applied sciences

MDPI

Article

# Generative Adversarial Networks for Zero-Shot Remote Sensing Scene Classification

**Zihao Li [1,2,3], Daobing Zhang [1,2,*], Yang Wang [1,2], Daoyu Lin [1,2] and Jinghua Zhang [1,2]**

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China; lizihao191@mails.ucas.ac.cn (Z.L.); ywang1@mail.ie.ac.cn (Y.W.); lindy@aircas.ac.cn (D.L.); zhangjh004940@aircas.ac.cn (J.Z.)
[2] Key Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China
[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: dbzhang@mail.ie.ac.cn

**Abstract:** Deep learning-based methods succeed in remote sensing scene classification (RSSC). However, current methods require training on a large dataset, and if a class does not appear in the training set, it does not work well. Zero-shot classification methods are designed to address the classification for unseen category images in which the generative adversarial network (GAN) is a popular method. Thus, our approach aims to achieve the zero-shot RSSC based on GAN. We employed the conditional Wasserstein generative adversarial network (WGAN) to generate image features. Since remote sensing images have inter-class similarity and intra-class diversity, we introduced classification loss, semantic regression module, and class-prototype loss to constrain the generator. The classification loss was used to preserve inter-class discrimination. We used the semantic regression module to ensure that the image features generated by the generator can represent the semantic features. We introduced class-prototype loss to ensure the intra-class diversity of the synthesized image features and avoid generating too homogeneous image features. We studied the effect of different semantic embeddings for zero-shot RSSC. We performed experiments on three datasets, and the experimental results show that our method performs better than the state-of-the-art methods in zero-shot RSSC in most cases.

**Keywords:** remote sensing scene classification; zero-shot learning; generative adversarial network

check for updates

**Citation:** Li, Z.; Zhang, D.; Wang, Y.; Lin, D.; Zhang, J. Generative Adversarial Networks for Zero-Shot Remote Sensing Scene Classification. *Appl. Sci.* **2022**, *12*, 3760. https://doi.org/10.3390/app12083760

Academic Editors: Qizhi Xu, Jin Zheng and Feng Gao

Received: 2 March 2022
Accepted: 6 April 2022
Published: 8 April 2022

## 1. Introduction

Over the past decades, the epoch of remote sensing big data arrived, and remote sensing technology has made excellent progress [1]. It has played an important role in environmental monitoring [2], urban construction planning [3], and land use classification [4]. Currently, the accuracy of remote sensing image classification is very high. Most of the current remote sensing scene classification approaches rely on supervised learning [4–6], which can obtain excellent results when trained on large-scale datasets [7]. Usually, it requires labeling a lot of image data for each remote sensing image class. When we classify an image, the classifier does not work well if the category corresponding to the image is not in the training set, which is referred to as a zero-shot problem [8]. With the explosive growth of remote sensing image classes, we will encounter many new classes, and it is unrealistic to collect sufficient image samples for each class. It makes sense to identify the unseen class images when the training set does not contain the corresponding class, for instance, a rare aircraft species.

In order to overcome these problems in remote sensing images, it is of great interest to study zero-shot learning methods for RSSC. Zero-shot classification methods for remote sensing images are designed to address the classification for unseen category images, and more and more researchers are focusing on this topic. This can alleviate manual labeling, reduce a lot of labor, and make the existing classification model more scalable.

Humans can recognize about 30,000 object categories of information [8], and humans can transfer their knowledge to identify new classes when only textual descriptions of the new classes are available. The key of zero-shot learning to identify unseen classes is that zero-shot learning uses semantic and visual information from all image classes. Semantic information builds a bridge of knowledge transfer between seen and unseen categories, breaking the boundary of category mutual exclusion between training and testing sets.

The concept of zero-shot learning can be traced back to 2008 when Larochelle first introduced zero-shot learning [8]. Lampert et al. [9] first extended this approach to the field of computer vision. We can summarize the ZSL methods into two types: The first is embedding-based methods that map image features and semantic embeddings into one space and perform metric learning for ZSL. Zhang et al. [10] solved the zero-shot learning task by using the semantic similarity embedding (SSE) method. SSE fuses multiple semantic information into the same semantic space and calculates the similarity of them. Frome et al. [11] used a novel visual-semantic embedding approach to use annotated image data and semantic information from the unannotated text to recognize visual objects. Li et al. [12] presented a dual visual-semantic mapping framework for zero-shot classification (DMaP), which studies the relationship between the semantic space manifold and the transferability of visual-semantic mapping. However, they cannot work well in the generalized zero-shot learning (GZSL) setting because they overfit to seen classes. The second is GAN-based methods which generate fake image features for unseen classes and train KNN [13] or softmax classifier to perform ZSL. Xian et al. [14] first used generative adversarial networks (GANs) [15] to convert semantic features to visual features, providing a new idea for zero-shot learning. Felix et al. [16] mainly added a cycle consistency loss term, allowing GAN to generate more realistic images. Li et al. [17] introduced soul samples for generative zero-shot learning and used the cascade classifier to classify the unseen class. Yu et al. [18] presented an episode-based training pattern to improve the model's performance for zero-shot learning. Generative adversarial networks (GANs) [15] is a popular zero-shot classification method. GAN-based methods generally work better than embedding-based methods.

For the remote sensing field, Li et al. [19] first proposed a zero-shot remote sensing scene classification (RSSC) method called ZSL-LP. ZSL-LP constructs a semantic-directed graph, then uses a label-propagation algorithm for zero-shot classification. Quan et al. [20] introduced a novel zero-shot RSSC method, which relies on Sammon embedding and spectral clustering. They modified semantic feature space class prototypes by Sammon embedding, which ensures the consistency with the visual feature space class prototypes. Wang et al. [21] introduced a distance-constrained semantic autoencoder for zero-shot remote sensing scene classification. Sumbul et al. [22] conducted a zero-shot study for fine-grained remote sensing image recognition. They first learned a compatible function and then showed how to transfer knowledge for unseen classes. Although these approaches have achieved promising results, most are devoted to designing visual-semantic embedding models only with seen classes, in which it is difficult to guarantee an excellent extension to unseen classes. In addition, these models trained only with seen data tend to misclassify unseen test instances into the seen category, generating a special imbalanced classification problem. Generative adversarial networks can alleviate the above issues to some extent [14,17,23]. As far as we know, no one has tried to apply generative adversarial networks for zero-shot RSSC. The GAN-based method designed for ordinary images cannot apply to remote sensing images well because remote sensing image scenes are complex. Remote sensing images generated by the GAN model may have instability and mode collapse problems. In addition, remote sensing images have inter-class similarity and intra-class diversity. Overall, zero-shot RSSC deserves more exploration.

We propose a novel approach for zero-shot RSSC with the mentioned considerations. Since the remote sensing image dataset does not directly provide class attribute information, we used four natural language processing models pre-trained on Wikipedia to obtain word vectors as the semantic information we need. Word2vec [24], Glove [25], Fasttext [26], and Bert [27] are the four natural language processing models. Because of the complexity of remote sensing scenes, we employed the conditional Wasserstein generative adversarial network [28] for generating image features directly instead of images to avoid instability and mode collapse problems in training. We trained a generator that can generate an arbitrary number of class image features from class semantic information, converting the zero-shot classification problem into a traditional classification problem. Following the characteristics of remote sensing images, we used the classification loss, semantic regression module, and class-prototype loss to constrain the generator to ensure the generator can generate image features close to the real image with the class semantic information. Our model is referred to as CSPWGAN. The classification loss is used to preserve inter-class discrimination. We used a semantic regression module to ensure that the image features generated by the generator can represent the semantic features. We introduced class-prototype loss when training the generator to constrain itself to ensure intra-class diversity of the synthesized image features and avoid generating too homogeneous image features. The contributions of this study are as follows:

1. We trained a generator that can generate class image features close to the real image through class semantic information. We propose well-designed modules to constrain the generator, including classification loss module, class-prototype loss module, and semantic regression module. To the best of our knowledge, we are the first to employ generative adversarial networks for zero-shot remote sensing scene classification (RSSC);
2. We explored the effect of different semantic embeddings for zero-shot RSSC. Specifically, we investigated various natural language processing models, i.e., Word2vec, Fasttest, Glove, and Bert, to extract semantic embeddings for each class either from the class name or from the class sentence descriptions. Our conclusion may help future work in understanding and choosing semantic embeddings for zero-shot RSSC;
3. We conduct experiments on three benchmark datasets, UCM21 [29], AID30 [30] and NWPU45 [7]. The experimental results show that our method performs better than most state-of-the-art methods in zero-shot RSSC.

