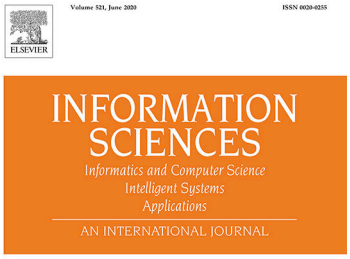
Journal Pre-proofs



Sample Generation based on a Supervised Wasserstein Generative Adversarial Network for High-resolution Remote-sensing Scene Classification

Wei Han, Lizhe Wang, Ruyi Feng, Lang Gao, Xiaodao Chen, Ze Deng, Jia Chen, Peng Liu

PII: S0020-0255(20)30606-X
DOI: https://doi.org/10.1016/j.ins.2020.06.018
Reference: INS 15581

Available online at www.sciencedirect.com
ScienceDirect

To appear in: *Information Sciences*

Received Date: 29 June 2019
Revised Date: 3 June 2020
Accepted Date: 6 June 2020

Please cite this article as: W. Han, L. Wang, R. Feng, L. Gao, X. Chen, Z. Deng, J. Chen, P. Liu, Sample Generation based on a Supervised Wasserstein Generative Adversarial Network for High-resolution Remote-sensing Scene Classification, *Information Sciences* (2020), doi: https://doi.org/10.1016/j.ins.2020.06.018

Sample Generation based on a Supervised Wasserstein Generative Adversarial Network for High-resolution Remote-sensing Scene Classification

Wei Han[a,b], Lizhe Wang[a,b,*], Ruyi Feng[a,b,*], Lang Gao[a,b], Xiaodao Chen[a,b], Ze Deng[a,b], Jia Chen[a,b], Peng Liu[c]

[a] *School of Computer Science, China University of Geosciences, Wuhan, 430074, P. R. China*
[b] *Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, P. R. China*
[c] *Institution of Remote Sensing and and Digital Earth, Chinese Academy of Sciences, Beijing, 100094, China*

Abstract

As high-resolution remote-sensing (HRRS) images have become increasingly widely available, scene classification focusing on the smart classification of land cover and land use has also attracted more attention. However, mainstream methods encounter a severe problem in that many annotation samples are required to obtain an ideal model for scene classification. In the remote sensing community, there is no dataset with a comparative scale to ImageNet (which contains over 14 million images) to meet the sample requirements of the convolutional neural network (CNN)-based methods. In addition, labeling new images is both labor intensive and time consuming. To address these problems, we present a new *generative adversarial network (GAN)-based remote-sensing image generation method (GAN-RSIGM)* that can be applied to create high-resolution annotated samples for scene classification. In GAN-RSIGM, the Wasserstein distance is used to measure the difference between the generator distribution and the real data distribution. This addresses the problem of the gradient disappearing during sample generation, and distinctly promotes a generator distribution close to the real data distribution. An auxiliary classi-

*Corresponding author
Email addresses: lizhe.wang@gmail.com (Lizhe Wang), fengry@cug.edu.cn (Ruyi Feng)

fier is added to the discriminator, guiding the generator to produce consistent and distinct images. With regard to the network structure, the discriminator and the generator are implemented by stacking residual blocks, which further stabilize the training process of the GAN-RSIGM. Extensive experiments were conducted to evaluate the proposed method with two public HRRS datasets. The experimental results demonstrated that the proposed method could achieve satisfactory performance for high-quality annotation sample generation, scene classification, and data augmentation.

Keywords: Generative adversarial network, Scene classification, High-resolution remote sensing, Sample generation

1. Introduction

As Earth observation technologies are continually developing, an increasing number of satellite and aerial images with higher spatial and spectral resolutions can now be obtained [**?**]. High-resolution remote-sensing (HRRS) scene classification, which focuses on the smart classification of land cover and land use according to the image content, is a fundamental task and greatly important for many remote sensing applications, such as urban planning [**?**] and resource management [**?**]. Generally, scenes are composed of diverse objects (including buildings and roads soil), which establish a spatial pattern that serves as a functional zone. Some objects or land-cover classes are generally shared among many scene categories. Moreover, if there are many kinds of objects and complex spatial patterns among objects, this makes scene classification a fairly challenging problem.

In the past few decades, HRRS scenes have attracted increasing research interest [**?**], and significant effort has been made to promote various recognition models and feature representation methods for HRRS scene classification. The bag-of-visual-word (BOVW)-based models [**?**] are some of the most notable approaches for solving the problem of scene classification and have achieved promising results. This type of methods represents images by the frequency

of "visual words," which are constructed by quantizing local features using a clustering method. The performance of BOVW-based methods relies heavily on "handcrafted" aspects, such as texture features [**?**] and color histogram or local structural information [**?**]. As the resolution and the types of the sensors for Earth observation increase, HRRS images present greater details of objects and more complex spatial patterns. However, handcrafted-feature-based methods encounter a severe problem in that they are limited in effectively representing massive HRRS images for scene classification. In addition, extensive engineering skills and domain expertise are needed to design robust handcrafted features. Driven by the development of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [**?**], deep-learning-based methods such as convolutional neural networks (CNNs) have achieved great improvements and have been applied in both image classification and object detection. The deep-learning features, which are automatically derived from massive images by a deep architecture neural model, which works with remarkable representation capability and can be an ideal solution to address the disadvantages of the handcrafted features. Therefore, a variety of popular CNN models such as VGG-Net [**?**], and ResNet [**?**] have been developed and have achieved good performance when compared with the previous state-of-the-art methods.

Owing to the impressive representation ability of deep-learning features, different CNN-based methods have been applied to remote-sensing image scene classification. Existing works mainly increased the performance of scene classification methods by improving feature representation. To adapt the objects with different scales and orientations in scene images, Liu et al. [**?**] applied spatial pyramid pooling to extract multiscale features of HRRS images and proposed a multi-scale deep-feature-learning method (SPP-Net) for scene classification. Yu et al. [**?**] improved the feature representation by fusing the features from multiple CNN layers, while Zhao et al. [**?**] combined spectral and structural information to enhance the representation ability of deep-learning features. Lu et al. [**?**] considered local features are more important than global features and proposed a rearranged local-features-based scene classification. To optimize the

loss function further, the Wasserstein distance was used to replace the maximum likelihood estimation, which more precisely measures the difference between the classifier distribution and the real data distribution [**?**]. In addition, the attention mechanism [**?**], topic models [**?**], metric learning [**?**], texture information [**?**], and spatial relationships of objects [**?**] were also exploited to improve the performance of HRRS scene classification methods. However, one challenge encountered using CNN-based methods for fully supervised learning is that they require a large-scale manually labeled dataset as a training set. Currently, there is no HRRS dataset with a comparative scale to ImageNet [**?**] (which contains over 14 million images) that can meet the requirements of training samples for the CNN-based methods in the remote sensing community. The annotation of remote sensing images is not only time-consuming and expensive but also highly dependent on the availability of expert observers. Developing an effective classification model that works with limited annotation samples or high-resolution sample generation methods can be a promising solution to address this challenge. Liu and Huang *et al.* [**?**] used weakly labeled images as input and a triplet network to address the limitation of clearly labeled datasets. Cheng et al. [**?**] utilized metric learning to reduce the requirement of labeled samples. Despite some excellent work to promote the development of scene classification in the case of limited annotation samples, there remains a gap between the state-of-the-art and satisfactory performance. Therefore, much work remains to be done.

