolutional layer	Size	feature dimension	stride
Conv_G1	4×4	512	1
Conv_G2	4×4	256	2
Conv_G3	4×4	128	2
Conv_G4	4×4	64	2
Conv_G5	4×4	3	2

The discriminator structure of the C-SAGAN model is shown in Figure 4.

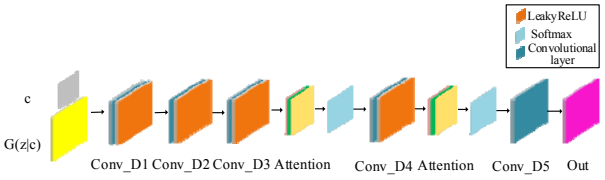


Figure 4. Discriminator model

The discriminator model consists of convolution layers, spectral normalization and LeakyReLU activation function[12]. The output of the generator is connected to the label c as the input of the discriminator. The first three layers of the discriminator are similar, outputs a tensor of (64, 256, 8, 8), and then enter the self-attention layer, the attention matrix is (64, 64, 64), and the output size is unchanged. Layer4 has the similar structure of the first three layers, output the tensor of (64, 512, 4, 5); then enter the self-attention layer, the attention matrix is (64, 16, 16). After the last convolution layer, output (64, 1, 1, 1) Tensor.

TABLE II. DISCRIMINATOR CONVOLUTION LAYER PARAMETER

convolutional layer	Size	feature dimension	stride
Conv_D1	4×4	64	2
Conv_D2	4×4	128	2
Conv_D3	4×4	256	2
Conv_D4	4×4	512	2
Conv_D5	4×4	1	1

Both the generator and discriminator used hinge loss function as the optimization loss, and Adam optimizer was selected to optimize the loss function. The loss of the generator is the average value that the discriminator discriminates against the generated samples; the loss of the discriminator is that the discrimination result is subtracted from "1", less than 0 is taken as 0, and the average value is finally taken. The discrimination results include the discrimination results of the real samples and the discrimination results of the generated samples, respectively. The model uses BN to prevent the gradient from disappearing or exploding, and uses spectral normalization to stabilize the training process.

IV. EXPERIMENT AND ANALYSIS

This experiment is based on Windows Server 2008 R2 operating system, using Pytorch 1.3 deep learning framework, python 3.6. The experimental environment is a processor Intel Xeon Silver 4110 CPU @ 2.10GHz, a graphics card NVIDIA GeForce GTX 1080 Ti, and a memory of 32GB.

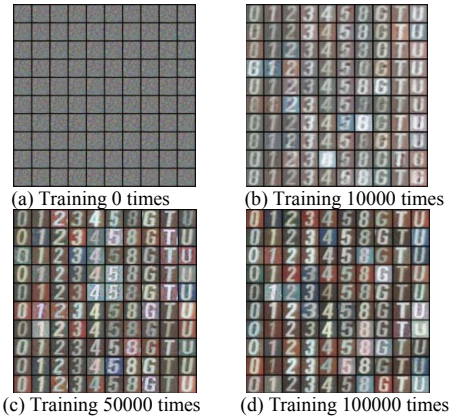
The data used in the experiment contained 1118 container box code images. The color of the container box was not uniform, but the color of the box number font was white.

Figure 5. Container box image examples

In the experiment, an image that can clearly identify the box code font is selected, and boxes of different colors are also selected, as shown in Figure 5. The container box code contains 36 characters from 0-9 and A-Z. The container box code is encoded by international standards and consists of 4 letters and 7 digits, different companies have different arrangement and order of the box code characters of containers. Box numbers include horizontal rows, multiple rows, vertical rows, and multiple rows of vertical rows. However, no matter how the container box numbers are sorted, the color of the box number font still has a clear contrast with the surroundings. In the experiment, the image characters are segmented first, and 36 types of characters are taken out to generate corresponding datasets, and a total of 12,298 characters are obtained.

A. Box number character generation sample

Use the structures of Figures 3 and 4 as generator and discriminator for generative adversarial networks to train the container box character dataset. Before training, the character image size is adjusted to uniform pixels. Figure 6 shows some experimental results, which are the results immediately after training, 10,000 times, 50,000 times, and 100,000 times.

