*remote sensing*

MDPI

*Article*

# Multi-Class Guided GAN for Remote-Sensing Image Synthesis Based on Semantic Labels

**Zhenye Niu [1], Yuxia Li [1], Yushu Gong [1], Bowei Zhang [2], Yuan He [2], Jinglin Zhang [1], Mengyu Tian [1] and Lei He [3,4,*]**

[1] School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; 202321060615@std.uestc.edu.cn (Z.N.); liyuxia@uestc.edu.cn (Y.L.); 2019060401011@std.uestc.edu.cn (Y.G.); 202221060613@std.uestc.edu.cn (J.Z.); 202421060515@std.uestc.edu.cn (M.T.)
[2] Southwest Institute of Technical Physics, Chengdu 610041, China; bowilzhang@163.com (B.Z.); heyuanyuan19812023@163.com (Y.H.)
[3] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China
[4] Sichuan Province Engineering Technology Research Center of Support Software of Informatization Application, Chengdu 610225, China
* Correspondence: helei1978@cuit.edu.cn

**Abstract:** In the scenario of limited labeled remote-sensing datasets, the model's performance is constrained by the insufficient availability of data. Generative model-based data augmentation has emerged as a promising solution to this limitation. While existing generative models perform well in natural scene domains (e.g., faces and street scenes), their performance in remote sensing is hindered by severe data imbalance and the semantic similarity among land-cover classes. To tackle these challenges, we propose the Multi-Class Guided GAN (MCGGAN), a novel network for generating remote-sensing images from semantic labels. Our model features a dual-branch architecture with a global generator that captures the overall image structure and a multi-class generator that improves the quality and differentiation of land-cover types. To integrate these generators, we design a shared-parameter encoder for consistent feature encoding across two branches, and a spatial decoder that synthesizes outputs from the class generators, preventing overlap and confusion. Additionally, we employ perceptual loss ($L_{VGG}$) to assess perceptual similarity between generated and real images, and texture matching loss ($L_T$) to capture fine texture details. To evaluate the quality of image generation, we tested multiple models on two custom datasets (one from Chongzhou, Sichuan Province, and another from Wuzhen, Zhejiang Province, China) and a public dataset LoveDA. The results show that MCGGAN achieves improvements of 52.86 in FID, 0.0821 in SSIM, and 0.0297 in LPIPS compared to the Pix2Pix baseline. We also conducted comparative experiments to assess the semantic segmentation accuracy of the U-Net before and after incorporating the generated images. The results show that data augmentation with the generated images leads to an improvement of 4.47% in FWIoU and 3.23% in OA across the Chongzhou and Wuzhen datasets. Experiments show that MCGGAN can be effectively used as a data augmentation approach to improve the performance of downstream remote-sensing image segmentation tasks.

**Keywords:** remote-sensing images; generative adversarial networks; image synthesis; data augmentation



Academic Editor: Chiman Kwan

Received: 18 December 2024
Revised: 15 January 2025
Accepted: 17 January 2025
Published: 20 January 2025

**Citation:** Niu, Z.; Li, Y.; Gong, Y.; Zhang, B.; He, Y.; Zhang, J.; Tian, M.; He, L. Multi-Class Guided GAN for Remote-Sensing Image Synthesis Based on Semantic Labels. *Remote Sens.* **2025**, *17*, 344. https://doi.org/10.3390/rs17020344



## 1. Introduction

Currently, deep-learning technology is rapidly advancing and achieving significant results across various fields through data-driven methods. High-quality training samples

are fundamental to achieving outstanding performance of network models, thus the demand for sufficient and high-quality datasets is becoming increasingly urgent. However, objects in remote-sensing images can exhibit variations in color, texture, and size across different regions, even within the same class. Consequently, samples from one region may not be effective in improving the network's performance for other regions. Unlike tasks in computer vision, many remote-sensing tasks, such as image segmentation [1], hyperspectral image classification [2], and disaster detection and assessment [3], suffer from a lack of diverse training samples due to regional differences, satellite sensor limitations, and other factors [4,5]. Using data augmentation to generate remote-sensing images can address this [6,7].

Transformation-based augmentation includes color space transformations that modify pixel intensity values (e.g., brightness and contrast adjustment) and geometric transformations that update the spatial locations of pixels (e.g., affine transformations), while synthetic-based augmentation resorts to generative methods (e.g., neural style transfer [8]) and mixing augmentations (e.g., MixUp [9] and CutMix [10]) [11]. Although these approaches increase the dataset size, they primarily focus on individual images or image pairs, utilizing only intrinsic image information or the mutual information between pairs. As a result, the augmented data introduce limited prior knowledge, producing new samples that closely mimic existing patterns with minimal informational novelty. This lack of diversity in the generated data reduces the effectiveness of these methods in enhancing model performance [12].

In recent years, methods based on generative models have introduced innovative approaches to data augmentation. These models generate new images by learning the overall distribution of the data. The mainstream generative models currently include GANs and diffusion models [13,14]. As an emerging method, diffusion has already been applied to various remote-sensing image-generation tasks, such as cloud removal [15], image super-resolution [16], and the conversion between SAR and optical images [17]. However, training diffusion models often requires large datasets. When paired remote-sensing and semantic-label datasets are scarce, training such models becomes challenging. Additionally, since diffusion models require iterative denoising at each timestep during prediction, generating a single image can take several seconds or even longer, making them less suitable for large-scale data augmentation [4,5]. In contrast, GAN-based methods offer a more efficient solution for data augmentation, as they can generate large amounts of data with relatively smaller training datasets [1].

Our research aims to generate remote-sensing images from semantic labels to enhance deep learning datasets. Currently, GAN-based label-to-image methods have achieved outstanding results on scene datasets such as COCO-Stuff [18], Cityscapes [19], and ADE-20K [20]. However, they are still limited in remote-sensing datasets. X. Pan et al. proposed a CGAN-TSIM model for generating remote-sensing images from semantic label, which improves the spatial diversity of the original sample set and enhances the accuracy of semantic segmentation networks [1]. CSEBGAN [21] generates realistic remote-sensing images by decoupling different semantic classes into independent embeddings, achieving finer-grained diversity. Zhang et al. introduced a BnGLGAN [22] translating image to image based on noise reconstruction, which has shown excellent performance in several image generation tasks, including generating remote-sensing images from label maps. However, generating remote-sensing images from semantic labels still faces several challenges.

Existing methods often rely on datasets with simple image forms and low data complexity. These methods may not be suitable when confronted with challenges such as high data imbalance and high semantic similarity among land-cover classes in remote-sensing images. As a result, a single generator structure struggles to effectively capture and gener-

ate the diverse characteristics of these land-cover types. Some methods have attempted to address this by designing dual-branch structures to separately focus on global and local information. However, existing dual-branch architectures still face challenges in effectively balancing the generation of samples for different land-cover types. This imbalance often leads to significant interference between generators, resulting in issues such as distorted building structures and unrealistic texture details in the generated images.

To address these challenges, we employ the Multi-Class Guided GAN (MCGGAN) to augment the dataset of remote-sensing images, specifically focusing on five typical land-cover classes: water, buildings, vegetation, roads, and background. The contributions of this paper are summarized as follows:

1. Our model features a dual-branch generator architecture that combines multi-scale features from both multi-class and global generators using a Pixel-Level Fusion Network. This design overcomes the global generator's limitation in capturing detailed features across different land-cover types.
2. We propose the shared-parameter encoder that ensures consistent feature encoding across both the global and multi-class branches. Additionally, we introduce a spatial decoder that effectively synthesizes outputs from the multi-class generator, preventing overlap and confusion. This design reduces the mutual influence among generators, ensuring more consistent outputs.
3. We employ perceptual loss ($L_{VGG}$) to compute the perceptual similarity of images and use texture matching loss ($L_T$) to assess the differences in texture details between generated and real images via the Gram matrix. By combining these two loss functions, we enhance the color, texture, and perceptual authenticity of the generated images, ensuring higher output quality.
4. The effectiveness of the generated images is assessed by analyzing how varying quantities of generated data impact the accuracy of the U-Net segmentation network across datasets of different sizes.