## 2. Methods

In this section, we first define the remote sensing image zero-shot classification task. Then we present our generative framework for zero-shot RSSC. Finally, we introduce each part of the model in detail.

### 2.1. Problem Definition

Let $D = \{x^i, y^i, e(y^i)\}_{i=1}^{N}$ where $D$ denotes the dataset, $x$ is the features of remote sensing images, $y$ is the class label of remote sensing images, $e(y)$ denotes the classes semantic feature. The dataset $D$ is split into the seen and unseen datasets. Let $D_s = \{x_s^i, y_s^i, e(y_s^i)\}_{i=1}^{N_s}$ represent the seen datasets and $D_u = \{x_u^i, y_u^i, e(y_u^i)\}_{i=1}^{N_u}$ represent the unseen datasets. Class label set $y$ includes the seen class label set $y_s$ and unseen class label set $y_u$. The seen class label set $y_s$ is disjointed $y_u$. The relationship can be represented by $y = y_s \cup y_u$ and $y_s \cap y_u = \phi$. The purpose of the ZSL task is to predict the class label $y_u$, and the purpose of the GZSL task is to predict the class label $y$.

### 2.2. Overall Framework

In this section, we present the overall framework of our CSPWGAN model. As shown in Figure 1, our model is composed of two parts, including the Class Semantic Feature Representation Module and the Feature Generation Module.

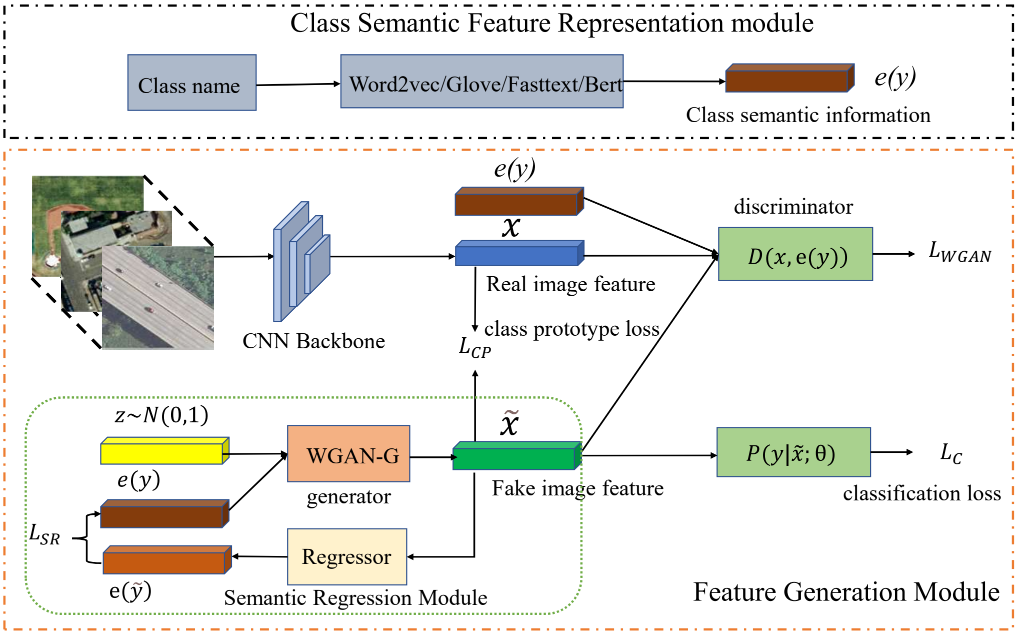


**Figure 1.** The framework of our proposed CSPWGAN model for zero-shot RSSC.

**Class Semantic Feature Representation Module.** Typical zero-shot task datasets usually contain class attribute vectors [9]. Since there are no class attribute vectors in the remote sensing image dataset, we used the natural language pre-training model to generate the word vectors we need.

**Feature Generation Module.** We used a generative model to train a generator that can generate class image features by class semantic features, which can convert the zero-shot remote sensing scene classification problem into a traditional classification problem and avoid the hubness problem [31]. In addition, using generative adversarial networks can prevent the class imbalance problem under the GZSL setting. Meanwhile, we considered that remote sensing images have inter-class similarity and intra-class diversity. We used the classification loss, semantic regression module, and class-prototype loss to constrain the generator. The classification loss was used to preserve inter-class discrimination. We used semantic regression module to ensure that the image features generated by the generator can represent the semantic features. To ensure intra-class diversity of the synthesized image features and avoid generating too homogeneous image features, we introduced class-prototype loss to constrain the generator. The key to solving this problem is to obtain a high-performance generator, and multiple losses to jointly constrain the generator to obtain better results.

### *2.3. Class Semantic Feature Representation Module*

Zero-shot learning aims to solve the problem of recognizing unseen classes that cannot be accomplished by traditional supervised learning. The key is that zero-shot learning uses not only visual features but also class semantic features. Class semantic information builds a bridge of knowledge transfer between seen and unseen categories, breaking the boundary of category mutual exclusion between training and testing sets. There are usually two ways to extract semantic features: attribute vectors and word embeddings.

Attributes vectors are manually annotated using expert knowledge and share attribute space among all classes of objects. It is the most common and efficient method of semantic feature construction. Attribute vectors can leverage prior human knowledge with good interpretability and accuracy. At the same time, the disadvantage of attribute vectors is that they are highly dependent on manual annotation and are challenging to annotate in the absence of prior human knowledge.

Using natural language pre-training models to generate the word vectors we need has the advantage of being fast, simple, and does not require prior knowledge. Word2vec [24], Glove [25], Fasttext [26], and Bert [27] are the most common methods used. These methods can be trained with open-source corpora (e.g., Wikipedia), which significantly saves the cost of manual annotation. We can use them to convert classes name or class text descriptions into word vectors. For example, we can use Word2vec to convert remote sensing image classes name into 300- or 500-dimensional word vectors and use Bert to convert the description of remote sensing image classes into 768-dimensional word vectors.

### *2.4. Feature Generation Module*

Since there are no samples of unseen categories in the training data, it is difficult to train a classifier for unseen categories. Our method synthesizes unseen classes' image visual features through class semantic information and noise vectors. We trained a classifier that can classify unseen class images based on the synthesized image features and their corresponding unseen class labels. Our approach was inspired by [14]. CSPWGAN is composed of a generative adversarial network. The generative adversarial network model converts the zero-shot RSSC problem into a traditional classification problem.

We used the conditional generative adversarial net [32] as our baseline model. The network is composed of a generative model $G$ and a discriminative model $D$ that compete in a two-player minimax game. $G$ generates the visual image representation $\hat{x}$ by its semantic feature $e(y)$ and a noise vector sampled from a normal distribution $z \sim N(0, I)$. The generation process can be represented as: $y \times z \rightarrow x$. We used the semantic feature $e(y)$ and a noise vector $z$ as inputs of $G$, and $\hat{x}$ of the class label $y$ as outputs of $G$. $\theta_G$ denotes the parameters of $G$. The discriminative model $D$ can be represented as: $e \times x \rightarrow [0, 1]$, $\theta_D$ denotes the parameters of $D$. $D$ aims to accurately distinguish the real image visual feature $x$ from the generated visual features $\hat{x}$. $\theta_D$ denotes the parameters of the discriminative model $D$. $G$ aims to cheat the discriminator $D$ by generating images that can be mistaken for the real ones. This is the training process for estimating the parameters of $\theta_D$ and $\theta_G$.

