An Edge Intelligence-based Generative Data Augmentation System for IoT Image Recognition Tasks

Wei-Jian Hu[1], Tang-Ying Xie[1], Bao-Shan Li[1], Yong-Xing Du[1], Neal N. Xiong[2]

[1] School of Information Engineering, Inner Mongolia University of Science and Technology, China
[2] Department of Mathematics and Computer Science, Northeastern State University, USA
hwj_llf@hotmail.com, xty_CGC@163.com, libaoshan@imust.edu.cn, dyxql@imust.edu.cn,
xiongnaixue@gmail.com

Abstract

To solve the problem of data scarcity in IoT image recognition tasks, an EI-based generative data augmentation system is designed in this paper. The system adopts hybrid architecture, and edge server and cloud data center participate in computing together, which is logically divided into the training phase and running phase. The training phase completes data augmentation of source data and training of Convolutional Neural Networks (CNNs), while the running phase processes information through the pretrained CNNs, and completes iteration of the CNNs through expert review and self-learning mechanism. It is worth mentioning that a generative data augmentation model, an Effective Deep Convolutional Generative Adversarial Network (E-DCGAN), has been proposed in the system. The experiments show that E-DCGAN is superior to the baseline model in image generation and data augmentation in both agricultural and medical fields. Compared with the baseline model, the FID values were reduced by 4.73% and 19.59%. Meanwhile, the use of E-DCGAN for data augmentation can significantly improve the image classification model (VGG19, AlexNet, ResNet50), and the average accuracy of agricultural and medical classification results has increased by 0.96% and 1.27% over the baseline.

Keywords: Deep Learning (DL), Data scarcity, Edge Intelligence (EI), Generative Adversarial Network (GAN), IoT

1 Introduction

Recent years, with the rapid development of artificial intelligence (AI) technology, many related applications and services have emerged. AI technology has achieved good performance in Computer Vision (CV), Natural Language Processing (NLP), etc. The use of AI technology to analyze information has become the most effective method now, mostly represented by Machine Learning (ML) technology and Deep Learning (DL) technology [1]. However, the computing resources required for traditional AI computing are usually concentrated in the cloud data center, which makes it difficult for the IoT system to generate many real-time data from edge devices to enjoy the strong support of AI technology, and it is also challenging to ensure that high quality and high-level service. The emergence of Edge Intelligence (EI) technology has successfully solved this problem. EI is a new computing model based on EC. It combines EC and AI and uses AI methods to process data in edge devices. To ensure the efficiency and real-time of data processing, and simultaneously promote the broad application of AI in IoT, and solve “the last mile” problem in AI. At present, IoT has become an important application field of EI, and the use of EI in IoT systems has become a hot research topic, attracting the attention of many scholars from academia and industry [2-4].

The deep Convolutional Neural Networks (CNNs) in DL is a type of feedforward neural network that includes convolution calculation and has a deep structure. It has achieved good results in CV tasks such as image recognition, semantic segmentation, and image generation [5]. In IoT systems that use EI technology, deep CNNs are also widely used, such as in the agricultural field [6], medicine field [7], etc.

Although CNNs has achieved good results in CV tasks, its training relies on many labeled data. Obtaining enough labeled data in actual application scenes requires many workforces and material resources, which makes data scarcity has become one of the main problems that limit CNN’s development. For example, it is challenging to collect image data of various crop diseases in crop disease identification due to multiple crops, broad geographic distribution, and different disease cycles. It is also challenging to collect massively labeled medical image data in the medical field due to personal privacy, disease diversity, and collection equipment differences. Unfortunately, the data scarcity is more prominent in IoT systems that use CNN. This is because the system’s visual information

*Corresponding Author: Yong-Xing Du; E-mail: dyxql@imust.edu.cn
DOI: 10.53106/160792642021072204005

is collected from a specific application scene, and it isn't easy to find the same data information as in the scene. The amount of information is usually difficult to meet the training requirements of CNN. CNN models often have problems such as training difficulties and weak generalization ability. It is difficult for the system to be put into use quickly, and it is also a big project to label enough data.

This article takes the agricultural and medical fields as the starting point. Aiming at data scarcity using CNN in the IoT image recognition tasks, based on EI and Generative Adversarial Network (GAN) technology, designs an EI-based generative data augmentation system. The system is constructed based on EI technology and uses a hybrid structure of "edge-cloud". The edge service and cloud data center jointly complete the DL computing tasks in center phases. The system solves the problem that cannot train the CNN model due to the lack of actual application scene data in the image recognition task of IoT system, avoids the restriction of the environment on the system to a certain extent, and is beneficial to the application of EI technology in the IoT system. At the same time, An Effective Deep Convolutional Generative Adversarial Network (E-DCGAN) is proposed to expand the image data in the system to solve the problems of model over-fitting and insufficient generalization ability caused by data scarcity in the IoT image recognition task. E-DCGAN is based on the Deep Convolutional Generative Adversarial Network (DCGAN) [8] model, the gradient instability problem is solved by changing the model loss function, and the network structure of the generator network and discriminator network is optimized, which can greatly improve the model performance ability, enhance the stability of the network, and reduce the occurrence of mode collapse as far as possible. In the experiment, the agricultural and medical datasets were used to verify the model, which not only verified the model generation effect but also verified the influence of the generated sample data on the classification effect through the classification model. The main contributions of this paper are listed as follows:

(1) Designed an EI-based generative data augmentation system. The system is constructed based on EI technology and uses a hybrid structure of "edge-cloud". The edge service and cloud data center jointly complete the DL computing tasks in phases. The whole system divides into the training phase and running phase, in the training phase, the public dataset close to the application field is used as the source data, and the GAN is used to enhance the source data. The data after the data augmentation is used to train the CNN in the system, and the effect of the model is comprehensively evaluated by the precision, accuracy, recall and F1-Measure; in the running phase, the system adopts the transfer learning method for the classification model obtained in the training phase, and establishes an expert review and self-learning mechanism that during the system running, the expert reviews and proofreads the real data, and iteratively train model to improve the model's effectiveness in actual use.

(2) Proposed a generative data augmentation method: E-DCGAN. Based on DCGAN, E-DCGAN uses the wasserstein distance loss function with the gradient penalty to guide unsupervised learning, which improves the feature extraction ability of the convolutional layer. Moreover, it introduces a Spectral Normalization (SN) layer in the generator and discriminator, respectively. The model's performance is greatly improved, and the dropout layer is introduced into the discriminator network structure to add randomness to avoid getting stuck during the training process (may be at a local minimum value or saddle point).

(3) In the fields of agriculture and medicine, using FID (Fréchet Inception Distance) as the criterion, AI Challenger 2018 crop disease fine-grained classification competition and Chest X-Ray Images datasets are used to verify the image effects generated by the model and compared with the current mainstream generative data augmentation methods. The results show that the E-DCGAN network has better image generation effects.

(4) In the fields of agriculture and medicine, use accuracy, precision, recall, and F1-Measure as the evaluation criteria, use E-DCGAN to generate image samples for data augmentation, and use the VGG19, AlexNet, ResNet50 models to compare the classification effects of different data augmentation methods. The results show that the samples generated by E-DCGAN have better image generation effect.