High-resolution sample generation is a different approach to address the problem of an insufficient number of labeled samples. This has been a long-standing challenge in machine learning [**?**] and has been greatly boosted by the development of generative adversarial networks (GANs) [**?**]. These can model high-dimensional, complex, real data distributions, offer a novel way to unlock additional information from a dataset by generating synthetic samples, and have attracted much attention in the machine-learning field. After several years of development, GAN models have been widely applied in high-resolution image generation [**?**], data augmentation [**?**], image super resolution [**?**],

and image classification [**?**] for unsupervised feature extraction [**?**]. Due to their powerful ability to learn and create data distribution, GANs have been applied to solve the problems among the Earth observation community, and have been applied in object detection [**?**], hyperspectral image processing, scene classification [**?**], and image fusion [**?**].

In light of the excellent performance of GANs in image generation and to address the problem of insufficient training samples, a sample and efficient *GAN-based remote-sensing image generation method* (*GAN-RSIGM*) is proposed in this paper. The GAN-RSIGM is not only able to generate samples with specific labels as a supplement to annotated images but also can be applied to scene classification with limited samples as a classifier. In the proposed method, supervised category information [**?**] and Wasserstein evaluation [**?** **?**] are utilized to obtain synthesized images with a high resolution and a specific class. The Wasserstein distance is a more effective metric for estimating the differences between two different distributions than either the Kullbck-Leibler (KL) divergence or Jensen-Shannon (JS) divergences, because it always provides a meaningful gradient for guiding the training process of the neural networks. Supervised label information is employed by an auxiliary classifier of the discriminator that guides the generator to produce consistent and distinct images, and to accelerate the convergence. The networks of the discriminator and the generator are simply implemented by stacking residual blocks [**?**], wherein the residual blocks prevent the problem of vanishing gradients. Here, extensive experiments were conducted to evaluate the proposed method according to three distinct stages: (1) the quality and diversity of the generative samples were analyzed; (2) the classification ability of the auxiliary classifier in the discriminator was established; and (3) the impact of the generative samples on the classification ability was also evaluated. All the experiments were performed using two public datasets [**?** **?**]. The contributions of this work are summarized as follows:

(a) The Wasserstein distance and the auxiliary classifier were used to develop

a GAN-based HRRS image generation method for scene classification. Accordingly, the Wasserstein distance provides an effective measure enabling the proposed method to learn the real data distribution while the auxiliary classifier learns the category information. The Wasserstein distance overcomes the problem of training instability within the process of the GAN-based sample generation process and generates consistent and distinct samples of high quality. To the best of our knowledge, this is the first time that the GAN-based method with Wasserstein distance has been used for HRRS image generation.

(b) Because GAN was adopted for supervised scene classification, the GAN-RSIGM is able to achieve comparable or better performance than other CNN-based methods with the same depth. In the case of limited samples, the experimental results demonstrated that the proposed method was better than the other CNN-based approaches.

(c) The impact of the generative samples on the classification accuracy was analyzed. The experimental results indicated that the generative samples greatly enhanced classification performance.

The rest of this paper is organized as follows. In Section 2, we briefly review some of the related works and summarize the principles and development of GANs. The proposed generation method is described in detail in Section 3. The experimental results are discussed in Section 4. Finally, our conclusions are presented in Section 5.

2. Related Work and Preliminary Knowledge

2.1. Related Work

GANs have attracted wide attention in the field of computer vision for their potential to model high-dimensional, complex, real data distributions. GANs introduces adversarial learning between the generator and the discriminator, which act as adversaries to each other to produce realistic samples [**?**]. Recently, much effort has been made to improve the objective functions and the

architectures of GANs to solve various issues, and achieve many milestones [**?** **?** **?** **?**]. GANs (and its variants) have been applied in image translation, image super resolution, object detection, and many other fields.

Among the Earth observation community, many scholars have recently applied GAN-based models for object detection [**?**], cloud removal [**?**], scene classification [**?**], and image fusion [**?**]. Howe et al. utilized conditional GANs to model and synthesize segmentation mask labels and corresponding remote-sensing imagery to augment available data. To address the problem of a lack of data in the target domain, Zhu et al. [**?**] designed a multi-branch conditional GAN (MCGAN) to generate diverse and high-quality images in optical remote-sensing images. Chen et al. [**?**] proposed a semi-supervised learning-based method for object detection, which trains the detection network with less annotated data and a massive amount of unannotated data. Due to their powerful learning ability to learn and create data distributions, some scholars have used GANs to learn the unsupervised and supervised representations of input data to boost the performance of image classification and object detection [**?**]. Jiang et al. [**?**] proposed edge-enhancement GAN (EEGAN) for robust satellite image super-resolution and reconstruction together with an adversarial learning strategy that is insensitive to noise. Wang et al. [**?**] explored the use of a graphical GAN to synthesize remote sensing images for data augmentation in scene classification. This introduced a new module (recognition network R) that infers the posterior distribution of latent variables given real remote sensing images. Despite these cited applications of GANs, the current work proposes a unique contribution—a simple and efficient GAN-based image generation model that focuses on generating high-quality images for HRRS scene classification. The proposed method implements annotation data generation by introducing an auxiliary classifier, which only introduces a few additional parameters. A new loss function is implemented to stabilize the training process of the neural networks by combining the Wasserstein distance and cross-entropy evaluation, which can learn the class-specific distributions and always provides a significant gradient.

2.2. Preliminary Knowledge

In this section, we first discuss the standard GAN and its associated problems and introduce two relevant variants used in this work.

2.2.1. Generative Adversarial Networks

In the framework of the standard GAN [**?**], the generator network maps a source of noise to a fake sample, and the discriminator network receives either a fake sample or a true data sample, from which each type must be determined. The generator is trained to fool the discriminator. Formally, the game between the generator G and the discriminator D is a the minimax objective function:

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}[log(D(x))] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(D(\tilde{x}))] \tag{1}$$

where $\mathbb{P}_r$ is the real data distribution, x is a real sample, and $\mathbb{P}_g$ is the model distribution. A fake sample $\tilde{x} = G(z), z \sim \mathbb{P}(z)$ (where the input z to the generator is sampled from a simple noise distribution, such as a spherical Gaussian distribution or a uniform distribution). If the discriminator is optimally trained before each generator parameter update, minimizing the value function amounts to minimizing the JS divergence between $\mathbb{P}_r$ and $\mathbb{P}_g$. The JS divergence is a notable metric for measuring the similarity between two probability distributions, which is based on Kullback–Leibler (KL) divergence. The $JS(\mathbb{P}_r, \mathbb{P}_g)$ in the GAN is calculated as follows:

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r, \mathbb{P}_m) + KL(\mathbb{P}_g, \mathbb{P}_m) \tag{2}$$

where $KL(\mathbb{P}_r, \mathbb{P}_m)$ is the defined as $\int log(\frac{P_r(x)}{P_g(x)})P_r(x)d\mu(x)$. Both $\mathbb{P}_r$ and $\mathbb{P}_g$ are assumed to be absolutely continuous and admit densities.

Although GANs have shown great success in image generation, the training process is slow and unstable. Because both the real distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$ rest in low dimensional manifolds to generate high dimensional data, they are almost certainly going to be disjoint in low dimensional manifolds [**?**]. A perfect discriminator that classifies real and fake samples 100%

correctly can always be found. In this case, KL divergence gives infinity, and the value of JS divergence becomes constant. Neither KL nor JS divergence can provide a meaningful gradient; hence, the training of the generator is terminal.