Our model only uses the seen class data for training, but it can also generate image features of unseen classes. Generating models are usually hard to train and not easy to stabilize. We used a stable training methods called WGAN [28], the loss function of WGAN is as follows:

$$L_{WGAN} = E[D(x, e(y)|\theta_D)] - E[D(\hat{x}, e(y)|\theta_D)] - \lambda E[(\|\bigtriangledown_{\tilde{x}} D(\tilde{x}, e(y))\|_2 - 1)^2] \quad (1)$$

where $E[.]$ denotes the expectation function. $\hat{x} = G(z, e(y)|\theta_G)$, $\tilde{x} = \beta x + (1 - \beta)\hat{x}$ with $\beta \sim U(0, 1)$. Both $G$ and $D$ are multilayer networks. $\lambda$ represents the penalty factor. We set $\lambda = 10$ in this study.

To achieve good optimization, we applied classification loss, semantic regression module and class-prototype loss to train the network.

**Classification Loss.** Because of the complexity of remote sensing images, the WGAN model cannot ensure that the generated samples are discriminative enough. Generating inaccurate samples results in the bad performance of the classifier. To alleviate this issue, we added a classifier to identify whether the generated image features can be classified into the correct class. By doing this, the generated features are more discriminative. We added a simple classifier that uses the negative log-likelihood:

$$L_C = -E_{\hat{x} \sim p_{\hat{x}}}[logP(y|\hat{x}; \theta)] \quad (2)$$

where $\hat{x} = G(z, e(y)|\theta_G)$, the class label of $\hat{x}$ is $y$, and $P(y|\hat{x}; \theta)$ represents the probability that the class label of $\hat{x}$ predicted by the classifier is its true label $y$. $\theta$ is pre-trained on the seen classes dataset. It can be represented as:

$$P(y|\hat{x}; \theta) = \frac{\exp((\theta)^T \hat{x})}{\sum_{c \in y} \exp((\theta)^T \hat{x})} \quad (3)$$

**Semantic Regression Module.** We used class semantic features to guide the generator to generate image features for the corresponding class. The generator is trained using seen class semantic features and visual features in the training phase, and using unseen class semantic features synthesizes unseen visual features in the testing phase. Finally, we converted the zero-shot classification problem into a traditional classification problem. Our model relies on a high-performance generator that aims to generate image features similar to real image features. We used the semantic features of the unseen class to synthesize image features; if it differs significantly from the real image feature distribution, the synthesized image features cannot represent the real image features of the class. The classifier trained with synthetic image features of unseen categories misclassifies the real unseen class images. We must ensure that the image features generated by the generator are similar to the real image features, which allows us to achieve higher recognition results for the unseen categories. We were inspired by [16], and used the semantic regression module to constrain the generator. This can be represented as:

$$L_{SR} = E_{x \sim P_r}[||e(y) - R(G(z, e(y)|\theta_G)|\theta_R)||_2^2] \tag{4}$$

where $R$ denotes the regressor, and $\theta_R$ denotes the parameters of $R$. $\theta_R$ is pre-trained on seen classes using the following function:

$$L_R = \min_{\theta_R} ||e(y) - R(x|\theta_R)||_2^2 \tag{5}$$

**Class-Prototype Loss.** In remote sensing images, the features of different sample images in the same class vary greatly and have diversity within the class. We used classification loss, the semantic regression module cannot guarantee that the image features synthesized by the generator are diverse. To ensure that the synthesized image features are more consistent with the distribution of real remote sensing image features and avoid the generated image features being too homogeneous, we introduced class-prototype loss in training to constrain the generator. We were inspired by [17]. First, we clustered each class of real remote sensing image features to obtain $k$ clusters. We obtained class prototype sample image features by averaging all sample image features in each group, and can obtain $k$ class prototype samples for each real remote sensing image class. In the training process, we hoped that the image features synthesized by the generator would be close to at least one class prototype vector in the same class.

Let $X_n^y$ denote the $n$th cluster of category $y$, each category has $k$ clusters. In this study, we set $k = 5$ for simplicity. Let $p_n^y$ denote the $n$th prototype vector of category $y$, each category has $k$ prototype vectors. The $p_n^y$ is defined as:

$$p_n^y = \frac{1}{N_n^y} \sum_{x_i \in X_n^y} x_i \tag{6}$$

where $N_n^y$ denotes the number of samples in cluster $X_n^y$. For the generated fake features $\hat{x}$, we also define the prototype vector $\hat{p}_n^y$ as:

$$\hat{p}_n^y = \frac{1}{N_n^y} \sum_{\hat{x}_i \in \hat{X}_n^y} \hat{x}_i \tag{7}$$

We hope that each sample $\hat{x}$ generated for the category $y$ is be close to at least one prototype vector $p^y$. This loss can be defined as:

$$L_1 = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in [1,k]} \left\| \hat{x}_i - p_j^y \right\|_2^2 \tag{8}$$

where $n$ represents the number of generated samples for the category $y$ and $k$ represents the number of prototype vectors per class. We also hope that the prototype vector of generated fake features $\hat{p}_n^y$ is close to at least a real prototype vector $p_n^y$ for the same class, which is formulated as:

$$L_2 = \frac{1}{n_y} \sum_{y=1}^{n_y} \min_{j \in [1,k]} \left\| \hat{p}_j^y - p_j^y \right\|_2^2 \tag{9}$$

where $n_y$ represents the number of all categories. Our class-prototype loss can be expressed by the following equation:

$$L_{CPL} = \lambda_1 L_1 + \lambda_2 L_2 \tag{10}$$

With this class-prototype loss for constraint, our model can avoid generating single-view image features and ensure the intra-class diversity of the synthesized image features for remote sensing images.

Finally, our model was trained with the following overall loss function:

$$\min_{\theta_G} \max_{\theta_D} L_{WGAN} + \alpha L_C + \beta L_{SR} + \lambda L_{CPL} \tag{11}$$

where $\alpha$, $\beta$, and $\lambda$ are hyperparameters of the model to balance the importance of each term.

### *2.5. Training and Testing*

In the training phase, the generator was trained using class semantic features and visual features from the seen categories. We used classification loss, semantic regression module, and class-prototype loss to constrain the generator. We used classification loss to allow the generator to learn to generate discriminative image features. We used the semantic regression module to ensure that the generated image features represent semantic features. Class-prototype loss ensures that the synthetic image features are more consistent with the distribution of real remote sensing image features. The model is trained with Equation (11).

After the training is complete, we repeated the generator to generate an arbitrary number of image features for each unseen category. We obtained a training set of synthetic images. In the ZSL task, we can classify the unseen images by the classifier trained on this dataset. In the GZSL task, we added the synthetic images dataset to the seen class dataset and trained a classifier to recognize both seen and unseen class images. We can choose a softmax classifier or KNN, etc. In this study, we used the softmax classifier.

## 3. Results

In this section, we designed experiments to evaluate our method. First, we introduced the remote sensing image datasets used in our experiments. Second, we introduced our experimental setup. Third, we exploited the effect of our method on different semantic vectors. Fourth, we compared our approach with the current state-of-the-art zero-shot RSSC method. Finally, we performed ablation studies.

### *3.1. Dataset*

We selected three datasets in the most widely used remote sensing images classification datasets, namely UCM21 [29], AID30 [30] and NWPU45 [7].

The UCM21 dataset has 21 classes, and each scene class consists of 100 images. For each scene class, the size of the image is 256 × 256 pixels. The AID30 dataset has 30 categories, each category has 200 to 400 samples, and each sample has a pixel size of 600 × 600. The NWPU45 dataset has 45 classes, and each scene class consists of 700 images. For each scene class, the size of the image is 256 × 256 pixels. As shown in Figure 2, we can see the diversity of images within the same class.