## 2. Related Work

### *2.1. GAN-Based Data Augmentation*

Numerous studies have utilized GANs to generate remote-sensing samples for data augmentation. Kuang et al. proposes a semantic-layout-guided collaborative framework for SAR sample generation to enhance sample diversity and improve detection performance [23]. Remusati et al. explores the use of GANs to enhance the explainability and performance of SAR Automatic Target Recognition (ATR) and classification models, ad-dressing both the generation of synthetic data and the development of methods for better understanding model decisions [24]. Fu et al. presents a novel denoising GAN for colorizing remote-sensing grayscale images, which outperforms existing techniques and im-proves building detection performance [25]. Rui et al. introduces DisasterGAN to synthesize diverse remote-sensing disaster images with multiple types of disasters and varying building damage, addressing the challenges of class imbalance and limited training data in existing datasets [3]. Simonyan and Zisserman enhanced the WGAN [26] to generate remote-sensing images of construction waste, ensuring realistic edge and texture representation [27]. Kong et al. utilized Pix2Pix [28] and PS-GAN [29] to generate pedestrian samples along railroad perimeters [30], improving safety monitoring by providing additional training data for detection algorithms. Similarly, Yang applied GANs to create water flow images across natural and artificial environments, augmenting datasets and improving the accuracy of flow rate estimation for classification networks [31]. Wang et al. refined a conditional GAN (cGAN) by incorporating perceptual loss and structural similarity metrics with masks, enhancing the quality of aircraft region generation and resolving

sample scarcity in aircraft recognition tasks [32]. Jiang Y et al. improved StyleGAN2 [33] and integrated the generated images into the YOLOv3 [34] training dataset, ultimately boosting the accuracy of object detection and recognition [35].

### *2.2. Label-to-Image Generative Models*

Initially, Generative Adversarial Networks (GANs) could only generate samples from noise. With the rapid development of GANs, the emergence of Conditional Generative Adversarial Networks (cGANs) has enabled the generation of images using category labels, text descriptions, and other images as conditions. Based on this, numerous advancements in GAN-based label-to-image translation techniques have emerged, providing strong technical support for generating remote-sensing images from labels. Pix2Pix [19] employs the U-Net [36] architecture for the generator and the PatchGAN discriminator, improving the quality of image synthesis. Pix2PixHD [37] further refines Pix2Pix by optimizing the model structure, loss function, and semantic processing, enabling the generation of high-resolution images. GauGAN overcomes the limitations of traditional methods, where semantic label maps are processed through convolution, normalization, and nonlinear layers [38], resulting in information loss. It achieves this by introducing Spatially Adaptive Normalization (SPADE), which preserves semantic information and enhances image quality. DAGAN [39] and DPGAN [40] enhance image-level synthesis by adding functional modules. SEAN [41] and CLADE [42] improve image synthesis by designing normalization methods that leverage semantic constraints, enhancing both image quality and model performance. DP-SIMS [43] integrates a pretrained backbone network as the encoder in the discriminator and employs cross-attention mechanisms for noise injection, significantly enhancing image quality and consistency while reducing computational costs. But these methods still struggle to capture category-specific characteristics, limiting the generation of detailed remote-sensing images.

Diffusion models are used in label-to-image tasks due to their powerful generative capabilities. Wang et al. proposed a DDPM-based semantic image synthesis method [44], where noise images are input into the encoder of a U-Net architecture, and semantic layouts are fed into the decoder through multi-layer spatial adaptive normalization. Stable diffusion [14] improves training and inference speed by introducing latent space, where noise is added and denoised, and labels are concatenated with noise images as inputs to the U-Net for label-to-image generation. BBDM [45] models use image-to-image translation as a random Brownian bridge process, directly learning the transformation between domains through a bidirectional diffusion process, rather than relying on a conditional-generation approach. However, diffusion models generally require large datasets for training to achieve high-quality results, limiting their application in remote-sensing tasks with smaller datasets.

### *2.3. Dual Branch GANs*

The use of dual-branch structures for modeling both global and local information has been widely applied in various generation tasks. In face-related tasks, Huang et al. propose TPGAN [46] for frontal view synthesis, which simultaneously captures global structures and local details. Gu et al. introduce MaskGAN [47]. The framework learns feature embeddings for every face component (e.g., mouth, hair, and eye), separately, contributing to better correspondences for image translation, and local face editing. Li et al. propose GLCA-GAN [48] for age synthesis, where a global network learns the overall facial structure and simulates the aging trend of the entire face. Meanwhile, three local networks focus on three key facial patches, either progressing or regressing them to capture subtle changes in crucial facial subregions. However, these methods are primarily

designed for face-related tasks, such as face rotation or editing, where domains exhibit significant overlap and similarity. In semantic-guided scene generation tasks, LGGAN [46] and CollageGAN [49] set up a separate generator for each category. However, LGGAN's l generators, which are based on simple convolutional layers, struggle to generate complex geographic objects. On the other hand, CollageGAN is not well-suited for remote-sensing images, as it assumes clear foreground–background segmentation, which is often not present in remote-sensing images.

These methods have demonstrated the unique advantages of dual-branch architecture in specific domains, achieving better generation results by integrating both global and local information. However, in the remote-sensing field, there is still a lack of specialized generators designed to address high data imbalance and high semantic similarity among land-cover classes.

## 3. Methods

### *3.1. MCGGAN Generator*

The MCGGAN generator adopts a dual-branch architecture. The global generator captures global context to produce coherent results, while the multi-class generator uses class generators to handle different features. The global generator result, $I_g^G$, and the multi-class generator result, $I_g^l$, are combined through a fusion network to produce the final generated image $I_g$.

Notably, we designed the shared-parameter encoder to extract feature maps, which are then fed into three modules: the global generator, the multi-class generator, and the pixel-level fusion network. The gradients from these modules jointly update the encoder's parameters. This approach allows the model to learn two branches of information, providing richer features for the generators and more effective fusion for the pixel-level network.

As shown in Figure 1, the MCGGAN generator consists of four main parts: the shared-parameter encoder, E; the global generator, $G_g$; the multi-class generator, $G_l$; and the fusion network, $G_w$.