The remainder of the paper is arranged as follows: In Section 2, EI and IoT, data augmentation, and GAN-based generative data augmentation methods are introduced. In Section 3, an EI-based generative data augmentation system is introduced, and the generative data augmentation method E-DCGAN is explained in detail. In Section 4, extensive experiments are described, and the results are analyzed. In Section 5, we conclude this shortcomings and future work are also proposed.

2 Related Work

2.1 EI and IoT

With the proliferation of mobile computing and IoT, billions of mobile and IoT devices are connected to the Internet. Meanwhile, these devices will generate zillions of bytes of data at the network edge, calling for instant data processing and intelligent data analysis to fully unleash the edge big data's potential. Both traditional cloud computing and on-device computing cannot sufficiently address this problem due to the high latency and limited computation capacity [9]. However, the emerging EC pushes the data processing from the remote network core to the local network edge, remarkably reducing the latency and improving the efficiency. In fact, EI integrates EC and AI, and intelligently processes data at the terminal, enabling fast, efficient, and reliable calculation and decision-making at the edge of the network. And EI is widely used in agriculture, medicine, industry, etc. For example, in agriculture, Castañeda-Miranda et al. [10] developed a reliable system for smart irrigation of greenhouses using artificial neural networks and IoT architecture, using transfer learning to reduce neural networks' processing power for the IoT edge devices. Namani et al. [11] used a Smart Drone for crop management where the real-time Drone data coupled with IoT and Cloud Computing technologies help in building a sustainable Smart Agriculture. In medicine, Khan et al [12] designed An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier, the MDCNN achieves an accuracy of 98.2 which is better than existing classifiers. Young et al. [13] designed wearable medical IoT devices based on DL to automate AI hearing aids. The system has achieved 92% accuracy in sound recognition and classification. In industry, Weimer et al. [14] designed a new industrial inspection architecture based on CNN-based automatic feature extraction with minimal human interaction. Wang et al. [15] proposed a DNN-based architecture to accurately predict the remaining energy and remaining lifetime of batteries, which further enables an informed power configuration among base stations. In other respects, Yan et al. [16] proposed a reinforcement learning-based method to identify critical attack sequences with consideration of physical system behaviors.

2.2 Data Augmentation

Data augmentation is one of the effective means to solve the data scarcity in DL. Currently, there are two methods for data augmentation: non-generative data augmentation and generative data augmentation. Non-generative data augmentation adopts preset rules to increase the number and types of data by geometric transformation and color change [17] of existing data. Commonly used methods include affine transformation [18], noise type, and fuzzy type, etc. The non-generative data augmentation method is simple to implement and can expand the number of datasets to a certain extent. Still, it does not produce substantial changes to the dataset, and the generated samples lack diversity, so non-generative data augmentation generalizes the model. The improvement in capacity is limited. Simultaneously, non-generative data augmentation methods need to adjust the generation method according to the characteristics of the dataset. Generally, the generation method that applies to a specific dataset is challenging to apply to other datasets. For example, on the CIFAR-10 dataset, rotation is a useful data augmentation method, but it does not perform well on the MNIST dataset because the classifier cannot correctly identify the numbers 6 and 9 [19]. The generative data augmentation method learns the distribution that the data obey through the model and randomly generates data consistent with the sample set's distribution. This method can make the dataset cover more patterns, which is more conducive to improving the model's generalization performance and makes up for the shortcomings of the non-generative data augmentation.

The generative model is the critical technology in generative data augmentation. At present, the more commonly used models are Variational Auto-Encoder (VAE) [20], Auto Regressive model (AR) [21], and GAN. Among them, VAE and AR are modeled based on display density, but AR is used to generate images on a pixel by pixel, which resulted in high computational cost and limited parallelism and time-consuming in processing large resolution images. VAE in the image generation can be parallel, but the generated images are fuzzy, lacking expression of complex models. Compared with display density modeling methods such as VAE and AR, GAN based on implicit density modeling avoids problematic inferences and generates high-quality images; due to its ability to fit high-dimensional data distribution and excellent image generation performance, GAN the best method for generative models. GAN is mainly composed of two parts, including the generator and the discriminator [22]. The generator learns mostly the distribution of original data samples to make the data it generates more real, while the discriminator is used to judge the input image data's authenticity. After multiple rounds of zero-sum games, the generator's performance and the discriminator continue to improve, and finally, the two will reach a dynamic equilibrium (Nash equilibrium) [23]. Scholars have done extensive research on data augmentation using GAN, such as Frid-Adar et al. [24] based on GAN synthetic medical images for data augmentation, the sensitivity increased from 78.6% to 85.7%, and the specificity increased from 88.4% to 92.4%. Zheng et al. [25] generated unmarked samples based on DCGAN, achieving a +0.6% increase on a strong baseline (CNN). Wang et al. [26] based on GAN, combined with noise-to-image and image-to-image, for image augmentation of brain MR tumor detection, increasing the sensitivity from 93.67% to 97.48%.

3 Our Proposed System

3.1 System Overview

The system is built on EI technology, which gives AI capabilities to edge services in the IoT system so that the IoT system's visual information can be processed quickly and efficiently. Most importantly, it provides a solution to the problem of data scarcity in IoT image recognition task. In the system, we use the generative data augmentation method based on GAN to enhance the data samples and use the CNN to classify the visual information. However, whether it is a generative data augmentation method or a deep convolution neural network, its model's training and operation require a lot of time and computing resources. It is not appropriate to place these tasks in edge services. Therefore, the system uses a hybrid architecture of "edge-cloud". Edge services and cloud data center work together to complete DL computing tasks in phases. Figure 1 shows the overall structure of the system. The cloud data center, the edge service nodes, and the edge devices make up the whole system, in which edge devices are divided into information collection devices and information display devices. The system uses wireless networks to connect edge devices and edge service nodes and wired networks to connect edge service nodes to the cloud data center. Information collection devices include cameras, unmanned aerial vehicles, and smartphones responsible for collecting visual information while the system is running. Information display devices include personal computers, tablets, and smartphones, which receive and display processed information. The edge server node consists of a GPU-capable server that receives the collected visual information, processes the data, and sends the CNN analysis results. The cloud data center comprises a cluster of servers with GPU-capable and is responsible for tasks that require a lot of computing, such as generative data augmentation, CNN model training, expert review, and self-learning mechanisms. The Cloud Data Center is composed of a cluster of servers with GPU computing and is responsible for tasks that require a lot of computing, such as data augmentation for generation, CNN model training, expert review, and self-learning mechanisms. Ultimately, the cloud data center sends trained CNN models to edge server nodes through transfer learning. The whole system can be logically divided into the training phase and running phase. The training phase describes the training process of the CNN model and data augmentation in the system, and the running phase describes the process of using the CNN model to process visual information.