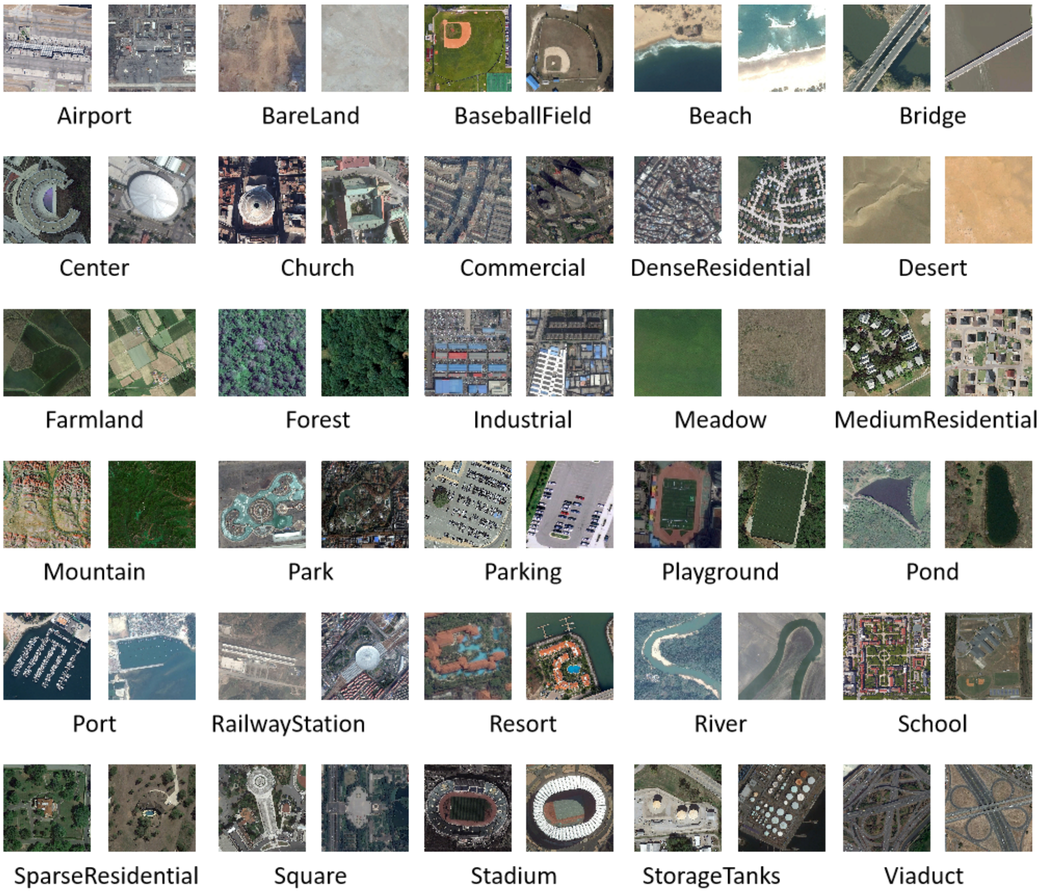


**Figure 2.** Some remote sensing sample images in UCM21, AID30, and NWPU45 datasets.

### 3.2. Evaluation Protocols

In the ZSL task, we used the class average accuracy [14] as the performance evaluation metric, i.e., the classification accuracy within a class is first counted for each class. Then the class average accuracy is calculated by finding the mean value.

In the GZSL task, harmonic mean accuracy [14] is used to evaluate the classification effect. The formula for calculating the harmonic mean accuracy is as follows:

$$H = \frac{2 \times u \times s}{u + s} \tag{12}$$

where $u$ denotes the accuracy of unseen classes, $s$ denotes the accuracy of seen classes, and $H$ denotes the harmonic mean accuracy.

### 3.3. Implementation Details

We first must extract image features, obtain the corresponding class semantic information, and then use the GAN network for training. We used the ResNet-101 model pre-trained on ImageNet to extract 2048-dimensional features of remote sensing images. We obtained the word vectors of the corresponding class by using Word2vec, Glove, Fasttext, and Bert. We implemented our model via multilayer neural networks. The generator $G$ contains a 4096 nodes hidden layer activated by LeakyRelu [33], the output layer has 2048 nodes activated by Relu [34]. The discriminator $D$ also contains a 4096 nodes hidden layer activated by LeakyRelu, and the output layer has no activation. The regressor $R$ contains a 2048 nodes hidden layer, the dimensionality of the output layer is the same as the semantic information dimension. We used the adam [35] optimizer to optimize our model , $lr = 0.0001$, $\beta_1$ = 0.5, and $\beta_2$ = 0.999. In this study, we set $\alpha = 0.001$, $\beta = 10$, $\gamma = 1$,

$\gamma_1 = 0.01$, $\gamma_2 = 0.001$. The above algorithms were implemented using the PyTorch, and the experiments were completed on four RTX2080s.

### *3.4. Ablations on Different Word Vectors*

We used four natural language processing models pre-trained on Wikipedia to obtain word vectors for remote sensing datasets. Moreover, we obtained different dimensional word vectors for each method as class semantic features. To find the word vectors with the best results, we compared the word vectors obtained by these four methods with different dimensions. The classification results on the UCM21 dataset are shown in Figure 3.

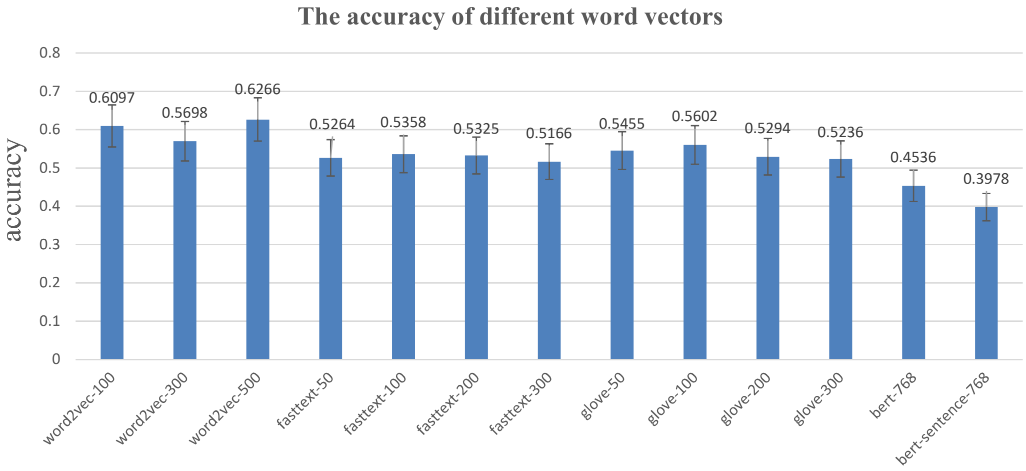


**Figure 3.** The classification accuracy of different word vectors on the UCM21 dataset.

The Word2vec model pre-trained on Wikipedia has three dimensions, i.e., 100, 300, and 500, and they correspond to an accuracy of 0.6097, 0.5698, and 0.6266, respectively.

The Glove model pre-trained on Wikipedia has four dimensions, i.e., 50, 100, 200, and 300, and they correspond to an accuracy of 0.5455, 0.5602, 0.5294, and 0.5236, respectively.

The Fasttext model pre-trained on Wikipedia has four dimensions, i.e., 50, 100, 200, 300, and they correspond to an accuracy of 0.5264, 0.5358, 0.5325, and 0.5166, respectively.

We used Bert in two ways to obtain word vectors, one using class names and the other using class description texts, which correspond to an accuracy of 0.4536 and 0.3978, respectively.

We also explored the effect of different word vectors in GZSL tasks. We used Word2vec-500, Glove-100, Fastetext-100, and Bert-768 as semantic information In Table 1. When we used the Word2vec method to extract 500-dimensional word vectors as semantic information for remote sensing image datasets, we achieved the best results in both ZSL and GZSL tasks.