2.2.2. Wasserstein Generative Adversarial Networks

The Wasserstein distance (also called earth-mover's distance) can be interpreted as the minimum energy cost of moving and transforming a pile of dirt in the shape of a probability distribution to the shape of the other distribution. It can always produce a meaningful gradient during the training process; even the generator distribution and the real data distribution are disjointed. Therefore, it is used to solve the problem of gradient vanishing in GAN and to implement a Wasserstein generative adversarial network (WGAN) [**?**].

In a WGAN, the Wasserstein distance measures the distance between the fake distribution $\mathbb{P}_g$ and the real data distribution $\mathbb{P}_r$, which is defined as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{3}$$

where $\prod(\mathbb{P}_r, \mathbb{P}_g)$ is the set of joint distribution $\gamma(x, y)$, with $\mathbb{P}_r$ and $\mathbb{P}_g$ as marginal distributions. Term $W(\mathbb{P}_r, \mathbb{P}_g)$ indicates the "mass" that must be transported from x to y to transform the distribution $\mathbb{P}_r$ into $\mathbb{P}_g$.

The Wasserstein distance enables us to provide a meaningful gradient value during backward computing in GANs. According to the Kantorovich–Rubinstein duality [**?**], the equation for the objective function of GAN, Equation (1) combined with the Wasserstein distance measure is optimized as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[f(\tilde{x})] \tag{4}$$

To meet the requirement of Lipschitz continuity, the weight parameters are clamped to a fixed box $W = [-0.1, 0.1]$. The objective function is defined as follows:

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{5}$$

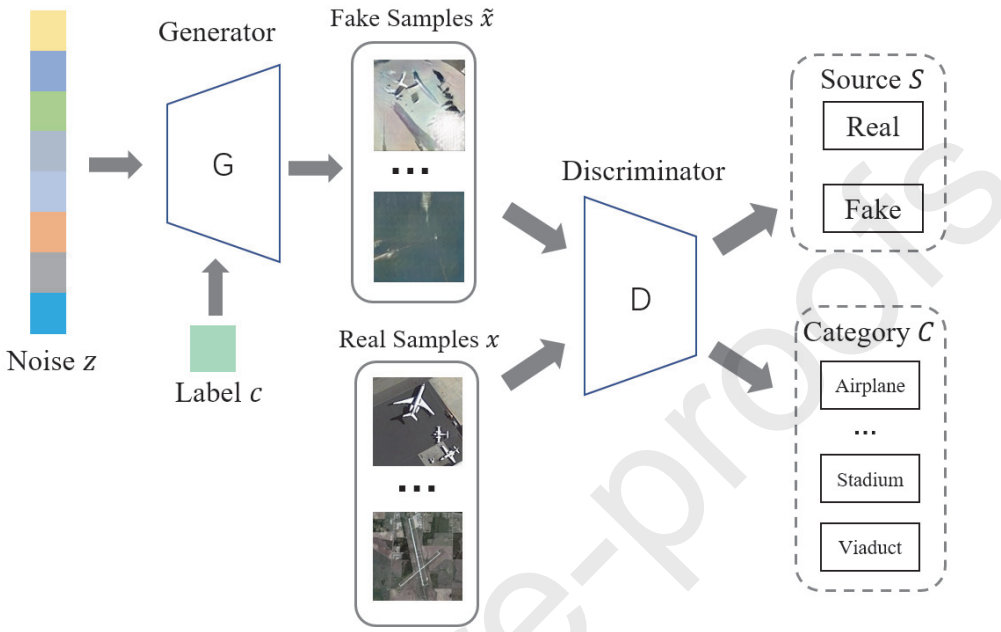


Figure 1: Illustration of ACGAN.

The problem of hard training in GANs is primarily addressed by the Wasserstein distance.

2.2.3. The Auxiliary Classifier Generative Adversarial Network (ACGAN)

The original GAN is based on unsupervised learning, and the categories of the generative samples are uncontrollable. To derive generative sample classes, label information has been added to promote the generated categories and training stability for some approaches, such as the conditional generative adversarial network (CGAN) [**?**] and the auxiliary classifier generative adversarial network (ACGAN) [**?**]. To guides the category and the quality of the generative samples, the CGAN adds label information to initialize the noise vector. However, the discriminator of the CGAN can only distinguish the source of the input images and can not explicitly recognize the class labels of the input images to update the model parameters using the backpropagation algorithm. Therefore, the effect of the category information for CGAN is limited. This problem is

addressed by an ACGAN, wherein an auxiliary classifier is added to the discriminator. Therefore, the discriminator has two branches to simultaneously predict the category and the source of an input image, as shown in Fig. 1. Accordingly, supervised category loss is added into the objective function, which guides the ACGAN to achieve the generated sample with class information.

In Fig. 1, each generative sample has a specific class label, $c \sim P_c$, in addition to the noise z. Here, G uses both to generate fake images $\tilde{x} = G(c, z)$. The discriminator provides a probability distribution over sources, as well as a probability distribution over category labels, $P(S|X), P(C|X) = D(X)$. Hence, the objective function has two parts: the log-likelihood of the correct source and the log-likelihood of the correct category. The entire objective function is as follows:

$$\begin{aligned} \max_{D,G} \ & \mathbb{E}_{x\sim\mathbb{P}_r}[log(P(S = real|x))] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(P(S = fake|\tilde{x}))] + \\ & \mathbb{E}_{x\sim\mathbb{P}_r}[log(P(C = c|x))] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(P(C = c|\tilde{x}))] \end{aligned} \tag{6}$$

where $\mathbb{E}_{x\sim\mathbb{P}_r}[log(P(C = c|x))]$ and $\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(P(C = c|\tilde{x}))]$ are the log-likelihood expectations of the probability of correct classification.

3. Methodology

To stably generate HRRS images stably and efficiently with consistent class information, a *GAN-based HRRS image generation method (GAN-RSIGM)* is implemented to meet the requirements. The main procedure of the proposed method is presented in Fig. 2. This contains two components: sample generation for the generator and sample judgement for the discriminator. In the first component, the input of the generator is a set of noise vectors z sampled from the Gaussian distribution by adding category labels c sampled from the random seed algorithm. Next, the input vectors are processed by a linear layer and a set of deconvolution-based residual blocks. Finally, high-resolution samples with specific labels are obtained as the output of the generator. In the component of sample judgement for the discriminator, the input is either a batch of real

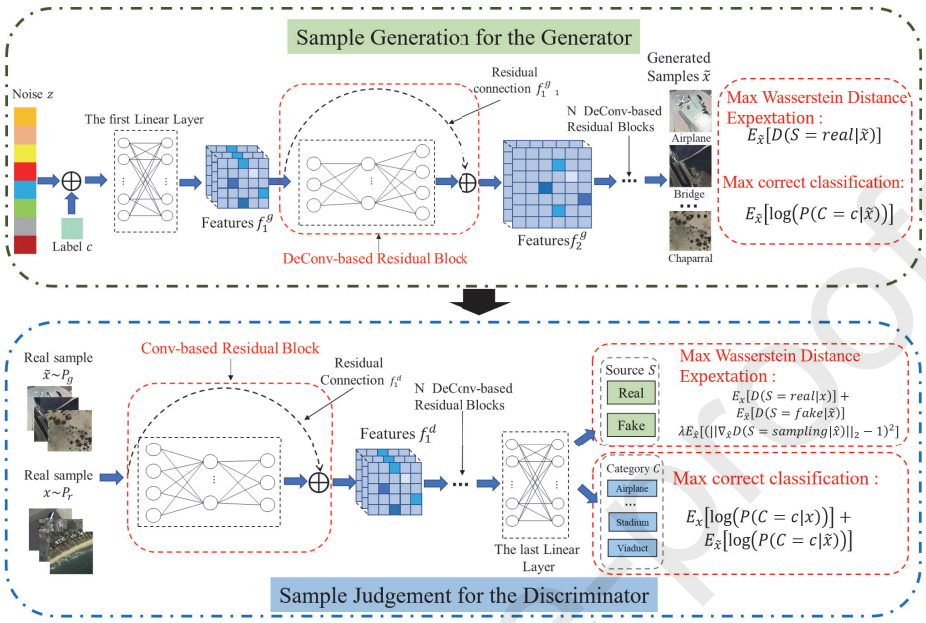


Figure 2: Flowchart of the proposed GAN-RSIGM.