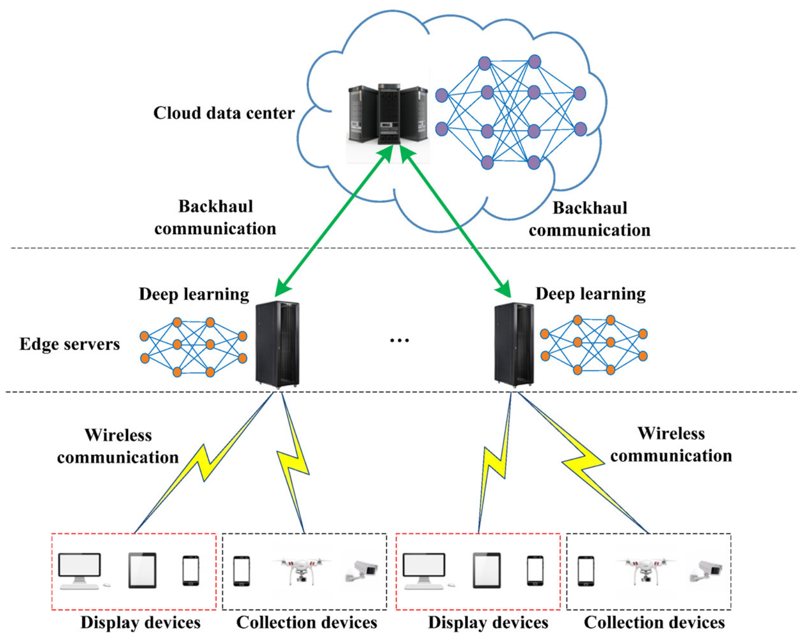

Figure 1. Structure of the system

In the training phase, the open dataset close to the system application field is used as the source data, and the data generated in the actual application field as the target data. The source data train the CNN model, and the CNN model put into operation after the training. Simultaneously, to further improve the training effect of the CNN model in the training process, the generative data augmentation method E-DCGAN (see Section III for the specific structure) enhances the source data. Finally, accuracy, precision, recall, and F1-Measure are used to investigate the model's training effect in many aspects. Figure 2 shows the training process of the system training phase with tomato leaf's disease and pneumonia X-ray as examples. The training phase process divides into three sub steps. First, select the tomato leaf's disease images and chest X-Ray images send them to E-DCGAN; after E-DCGAN data augmentation, the generated images and the real images are mixed as a training set and sent to the CNN model. CNN model consists of VGG19, AlexNet, ResNet50. And judge the performance of the model contains Accuracy, Precision, Recall, F1-Measure. Figure 3 shows the running phase of the system. The running phase process divides into three sub steps. First, the data collected by the visual information collector is preprocessed, and then sent to the classification model for recognition, and the recognition result is pushed to the display devices, such as computer, ipad and smartphone. Second, the preprocessed images and the classification model's recognition results review by domain experts. The experts judge the classification, and give the correct recognition results as the label of the images, and finally these data will be stored in the database. Third, the self-learning mechanism regularly obtains the labeled real data from the database, and uses the real data to continue training the classification model, to improve the performance of the classification model in real tasks. The process of self-learning is the same as that of system training phase. Finally, replace the original model with a new classification model.