**Table 1.** The experimental results of different word vectors. Bold font denotes the best result.

| Word2vec | Glove | Fasttext | Bert | UC Merced Land Use Dataset | | | |
|---|---|---|---|---|---|---|---|
| | | | | acc | $u$ | $s$ | $H$ |
| ✓ | | | | **0.6266** | **0.4869** | 0.7348 | **0.5857** |
| | ✓ | | | 0.5602 | 0.4303 | **0.7389** | 0.5439 |
| | | ✓ | | 0.5358 | 0.4007 | 0.6875 | 0.5063 |
| | | | ✓ | 0.4536 | 0.3507 | 0.7138 | 0.4704 |

### *3.5. Comparison with State-of-the-Art*

To show the superiority of CSPWGAN, we present a comparison of our method with some previous methods. We chose these baselines: SSE [10], DMaP [12], SAE [36], ZSL-LP [19], ZSC-SA [20], DASE [21], VSC [37], VSOP [38], f-CLSWGAN [14], CYCLEWGAN [16], and RBGN [39].

SSE [10] fuses multiple semantic information into the same semantic space and calculates the similarity of them. DMaP [12] studies the relationship between the semantic space manifold and the transferability of visual-semantic mapping. SAE [36] is based on learning a Semantic autoencoder for solving the label prediction problem. SAE adds a constraint from visual mapping to semantic features to alleviate the domain drift problem. ZSL-LP [19] constructs a semantic-directed graph, then uses a label-propagation algorithm for zero-shot classification. ZSC-SA [20] is a novel zero-shot RSSC method, which relies on Sammon embedding and spectral clustering. VSC [37] employs a novel visual architecture constraint for ZSL. VSOP [38] proposes to match latent visual and semantic representations in a shared subspace. DASE [21] proposes a distance-constrained semantic autoencoder to handle ZSRSSC. RBGN [39] employs an adversarial attack and bidirectional generation into GZSL to improve the generalizability and robustness of the model.

For a fair comparison with prior methods, we divided the dataset into four seen/unseen ratios, and each ratio was randomly divided 25 times, and 25 zero-shot classification experiments were performed. The average accuracy of the categories is taken as the result. Regarding the splitting ratio setting of the remote sensing dataset, our method is consistent with previous zero-shot classification methods for remote sensing images, such as articles DSAE, ZSC-SA, and ZSL-LP. Tables 2–4 show the classification results for UCM21, AID30, and NWPU45 datasets. As shown in Tables 2–4, our method significantly improved the classification accuracy compared with the state-of-the-art(sota) approaches for zero-shot RSSC in most cases. On the UCM21 dataset, our method improved 4.03%, 8.69%, 9.58%, and 5.99% under different seen/unseen ratios (e.g., 16/5, 13/8, 10/11, and 7/14, respectively). On the AID30 dataset, our method improves 2.37%, 2.61%, 1.05%, and 1.78% under different seen/unseen ratios (e.g., 25/5, 20/10, 15/15, and 10/20, respectively). On the NWPU45 dataset, our method improves 0.24% and 1.45% under two seen/unseen ratios (e.g., 25/20 and 20/25, respectively). The standard deviation of our method is smaller than DSAE in most cases, which also proves the superiority of our method. In conclusion, the results show that our method is more adapted for remote sensing images. The results also show that our method works better with a higher ratio of unseen classes and higher accuracy.

**Table 2.** With different seen/unseen ratios, the average ZSL classification accuracies and standard deviation (%) of our CSPWGAN model and the sota methods on the UCM21 dataset. Bold font denotes the best result.

| Method | 16/5 | 13/8 | 10/11 | 7/14 |
|---|---|---|---|---|
| SSE [10] | 35.59 ± 5.90 | 23.42 ± 3.81 | 17.07 ± 3.56 | 10.82 ± 2.10 |
| DMaP [12] | 48.92 ± 8.71 | 30.91 ± 4.77 | 22.99 ± 4.81 | 17.30 ± 3.04 |
| SAE [36] | 49.50 ± 8.42 | 32.71 ± 6.49 | 24.04 ± 4.36 | 18.63 ± 2.76 |
| ZSL-LP [19] | 49.01 ± 8.85 | 31.26 ± 5.09 | 23.28 ± 4.13 | 17.55 ± 2.90 |
| ZSC-SA [20] | 50.42 ± 8.84 | 34.12 ± 6.10 | 24.68 ± 4.22 | 18.38 ± 2.74 |
| VSC [37] | 55.91 ± 11.77 | 36.26 ± 7.31 | 25.97 ± 5.79 | 19.53 ± 3.05 |
| VSOP [38] | 46.48 ± 7.83 | 29.81 ± 4.56 | 21.97 ± 4.11 | 16.14 ± 2.59 |
| f-CLSWGAN [14] | 56.97 ± 11.06 | 36.47 ± 6.28 | 27.89 ± 4.99 | 19.34 ± 3.96 |
| CYCLEWGAN [16] | 58.36 ± 10.04 | 36.81 ± 5.53 | 28.37 ± 4.53 | 21.15 ± 3.51 |
| RBGN [39] | 57.93 ± 11.56 | 36.95 ± 5.99 | 27.74 ± 5.16 | 20.67 ± 3.95 |
| DSAE [21] | 58.63 ± 11.23 | 37.50 ± 7.79 | 25.59 ± 5.24 | 20.18 ± 3.07 |
| CSPWGAN (our) | **62.66 ± 10.79** | **46.19 ± 5.52** | **35.17 ± 4.93** | **26.17 ± 3.87** |

**Table 3.** With different seen/unseen ratios, the average ZSL classification accuracies and standard deviation (%) of our CSPWGAN model and the sota methods on the AID30 dataset. Bold font denotes the best result.

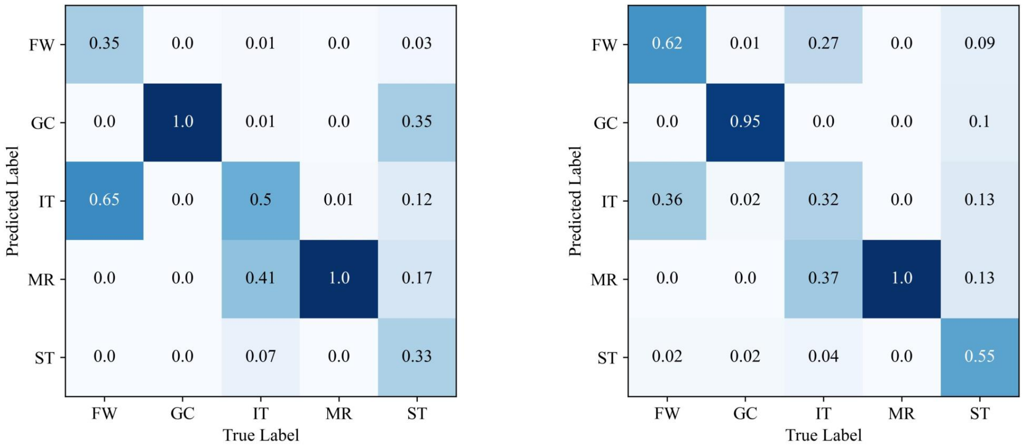
| Method | 25/5 | 20/10 | 15/15 | 10/20 |
|---|---|---|---|---|
| SSE [10] | 46.11 ± 7.21 | 30.28 ± 4.90 | 19.94 ± 2.43 | 12.73 ± 1.27 |
| DMaP [12] | 43.40 ± 7.29 | 28.29 ± 4.78 | 19.38 ± 2.62 | 11.56 ± 1.29 |
| SAE [36] | 47.34 ± 8.42 | 32.12 ± 4.45 | 23.73 ± 3.28 | 13.77 ± 1.17 |
| ZSL-LP [19] | 46.77 ± 7.65 | 30.82 ± 4.90 | 21.78 ± 3.37 | 12.97 ± 1.06 |
| ZSC-SA [20] | 50.87 ± 8.74 | 33.46 ± 5.99 | 24.41 ± 3.83 | 15.89 ± 2.03 |
| VSC [37] | 52.61 ± 8.37 | 35.85 ± 5.52 | 26.11 ± 3.76 | 17.50 ± 2.19 |
| VSOP [38] | 48.56 ± 7.90 | 32.95 ± 5.52 | 24.84 ± 3.04 | 14.03 ± 2.47 |
| f-CLSWGAN [14] | 50.68 ± 11.25 | 33.89 ± 5.72 | 24.95 ± 2.96 | 17.26 ± 3.06 |
| CYCLEWGAN [16] | 52.37 ± 10.47 | 35.94 ± 5.46 | 25.28 ± 2.66 | 17.89 ± 2.86 |
| RBGN [39] | 51.99 ± 11.32 | 36.27 ± 5.65 | 24.83 ± 3.07 | 16.83 ± 3.14 |
| DSAE [21] | 53.49 ± 8.58 | 35.32 ± 5.17 | 25.92 ± 3.92 | 17.65 ± 2.52 |
| CSPWGAN (our) | **55.86 ± 10.60** | **37.93 ± 5.26** | **26.97 ± 2.53** | **19.43 ± 3.02** |