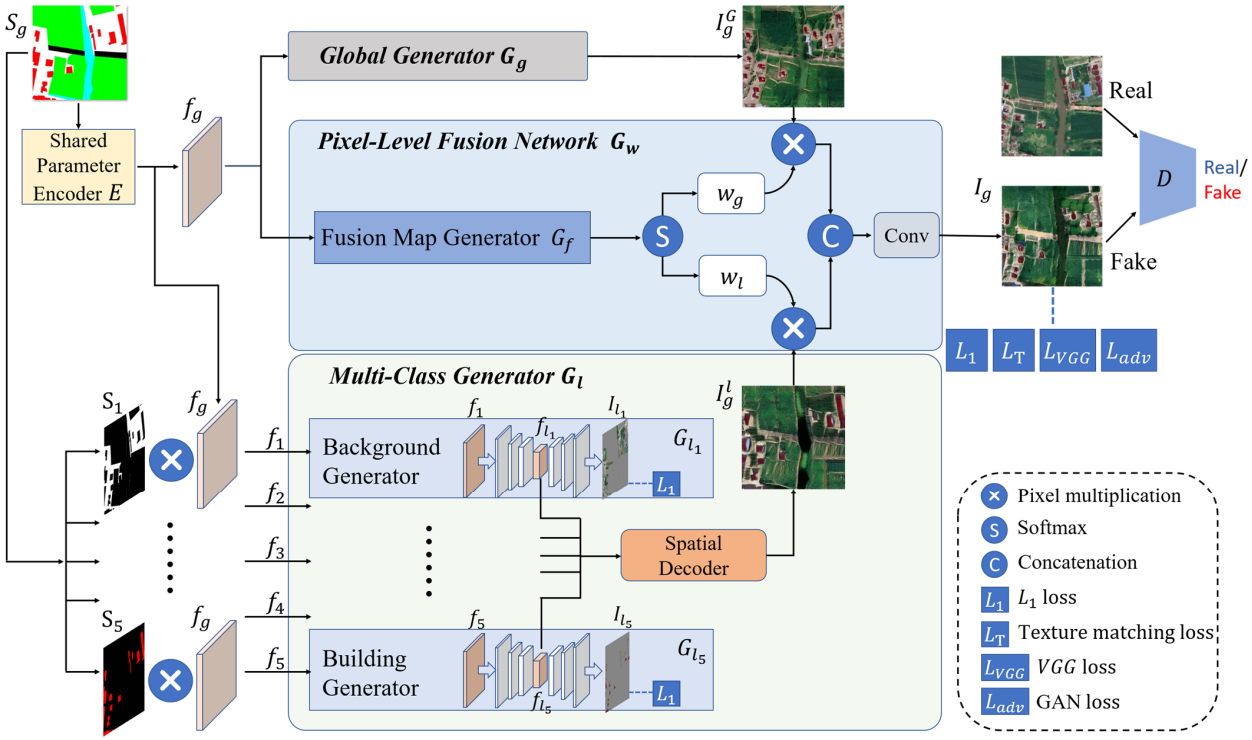



**Figure 1.** The network structure of MCGGAN.

### 3.1.1. Shared-Parameter Encoder

The shared-parameter encoder, $E$, serves two critical functions in this framework. First, it maintains the balance between the global and the multi-class generator during training. Second, it extracts comprehensive semantic information from the semantic label, ensuring optimal utilization of semantic information. The structure of encoder E is illustrated in Figure 2, comprising three core sub-modules: the convolutional module, residual module, and inverse convolutional module.

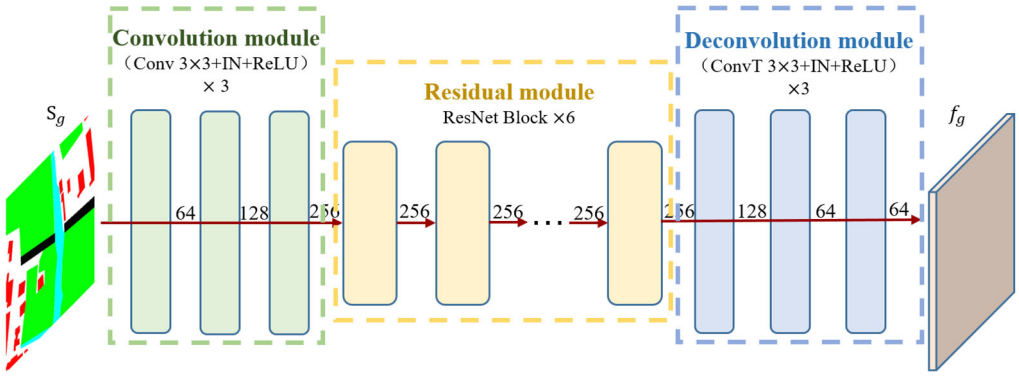



**Figure 2.** The structure of shared-parameter encoder.

The convolutional module contains three convolutional layers. The spatial dimensions of the feature maps are halved, while the channel dimensions are progressively doubled through the layers. The features processed by the convolutional module are then passed into the residual module, which consists of six ResNet blocks. Notably, the size and channel dimensions of the features remain unchanged throughout the residual module. Following this, the processed features are fed into the inverse convolutional module. With each inverse convolution, the spatial dimensions of the feature maps are doubled.

### 3.1.2. Multi-Class Generator

Considering that the diverse spatial patterns and distributions of land-cover types, along with the high semantic similarity observed in certain categories. A single generator may result in the generated samples being of poor quality and may make it difficult to distinguish between types with high semantic similarity.

The structure of the multi-class generator is shown in Figure 3. It consists of multiple class generators and the spatial decoder. The class generators target specific land-cover types, improving both the quality and differentiation of these types. The spatial decoder synthesizes outputs from the class generators, minimizing overlap and confusion.

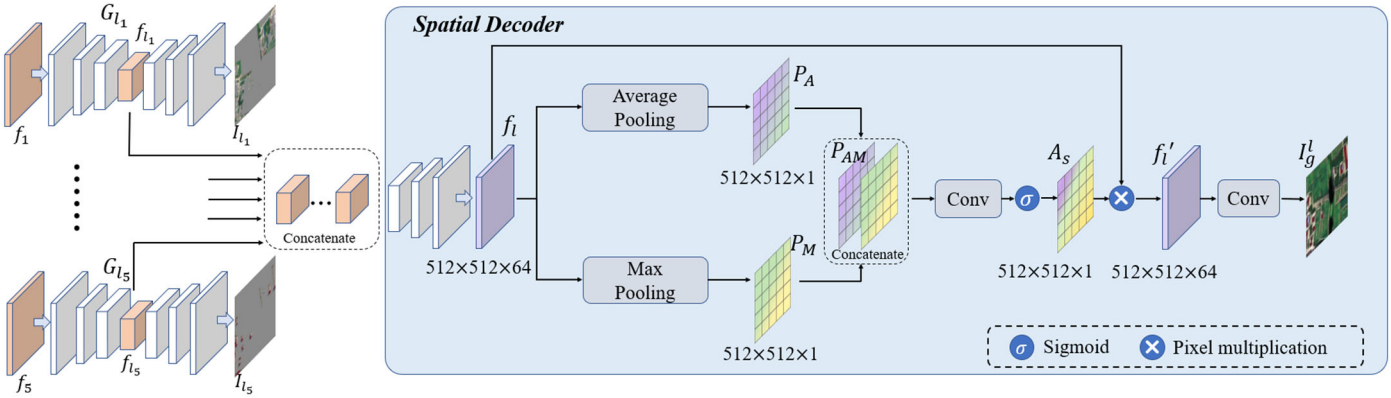



**Figure 3.** The module structure of the multi-class generator.

In the MCGGAN framework, five dedicated class generators are designed for key feature classes: the background generator, $G_{l_1}$; the water generator, $G_{l_2}$; the vegetation generator, $G_{l_3}$; the road generator, $G_{l_4}$; and the building generator, $G_{l_5}$. Each class generator, $G_{l_i}, i \in \{1, 2, 3, 4, 5\}$, targets the generation of its corresponding feature class, enhancing the synthesis quality of those classes and improving the overall generation.

Each class generator is optimized using an L1 loss to ensure the generated images resemble their corresponding real class images. The L1 loss is defined as follows:

$$L_1^l = \mathbb{E}_{I_{r_i}, I_{l_i}} \sum_1^5 \| I_{r_i} - I_{l_i} \|_1 \tag{1}$$

Since the class generators follow an encoder–decoder architecture, some extracted information is inevitably lost during the decoding process. Combined with the shallow depth of these generators, the capacity to effectively capture and utilize class semantic information is weakened. To address these, intermediate features from each class generator are incorporated into the multi-class generator result synthesis. This strategy strengthens the multi-class generator's capability to better capture characteristics, ultimately enhancing the quality of both the individual class images and the final composite output.

The specific network structure of the multi-class generator, $G_l$, in the MCGGAN is illustrated in Figure 3. The multi-class generation result, $I_g^l$, is synthesized using intermediate features from each class generator. Specifically, intermediate features, $f_{l_i}, \mathrm{i} \in \{1, 2, 3, 4, 5\}$, are extracted and concatenated along the channel dimension. These concatenated features are then processed through the spatial decoder, ultimately producing the multi-class generation result, $I_g^l$. Extracting these intermediate features is essential, as they originate from the deeper layers of the encoder and encapsulate the richest semantic information. In the spatial decoder, the features are first upsampled to recover spatial resolution. The upsampling module comprises three inverse convolutional layers. This computational process can be succinctly expressed as Equation (2):

$$f_l = UpSample\big(Concat\big(f_{l_1}, f_{l_2}, \ldots, f_{l_5}\big)\big) \tag{2}$$

where $UpSample(\cdot)$ denotes the upsampling operation.