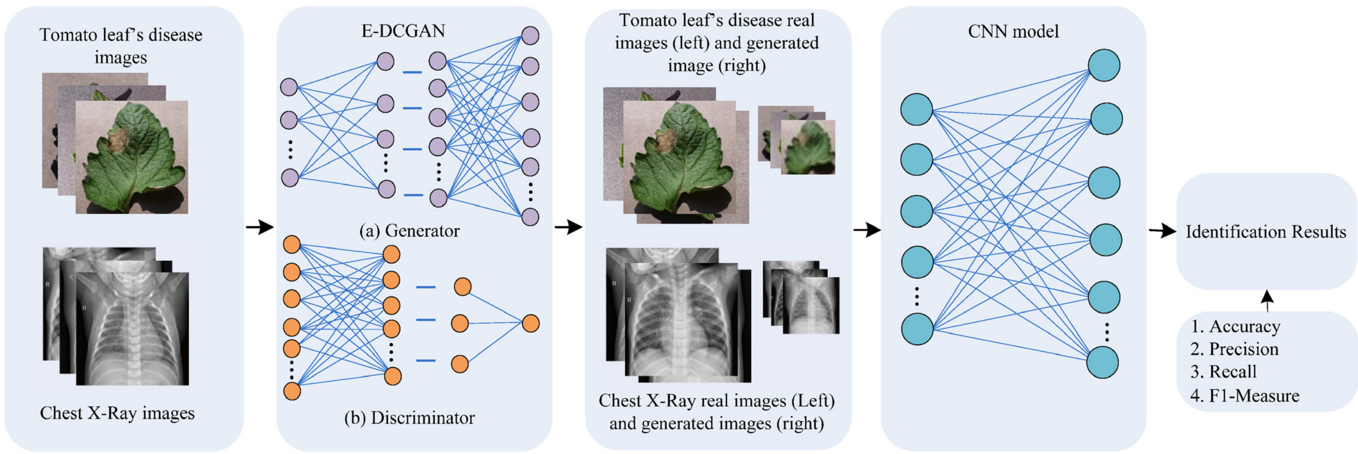

Figure 2. The training phase of the system

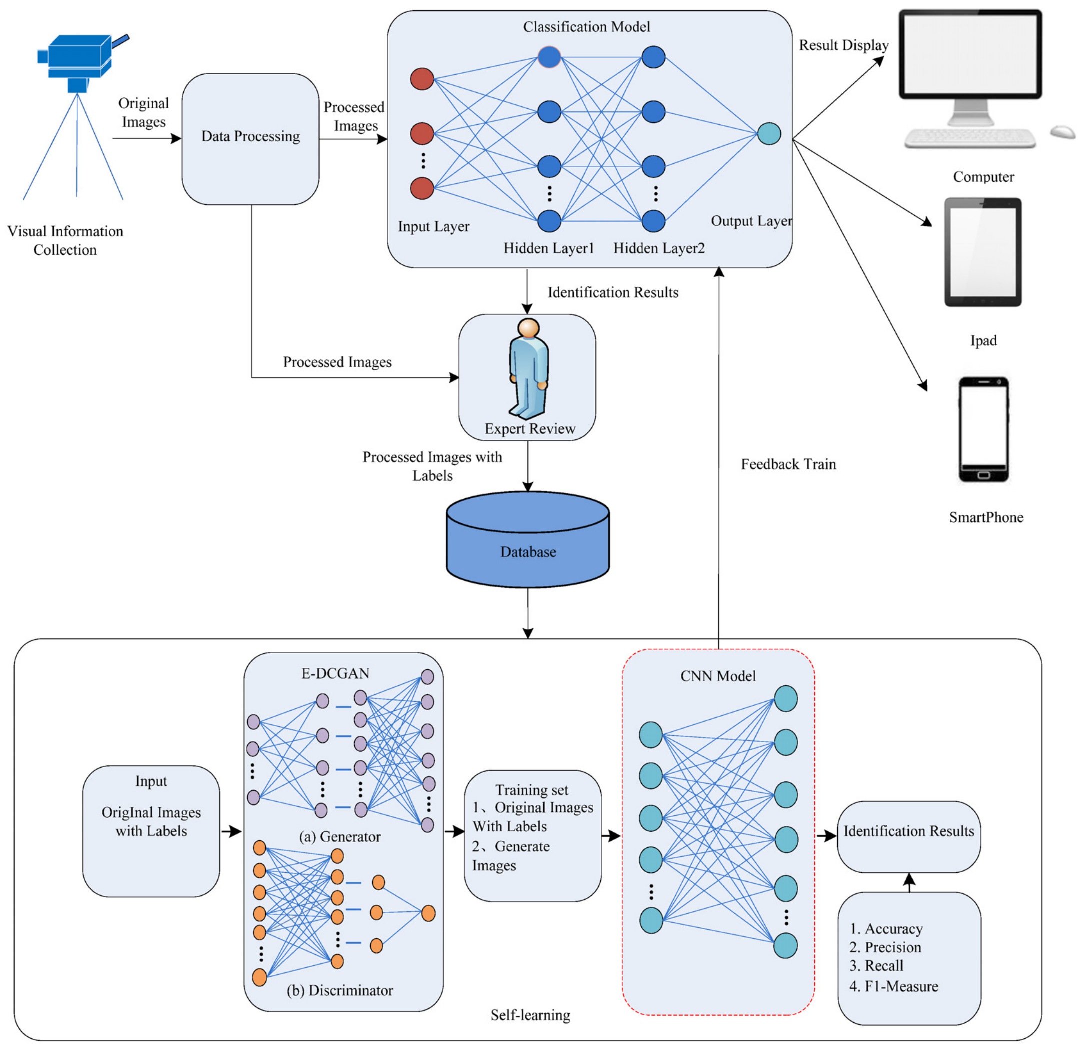

Figure 3. The running phase of the system

3.2 E-DCGAN

E-DCGAN is based on the standard DCGAN and has been improved from three aspects: generator, discriminator, and loss function. The improved model is composed of a generator and a discriminator. The generator is used to learn the distribution of real data to generate images, and the discriminator is used to judge whether the input image is true or false. The overall structure of the E-DCGAN model is shown in Figure 4.

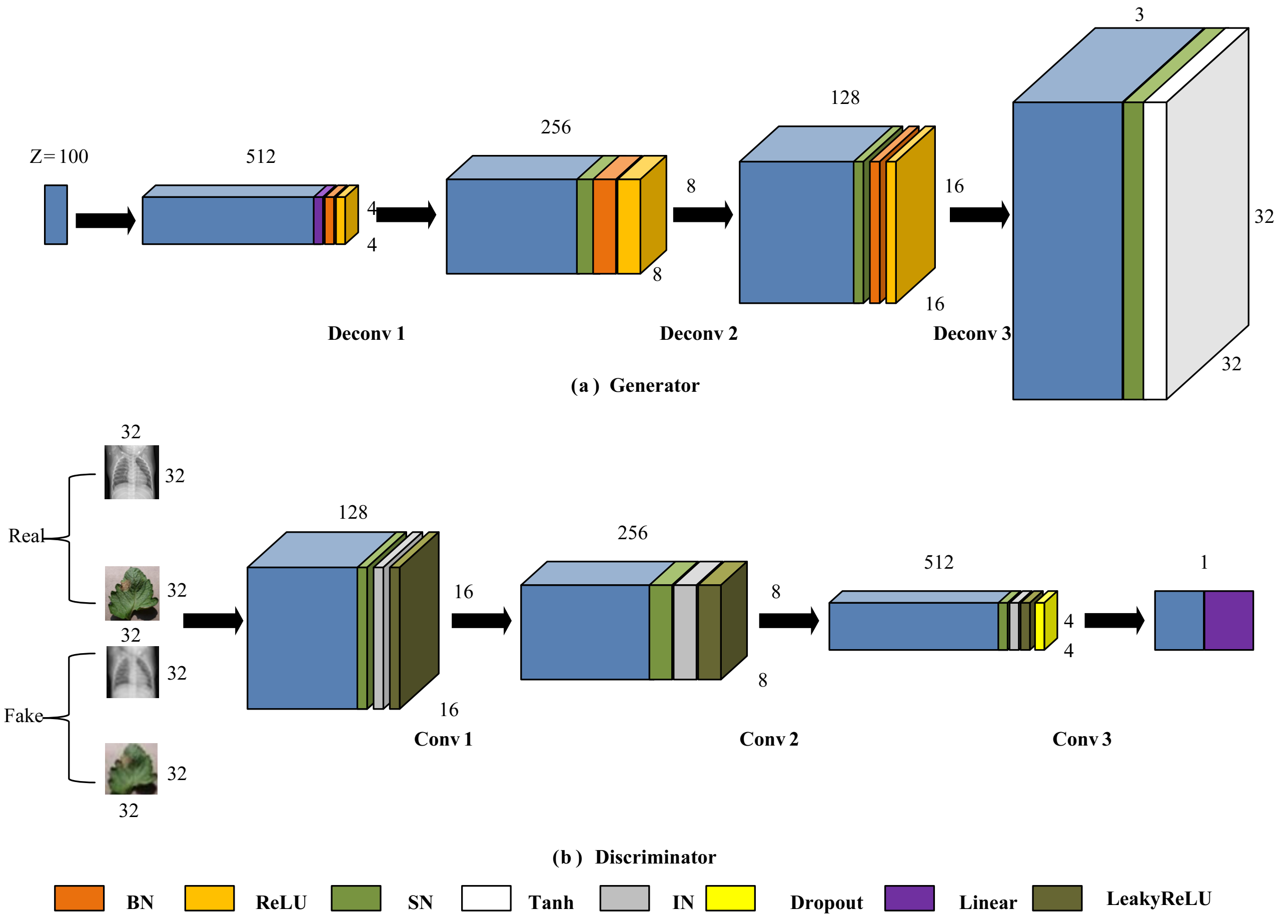

Figure 4. Structure of the network

3.2.1 Generator Network Structure

The generator network architecture shows in Figure 4(a). It is divided into five layers, from left to right are the input layer, the fully connected layer, and the three-layer transposed convolutional layer (Deconv1, Deconv2, Deconv3). The fully connected layer consists of Linear Layer, Batch Normalization (BN), and ReLU activation function; Deconv1 and Deconv2 layers consist of SN [27] and BN. The Deconv3 layer is consists of the SN layer and the Tanh activation function. The size of each convolution kernel of the generator network is 4 × 4, and the convolution step size is [2, 2, 1]. The generator network takes 100-dimensional (z=100) random, customarily distributed noise as input, after the input linear mapped to a tensor with a shape of 4 × 4 × 512, and then it is up-sampling using a transposed convolutional layer to improve the learning rate, speed up the convergence speed and stabilize the training process. Each transposed convolutional layer will perform an SN operation. The output will be batch normalized and processed by the non-linear ReLU activation function as the next layer's input. After the last transposed convolutional layer's output, through the Tanh activation function, a three-channel RGB image with 32 × 32 pixels is finally output.