**Table 4.** With different seen/unseen ratios, the average ZSL classification accuracies and standard deviation (%) of our CSPWGAN model and the sota methods on the NWPU45 dataset. Bold font denotes the best result.

| Method | 35/10 | 30/15 | 25/20 | 20/25 |
|---|---|---|---|---|
| SSE [10] | 33.36 ± 3.58 | 23.30 ± 2.48 | 16.88 ± 2.29 | 12.94 ± 1.46 |
| DMaP [12] | 49.53 ± 6.31 | 38.07 ± 4.83 | 28.15 ± 3.86 | 23.95 ± 2.60 |
| SAE [36] | 44.81 ± 4.73 | 35.07 ± 3.91 | 24.65 ± 3.71 | 20.77 ± 2.02 |
| ZSL-LP [19] | 47.00 ± 6.64 | 36.45 ± 4.58 | 26.71 ± 3.43 | 22.90 ± 2.47 |
| ZSC-SA [20] | 48.40 ± 6.36 | 37.55 ± 4.54 | 28.27 ± 3.47 | 23.69 ± 2.38 |
| VSC [37] | 50.68 ± 6.60 | 40.92 ± 4.59 | 30.62 ± 3.10 | 25.51 ± 2.04 |
| VSOP [38] | 45.32 ± 5.71 | 36.09 ± 4.63 | 25.44 ± 3.13 | 22.18 ± 2.00 |
| f-CLSWGAN [14] | 45.35 ± 6.37 | 38.97 ± 4.93 | 30.06 ± 2.96 | 24.31 ± 2.57 |
| CYCLEWGAN [16] | 46.87 ± 5.99 | 39.85 ± 4.71 | 31.17 ± 2.66 | 25.06 ± 2.74 |
| RBGN [39] | 44.68 ± 6.14 | 40.31 ± 4.89 | 31.91 ± 3.07 | 24.89 ± 2.44 |
| DSAE [21] | **51.52 ± 6.91** | **41.94 ± 4.61** | 31.85 ± 3.32 | 25.20 ± 2.17 |
| CSPWGAN (our) | 50.66 ± 5.86 | 41.61 ± 4.48 | **32.09 ± 2.96** | **26.65 ± 2.33** |

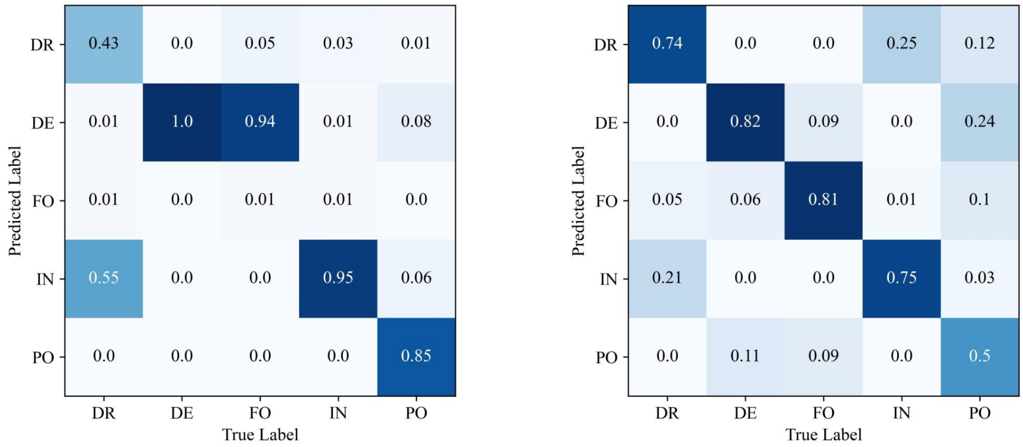
To compare the classification performance for each unseen category, we fixed the unseen classes for a given split ratio according to the DSAE setting. We present the confusion matrix of our approach and the DASE method. Notably, each column of the matrix represents an instance of the real class, and each row of the matrix represents an instance of the predicted class. On the UCM21 dataset, we have five unseen classes with the seen/unseen ratio set to 16/5, i.e., “freeway”, “golf course”, “intersection”, “medium residential”, and “storage tanks”. Figure 4 shows the confusion matrix of the DASE method and our method on the UCM21 datasets. We found that we obtained 27% and 22% improvement on the classes “freeway” and “storage tanks”, respectively. Our method has a decrease in classification accuracy on other unseen classes, but the average accuracy of the five unseen classes is higher than DASE. The average accuracy of our method improved from 63.6% to 68.8% on the five unseen categories. On the AID30 dataset, we have five unseen classes with the seen/unseen ratio set to 25/5, i.e., “dense residential”, “desert”, “forest”, “industrial”, and “pond”. Figure 5 shows the confusion matrix of the DASE method and our method on the AID30 datasets. We achieved 31% and 80% improvement on the classes “dense residential” and “forest”, respectively. The average accuracy of our method improved from 64.8% to 72.4% on the five unseen categories. On the NWPU45 dataset, we have ten unseen classes with the seen/unseen ratio set to 35/10, i.e., “airport”, "basketball court", "circular farmland", "cloud", "dense residential", "desert","harbor", "intersection", "medium residential" and "sparse residential". Figure 6 shows the confusion matrix of the DASE method and our method on the NWPU45 datasets. We achieved 2%, 5%, 60%, 16%, and 1% improvement on the classes "airport", "circular farmland", "dense residential", "medium residential" and "sparse residential", respectively.

To visually display the efficiency of CSPWGAN, we visualized the synthesized image features through our CSPWGAN method and the original image features of the unseen class in Figure 7. We used the popular t-SNE [40] tool to display the synthesized image features and the real image features. From Figure 7, we can observe that the unseen image features synthesized by our method are relatively close to the real image feature distribution, which proves that our approach is meaningful.



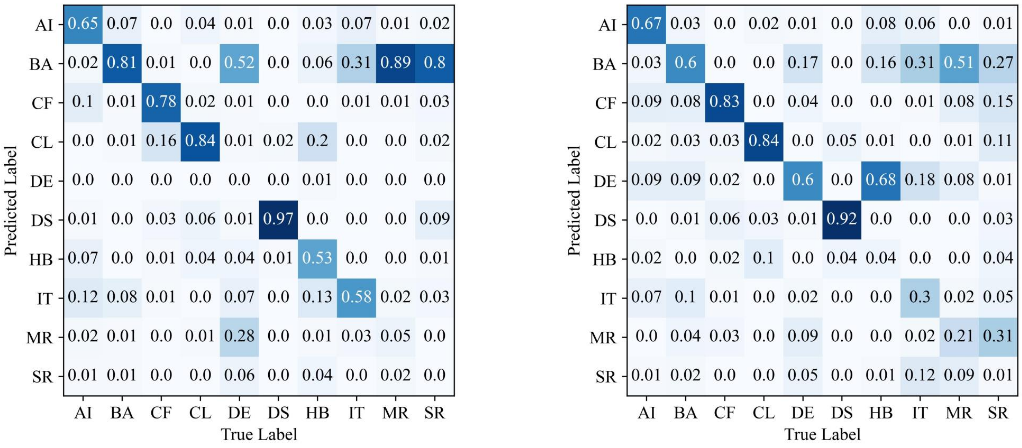
(a) DASE (b) Ours

**Figure 4.** Confusion matrix of the two methods on the UCM21 dataset. Where "FW", "GC", "IT", "MR", and "ST" denotes "freeway", "golf course", "intersection", "medium residential", and "storage tanks", respectively.