The upsampled features are further refined using a spatial attention module. The feature $f_l$ is computed by applying both average pooling and maximum pooling to obtain the feature maps $P_A$ and $P_M$. $P_A$ and $P_M$ are spliced in channel dimension to obtain a feature map called $P_{AM}$. The spatial attention map, $A_S$, is generated by passing $P_{AM}$ through a convolutional layer with a $3 \times 3$ kernel and the sigmoid activation function. The resulting spatial attention map, $A_S$, is then matrix-multiplied with the feature $f_l$ to produce the processed feature, $f_l{}'$.

$$f_l{}' = A_s \otimes f_l = \sigma(Conv(Concat(P_A, P_M))) \otimes f_l \tag{3}$$

The introduction of the spatial decoder not only aims to extract features from the intermediate layers of each class generator to enhance generation quality but also serves to balance the differences in generation capacity between various class generators. This unified approach prevents interference between generators, ensuring that features from different categories are harmoniously integrated, thus avoiding confusion or overlapping in the generated images.

#### 3.1.3. Pixel-Level Fusion Network

Both the global generation result ($I_g^G$) and the multi-class generation result ($I_g^l$) are passed through the fusion network ($G_w$) to obtain the final image, $\mathrm{I}_g$. The pixel-level fusion

network consists of a fusion map generator. The outputs of the fusion map generator are computed using the Softmax function to generate the pixel-level weight maps, $w_g$ and $w_l$, as shown in Equation (4):

$$w_g, w_l = G_w(f_g) \tag{4}$$

The final fusion process is expressed in Equation (5):

$$I_g = Conv\left(Concat\left(w_g * I_g^G, w_l * I_g^l\right)\right) \tag{5}$$

### 3.2. MCGGAN Discriminator

MCGGAN employs PatchGAN as its discriminator. PatchGAN divides the input image into N × N blocks and performing real or fake classification on each block, producing an N × N matrix of probabilities. The overall classification probability of the image is then obtained by averaging the values in this matrix. This block-based approach enables PatchGAN to focus more on local details, enhancing image quality while also improving computational efficiency compared to the original GAN.

### 3.3. Loss Function

During the training process of the Generative Adversarial Network, the generator, G, aims to produce images that closely resemble real images, while the discriminator, D, strives to accurately distinguish whether the output image originates from the generator or the real dataset. The optimization objective for the adversarial loss in MCGGAN is grounded in the CGAN architecture, with the adversarial loss function defined as shown in Equation (6):

$$L_{adv} = \mathbb{E}_{S_g, I_r}\left[logD(S_g, I_r)\right] + \mathbb{E}_{S_g, I_g}\left[log(1 - D(S_g, I_g))\right] \tag{6}$$

To recover the low-frequency components of the image and minimize the influence of outliers, the L1 loss is employed to constrain the global generation result, as expressed in Equation (7):

$$L_1^g = \mathbb{E}_{I_r, I_g} \| I_r - I_g \|_1 \tag{7}$$

In this study, image features are extracted using a deep neural network based on the ImageNet-pretrained VGG19 architecture. The distance between real images ($I_r$) and generated images ($I_g$) is measured in the image feature space as a loss function. Both images are sequentially input into the VGG19 network, and the output features $\Phi_j(I_r)$ and $\Phi_j(I_g)$ are selected after the ReLU layers from different modules of the network. To enhance perceptual similarity between generated images, features from the 2nd, 4th, 8th, 12th, and 16th convolutional modules of the VGG19 network are chosen. The distances between these selected features are then calculated as part of the VGG loss. This relationship is expressed in Equation (8):

$$L_{VGG} = \sum_{j \in \mathcal{V}} \| \Phi_j(I_r) - \Phi_j(I_g) \|_2^2 \tag{8}$$

where $j$ represents the module serial number, and $\mathcal{V} = \{2, 4, 8, 12, 16\}$ represents the selected set of VGG19 module serial number.

Additionally, the Gram matrix serves as a texture synthesis method, constructed from feature vectors extracted from the input image features. It enables the measurement of correlations between features while disregarding spatial information within the feature maps. By comparing the differences between the features of generated images and real images, it aids the model in learning how to effectively generate image textures. We select the output features after the ReLU layers following the 2nd and 16th convolutional

modules of the VGG19 network to compute the Gram matrix. The difference in Gram matrices between real and generated images is then used to optimize the training process of the generator. The texture matching loss function is defined in Equation (9):

$$L_T = \sum_{j=2,16} \| G(\Phi_j(I_r)) - G(\Phi_j(I_g)) \|_2^2 \quad (9)$$

To summarize, the loss function of MCGGAN includes the adversarial loss ($L_{adv}$), the $L_1$ loss ($L_1^g$) of the generated image ($I_g$), the $L_1$ loss ($L_1^l$) of the category-generated image ($I_{l_i}$), the perceptual loss ($L_{VGG}$) of the generated image ($I_g$), and the texture-matching loss ($L_T$). The synthesized expression is shown in Equation (10):

$$L_{MCGGAN} = L_{adv} + \lambda_1 L_1^g + \lambda_2 L_1^l + \lambda_3 L_{VGG} + \lambda_4 L_T \quad (10)$$

$\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ are the weight coefficients.

## 4. Results

### *4.1. Experimental Dataset*

We build two remote-sensing image datasets with distinct styles: the Chongzhou area in Sichuan Province of China, covering longitudes 103°37′ to 103°45′ E and latitudes 30°35′ to 30°40′ N, and the Wuzhen area in Zhejiang Province of China, covering longitudes 120°26′ to 120°33′ E and latitudes 30°43′ to 30°47′ N. The original images are satellite optical orthorectified images captured over two time periods, with each image measuring 5826 × 3884 pixels and a spatial resolution of 0.51 m. The Chongzhou dataset features complex land characteristics, including large factories, intricate residential structures, and rural clusters. In contrast, the Wuzhen dataset primarily consists of water bodies surrounded by villages, with a landscape dominated by vegetation and rural buildings.

Both datasets exhibit data imbalance, each with its own characteristics. The Chongzhou dataset features complex and diverse buildings, posing a challenge for the model to generate these underrepresented classes, especially for buildings. The Wuzhen dataset contains water and vegetation with high semantic similarity, requiring the model to have strong distinguishing capability.

We also conducted comparative experiments on the publicly available LoveDA [50] dataset to further validate the advantages of MCGGAN across different geographic locations and satellite sensors. The LoveDA dataset covers three cities—Nanjing, Changzhou, and Wuhan. It features inconsistent sample distributions between urban and rural areas, posing significant challenges for generative models.

In total, as shown in Figure 4, we utilized two custom datasets and one public dataset, covering multiple cities in China and different satellite sensors. We focus on generating five typical remote-sensing land features: background, water, vegetation, buildings, and roads. To facilitate this, we annotated the Chongzhou and Wuzhen dataset with the corresponding categories to create semantic labels. For the LoveDA dataset, we merged forest and agriculture into vegetation, and barren areas into background, resulting in five label categories.

The images are cropped into 512 × 512. Following cropping, both datasets are randomly divided into training and testing sets in a 4:1 ratio. The final Chongzhou dataset comprises 845 training samples and 211 testing samples, while the Wuzhen dataset contains 704 training samples and 176 testing samples. To maintain consistency in dataset size, we randomly selected 1000 512 × 512 images from the LoveDA dataset and then split them at a 4:1 ratio to obtain 800 training samples and 200 testing samples.