samples, denoted as x, or fake samples, denoted as $\tilde{x}$. The units to construct the discriminator are a series of convolution-based residual blocks and linear layers. The discriminator has two branches for outputting two predicted results: one prediction for the sample source and the other for the category class. The proposed method can be summarized as follows:

(1) The noise z and the label c are fed into the generator to generate fake samples $\tilde{x}$.

(2) Real samples x and generative samples $\tilde{x}$ are jointly fed to the discriminator. The discriminator judges the data source and predicts the category labels of the input samples according to the auxiliary classifier.

(3) The source loss (Wasserstein distance [**?**]) and classifier loss (cross-entropy) are calculated. The discriminator is updated based on the back-propagation algorithm.

(4) The generator is also modified according to the loss of fake samples.

High-quality generative samples can be obtained by performing sufficient

training iterations. In addition, the discriminator is available for remote sensing scene classification. Within the whole process flow, there are two factors that significantly affect the performance: (1) the objective function, and (2) the structure of the neural network. In the following part of this section, a more detailed discussion is provided with regard to the innovative contributions of these two factors.

3.1. The Objective Function

To generate high-resolution and annotated remote sensing images, the Wasserstein distance is used to measure the difference between the real and the fake data distributions. Meanwhile, supervised likelihood loss is used to guide the generative samples of the generator. The advantages of these two metrics are as follows:

(1) The Wasserstein distance, which is set as the metric to indicate the difference between the real data distribution $\mathbb{P}_r$ and model distribution $\mathbb{P}_g$, can overcome the drawback of gradient dispersion in JS divergence. Only the discriminator meets the Lipschitz continuity, and the Wasserstein distance is able to provide a meaningful gradient value during backward computing. Therefore, the optimization process of the generator is more efficient and stable [**?**].

(2) The log-likelihood loss of correct classification for the auxiliary classifier guides the generator to create consistent and distinct categories as well as to accelerate the convergence. In the ACGAN [**?**], supervised category loss is first added into the objective function to achieve consistency of the generated categories and training stability.

However, due to the problem of difficult convergence for the gradient clip in WGAN [**?**], the weight penalty method is proposed to replace the gradient clip [**?**]. The objective function is expressed as follows:

$$\min_G \max_D \ \mathbb{E}_{x\sim\mathbb{P}_r}\left[D(x)\right] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}\left[D(\tilde{x})\right] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\| \bigtriangledown_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (7)$$

where $\lambda \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\| \bigtriangledown_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$ is denoted as the gradient penalty. Term $\hat{x} \sim \mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points sampled from the real data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$. Because of this, the graph of the optimal discriminator consists of straight lines connecting points $\mathbb{P}_r$ and $\mathbb{P}_g$. On the sampling points, $\bigtriangledown_{\hat{x}} D(\hat{x})$ is used as the gradient norm and is constrained to 1. Because it would be too constraining to oblige the unit gradient norm everywhere, the unit gradient norm is only obliged on these straight lines, which yields satisfactory results. By adding the gradient penalty, the objective function satisfies the limitation of the Lipschitz continuity, and the advantages of the Wasserstein distance can be made available. Empirically, the penalty coefficient λ is set to 10 [**?**]. After replacing the gradient clip with the gradient penalty, the improved WGAN overcomes the problems of the original version. Furthermore, the GAN problem of difficult convergence is resolved. The improved WGAN can then be utilized to model more complex distributions and generate high-quality and high-resolution image samples.

According to Equations (6) and (7), a new objective function is proposed that combines the advantages of WGAN and ACGAN. For the discriminator, the objective function is presented as follows:

$$\begin{aligned}\max\ \mathbb{E}_{x\sim\mathbb{P}_r}\ [P(S = real|x)] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}\ [P(S = fake|\tilde{x})] + \mathbb{E}_{x\sim\mathbb{P}_r}[log(P(C = c|x))] + \\ \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(P(C = c|\tilde{x}))] + \lambda \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\| \bigtriangledown_{\hat{x}} D(S = sampling|\hat{x})\|_2 - 1)^2]\end{aligned} \tag{8}$$

where $x \sim \mathbb{P}_r$ is the real data distribution, $\tilde{x} \sim \mathbb{P}_g$ is the model distribution, and the $\mathbb{P}_{\hat{x}}$ sample is uniform along straight lines between pairs of points sampled from the real data distribution $\mathbb{P}_r$ and generator distribution $\mathbb{P}_g$. $P(S|X), P(C|X) = D(X)$ indicates a probability distribution over the sources and a probability distribution over the category labels. Here, $(\| \bigtriangledown_{\hat{x}} D(S = sampling|\hat{x})\|_2 - 1)^2$ is the gradient penalty for the discriminator to meet the Lipschitz continuity. The penalty coefficient λ is consistent with the original work [**?**].

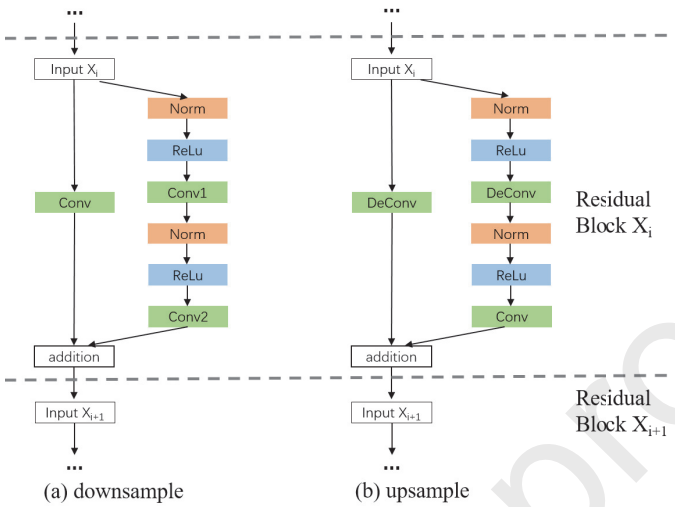


Figure 3: Illustrations of the two kinds of residual blocks. (a) A downsampling residual block. (b) An upsampling residual block.

For the generator, the objective function is shown as follows:

$$\max \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(P(C=c|\tilde{x}))] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[P(S=real|\tilde{x})] \tag{9}$$

For the generator, the objective function only maximizes the expectation of the possibility that fake data is judged to be real, and the possibility that the categories of fake data are classified correctly.