(a) DASE (b) Ours

**Figure 5.** Confusion matrix of the two methods on the AID30 dataset. Where "DR", "DE", "FO", "IN", and "PO" denotes "dense residential", "desert", "forest", "industrial", and "pond", respectively.



(a) DASE

| | AI | BA | CF | CL | DE | DS | HB | IT | MR | SR |
|---|---|---|---|---|---|---|---|---|---|---|
| AI | 0.65 | 0.07 | 0.0 | 0.04 | 0.01 | 0.0 | 0.03 | 0.07 | 0.01 | 0.02 |
| BA | 0.02 | 0.81 | 0.01 | 0.0 | 0.52 | 0.0 | 0.06 | 0.31 | 0.89 | 0.8 |
| CF | 0.1 | 0.01 | 0.78 | 0.02 | 0.01 | 0.0 | 0.0 | 0.01 | 0.01 | 0.03 |
| CL | 0.0 | 0.01 | 0.16 | 0.84 | 0.01 | 0.02 | 0.2 | 0.0 | 0.0 | 0.02 |
| DE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.01 | 0.0 | 0.0 | 0.0 |
| DS | 0.01 | 0.0 | 0.03 | 0.06 | 0.01 | 0.97 | 0.0 | 0.0 | 0.0 | 0.09 |
| HB | 0.07 | 0.0 | 0.01 | 0.04 | 0.04 | 0.01 | 0.53 | 0.0 | 0.0 | 0.01 |
| IT | 0.12 | 0.08 | 0.01 | 0.0 | 0.07 | 0.0 | 0.13 | 0.58 | 0.02 | 0.03 |
| MR | 0.02 | 0.01 | 0.0 | 0.01 | 0.28 | 0.0 | 0.01 | 0.03 | 0.05 | 0.0 |
| SR | 0.01 | 0.01 | 0.0 | 0.0 | 0.06 | 0.0 | 0.04 | 0.0 | 0.02 | 0.0 |

(b) Ours

| | AI | BA | CF | CL | DE | DS | HB | IT | MR | SR |
|---|---|---|---|---|---|---|---|---|---|---|
| AI | 0.67 | 0.03 | 0.0 | 0.02 | 0.01 | 0.0 | 0.08 | 0.06 | 0.0 | 0.01 |
| BA | 0.03 | 0.6 | 0.0 | 0.0 | 0.17 | 0.0 | 0.16 | 0.31 | 0.51 | 0.27 |
| CF | 0.09 | 0.08 | 0.83 | 0.0 | 0.04 | 0.0 | 0.0 | 0.01 | 0.08 | 0.15 |
| CL | 0.02 | 0.03 | 0.03 | 0.84 | 0.0 | 0.05 | 0.01 | 0.0 | 0.01 | 0.11 |
| DE | 0.09 | 0.09 | 0.02 | 0.0 | 0.6 | 0.0 | 0.68 | 0.18 | 0.08 | 0.01 |
| DS | 0.0 | 0.01 | 0.06 | 0.03 | 0.01 | 0.92 | 0.0 | 0.0 | 0.0 | 0.03 |
| HB | 0.02 | 0.0 | 0.02 | 0.1 | 0.0 | 0.04 | 0.04 | 0.0 | 0.0 | 0.04 |
| IT | 0.07 | 0.1 | 0.01 | 0.0 | 0.02 | 0.0 | 0.0 | 0.3 | 0.02 | 0.05 |
| MR | 0.0 | 0.04 | 0.03 | 0.0 | 0.09 | 0.0 | 0.0 | 0.02 | 0.21 | 0.31 |
| SR | 0.01 | 0.02 | 0.0 | 0.0 | 0.05 | 0.0 | 0.01 | 0.12 | 0.09 | 0.01 |

**Figure 6.** Confusion matrix of the two methods on the NWPU45 dataset. Where "AI", "BA", "CF", "CL", "DE", "DS", "HB", "IT", "MR", and "SR" denotes "airport", "basketball court", "circular farmland", "cloud", "dense residential", "desert","harbor", "intersection", "medium residential" and "sparse residential", respectively.

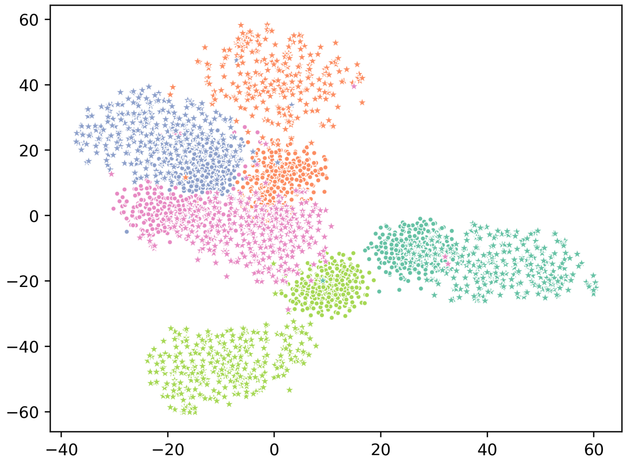


**Figure 7.** t-SNE visualizations of image features for five unseen classes on the AID30 dataset. The pentagram denotes the real features of the unseen class. The solid circle denotes the synthetic features of the unseen class. The real and synthetic features of the same unseen class have the same color.

### 3.6. Ablation Studies

Our approach is based on generative adversarial networks. Following the characteristics of remote sensing images, we used classification loss, class-prototype loss, and semantic regression loss to constrain the generator, to ensure that we can obtain a high-performance generator. We constructed ablation experiments to understand our model further and to evaluate their effects. We set $\lambda = 0$ in (12), our model without classification loss term can be referred to as CSPWGAN-$\lambda$; we set $\beta = 0$ in (12), our model without semantic regression term can be referred to as CSPWGAN-$\beta$; we set $\gamma = 0$ in (12), our model without class-prototype loss term can be referred to as CSPWGAN-$\gamma$. We executed ablation experiments on the UCM21, AID30, and NWPU45 datasets. Table 5 shows the result of ablation studies on the UCM21, AID30, and NWPU45 datasets. On the UCM21 dataset, when we removed the classification loss item, the model accuracy decreased by 2.82%; when we removed the semantic regression term, the model accuracy decreased by 2.23%; when we removed the semantic regression term, the model accuracy decreased by 2.99%. The results show the effectiveness of the three items for zero-shot RSSC. We can find the same conclusion on the AID30 and NWPU45 datasets.