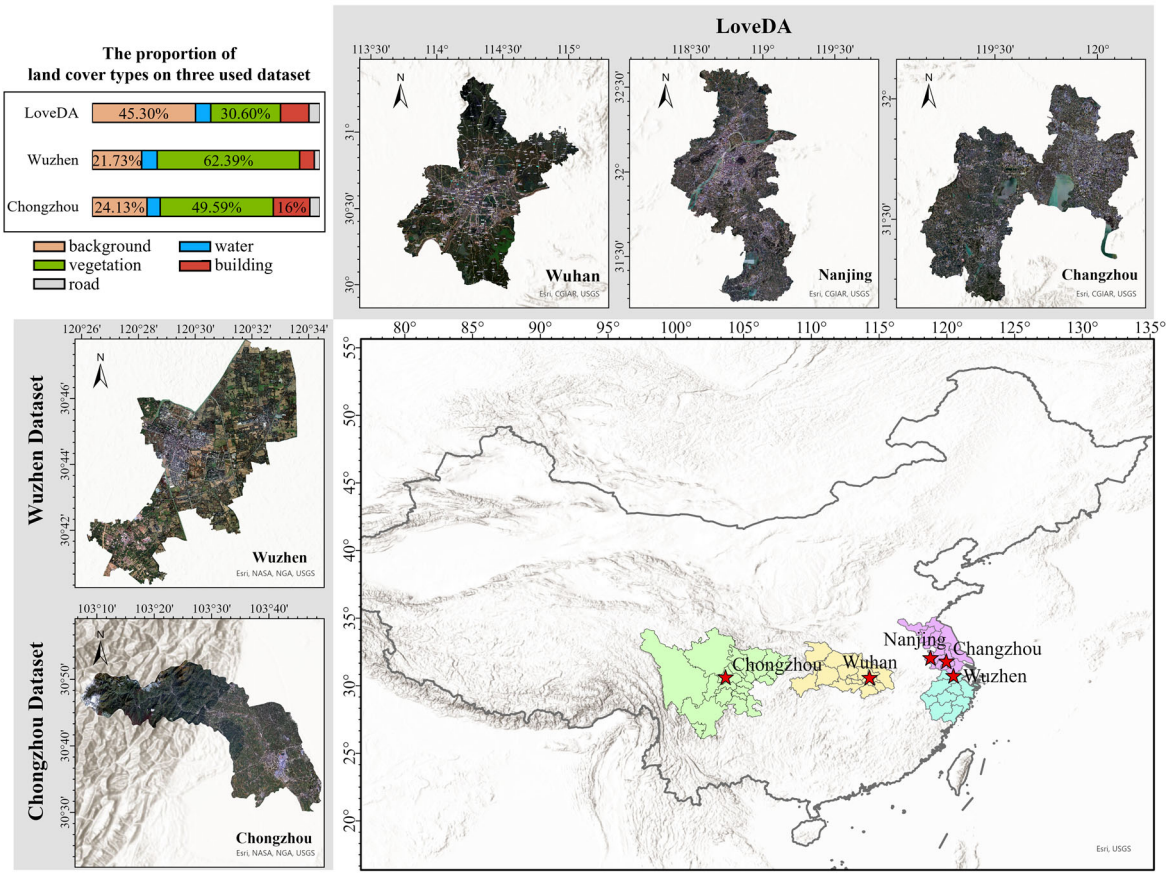



**Figure 4.** The three datasets used for MCGGAN.

Figure 4 illustrates the percentage of each feature in the Chongzhou, Wuzhen, and LoveDA datasets. In Chongzhou, buildings occupy a large proportion, predominantly representing urban scenes. In contrast, Wuzhen features abundant vegetation and water bodies, has fewer buildings, and is predominantly characterized by rural farmland. The LoveDA dataset contains a higher proportion of background and includes both urban and rural sample distributions. It features a diverse range of buildings, thus posing a greater challenge for generative models.

### *4.2. Evaluation Metrics*

To assess the perceptual similarity between generated images and real images, we employ three representative metrics: Fre'chet Inception Distance (FID) [51,52], Learned Perceptual Image Patch Similarity (LPIPS) [53], and Structural Similarity (SSIM) tailored to the characteristics of remote-sensing images.

FID is used to measure the distribution differences between generated images and real images in feature space. The process begins by extracting features using the Inception network, followed by modeling the feature space with a Gaussian model, and finally calculating the distance between the two feature distributions. A lower FID indicates higher image quality and diversity. The formula is as follows:

$$FID(x,y)^2 = \|\mu_x - \mu_y\|_2^2 + Tr(C_x + C_y - 2(C_xC_y)^{\frac{1}{2}}) \quad (11)$$

where $x$ denotes the generated image; $y$ denotes the original image; $\mu_x$, $\mu_y$, $C_x$, and $C_y$ represent the mean and covariance matrices of the image features; and $Tr(\cdot)$ represents the trace of a matrix.

LPIPS is a metric used to assess the perceptual similarity between images. Unlike traditional metrics such as PSNR and SSIM, which primarily focus on pixel-level differences, LPIPS aligns more closely with human visual perception. The formula is as follows:

$$LPIPS(I_1, I_2) = \sum_l \frac{1}{N_l} \left\| F_1^l - F_2^l \right\|_2^2 \tag{12}$$

where $l$ denotes different layers in the network (e.g., convolutional layers); $F_1^l$ and $F_2^l$ are the feature maps of images $I_1$ and $I_2$ at layer $l$; $N_l$ is the number of elements in the feature map at layer $l$; and $\|\cdot\|_2$ represents the L2 norm (Euclidean distance).

SSIM is an index that estimates the resemblance of two images. One is the undistorted, uncompressed image, and the other is the distorted image of the two images used in SSIM. The specific calculation is as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{\left(\mu_x^2 + \mu_y^2 + C1\right)\left(\sigma_x^2 + \sigma_y^2 + C2\right)} \tag{13}$$

where the mean, $\mu$, represents the estimate of lightness; the standard deviation, $\sigma$, represents the estimate of contrast; and the covariance, $C$, represents the evaluation of the degree of structural correspondence.

The ultimate goal of generating these images is to augment the dataset for deep learning tasks. To evaluate the quality of the generated images, we utilize a U-Net network trained on the three datasets for semantic segmentation. The model's accuracy is quantified using two metrics: Frequency-Weighted Intersection over Union (FWIoU) and Overall Accuracy (OA). If the generated images are highly realistic and closely resemble real images, a segmentation network trained on real images should accurately segment the generated outputs.

### *4.3. Implementation Details*

We train and test the model using PyTorch 1.13.0 on the Chongzhou, Wuzhen, and LoveDA datasets, respectively, employing an NVIDIA RTX 4090 GPU (NVIDIA, Santa Clara, CA, USA) as the training tool. For model training parameters, the batch size is set to 4, with a total of 200 epochs. The learning rate is initially set to 0.0002 for the first 100 epochs and decays linearly to 0 over the subsequent 100 epochs. The optimization algorithm used for network parameters is Adam.

### *4.4. Hyperparameter Settings*

The loss function used in MCGGAN is defined as $L_{MCGGAN} = L_{adv} + \lambda_1 L_1^g + \lambda_2 L_1^l + \lambda_3 L_{VGG} + \lambda_4 L_T$, where $L_{adv}$, $L_1^g$, $L_1^l$ are the basic losses in the dual-branch network. For $\lambda_1$ and $\lambda_2$, we refer to the empirical values used in LGGAN [54], setting $\lambda_1 = 1$ and $\lambda_2 = 1$. The terms $L_{VGG}$ and $L_T$ correspond to the perceptual loss and texture matching loss used in MCGGAN. We perform ablation studies and sensitivity analysis to investigate the impact of the perceptual loss and texture matching loss on the generation quality.

The experimental results are shown in Table 1. The first three rows present the ablation study of the losses. As observed, introducing the losses leads to improvements in all metrics, indicating that both the perceptual loss and texture matching loss positively contribute to model training.