3.2. The Architecture of the Neural Network

ResNet [**?**], as one of the more widely used neural networks, has aroused considerable attention in the computer vision and scene classification fields. Its most impressive characteristic of ResNet is its depth. ResNet is implemented with layer numbers of 50, 101, and more than 1,000, which is much bigger than the existing CNN models. The performance of ResNet can be attributed to the use of residual blocks. A residual block enables us to pass the important information from one layer to the next, which speeds up the training process for the CNN and prevents the problem of gradient dispersion. Some works that

Table 1: **Parameter settings for the generator**

	Kernel size	**Sample**	**Output size**
z			128
Linear			512×4×4
Residual Block 1	[3×3]×2	Up	256×8×8
Residual Block 2	[3×3]×2	Up	256×16×16
Residual Block 3	[3×3]×2	Up	128×32×32
Residual Block 4	[3×3]×2	Up	64×64×64
Residual Block 5	[3×3]×2	Up	64×128×128
Conv2D tanh	3×3	-	3×128×128

have used the residual structure in the GAN for enhanced non-linear ability. A residual-block-based generator and discriminator are applied in the proposed GAN-RSIGM. Differing from the residual blocks in classification models, the two types of residual block structures are shown in Fig. 3, where (a) is a downsampling block used for the discriminator and (b) is an upsampling block used for the generator. We let $X_l(l = 1, 2, ..., N)$ be the input of the i_{th} residual block. Therefore, the output X_{l+1} is as follows:

$$X_{l+1} = f_1(X_l) + f_2(X_l) \tag{10}$$

where $f_1(x)$ is the standard feed-forward convolution operation, and $f_2(x)$ is the convolutional operation in the shortcut connection. $f_1(x)$ is defined as follows:

$$f_1(x) = \begin{cases} F(r(up\_F(r(X_l)))), & G \\ F(r(down\_F(r(X_l)))), & D \end{cases} \tag{11}$$

where $r(x)$ is the non-linear activation function, $F(x)$ is the convolution operation, $up\_F(x)$ is the upsampling convolution operation, and $down\_F(x)$ is the downsampling convolution operational. G is the generator, and D is the

Table 2: **Parameter settings for the discriminator**

	Kernel size	**Sample**	**Output size**
Input			3×128×128
Residual Block 1	[3×3]×2	Down	64×64×64
Residual Block 2	[3×3]×2	Down	128×32×32
Residual Block 3	[3×3]×2	Down	256×16×16
Residual Block 4	[3×3]×2	Down	256×8×8
Residual Block 5	[3×3]×2	Down	512×4×4
ReLU, Linear	-	-	1
ReLU, Linear	-	-	n_class

discriminator. Similarly, $f_2(X_l)$ is formulated as follows:

$$f_2(x) = \begin{cases} up\_F(X_l), & G \\ down\_F(X_l), & D \end{cases} \tag{12}$$

Owing to the advantages of the residual block, the layer number of the generator and the discriminator can be relatively large. The two large neural networks are effective in improving the quality of the generative samples. In the proposed GAN-RSIGM, the generator and the discriminator networks both contain five residual blocks (10 convolutional layers). The kernel size, the sample operation, and the output size of each block in the generator and the discriminator are shown in Tables 1 and 2. Each line in the two tables shows a part of the structure of the generator or the discriminator network. For example, the input of the discriminator network is a set of images with RGB channels and a resolution of 128×128. As shown in line 2 of Table 2, Residual Block 1 performs a downsampling operation to generate the feature maps with the dimensions of 64×64×64. An identical downsampling operation is undertaken to obtain more high-level features in the following residual blocks. As shown in the last two lines in Table 2, two fully connected layers are used to output the category label and the Wasserstein distance, respectively. The noise z is the input of the generator network, and the upsampling operations are executed in

Figure 4: Example images from the UCM dataset. Each class shows two images.

Figure 5: Example images from the NWPU-RESISC45 dataset. Each class shows two images.

the generator network, as shown in Table 1.

4. Experimental Results

A series of experiments was conducted to evaluate the performance of the GAN-RSIGM with two public HRRS image datasets. The experimental settings and results, along with an analysis of the results, are presented in detail in the following section.

4.1. Dataset Description

The proposed method was evaluated using two public datasets designed for HRRS image scene classification: the University of California-Merced (UCM) dataset [**?**] and the NWPU-RESISC45 dataset [**?**].

The UCM dataset [**?**] was extracted from large optical images (RGB color space) of the US Geological Survey taken over various regions of the United States (and is named after the institution that created it for research purposes). A total of 2,100 images with 256×256 pixels were manually labeled according to 21 land-use classes, with 100 images for each class. Fig. 4 presents two example images for each class. Due to their nature and relatively high resolution (30 cm), these images share many low-level features with general-purpose optical images. This makes the features extracted by a CNN pre-trained on ImageNet efficient for scene classification of the UCM dataset.

The NWPU-RESISC45 dataset [**?**] is composed of 31,500 remote sensing images divided into 45 scene classes. Each scene class contains 700 images, each of which is of 256×256 pixels in the RGB color space. Some example images are shown in Fig. 5. The dataset was extracted by remote sensing field specialists using Google Earth, with the spatial resolutions ranging from approximately 30 m to 0.2 m per pixel. The NWPU-RESISC45 dataset is a large-scale dataset of different scene classes. It involves several variations in term of translation, viewpoint, object pose, spatial resolution, illumination, background, and occlusion, and features significant inter-class similarity and intra-class diversity, which means that this dataset is suitable for the development and evaluation of various data-driven algorithms.

4.2. Dataset Split and Metrics

As shown in Table 3, the datasets were randomly split into a training set and test set using three different settings: (1) 20% for training and 80% for testing, (2) 40% for training and 60% for testing, and (3) 60% for training and 40% for testing. To match the proposed model, all the images were resized to 128×128 pixels.

Five metrics were adopted to evaluate the convergence properties of the GAN-RSIGM and the scene classification results: the Wasserstein distance, multi-scale structural similarity (MS-SSIM), overall accuracy (OA), kappa coefficient (kappa), and F1 score (F1_score). More information about these metrics

Table 3: **The three dataset settings**

Dataset Setting	Training Set Ratio	Test Set Ratio	No. per Class for UCM		No. per Class for NWPU45	
			training	test	training	test
(1)	20%	80%	20	80	140	560
(2)	40%	60%	40	60	280	420
(3)	60%	40%	60	40	420	280

has been provided in the following:

(1) *Wasserstein distance*: This enables us to measure the difference between two distributions. In [**?**], it was first applied to a GAN to show the degree of convergence. However, it should be noted that the Wasserstein distance is not a quantitative evaluation measure. The value depends on the discriminator's architecture and model capacity, which means that it is difficult to compare models with different discriminators.

(2) *MS-SSIM*: This metric [**?**] is a multi-scale variant of a well-characterized perceptual similarity metric that attempts to discount aspects of an image that are not important for human perception. The values of MS-SSIM range between 0.0 and 1.0; higher MS-SSIM values correspond to more perceptually similar images.

(3) *Overall accuracy (OA)*: This is defined as the number of correctly classified images, regardless of their classes, divided by the total number of images.

(4) *Kappa coefficient (kappa)*: This is a statistical calculation that measures the inter-rater agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than a simple percentage agreement calculation,, as the value takes into account the possibility of the agreement occurring by chance.

(5) *F1 score (F1_score)*: Also called the F-measure or F-score, this considers both the precision and the recall of the test in computing the score. The F1 score achieves a best value at 1 and the worst at 0.