3.2.2 Discriminator Network Structure

The discriminator network structure shows in Figure 4(b). It is divided into five layers, from left to right are the input layer, the 3-layer convolutional layer (Conv1, Conv2, and Conv3), and the fully connected layer. Conv1 layer and Conv2 layer are consists of SN, instance normalization (IN), LeakyReLU activation function; Conv3 layer is consists of SN, IN, LeakyReLU activation function, and dropout layer; the fully connected layer consists of a linear layer. Each convolution kernel of the discriminator network is 4 × 4, and the convolution step size is [1, 2, 2]. The discriminator network inputs a 32 × 32 × 3 RGB image and then uses the convolutional layer to downsampling it. To more improve the performance and learning ability of the model, each convolutional layer performs SN. At the same time, we retain the IN layer to constrain the output of the samples and convolutional layers to a better range, such as a Gaussian distribution with a mean of 0 and a variance of 1. And the last layer of activation function sigmoid is removed; a layer of dropout is added after the LeakyReLU activation function at the end of the network to increase its randomness and avoid getting stuck during training (may be at a local minimum value or saddle point). Finally, it is passed to the fully connected layer to determine whether the input image is true or false.

3.2.3 Loss Function

The loss function of the original DCGAN is showing in formula (1):

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad \textbf{(1)}$$

G is the generator, D is the discriminator, p_{data} is the real data sample, and p_z is the noise distribution. Formula (1) shows that the solving process of the model is maximizing the discriminator and minimizing the generator. For the discriminator, we maximize $\log(D(x)) + \log(1-D(G(z)))$ the generator; we minimize $\log(1-D(G(z)))$ by gradient descent. Unfortunately, such an alternating gradient rise and descent make the model difficult to learn. It is easy to cause problems such as model collapse and insufficient diversity of generated samples in practical application. In response to the above problems, wasserstein's distance with the gradient penalty term is used to replace the original loss function, which can effectively solve gradient instability and improve the training process's stability.

The updated loss function is shown in formula (2), where L represents the updated loss function, which is consists of the arithmetic sum of the discriminator's loss term L_D and generator's loss term L_G. The discriminator loss term L_D is consists of the arithmetic sum of unsupervised loss term L_{UL} and gradient penalty term L_{GP}, L_{UL} as shown in formula (4), where is p_{data} the real data sample, p_g is the generated data sample, $x \sim p_{data}$ and $x_g \sim p_g$ follows the distribution of [0, 1]. As shown in formula (5), where is $\bar{x}$ random interpolation sample on x and x_g, $\bar{x}$ can be expressed as $\bar{x} = \theta x + (1-\theta x)x_g$. Besides, $x \sim p_{data}$, $x_g \sim p_g$, and θ also follows the distribution of [0, 1]. And λ is the weight coefficient. The generator loss term L_G is shown in formula (6), p_z is the noise distribution and $G(z)$ generate data samples.

$$L = L_D + L_G \quad \textbf{(2)}$$

$$L_D = L_{UL} + L_{GP} \quad \textbf{(3)}$$

$$L_{UL} = -E_{z\sim p_{data}(x)}[D(x)] + E_{x\sim p_g}[D(x)] \quad \textbf{(4)}$$

$$L_{GP} = \lambda E_{\bar{x}\sim p_{\bar{x}}}[(\| \nabla_{\bar{x}} D(\bar{x}) \|_2 - 1)^2] \quad \textbf{(5)}$$

$$L_G = -E_{z\sim p_z(z)}[D(G(z))] \quad \textbf{(6)}$$

4 Performance Analysis

4.1 Dataset

To verify the validity and advancement of the E-DCGAN model, we selected two different types of datasets for experiments in the fields of agriculture and medicine, and subsequent experiments will be conducted on these two datasets. The agricultural field selected AI Challenger 2018 Crop Disease Fine-Grained Classification Contest dataset, containing 45285 images, of which 40745 is training set, and 4540 is test set can be divided into 61 categories. To explore the impact of various data augmentation methods on the small-scale dataset, we selected tomato species as dataset1 from this dataset, including seven diseases and 11100 images in 14 categories. The medical field selected the public medical image Chest X-Ray Images dataset on Kaggle, including two types of NORMAL and PNEUMONIA, 5883 X-ray images. We selected 2500 from this dataset as dataset2. Dataset1 and dataset2 divides into a training set and test set according to 8:2. The specific distribution of the dataset shows in Table 1.

Table 1. Dataset distribution

Dataset	Training set	Test set
dataset1	8880	2220
dataset2	2000	500

4.2 Contrast Experimental of Image Generation

4.2.1 Experimental Description

In order to verify the quality and diversity of the images generated by E-DCGAN (method4) proposed in this paper, we use DCGAN (method1) as the baseline to generate images for dataset1 and dataset2, and set up multiple groups of comparison experiments, mainly comparing method1, DCGAN+WGAN [28] +dropout (method2), DCGAN+WGAN-GP [29] +dropout (method3), and evaluating the advantages and disadvantages of the four image generation methods through FID. To reduce the FID may have errors, each method selected the same number of images for four sets of experiments. The specific experimental results are showing in Table 4.

4.2.2 Evaluating Indicator

It is a difficult problem to evaluate the quality of the generated image. The traditional method is to judge the quality of the model's rendered image through visual intuition to assess the GAN model's performance. This method has specific errors in the generated image and the real image, and fine-tuning of parameters is not applicable. Therefore, in this paper, FID (Fréchet Inception Distance) [30] was used to evaluate the generated images' quality and diversity. FID is based on the convolution feature layer of the Inception network to model the real data distribution p_{data} and the generated data distribution p_g as a multivariate Gaussian distribution with a mean value μ_r, μ_g and variance Σ_r, Σ_g and then solve the distance between the two features. The calculation formula of FID is shown in formula (7).

$$FID = \| \mu_r - \mu_g \|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad \textbf{(7)}$$

The formula represents the sum of the elements along the diagonal of the matrix. FID represents the distance between the feature vector of the generated image and the real image's feature vector. The closer the distance, the better the generated model's effect. That is, the higher the definition of the image and the richer the diversity. Simultaneously, the measurement method can reflect the difference between the real data distribution and the generated data distribution, and has a better performance in terms of diversity, robustness, and efficiency.

4.2.3 Experimental Environment And Parameter Setting

The experiment carries out under the GPU environment. Based on the Pytorch framework, parameters such as learning rate (LR), epoch, and dropout model were adjusted, and default parameter settings are used for the others. Table 2 and Table 3 shows the experimental environment and parameter configuration.

Table 2. Experimental environment configuration

Parameters	Values
Operating System	Windows10
Processor	Core (TM) i7-7800X
Memory	32G
GPU	CUDA9.0
Development Environment	Pytorch 1.1.0, torvision 0.3.0

Table 3. Parameter configuration

Parameters	Values
optim	Adam
LR	0.0002
λ	10
(β_1, β_2, n_d)	(0.5, 0.999, 5)
epoch	300
LeakyReLU	0.2
batchsize	32
dropout	0.2