**Table 1.** The sensitivity analysis of $L_{VGG}$ and $L_T$.

| $\lambda_3$ | $\lambda_4$ | LPIPS↓ | FID↓ | SSIM↑ | FWIoU(%)↑ | PA(%)↑ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0.6104 | 166.25 | 0.2201 | 55.79 | 68.17 |
| 0 | 10 | 0.5791 | 148.08 | 0.2247 | 56.77 | 69.97 |
| 10 | 0 | 0.5922 | 159.31 | 0.2257 | 57.36 | 69.83 |
| 10 | 0.5 | 0.5810 | 157.05 | 0.2180 | 57.36 | 71.57 |
| **10** | **1** | **0.5783** | **123.42** | **0.2324** | **60.80** | **72.88** |
| 10 | 2 | 0.5840 | 135.27 | 0.2283 | 59.69 | 71.35 |
| 10 | 10 | 0.5897 | 145.63 | 0.2121 | 58.03 | 70.74 |
| 2 | 10 | 0.5844 | 147.42 | 0.2129 | 58.43 | 70.80 |
| 1 | 10 | 0.5807 | 140.18 | 0.2242 | 58.38 | 70.57 |
| 0.5 | 10 | 0.5813 | 152.41 | 0.2171 | 58.54 | 71.06 |

After introducing the two losses, we conducted a sensitivity analysis. By fixing $\lambda_3$ and gradually increasing $\lambda_4$, we observed that the metrics initially improved and then gradually declined, with significant fluctuations. When $\lambda_4$ was fixed and $\lambda_3$ was gradually decreased, the metrics followed a similar trend. But the sensitivity to $\lambda_3$ was lower, as the fluctuations were less pronounced. Based on the experimental results, we ultimately chose $\lambda_3 = 10$ and $\lambda_4 = 1$.

### *4.5. Ablation Experiments*

Ablation experiments are conducted to decompose the generator model and evaluate how various structures influence image quality. This approach allows us to verify the contribution of each functional module within the MCGGAN generator to the enhancement of generated image quality.

The ablation experiments are structured around five schemes as shown in Table 2 and Figure 5: Pix2Pix serves as the baseline model. Pix2Pix++ incorporates perceptual loss ($L_{VGG}$) and texture-matching loss ($L_T$) into Pix2Pix, resulting in the loss function $L_{MCGGAN}$. DBGAN employs Pix2Pix as the global generator and includes the multi-class generator for different features within the dual-branch generative model. DBGAN++ builds upon DBGAN by introducing the shared-parameter encoder, thereby balancing the training process. MCGGAN enhances the class generators by introducing the spatial decoder to form the final proposed model. In this context, the loss functions for DBGAN, DBGAN++, and MCGGAN are all defined as $L_{MCGGAN}$.

**Table 2.** The ablation experiment plan.

| Model | Description |
|---|---|
| Pix2Pix | Baseline model |
| Pix2Pix++ | $L_{VGG}$ and $L_T$ are added on Pix2Pix model |
| DBGAN | Dual-branch generative model |
| DBGAN++ | A shared-parameter encoder is added on DBGAN |
| MCGGAN | A spatial decoder is added on DBGAN++ |

Ablation experiments are conducted on the Chongzhou and Wuzhen datasets. Table 3 presents the evaluation metrics for each program on the respective datasets.

**Table 3.** The results of ablation experiment on the Chongzhou and Wuzhen datasets.

| Methods | Chongzhou | | | | | Wuzhen | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LPIPS | FID | SSIM | FWIoU (%) | OA (%) | LPIPS | FID | SSIM | FWIoU (%) | OA (%) |
| Pix2Pix | 0.6080 | 176.28 | 0.1603 | 49.56 | 62.50 | 0.6270 | 225.25 | 0.2462 | 57.22 | 71.18 |
| Pix2Pix++ | 0.6008 | 159.23 | 0.2210 | 53.26 | 66.02 | 0.6087 | 213.08 | 0.2602 | 59.34 | 72.57 |
| DBGAN | 0.6132 | 169.66 | 0.2174 | 54.86 | 67.69 | 0.6273 | 236.80 | 0.2497 | 61.63 | 74.20 |
| DBGAN++ | 0.5961 | 154.27 | 0.2309 | 54.50 | 67.59 | 0.5983 | 179.22 | 0.2744 | 59.63 | 73.47 |
| **MCGGAN** | **0.5783** | **123.42** | **0.2324** | **60.80** | **72.88** | **0.5551** | **137.96** | **0.2793** | **65.98** | **77.35** |

Table 3 indicates that compared to the baseline model Pix2Pix, Pix2Pix++ shows significant improvements in the Chongzhou dataset, achieving a 3.61% improvement in FWIoU, a 3.52% improvement in OA, a 0.0072 improvement in LPIPS, a 0.0607 improvement in SSIM and a remarkable 17.05% improvement in FID. Similarly, for the Wuzhen dataset, FWIoU and OA improvement by 2.12% and 1.39%, with LPIPS improves by 0.0183, SSIM by 0.014 and FID by 12.27. Figure 6b,c and Figure 7c illustrate that the incorporation of VGG loss and texture matching loss effectively mitigates issues in water generation. Additionally, this enhancement improves the model's capacity to learn color textures, particularly evident in the extraction of urban building colors in the Wuzhen dataset.