Table 4: **Workstation configurations**

Hardware Platforms	**Parameter Name**
CPU	i7-5820k (3.3 Ghz, 6 cores)
Memory	32GB DDR4
GPU	TITAN X Pascal
Memory	12GB DDR5
Software Platforms	**Parameter Name**
OS	Ubuntu 16.10
DL Framework	Tensorflow & Pytorch
CUDA	SDK 8.0
CuDNN	SDK 6.1

4.3. Experiment Setting

All the experiments were carried out on a workstation equipped with two NVIDIA GeForce Pascal Titan X GPUs. The hardware and software configurations are listed in Table 4. To be specific, two CNNs were constructed as comparison algorithms to evaluate the classification ability for images of 128×128 pixels. One CNN was based on VGGNet [**?**] to achieve a 10-layer network structure (seven convolutional layers and three fully-connected layers)—denoted as VGG-10—and the other CNN was a nine-layer model based on residual block [**?**] (nine convolutional layers and one fully-connected layer), denoted as ResNet-10. To adapt to the image resolution, the depths of the comparison algorithms were relatively small. One shallow-feature-based method, BOVW, was applied as a comparison method; this constructs a word vector by extracting the "bag of features and colors" (BOFC) information from an input image, and a linear support vector machine (SVM) is used to finish the classification task.

The training settings of the proposed method were as described as follows. The batch size was set to 64 with a learning rate of 0.0002. A learning rate decay and the batch normalization were used to accelerate the learning process and avoid overfitting. An Adam optimizer was utilized to update the weight parameters of the deep neural networks. All the weight and bias parameters were

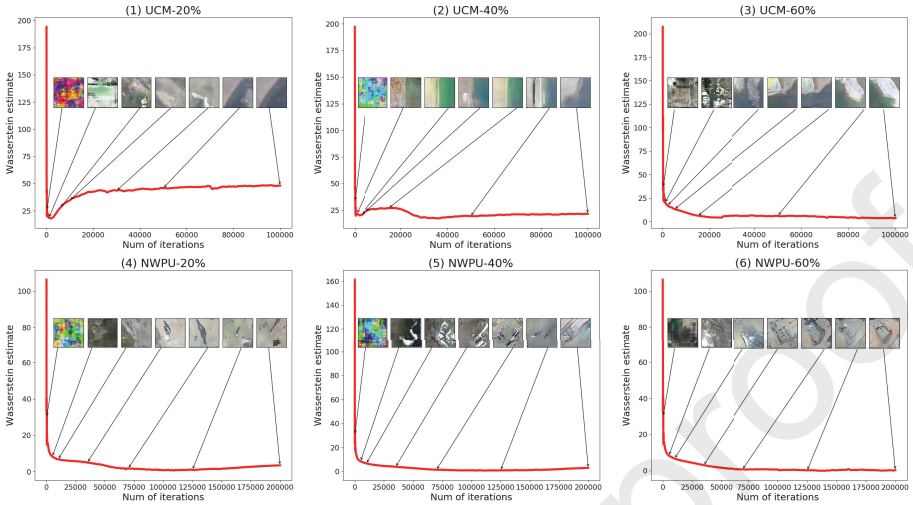


Figure 6: Wasserstein estimation for the proposed method on the UCM and NWPU-RESISC45 datasets. The x-axis is the number of iterations and the y-axis is the Wasserstein distance.

initialized by a Gaussian distribution. The training settings of the comparison methods were as follows. The batch size was also set to 64, and the learning rate was initialized to 0.01. A stochastic gradient descent (SGD) optimizer was used to update the CNN models.

The experiments were organized into three parts. First, the Wasserstein distance of the proposed GAN-RSIGM was observed during the training process for the two datasets. The final generative samples were then analyzed. Second, the classification ability of the proposed method was evaluated with the three comparison algorithms of BOVW, ResNet-10, and VGG-10. Third, the effect of the generative samples on the classification accuracy of the classifier model was investigated. It should be noted that once the dataset split was completed for the three settings was completed, the training set and the test set were the same for the proposed method and the comparison methods in all the experiments. In addition, to obtain reliable results for both datasets, the experiments were repeated five times for each experimental setting, with the mean and standard deviations of all the results being reported.

4.4. Experiment I:Analysis of the Training Process and Generative Samples

First, we evaluated the stability of the training and the convergence for the two datasets. The Wasserstein distance enabled us to determine the convergence of the training process. Fig. 6 shows the relationship between the quality of the generative samples, the Wasserstein distance, and the iteration number. Due to the difference in sample numbers, we trained the GAN-RSIGM on the UCM dataset for 100,000 iterations and on the NWPU-RESISC45 dataset for 200,000 iterations. Initially, the Wasserstein distance between the generator distribution and the real data distribution was large (more than 200 for UCM and 100 for NWPU-RESISC45). As the number of iterations increased, the value dramatically decreased. After 25% of the training iterations, the curve of the Wasserstein distance became stable. The curve of the Wasserstein distance for the UCM dataset converged to about 45 over 20% of the training set, see Fig. 6(1), 17 over 40% of the training set, and 7 over 60% of the training set, see Fig. 6(2-3). The results for the NWPU-RESISC45 dataset were even lower, with values of less than 1, see Fig. 6(5-6). At first, the generative samples were first confusing and did not contain category information. As the number of training iterations increased, the generative images started to reveal category information. Fig. 6(1-3) shows the changes of a beach image during the training process, and Fig. 6(4-6) depicts an airplane sample. In the subsequent training, the generative images gradually contained greater details, and the quality was improved. Therefore, the proposed GAN-RSIGM can clearly achieve a high degree of convergence and generate HRRS images with specific categories. Meanwhile, the Wasserstein distance decreased as the number of training samples increased.

The generated samples of the proposed method were compared with some widely-used GAN-based models, including GAN [**?**], RSGAN [**?**], WGAN [**?**], LSGAN [**?**], and ACGAN [**?**]. Unsupervised learning training was used in GAN, RSGAN, WGAN, and LSGAN, while ACGAN and the proposed method utilize supervised learning training. The UCM dataset was used in this experiment, and the settings for the dataset split were 60% for the training set

Figure 7: Comparison of generative samples for standard GAN [**?**], RSGAN [**?**], WGAN-GP [**?**], LSGAN [**?**], ACGAN [**?**] and the proposed GAN-RSIGM for the UCM dataset.

and 40% for the test set. The iteration number for the comparison methods was set at 500,000. As shown in Fig. 7 (a), the generated GAN samples were unrecognizable and fuzzy. A possible reason is that the complexity of the HRRS images caused a disappearing gradient, while the training of GAN did not reach convergence. In Fig. 7, the generated results of RSGAN and LSGAN showed a higher degree of convergence; however, this was with ambiguous category labels and some details. Furthermore, the image quality of the results was unsatisfactory. This result illustrated that it might not be possible to apply GAN-based models directly in computer vision to generate HRRS images.

Because the loss metrics of WGAN and ACGAN were adopted to form the overall loss function in the proposed method, we focused on comparing the results of the proposed method with WGAN and ACGAN. In Fig. 7 (c), the results generated by WGAN did not reach a satisfactory degree of convergence, in that it did not show specific information regarding category. In Fig. 7 (e), the ACGAN generated samples did not present sufficient details and category information, while the generative samples of the proposed method contained sufficient details, and the category information was easy to recognize in Fig. 7 (f). The results demonstrated that one measure of cross-entropy loss (or Wasserstein distance) was limited for HRRS image generation because of the many categories of objects, the scale variations, and the complex backgrounds in the HRRS images. Thus, the proposed method using the Wasserstein distance evaluation and supervised cross-entropy loss was able to learn the distribution of the HRRS images and generate high-quality samples effectively.