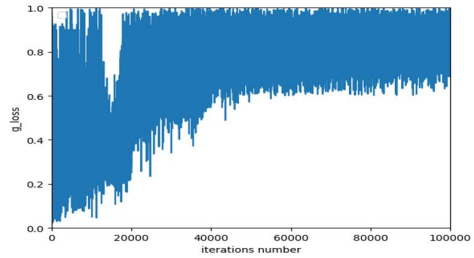


(a) Training 0 times (b) Training 10000 times

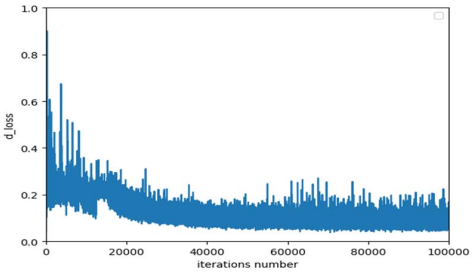
(c) Training 50000 times (d) Training 100000 times

Figure 6. Generate image samples

It can be seen that different background colors are generated in the same type of character image, but the font color is white, which is obviously different from the background color. With the increase of the number of iterations, the generated samples are also constantly changing. By 100,000 training, the generated characters can be clearly distinguished.



(a) Trend of generator loss function



(b) Trend of discriminator loss function

Figure 7. Change trend of loss function

Figure 7(a) shows the change of the loss function of the samples generated by the generator as the number of training increases. Figure 7(b) shows the change of the discriminator discriminating the loss function of image samples with the increase of training times. It can be seen from Figures 7 that with the increase of the number of trainings, the loss function of the generator gradually increases and the loss function of the discriminator gradually decreases under the general trend. Throughout the confrontation process, the model gradually stabilized.

B. Container number identification

A total of 28,800 character image samples and 1,118 real image samples of 36 characters were used as training datasets, and a convolutional recurrent neural network[13] (CRNN) was used to train the recognition model. Convolutional recurrent neural network is a natural scene text recognition model, natural scene text recognition does not require binarization of images, it can perform text recognition on images with different backgrounds[14][15], while CRNN model is the model that can get the best effect in the current end-to-end text recognition. Take 5500 character images of 500 real image samples as the test set. In order to test whether the recognition model maintains a similar recognition rate for different test sets, the test set is divided into 10 small test sets for recognition, and finally the similarity is obtained, the recognition rate is 98.2%. Compared with the use of convolutional neural networks in literature[1] to obtain a recognition accuracy of 94.3%, and the combination of convolutional neural network algorithm and template matching algorithm in literature[2] to obtain a recognition accuracy of 95%, the recognition Accuracy has improved.

V. CONCLUSION

There have been many researches on container box number recognition technology, but in the actual scene, the image data of the container box number of the port is not enough for the training of the recognition model to obtain more accurate results. Based on generative adversarial networks, this paper proposes an image generation model combining conditional adversarial networks and self-attention adversarial networks to generate container box code character images for training CRNN recognition models to recognize box code characters. The next step will be an in-depth study on the distinction between similar characters.

REFERENCES

[1]. Yang Mei, “Box code recognition technology based on convolutional neural network”, Shanghai Jiao Tong University, 2014.

[2]. Huang Shengguang, Weng Maonan, Shi Yu, etc. “Container Code Identification Based on Computer Vision”, Port Operation, vol 238, pp. 1-4, January 2018.

[3]. Kingma, Diederik P, Max Welling, “Auto-Encoding Variational Bayes”, CoRR, abs/1312.6114, 2013.

[4]. Oord, Aäron van den, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, Alex Graves, “Conditional Image Generation with PixelCNN Decoders”, NIPS, 2016.

[5]. Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C, Courville, Yoshua Bengio, “Generative Adversarial Nets”, NIPS, 2014.

[6]. Mirza Mehdi, Simon Osindero, “Conditional Generative Adversarial Nets”, Computer Research Repository, abs/1411.1784, 2014.

[7]. Zhang, Han, Ian J. Goodfellow, Dimitris N. Metaxas, Augustus Odena, “Self-Attention Generative Adversarial Networks”, ICML, 2018.

[8]. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, “Attention is All you Need”, NIPS, 2017.

[9]. Ioffe, Sergey, Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ArXiv, abs/1502.03167, 2015.

[10]. Miyato, Takeru, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, “Spectral Normalization for Generative Adversarial Networks”, ArXiv, abs/1802.05957, 2018.

[11]. Nair Vinod, Geoffrey E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines”, ICML, 2010.

[12]. Maas, Andrew L, “Rectifier Nonlinearities Improve Neural Network Acoustic Models”, In ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[13]. Shi, Baoguang, Xiang Bai, Cong Yao< “An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 39, pp. 2298-2304, November, 2017.

[14]. Wang Runmin, Sang Nong, Ding Ding, Chen Jie, Ye Qixiang, Gao Changxin, Liu Li, “Summary of Text Detection in Natural Scene Images”, ACTA AUTOMATICA SINICA, vol 44, pp. 2113-2141, December, 2018.

[15]. Zhang Heng, “Research on text location and recognition algorithm of natural scene based on convolutional neural network”, Xi'an University of Science and Technology, 2018.