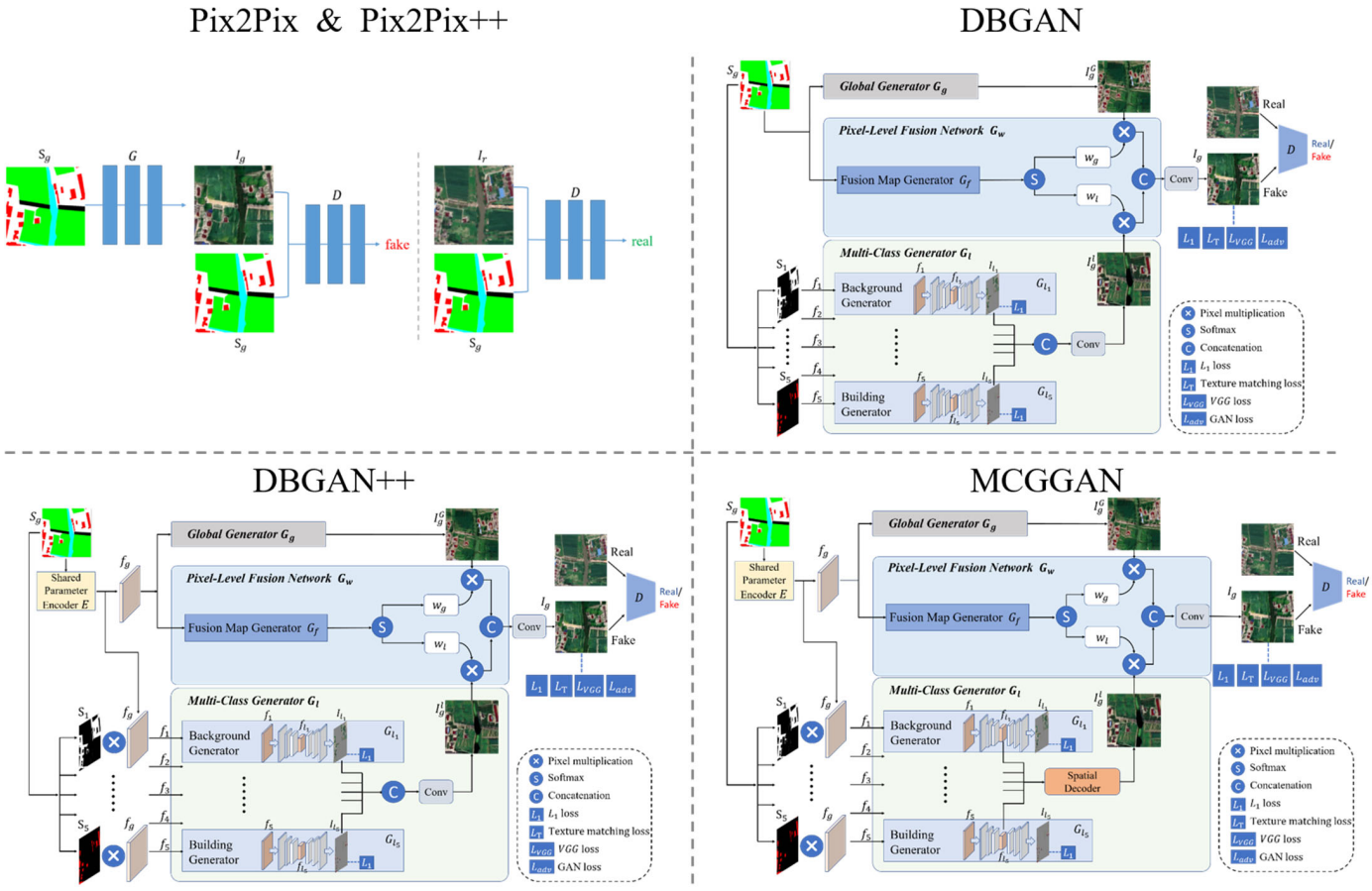



**Figure 5.** The schematic diagram of ablation experiment plan.

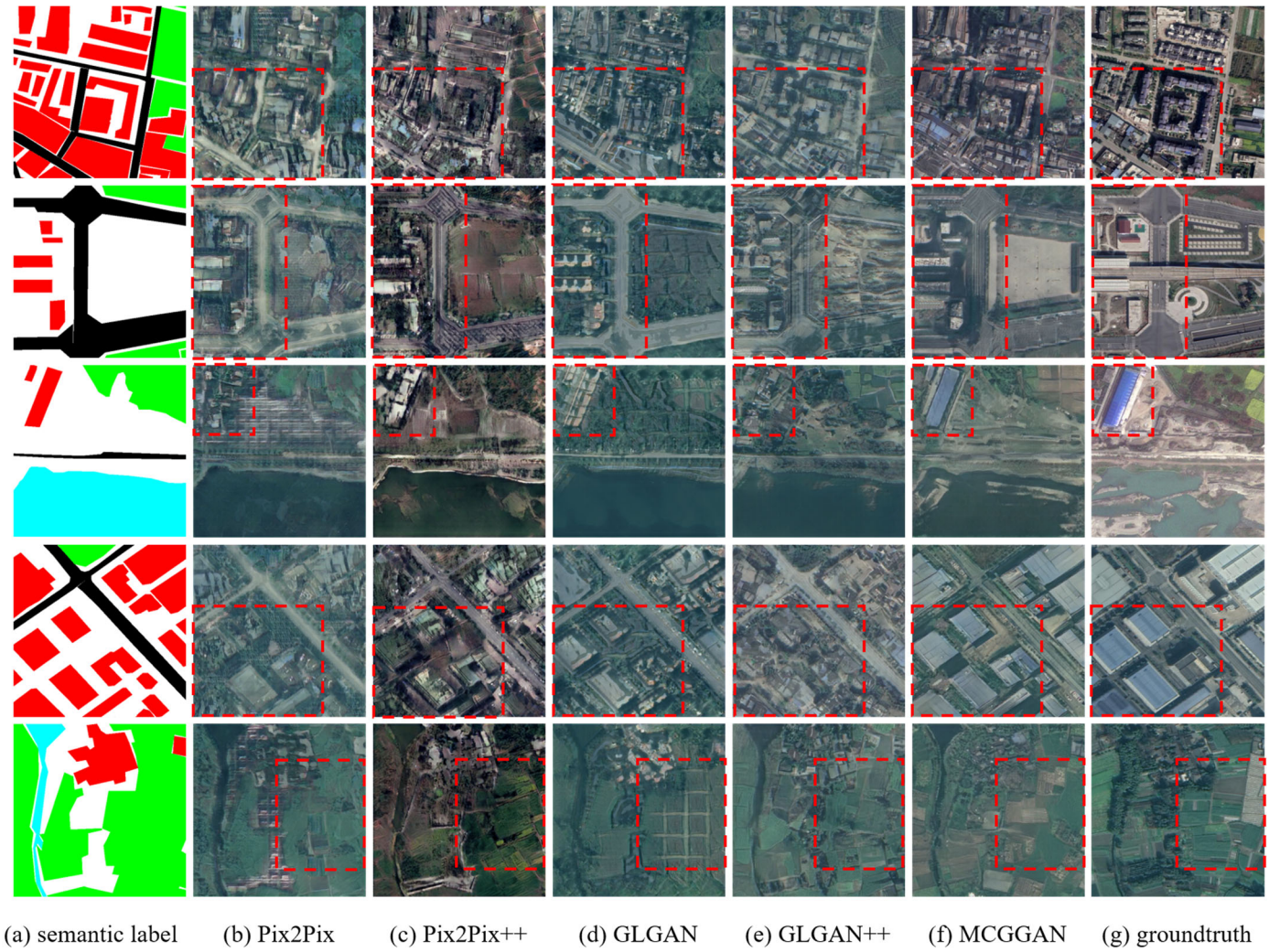



**Figure 6.** The ablation experiment on the Chongzhou dataset.

When comparing the DBGAN model to Pix2Pix++ on the Chongzhou dataset, DBGAN demonstrates improvements with a 1.60% increase in FWIoU and a 1.67% increase in OA. However, LPIPS decreases by 0.0124, SSIM decreases by 0.0036 and FID decreases by 10.43. Figure 6c,d visually illustrate that DBGAN produces architecture with clearer outlines compared to Pix2Pix++. Moreover, DBGAN's road generation results feature contours and textures that more closely resemble real road characteristics. On the Wuzhen dataset, DBGAN outperforms Pix2Pix++ with a 2.29% increase in FWIoU and a 1.63% increase in OA. Conversely, LPIPS decreases by 0.0186, SSIM decreases by 0.0105 and FID decreases by 23.72. Figure 7c,d show that DBGAN enhances the generation of colors and contours for small-scale buildings in the Wuzhen dataset. The above illustrates that the use of a dual-branch structure, along with the introduction of multi-class generator, can effectively enhance the model's ability to generate objects for underrepresented land-cover classes.

Table 4 shows the complexity of different modules. With the introduction of the shared-parameter encoder, DBGAN++ successfully addresses the issues present in DBGAN. Although introducing the shared-parameter encoder increases computational cost, DBGAN++ achieves a lower overall loss compared to DBGAN and demonstrates significantly faster convergence. This indicates that the shared-parameter encoder successfully balances the training of the two generators, effectively accelerating convergence and reducing training difficulty. In terms of image quality, DBGAN++ has also achieved significant improvements. As indicated by the metrics in Table 3, DBGAN++ shows improvements

in LPIPS, SSIM and FID by 0.0178, 0.0235, and 30.85 on the Chongzhou dataset, and by 0.0290, 0.0247, and 57.58 on the Wuzhen dataset, respectively. Visual comparisons in Figures 6d and 7d demonstrate that DBGAN++ effectively mitigates the pattern collapse and noise issues found in DBGAN, resulting in images that closely resemble real ones. However, there is a slight decrease in FWIoU and OA metrics with the introduction of the shared-parameter encoder. Specifically, on the Chongzhou dataset, FWIoU and OA decreased by 0.36% and 0.1%, while on the Wuzhen dataset, they decreased by 2% and 0.73%. This reduction can be attributed to the interference introduced during the convolution process of the shared-parameter encoder, which complicates the multi-class generator's ability to generate specific categories.