We also evaluated the diversity. Fig. 8 compares the comparison of the real samples and fake samples, and the MS-SSIM values for baseball diamond, beach, forest, and harbor using the UCM dataset. Although the generative samples showed certain differences from real images, the color, shape, and texture characteristics of the generative samples were similar to those of the real images. From Fig. 8, each fake image shows the objects, their spatial relationships between the objects, and the background information for the specific class. For several generative pictures in the same category, the shape, color, texture

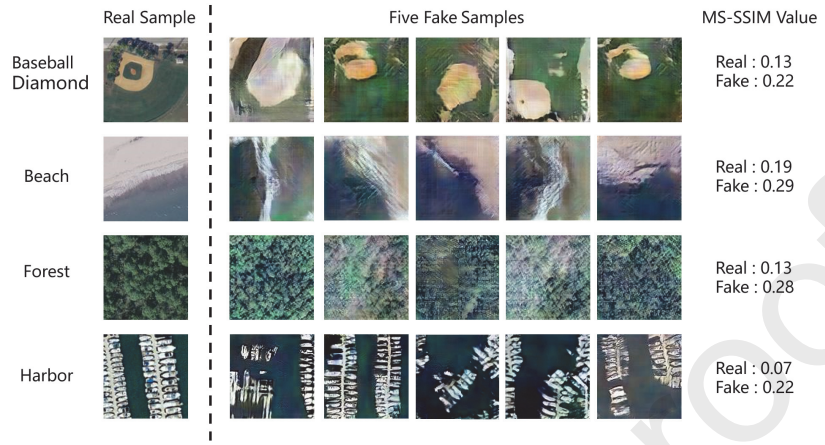


Figure 8: Analysis of the diversity of the generative samples. The classes of baseball diamond, beach, forest, and harbor are used for the evaluation, with one real sample and fives fake samples shown per class. The MS-SSIM value of real and fake samples for different classes are listed on the right.

information, and the relationships of the objects were consistently varied. One possible reason was that the input vector randomly sampling from the Gaussian distribution caused the diversities among the generative samples. For the similarity indicator MS-SSIM, the value of the generative samples was slightly higher than the real samples, and the range was 0.20–0.30. The experimental results showed that the generative samples produced by the proposed GAN-RSIGM had sufficient in-class diversity, rather than a fixed mode for the same class.

4.5. Experiment II: Evaluation of the Classification Capability

In the second experiment, the classification ability of the proposed GAN-RSIGM was evaluated using one shallow-feature-based method, BOVW, and two deep-feature-based methods, ResNet-10 and VGG-10. The UCM and NUPW-RESISC45 datasets were used. The classification ability of all the methods was evaluated under three dataset settings, which were the same as those in Experiment I. All the experimental results are listed in Table 5.

In Table 5, it can be seen that BOVW achieved the worst results in six different settings, which produced a significant gap between the deep learning

Table 5: **Classification results of the three classifiers for the UCM and the NWPU-RESISC45 datasets**

DataSet Name	Method Name	20% training set			40% training set			60% training set		
		OA(%)	Kappa	F1_Score	OA(%)	Kappa	F1_Score	OA(%)	Kappa	F1_Score
UC Merced	BOVW	61.31±0.92	0.601±0.019	0.608±0.023	70.03±2.05	0.682±0.015	0.693±0.019	76.36±1.13	0.751±0.009	0.758±0.012
	VGG	66.15±1.45	0.645±0.014	0.658±0.007	76.93±0.54	0.752±0.009	0.762±0.005	82.75±0.28	0.808±0.004	0.818±0.011
	ResNet	66.17±1.35	0.644±0.014	0.658±0.014	76.87±1.50	0.757±0.018	0.759±0.013	83.25±0.16	0.824±0.003	0.826±0.012
	GAN-RSIGM	71.40±3.20	0.697±0.021	0.709±0.020	79.40±2.57	0.783±0.025	0.791±0.022	87.05±2.00	0.867±0.016	0.865±0.018
NWPU RESISC45	BOVW	61.97±1.62	0.608±0.021	0.612±0.019	68.45±0.93	0.673±0.021	0.681±0.015	72.81±0.75	0.706±0.012	0.719±0.015
	VGG	67.15±0.64	0.664±0.022	0.665±0.007	74.92±0.87	0.742±0.010	0.756±0.011	79.19±0.08	0.786±0.002	0.780±0.005
	ResNet	65.37±0.34	0.646±0.004	0.650±0.002	73.20±0.69	0.726±0.008	0.724±0.010	76.93±0.38	0.764±0.004	0.764±0.009
	GAN-RSIGM	69.35±0.64	0.657±0.022	0.688±0.019	75.56±1.58	0.732±0.013	0.751±0.016	80.41±1.25	0.786±0.008	0.798±0.007

methods. The VGG-10 achieved a better classification performance than the ResNet-10 in five out of six settings. The GAN-RSIGM achieved the best classification results in all six settings, and it outperformed the comparison methods by 4.2% with the UCM dataset and 2.2% with the NWPU-RESISC45 dataset over the same dataset settings (20% training set and 80% test set). When the ratio of the training set was set to 60%, the proposed method obtained the highest classification accuracies, an OA of 87.05%, a kappa of 0.867, and an F1_score of 0.865 for the UCM dataset and an OA of 80.71%, a kappa of 0.786, and an F1_score of 0.789 for the NWPU-RESISC45 dataset. The experimental results demonstrated that the GAN-based method could achieve similar or even higher classification performance than the end-to-end convolutional networks. In addition, the increase in performance for the UCM dataset was greater than that for the NWPU-RESISC45 dataset. For a small dataset containing a small number of samples, the training set could not meet the training requirement of a CNN. In this case, the GAN-based method generated fake samples with some diversity, which were different from the original samples. Therefore, the training set was expanded, and the classification performance was improved. For a large-scale dataset, there was abundant variability and diversity. The neural networks, including both convolutional networks and adversarial networks, could undergo sufficient training with the large training set, and thus similar classification accuracies were achieved.

Table 6: **Evaluation of the effect of the generative samples on the classification accuracy**

Ratio of Labeled Set	Classification Ratio (%)					
	0	1	2	3	4	5
UCM-20%	66.15±1.45	68.95±1.85	73.59±0.35	72.50±0.89	71.37±0.98	72.22±0.59
UCM-40%	76.93±0.54	79.84±1.27	77.62±0.95	77.58±0.28	77.94±0.09	76.18±0.62
UCM-60%	82.75±0.28	83.55±0.70	83.56±0.23	83.87±0.07	84.52±0.72	84.46±0.6

4.6. Experiment III: Evaluation the Effect of the Generative Samples on Classification Accuracy.

Finally, we performed three experiments to evaluate the effect of the generative samples on the classification accuracy of the classifier model. The VGG-10 was used as the baseline model, and the UCM dataset was used, with the three different dataset settings described above. First, by fixing the original training set and increasing the number of generative samples as an extended training set, a change of the classification accuracy of the VGG-10 with the test set was observed. The results are shown in Table 6, where the only variable was the ratio of generative samples to original samples. For instance, when the ratio was set to 2, there were 20 real samples per class, and there were 40 generative samples per class for the UCM dataset. It should be noted that the GAN-RSIGM was trained on the same real training set as the VGG-10. Therefore, the generative samples were not affected by the samples in the test set. We then investigated the accuracy curve of the VGG-10 with the generative training set, where the VGG-10 without generative samples was set as the comparison method. The experimental results are listed in Table 6, and shown in Fig. 9.