**Table 5.** Ablation results on the UCM21, AID30, and NWPU45 datasets.

| Method | UCM21 | AID30 | NWPU45 |
|---|---|---|---|
| CSPWGAN-$\lambda$ | 59.84 ± 9.64 | 53.05 ± 10.07 | 47.82 ± 6.24 |
| CSPWGAN-$\beta$ | 60.43 ± 10.32 | 53.79 ± 10.34 | 48.79 ± 5.67 |
| CSPWGAN-$\gamma$ | 59.67 ± 9.85 | 52.89 ± 9.59 | 47.07 ± 5.96 |
| CSPWGAN | 62.66 ± 10.79 | 55.86 ± 10.60 | 50.66 ± 5.86 |

## 4. Conclusions

This study presents a novel method for zero-shot remote sensing scene classification (RSSC) named CSPWGAN. We are the first to apply generative adversarial networks for zero-shot RSSC. Since the remote sensing image dataset does not directly provide class attribute information, we used four natural language processing models pre-trained on Wikipedia to obtain word vectors as the class semantic information we need. We used generative adversarial networks to train a generator that can generate class image features through class semantic features, converting the zero-shot RSSC problem into a traditional classification problem. Following the characteristics of remote sensing images, we used the classification loss, semantic regression module, and class-prototype loss to constrain the generator. The classification loss was used to preserve inter-class discrimination. We used a semantic regression module to ensure the image features generated by the generator can represent the semantic features. We introduced class-prototype loss when training the generator to constrain itself to ensure intra-class diversity of the synthesized image features and avoid generating too homogeneous image features. We conducted experiments on two benchmark datasets. The results demonstrate the superiority of our proposed CSPWGAN method for remote sensing images. Our method can work better with a high ratio of unseen classes. In the experiments, we found that the class semantic information significantly affects the classification results. In future work, we will try to obtain better class semantic vectors and use a better approach to reduce classification standard deviation and improve the classification accuracy for remote sensing images.

In future work, we can also try to use active learning [41] to solve the problem of unseen remote sensing image classes in the training data. The key idea of active learning (AL) is that if a machine-learning algorithm is allowed to select the data it learns, then it can achieve higher accuracy with fewer labeled training instances. Active learning retrieves the most useful unlabeled data from a large set of unlabeled data, hands it over to a professional for labeling, and then uses that sample to train the model to improve its accuracy. Many excellent methods have emerged in the field of remote sensing and deep learning using active learning, such as [42–44]. Our approach uses semantic and visual information from all image classes. Class semantic information builds a bridge of knowledge transfer between seen and unseen categories, breaking the boundary of category mutual exclusion between training and testing sets. Active learning and our model are two different ideas for solving the problem of addressing unseen classes in the training data.

**Author Contributions:** Conceptualization, Z.L., D.Z., Y.W. and D.L.; methodology, Z.L. and D.Z.; software, Z.L., Y.W. and J.Z.; validation, Z.L., D.L. and J.Z.; formal analysis, Z.L.; investigation, Z.L.; resources, Z.L. and J.Z.; data curation, Z.L. and D.Z.; writing—original draft preparation, Z.L., Y.W. and D.L.; writing—review and editing, Z.L. and D.Z.; visualization, Z.L.; supervision, D.Z.; project administration, D.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** http://weegee.vision.ucmerced.edu/datasets/landuse.html (accessed on 7 April 2022), https://captain-whu.github.io/AID/ (accessed on 7 April 2022), https://1drv.ms/u/s!AmgKYzARBl5ca3HNaHIlzp_IXjs (accessed on 7 April 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chi, M.; Plaza, A.; Benediktsson, J.A.; Sun, Z.; Shen, J.; Zhu, Y. Big data for remote sensing: Challenges and opportunities. *Proc. IEEE* **2016**, *104*, 2207–2219. [CrossRef]
2. Cheng, G.; Guo, L.; Zhao, T.; Han, J.; Li, H.; Fang, J. Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA. *Int. J. Remote Sens.* **2013**, *34*, 45–59. [CrossRef]
3. Qi, K.; Yang, C.; Hu, C.; Shen, Y.; Shen, S.; Wu, H. Rotation invariance regularization for remote sensing image scene classification with convolutional neural networks. *Remote Sens.* **2021**, *13*, 569. [CrossRef]
4. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land use classification in remote sensing images by convolutional neural networks. *arXiv* **2015**, arXiv:1508.00092.
5. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
6. Penatti, O.A.; Nogueira, K.; Dos Santos, J.A. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 44–51.
7. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
8. Larochelle, H.; Erhan, D.; Bengio, Y. *Zero-Data Learning of New Tasks*; AAAI: Menlo Park, CA, USA, 2008; Volume 1, p. 3.
9. Lampert, C.H.; Nickisch, H.; Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 453–465. [CrossRef]
10. Zhang, Z.; Saligrama, V. Zero-shot learning via semantic similarity embedding. In Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015; pp. 4166–4174.
11. Frome, A.; Corrado, G.S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; Mikolov, T. Devise: A deep visual-semantic embedding model. In Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013), Stateline, NV, USA, 5–10 December 2013; Volume 26.
12. Li, Y.; Wang, D.; Hu, H.; Lin, Y.; Zhuang, Y. Zero-shot recognition using dual visual-semantic mapping paths. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3279–3287.
13. Fix, E.; Hodges, J.L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Int. Stat. Rev. Int. Stat.* **1989**, *57*, 238–247. [CrossRef]
14. Xian, Y.; Lorenz, T.; Schiele, B.; Akata, Z. Feature generating networks for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5542–5551.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montréal, QC, Canada, 8–13 December 2014; Volume 27.
16. Felix, R.; Reid, I.; Carneiro, G. Multi-modal cycle-consistent generalized zero-shot learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 21–37.
17. Li, J.; Jing, M.; Lu, K.; Ding, Z.; Zhu, L.; Huang, Z. Leveraging the invariant side of generative zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7402–7411.
18. Yu, Y.; Ji, Z.; Han, J.; Zhang, Z. Episode-based prototype generating network for zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14035–14044.
19. Li, A.; Lu, Z.; Wang, L.; Xiang, T.; Wen, J.R. Zero-shot scene classification for high spatial resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4157–4167. [CrossRef]
20. Quan, J.; Wu, C.; Wang, H.; Wang, Z. Structural alignment based zero-shot classification for remote sensing scenes. In Proceedings of the 2018 IEEE International Conference on Electronics and Communication Engineering (ICECE), Xi'an, China, 10–12 December 2018; pp. 17–21.
21. Wang, C.; Peng, G.; De Baets, B. A Distance-Constrained Semantic Autoencoder for Zero-Shot Remote Sensing Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12545–12556. [CrossRef]
22. Sumbul, G.; Cinbis, R.G.; Aksoy, S. Fine-grained object recognition and zero-shot learning in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 770–779. [CrossRef]
23. Verma, V.K.; Arora, G.; Mishra, A.; Rai, P. Generalized zero-shot learning via synthesized examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4281–4289.
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
25. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

26. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
29. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
30. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
31. Shigeto, Y.; Suzuki, I.; Hara, K.; Shimbo, M.; Matsumoto, Y. Ridge regression, hubness, and zero-shot learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Porto, Portugal, 7–11 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 135–151.
32. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
33. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
34. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010.
35. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Kodirov, E.; Xiang, T.; Gong, S. Semantic autoencoder for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3174–3183.
37. Wan, Z.; Chen, D.; Li, Y.; Yan, X.; Zhang, J.; Yu, Y.; Liao, J. Transductive zero-shot learning with visual structure constraint. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
38. Wu, H.; Yan, Y.; Chen, S.; Huang, X.; Wu, Q.; Ng, M.K. Joint visual and semantic optimization for zero-shot learning. *Knowl.-Based Syst.* **2021**, *215*, 106773. [CrossRef]
39. Xing, Y.; Huang, S.; Huangfu, L.; Chen, F.; Ge, Y. Robust bidirectional generative network for generalized zero-shot learning. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
40. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
41. Settles, B. *Active Learning Literature Survey*; University of Wisconsin-Madison: Madison, WI, USA, 2009.
42. Fu, M.; Yuan, T.; Wan, F.; Xu, S.; Ye, Q. Agreement-Discrepancy-Selection: Active Learning with Progressive Distribution Alignment. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; Volume 35, pp. 7466–7473.
43. Wang, S.; Li, Y.; Ma, K.; Ma, R.; Guan, H.; Zheng, Y. Dual adversarial network for deep active learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 680–696.
44. Ahmad, M.; Khan, A.; Khan, A.M.; Mazzara, M.; Distefano, S.; Sohaib, A.; Nibouche, O. Spatial prior fuzziness pool-based interactive classification of hyperspectral images. *Remote Sens.* **2019**, *11*, 1136. [CrossRef]