4.2.4 Experimental Results And Analysis

Table 4 shows the FID comparison of the four models in dataset1 and dataset2. Table 4 shows that in the two different datasets, the FID value of the method proposed in this paper is lower than that of other methods. Compared with the method1 drops by 6.47% and 19.59%, respectively. It is because method4 changes the loss function, introduce SN, GP, and dropout, which solves the problems of unstable training and insufficient diversity of generated images; compared with the method2 drops by 2.27% and 4.17%. It is because method4 uses gradient penalty instead of weight clipping to solve the problem of gradient disappearance and gradient explosion; compared with the method3 drops by 1.30% and 2.87%. It is because the method4 uses SN combined with GP to improve the model's quality. Besides, method4 has better adaptability on the two datasets and is more robust than other GAN types. Figure 5 Generated image comparison. The images from left to right are Tomato Powerly Mildew general and serious; Tomato_Early Blight Fungus general and serious; Tomato_Late Blight Water Mold general and serious; Tomato_Leaf Mold Fungus general and serious; Tomato_Septoria Leaf Spot Fungus general and serious; Tomato Spider Mite Damage general and serious; Tomato YLCV virus general and serious. Figure 5(a) The original images. Figure 5(b) All kinds of images generated by method1. Figure 5(c) All kinds of images generated by method2. Figure 5(d) All kinds of images generated by method3. (e) All kinds of images generated by method4. Figure 6 Generated image comparison. The images from left to right are NORMAL (the first three columns) and PNEUMONIA (the last three columns). Figure 6 (a) The original images. Figure 6(b) All kinds of images generated by method1. Figure 6(c) All kinds of images generated by method2. Figure 6(d) All kinds of images generated by method3. Figure 6(e) All kinds of images generated by method4. It can be seen from Figure 5, Figure 6 that the difference between the generated image and the original image is tiny, and it is difficult for the naked eye to judge the true or false, indicating that method4 has a good generated effect.

Table 4. Comparison of the average FID

Model	dataset1	dataset2
method1	250.28	149.30
method2	243.96	145.71
method3	241.58	143.77
method4	238.43	139.64

Figure 5. Dataset1 generated image comparison

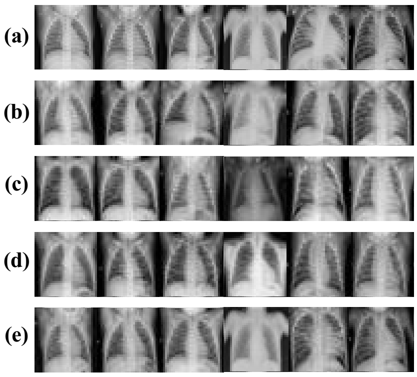

Figure 6. Dataset2 generated image comparison

In summary, given the unstable training of mehod1 and insufficient diversity of generated images, this paper's method solves the problems of method1 to a certain extent, and the method4 has improved the quality of image generation and can be used in data augmentation tasks.

4.3 Image Recognition Comparison Experiment

4.3.1 Experimental Description

To validate the proposed E-DCGAN for data practical application effect, enhanced image recognition tasks is presented in this paper, using the four different data augmentation method for dataset1 and dataset2, and to set up multiple sets of contrast experiment, validating data before and after augmentation, the influence of different data augmentation method for image recognition effect. This experiment mainly compares the following four different data augmentation methods: (1) do not use any data augmentation methods (N); (2) traditional data augmentation methods based on affine transformation (T_AUG); (3) DCGAN generation data (DCGAN); (4) E-DCGAN generation data (E-DCGAN). To solve the dataset imbalance caused by the imbalance of various types of data, the generated data is mixed into the corresponding category data to achieve global data balance. How to do it: (1) For the dataset1, the number of more than 1000 reduce to 1000, and the number of fewer than 1000 is expanded to 1000, while ensuring the same number of images for each data augmentation. (2) The Chest X-ray image dataset to expand 2000 images per category with the same number of images enhanced for each data type. In the experiment, the classification effects of four data augmentation methods on different classifiers (VGG19, AlexNet, ResNet50) compare by accuracy, precision, recall, and F1-Measure.

4.3.2 Evaluating Indicator

To ensure the experiment's scientific, this article uses accuracy, precision, recall, F1-Measure, etc. to evaluate our model. The accuracy is used to measure the model's overall recognition effect on the test set. The precision rate measures the precision of the model, that is, the probability that the classifier predicts a specific category. The recall measures the recall rate of the model, that is, the probability that the classifier is correctly classified into a specific class. The F1-Measure is the harmonic average of precision and recall used to measure the model's performance, defined by the formula such as (8)-(11).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(8)}$$

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(9)}$$

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(10)}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \textbf{(11)}$$

Where TP is the number of times the model correctly classified the samples with true positive, TN is the number of times the model correctly classified the samples with real negative, FP is the number of times the model misclassified the samples with false-positive, FN is the number of times the model misclassified the samples with false-positive, is the number of times the model misclassified the samples with false-negative.

4.3.3 Experimental Environment And Parameter Setting

This article's experiment is carried out in a GPU environment, using the Keras framework based on TensorFlow. The parameters of the model are mainly adjusted in the experiment. Epoch set to 40, batchsize set to 8, and the adam optimizer learning rate of AlexNet and ResNet50 set to 0.0001, and the learning rate of the adam optimizer of VGG19 set to 0.00001. The specific experimental environment parameter configuration shows in Table 5.

Table 5. Experimental environment configuration

Parameters	Values
Operating System	Ubuntu 18.04.3
Processor	GeForce RTX 2080 Ti
Memory	32G
GPU	CUDA10.0.1
Development Environment	Keras framework based on TensorFlow

4.3.4 Experimental Results and Analysis

Table 6 and Table 7 show the performance comparison of different data augmentation methods in dataset1 and dataset2. It can be seen from the table that E-DCGAN proposed in this paper has the following advantages compared with the three methods adopted:

(1) In the VGG19, AlexNet, ResNet50 classification model, data augmentation using T_AUG, DCGAN, and E-DCGAN is more significant than N in the classification model. Specific performance comparisons show in Table 6, Table 7. Based on the T_AUG and CNN models, the average accuracy of dataset1 and dataset2 is 87.05% and 84.07%. Compared with the N data augmentation method, the T_AUG data augmentation method's average accuracy is 0.67% and 1.69% higher. Based on DCGAN and CNN models, the average accuracy of dataset1 and dataset2 is 87.55% and 85.33%. Compared with the N data augmentation method, the average accuracy of the data augmentation method of DCGAN is 1.18% and 2.80% higher. The average accuracy of E-DCGAN and CNN based models on dataset1 and dataset2 is 88.51% and 86.60%, respectively. Compared with the N data augmentation method, the E-DCGAN data augmentation method's average accuracy is 2.14% and 4.07% higher. It can be seen from the improvement of classification model accuracy that the three methods of data augmentation are superior to the method N without data augmentation. Besides, data augmentation methods of T_AUG, DCGAN, and E-DCGAN have tremendous improvement in precision rate, recall rate, and F1-Measure, indicating that data augmentation can effectively solve data shortage and improve the generalization ability and robustness of classification models.

(2) Although the data augmentation method can improve the classification model's performance to a certain extent, different methods' improvement effect is different. It can be concluded from Table 9 that compared with N, the accuracy of the T_AUG data augmentation method is improved by 0.67% and 1.54%, the accuracy improves by 1.16%, 0.47%, and the recall rate on dataset1 and dataset2. The average increase was 0.36% and 1.16%, and the F1-Measure increased by 1.04% and 0.81% on average. Although the T_AUG data augmentation method can improve the model's classification effect to a certain extent, the improvement effect is not apparent due to the lack of diversity of samples generated by non-generative data augmentation and insufficient model generalization ability. The average of DCGAN and E-DCGAN methods for data augmentation is significantly better than the T_AUG method, indicating that the generative data augmentation method learns the original data distribution and randomly generates data consistent with the sample distribution, which is more conducive to improving the diversity of samples and generalization ability of the model.