Evaluating of the impact of the ratio of generative samples, we can see in Table 6 that the generative samples helped to improve the classification accuracy of the VGG-10 with the UCM dataset, by approximately 6% with the 20% training set, 1.9% with the 40% training set, and 1.7% with the 60% training set. The positive effect on the classification improvement of the generative samples was more evident in the case of a smaller training set. One possible reason for this was that, in the case of sufficient training samples, the generative samples could not provide more beneficial information to promote the ability

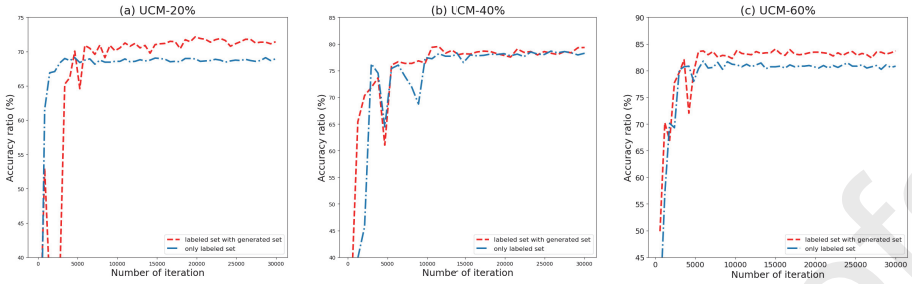


Figure 9: Accuracy curves of VGG-10 with the UCM dataset under three different training settings.

to generalize. Meanwhile, increasing the number of generative samples did not always improve the accuracy of the model. When the ratio of the generative sample set was 2, the improvement was satisfactory. Therefore, we set the ratio to 2 to analyze the changes in the accuracy curves.

Second, we analyzed the changes in accuracy curves during training. In Fig. 9, it can be seen that the accuracy curves for the VGG-10 with two different training sets increased quickly at the beginning. After the number of iterations was greater than 7,000, the two curves remained stable. In Fig. 9(a) and (c), the VGG-10 (with generative and original samples) converged to higher classification accuracy. In Fig. 9(b), the two accuracy curves showed similar precision. The experimental results demonstrated that the generative samples were effective in enhancing the classification performance of the CNN-based method.

Third, we compared the effect of data augmentation. We selected several data augmentation and regularization methods, including Cycle-GAN [**?**], graphical GAN [**?**], SamplePairing [**?**], and virtual adversarial training (VAT) [**?**], as comparison methods. These are described further as follows:

(a) Cycle-GAN [**?**], which is a variant of GAN, which executes the task of transforming an image from one domain to another. Ideally, other features of the image (anything not directly related to either domain, such as the background) should stay recognizably the same. Two kinds of pre-trained Cycle-GAN models (Cycle-GAN-Monet, and Cycle-GAN-

VanGogh) were used to generate HRRS images for data augmentation.

(b) Graphical GAN is a GAN-based model in the remote sensing community that pairs a generative network with a recognition network. Both of them are adversarially trained with a discriminative network. Graphical-GAN has advantages for synthesizing multiple categories of high-quality remote sensing images for data augmentation.

(c) SamplePairing [**?**] is a new data augmentation method in machine learning that synthesizes a new sample from one image by overlaying another image randomly chosen from the training data. By using two images randomly selected from the training set, the method can generate N^2 new samples from N training samples.

(d) VAT [**?**] is a regularization method based on virtual adversarial loss, which is a new measure of local smoothness of the conditional label distribution given input. Virtual adversarial loss is defined as the robustness of the conditional label distribution around each input data point against local perturbation. Because the directions in which we smooth the model are only "virtually" adversarial, this is called virtual adversarial training (VAT).

The 20% UCM dataset was used as labeled samples, the VGG-10 was set as the baseline method, and the number of training epochs was 300. The proposed method generated two types of labeled samples for data augmentation, while Cycle-GAN and Graphical GAN were set to the same settings. SamplePairing created a synthesized dataset with N^2 new samples. In VAT, the virtual adversarial loss was set to the cross-entropy loss to improve of the generalization ability of the classification model. The classification results of the VGG-10 combined with different data augmentation or regularization methods on the test set are shown in Table 7.

In Table 7, the VGG-10 without any data augmentation and regularization methods achieved an OA of 66.15%, a kappa of 0.645, and an F1_score of 0.658. Cycle-GAN, SamplePairing, graphical GAN, and VAT were able to

Table 7: **The classification results of the data augmentation methods and the data regularization method**

Method Name	OA(%)	Kappa	F1_score
Without DataAug	66.15±1.45	0.645±0.014	0.658±0.007
Cycle-GAN-Monet [?]	69.70±0.74	0.682±0.021	0.688±0.015
Cycle-GAN-VanGogh	70.07±1.25	0.693±0.026	0.701±0.021
Graphical GAN [?]	71.43±2.31	0.709±0.020	0.711±0.007
Sampleparing [?]	69.52±2.12	0.687±0.023	0.691±0.012
VAT [?]	73.39±1.73	0.721±0.017	0.727±0.009
The proposed method	73.59±0.35	0.721±0.012	0.732±0.007
VAT & the proposed method	77.97±1.38	0.769±0.018	0.777±0.011

improve their test performances. Graphical GAN, which was specially designed for HRRS image generation, achieved better performance than the two types of Cycle-GAN-based methods. Meanwhile, VAT achieved the best results in the comparison (i.e., existing) methods. The proposed method generated the best results of all the examined methods, with an OA of 73.59%, a kappa of 0.721, and an F1_score of 0.732. In addition, we combined our approach with VAT to boost the classification performance, which was further improved to an OA of 77.97%, kappa of 0.769, and F1_score of 0.777. The experimental results demonstrated that the generated samples of the proposed method were effective for data augmentation. The combination of regularization and generated samples could further improve the generalization ability of the classifier.

5. Conclusion

Focusing on the poor classification performance of the CNN-based methods for the case of limited samples and the insufficient manually labeled datasets in remote sensing, this paper presented a new *GAN-based remote sensing image generation method* for scene classification (denoted as *GAN-RSIGM*). The proposed method can be used not only for scene classification, but also to generate an annotated dataset with specific label information. The Wasserstein distance is used to measure the difference between the real data distribution and the generator distribution. The auxiliary classifier, which is used to recognize the

category information of the generative and real samples, guides the generator to produce consistent and distinct samples.

Extensive experiments were conducted to verify the effectiveness of the proposed method on two public datasets resulting in a number of important findings. First, the experimental results indicated that the GAN-RSIGM was easy to train. The generative samples produced by the generator contained specific labels and diversity. Second, in the evaluation of the classification ability, the proposed method outperformed the other CNN-based methods by approximately 4% for the UCM dataset and by approximately 2.2% on the NWPU-RESISC45 dataset. Third, the impact of the generative samples on the classification accuracy was analyzed, and the experimental results confirmed that the generative samples were able to enhance the classification ability.

In our future studies, we plan to implement more complex neural networks and find an optimal training category to promote the development of the remote sensing generation method, which will enable us to obtain massive HRRS images and generate more high-resolution samples for scene classification.

Acknowledgment

We gratefully acknowledged the support of the National Natural Science Foundation of China (Grant No. U1711266 and No. 41925007); the GF Innovative Research Program; the Fundamental Research Founds for National University, China University of Geosciences (Wuhan).