(3) The generative data augmentation methods are superior to other augmentation methods, but there are differences between them. From Table 8, the comparison table of the improvement degree of E-DCGAN compared to the other three methods, and compared with the DCGAN method, the accuracy of the E-DCGAN method on the dataset1 and dataset2 has increased by 0.96%, 1.27%, and accuracy on average. The precision has increased by 1.23%, 0.72%, the recall has increased by an average of 2.16%, 2.74%, and the F1-Measure has increased by an average of 1.67% and 1.46%. It shows the effect of E-DCGAN through the improvement of network structure and loss function, and further explains the progressive nature of E-DCGAN.

Table 6. Performance comparison of different data augmentation methods in dataset1

Classification models	Data augmentation methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)	Average training time	Average recognition time
VGG19	N	82.27	78.35	78.25	77.30	111m58s	0.480s
	T_AUG	83.42	79.16	78.26	78.71		
	DCGAN	84.43	79.37	80.87	80.11		
	E-DCGAN	85.51	80.30	85.02	82.59		
AlexNet	N	87.60	83.66	84.34	84.00	61m36s	0.264s
	T_AUG	87.93	84.24	85.22	84.73		
	DCGAN	88.25	84.31	85.34	84.82		
	E-DCGAN	88.54	85.68	85.76	85.72		
ResNet50	N	89.26	85.05	87.99	86.50	67m36s	0.290s
	T_AUG	89.80	87.14	88.20	87.47		
	DCGAN	89.98	87.77	88.22	87.99		
	E-DCGAN	91.49	89.16	90.12	89.63		

Table 7. Performance comparison of different data augmentation methods in dataset2

Classification models	Data augmentation methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)	Average training time	Average recognition time
VGG19	N	77.56	77.67	72.64	75.07	39m33s	0.472s
	T_AUG	80.61	79.0	74.62	76.75		
	DCGAN	81.89	86.40	82.05	84.99		
	E-DCGAN	83.81	88.15	87.95	88.04		
AlexNet	N	83.81	85.50	89.5	87.45	21m32s	0.251s
	T_AUG	84.57	85.54	89.74	87.59		
	DCGAN	85.47	85.73	90.16	87.89		
	E-DCGAN	86.72	85.96	92.24	88.99		
ResNet50	N	86.22	88.0	92.0	89.96	24m6s	0.284s
	T_AUG	87.02	88.04	93.26	90.57		
	DCGAN	88.62	88.28	94.36	91.22		
	E-DCGAN	89.26	88.46	94.62	91.44		

Table 8. Performance improvement of E-DCGAN compared with other methods

Dataset	Classification models	Data augmentation methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
dataset1	VGG19	N	3.24	1.95	6.77	5.29
		T_AUG	2.09	1.14	6.76	3.88
		DCGAN	1.08	0.93	4.15	2.48
	AlexNet	N	0.94	2.02	1.42	1.72
		T_AUG	0.61	1.44	0.54	0.99
		DCGAN	0.29	1.37	0.42	0.90
	ResNet50	N	2.23	4.11	2.13	3.13
		T_AUG	1.69	2.02	1.92	2.16
		DCGAN	1.51	1.39	1.90	1.64
dataset2	VGG19	N	6.25	10.48	15.31	12.97
		T_AUG	3.20	9.15	13.33	11.29
		DCGAN	1.92	1.75	5.90	3.05
	AlexNet	N	2.91	0.46	2.74	1.54
		T_AUG	2.15	0.42	2.50	1.40
		DCGAN	1.25	0.42	2.08	1.10
	ResNet50	N	3.04	0.46	2.62	1.48
		T_AUG	2.24	0.23	1.36	0.87
		DCGAN	0.64	0.38	0.26	0.22

Table 9. Average performance of data augmentation methods on different classification models

Dataset	Data augmentation methods	Average Accuracy (%)	Average Precision (%)	Average Recall (%)	Average F1-Measure (%)
dataset1	N	86.38	82.35	83.53	82.60
	T_AUG	87.05	83.51	83.89	83.64
	DCGAN	87.55	83.82	84.81	84.31
	E-DCGAN	88.51	85.05	86.97	85.98
dataset2	N	82.53	83.72	84.71	84.16
	T_AUG	84.07	84.19	85.87	84.97
	DCGAN	85.33	86.80	88.86	88.03
	E-DCGAN	86.60	87.52	91.60	89.49

In summary, given the relative data scarcity for crop disease images and medical images in training CNN, the E-DCGAN data augmentation method, used in this paper can improve the classifier and break performance the bottleneck of data augmentation technology compared with other methods. The accuracy of the model on the test set has been dramatically improved, further verify the feasibility and effectiveness of E-DCGAN for data augmentation.

5 Conclusion

Based on the background of using EI to process the visual information in IoT system, aiming at the problem of data scarcity in IoT image recognition task, this paper proposes an EI-based generative data augmentation system. The system adopts the hybrid architecture of "Edge- Cloud", applies EI technology to the edge computing nodes of the system, and expands the data through the generative data augmentation model to solve the problem of data scarcity. The system is logically divided into training phase and operation phase. In the training phase, transfer learning is used to perform generative data augmentation on the source data close to the target field, and use the enhanced data to complete the CNN model training; the pre-trained CNN is used in the running phase, and the CNN model is continuously improved in actual use scenes by establishing an expert review and self-learning mechanism. It is worth mentioning that this article proposes the E-DCGAN model as a generative data augmentation method. The model performs spectral normalization on the discriminator and generator convolution layer of the DCGAN model to improve the performance of the model. And use the wasserstein distance loss function with gradient penalty to guide unsupervised learning, which enhances the feature extraction ability of the convolutional layer; at the same time, a dropout layer is added at the end of the discriminator network to avoid being stuck during the training process (may be at a local minimum value or saddle point). In the training phase, the E-DCGAN model generated image data is applied to the image recognition of the CNNs; the data samples are generated through the generative network and then mixed into the category samples with less original data. Then the balanced dataset is input into the classification model for training. In the experiment, both agricultural and medical datasets are used, and E-DCGAN is investigated from image generation effect and image recognition. Experiments show that in terms of image generation, compared with the other three GAN, E-DCGAN improves the quality and diversity of the generated images, theoretically solves the phenomenon of model collapse, and stabilizes the training process of the model. In terms of image recognition, the classification model using E-DCGAN for data augmentation is significantly higher than other data augmentation methods in terms of accuracy, precision, recall, and F1-Measure, indicating that E-DCGAN improves the generalization ability and robustness of the model. Not only that, E-DCGAN is not limited to a single field. It has good adaptability and production effects in agriculture and medicine, and it makes up for the vacancy of the IoT system in this respect.

Acknowledgments

This work was supported in part by the Inner Mongolia Autonomous Region's major science and technology: AI application technology and product research and development; application research and demonstration in modem pastures, fund number 2019ZD025. It was also supported by the Inner Mongolia Natural Science Foundation on livestock big data, livestock grazing trajectory mining, and optimized production decision-making research fund number 2019MS06021. It was supported by Inner Mongolia's special project on the transformation of scientific and technological achievements, Xiaoweiyang's entire industry chain quality traceability big data service platform. It has obtained the innovation fund of Inner Mongolia University of science and technology, the construction of chronic disease knowledge map based on natural language processing, fund number: 2019QDL-S09.

References

[1] M. Asim, Y. Wang, K. Wang and P.-Q. Huang, A Review on Computational Intelligence Techniques in Cloud and Edge Computing, *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 4, No. 6, pp. 742-763, December, 2020.

[2] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina and F. Kawsar, Multimodal Deep Learning for Activity and Context Recognition, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* Vol. 1, No. 4, Article No. 157, December, 2017.

[3] S. K. Sharma and X. Wang, Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks, *IEEE Access*, Vol. 5, pp. 4621-4635, March, 2017.

[4] Y. Shen, T. Han, Q. Yang, X. Yang, Y. Wang, F. Li and H. Wen, CS-CNN: Enabling Robust and Efficient Convolutional Neural Networks Inference for Internet-of-Things Applications, *IEEE Access*, Vol. 6, pp. 13439-13448, February, 2018.

[5] S. Wu, G. Wang, P. Tang, F. Chen and L. Shi, Convolution with even-sized kernels and symmetric padding, *Computer Science*, https://arxiv.org/pdf/1903.08385.pdf, May, 2019.

[6] W. Hu, J. Fan, Y. Du, B. Li, N. Xiong and E. Bekkering, MDFC-ResNet: An Agricultural IoT System to Accurately Recognize Crop Diseases, *IEEE Access*, Vol. 8, pp. 115287-115298, June, 2020.

[7] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou and C. Tsatsoulis, Review of Security and Privacy for the Internet of Medical Things (IoMT), *2019 15th International Conference on Distributed Computing in Sensor Systems*, Santorini, Greece, 2019, pp. 457-464.

[8] A. Radford, L. Metz and S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *Computer Science*, https://arxiv.org/pdf/1511.06434.pdf, January, 2016.

[9] F. Wang, M. Zhang, X. Wang, X. Ma and J. Liu, Deep Learning for Edge Computing Applications: A State-of-the-Art Survey, *IEEE Access*, Vol. 8, pp. 58322-58336, March, 2020.

[10] A. Castañeda-Miranda and V. M. Castaño-Meneses, Smart frost measurement for anti-disaster intelligent control in greenhouses via embedding IoT and hybrid AI methods, *Measurement*, Vol. 164, Article No. 108043, November, 2020.

[11] S. Namani and B. Gonen, Smart Agriculture Based on IoT and Cloud Computing, 2*020 3rd International Conference on Information and Computer Technologies*, San Jose, CA, USA, 2020, pp. 553-556.

[12] M. A. Khan, An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier, *IEEE Access*, Vol. 8, pp. 34717-34727, February, 2020.

[13] F. Young, L. Zhang, R. Jiang, H. Liu and C. Wall, A Deep Learning based Wearable Healthcare IoT Device for AI-enabled Hearing Assistance Automation, *Computer Science*, https://arxiv.org/abs/2005.08076, May, 2020.

[14] D. Weimer, B. Scholz-Reiter and M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, *Cirp Annals Manufacturing Technology*, Vol. 65, No. 1, pp. 417-420, 2016.

[15] F. Wang, X. Fan, F. Wang and J. Liu, Backup Battery Analysis and Allocation against Power Outage for Cellular Base Stations, *IEEE Transactions on Mobile Computing*, Vol. 18, No. 3, pp. 520-533, March, 2019.

[16] J. Yan, H. He, X. Zhong and Y. Tang, Q-Learning-Based Vulnerability Analysis of Smart Grid Against Sequential Topology Attacks, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 1, pp. 200-210, January, 2017.

[17] P. Enkvetchakul and O. Surinta, Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition, *Applied Science and Engineering Progress*, January, 2021. DOI: 10.14416/j.asep.2021.01.003.

[18] E. J. Bjerrum, SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules, *Computer Science*, https://arxiv.org/pdf/1703.07076.pdf, May, 2017.

[19] I. S. A. Abdelhalim, M. F. Mohamed and Y. B. Mahdy, Data augmentation for skin lesion using self-attention based progressive generative adversarial network, *Expert Systems with Applications*, Vol. 165, Article No. 113922, March, 2021.

[20] T. Hahn, C. Mechefske, Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder, *International Journal of Hydromechatronics*, Vol. 4, No. 1, pp. 69-98, March, 2021.

[21] A. V. D. Oord, N. Kalchbrenner and K. Kavukcuoglu, Pixel Recurrent Neural Networks, *Proceedings of the 33nd International Conference on Machine Learning*, New York, NY, USA, 2016, pp. 1747-1756.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative Adversarial Nets, *Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, 2014, pp. 2672-2680.

[23] W. Li, M. Cao, Y. Wang, C. Tang and F. Lin, Mining Pool Game Model and Nash Equilibrium Analysis for PoW-Based Blockchain Networks, *IEEE Access*, Vol. 8, pp. 101049-101060, May, 2020.

[24] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification, *Neurocomputing*, Vol. 321, pp. 321-331, December, 2018.

[25] Z. Zheng, L. Zheng and Y. Yang, Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro, *2017 IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 3774-3782.

[26] W. Wang, Y. Sun and S. Halgamuge, Improving MMD-GAN Training with Repulsive Loss Function, *Computer Science*, https://arxiv.org/pdf/1812.09916.pdf, February, 2019.

[27] C. Han, L. Rundo, R. Araki, Y. Nagano, Y. Furukawa, G. Mauri, H. Nakayama and H. Hayashi, Combining Noise-to-Image and Image-to-Image GANs: Brain MR Image Augmentation for Tumor Detection, *IEEE Access*, Vol. 7, pp. 156966-156977, October, 2019.

[28] M. Arjovsky, S. Chintala and L. Bottou, Wasserstein GAN, *Statistics*, https://arxiv.org/pdf/1701.07875.pdf, December, 2017.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, Improved Training of Wasserstein GANs, *Computer Science*, https://arxiv.org/pdf/1704.00028.pdf, December, 2017.

[30] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, *Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 6629-6640.

Biographies

Wei-Jian Hu Lecturer of Inner Mongolia University of Science and Technology. He researches interest is computer vision and deep learning.

Tang-Ying Xie Student of Inner Mongolia University of Science and Technology. His research interests include pattern recognition and computer vision.

Bao-Shan Li Professor of Inner Mongolia University of Science and Technology, is mainly engaged in the research of RFID and Internet of things technology.

Yong-Xing Du Professor of Inner Mongolia University of Science and Technology, is mainly engaged in the research of fields are numerical calculation of electromagnetic fields, new microwave radiators, attenuation of black barrier effects, beam synthesis, etc.

Neal N. Xiong Professor (5th year) at Department of Mathematics and Computer Science, Northeastern State University, OK, USA. His research interests include Cloud Computing, Security and Dependability, Parallel and Distributed Computing, Networks, and Optimization Theory.