**Figure 7.** The ablation experiment on the Wuzhen dataset.

**Table 4.** Module complexity.

| Model | Shared-Parameter Encoder | Spatial Decoder | #Params (M) | Inference Time (ms) |
|---|---|---|---|---|
| DBGAN | | | 89.98 | 25.6 |
| DBGAN++ | ✔ | | 96.88 | 34.2 |
| MCGGAN | ✔ | ✔ | 101.74 | 38.1 |

MCGGAN leverages the spatial decoder to balance the influences among the class generators. This approach significantly enhances the performance of the multi-class gen-

erator. As shown in Figure 8, DBGAN++ converges faster but exhibits some fluctuations in the later stages of training, suggesting potential instability. MCGGAN shows the best stability, with the loss steadily decreasing and minimal fluctuations. Stable training leads to higher generation quality, MCGGAN achieves the best overall metrics. On the Chongzhou dataset, compared to DBGAN++, MCGGAN's FWIoU and OA improve by 6.3% and 5.29%, respectively; LPIPS improves by 0.0178, SSIM improves by 0.0015, and FID improves by 30.85. When compared to the baseline model Pix2Pix, MCGGAN demonstrates improvements of 11.24% and 10.38% in FWIoU and OA, respectively; LPIPS improves by 0.0297, SSIM improves by 0.0821, and FID improves by 52.86. On the Wuzhen dataset, MCGGAN outperforms DBGAN++ with improvements of 6.35% and 3.88% in FWIoU and OA, respectively; LPIPS improves by 0.0432, SSIM improves by 0.0049, and FID improves by 41.26. Compared to the baseline model Pix2Pix, MCGGAN shows improvements of 8.76% and 6.17% in FWIoU and OA, respectively; LPIPS improves by 0.0719, SSIM improves by 0.0331, and FID improves by 87.29.

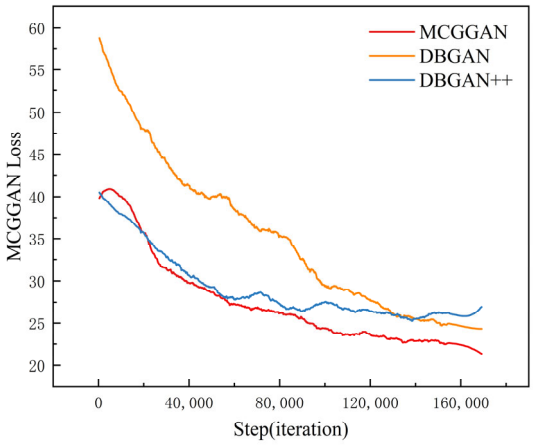


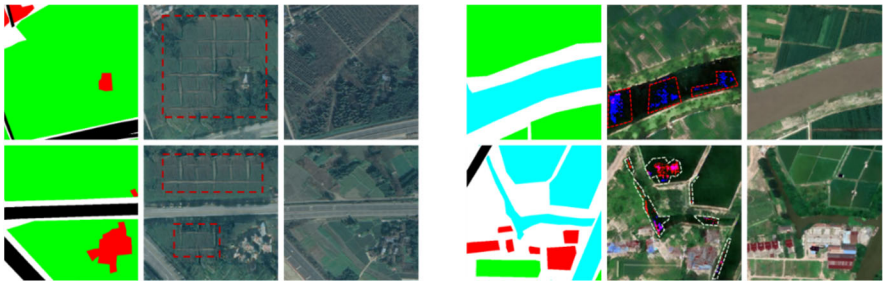

**Figure 8.** The loss function for the three dual-branch models in ablation experiments.

Visually, MCGGAN demonstrates significant enhancements in generating building outlines on the Chongzhou dataset. As illustrated in the first row of Figure 6, the model produces more realistic representations of complex residential buildings, while the fourth row shows improved generation of factory buildings. Additionally, MCGGAN effectively captures vegetation textures and rural buildings that closely resemble real remote-sensing images, as seen in the fifth row of Figure 6. The model also excels in generating roads and complex backgrounds, aligning better with the inherent characteristics of these features, highlighted in the second and third rows of Figure 6.

In summary, the improvements provided by MCGGAN not only ensure network stability but also enhance the generation of fine details across various land-cover categories. This results in a higher quality of generated samples for underrepresented land-cover classes. Moreover, MCGGAN strengthens the depiction of complex features like building outlines, leading to samples that more closely match real remote-sensing images.

On the Wuzhen dataset, MCGGAN's superior understanding of global context enables it to generate diverse remote-sensing images, reflecting both lush spring/summer scenes and darker autumn/winter tones based on the layout features of the semantic image.

However, DBGAN exhibits certain shortcomings, as evidenced by the generated images depicted in Figure 9. Specifically, the red dashed box in Figure 9a highlights a texture replication issue in the Chongzhou generated image, while the white dashed box in Figure 9b indicates the presence of noise in the Wuzhen generated image. These problems primarily arise from the challenges in maintaining a balance between the global generator and the multi-class generator during the training process.



(**a**) (**b**)

**Figure 9.** The partial DBGAN-generated images: left, semantic label; middle, generated image; right, real image. (**a**) Chongzhou and (**b**) Wuzhen.

### *4.6. Comparison Experiments*

In order to further verify the effectiveness of MCGGAN model, we respectively compare Pix2PixHD [37], DAGAN [39], DPGAN [40], stable diffusion model [14], LGGAN [54], and Lab2Pix-V2 [55] on the dataset of Chongzhou, Wuzhe ann and LoveDA. The evaluation indexes of the experimental results are shown in Tables 5–7. To visualize the generating effect of different models, some of the experimental result images are given in this paper, as shown in Figures 10–12.

**Table 5.** The comparison of experimental results of six models on the Chongzhou dataset.

| Methods | LPIPS↓ | FID↓ | SSIM↑ | FWIoU (%)↑ | OA (%)↑ |
|---|---|---|---|---|---|
| Pix2PixHD [37] | 0.6007 | 145.44 | 0.1922 | 57.60 | 69.10 |
| DAGAN [39] | 0.5901 | 170.19 | 0.2225 | 57.85 | 69.52 |
| DPGAN [40] | 0.6147 | 167.84 | 0.2257 | 56.12 | 69.50 |
| Lab2Pix-V2 [55] | 0.5844 | 137.88 | 0.2209 | 57.51 | 69.90 |
| SDM [14] | 0.5851 | 183.37 | 0.2008 | 56.14 | 70.39 |
| LGGAN [54] | 0.5874 | 192.85 | 0.1829 | 57.99 | 70.30 |
| MCGGAN [ours] | **0.5783** | **123.42** | **0.2324** | **60.80** | **72.88** |

**Table 6.** The comparison of experimental results of six models on the Wuzhen dataset.

| Methods | LPIPS↓ | FID↓ | SSIM↑ | FWIoU (%)↑ | OA (%)↑ |
|---|---|---|---|---|---|
| Pix2PixHD [37] | 0.5707 | 173.17 | 0.2418 | 62.03 | 74.55 |
| DAGAN [39] | 0.5670 | 150.34 | 0.2208 | 62.29 | 74.75 |
| DPGAN [40] | 0.5917 | 153.51 | 0.2551 | 57.60 | 70.72 |
| Lab2Pix-V2 [55] | 0.5590 | 148.15 | 0.2548 | 63.43 | 75.91 |
| SDM [14] | 0.5613 | 169.61 | 0.2422 | 62.35 | 75.20 |
| LGGAN [54] | 0.5611 | 193.68 | 0.2247 | 62.89 | 75.43 |
| MCGGAN [ours] | **0.5551** | **137.96** | **0.2793** | **65.98** | **77.35** |

**Table 7.** The comparison of experimental results of six models on the LoveDA dataset.

| Methods | LPIPS↓ | FID↓ | SSIM↑ | FWIoU (%)↑ | OA (%)↑ |
|---|---|---|---|---|---|
| Pix2PixHD [37] | 0.6651 | 199.79 | 0.2747 | 60.71 | 70.75 |
| DAGAN [39] | 0.6708 | 186.24 | 0.2563 | 60.26 | 70.58 |
| DPGAN [40] | 0.6862 | 188.28 | 0.2648 | 56.34 | 67.67 |
| Lab2Pix-V2 [55] | 0.6610 | 192.65 | **0.2906** | 63.74 | 73.79 |
| SDM [14] | 0.6877 | 199.77 | 0.2463 | 64.74 | 74.20 |
| LGGAN [54] | 0.6608 | 228.59 | 0.2201 | 60.48 | 70.62 |
| MCGGAN [ours] | **0.6551** | **184.63** | 0.2837 | **64.84** | **75.51** |

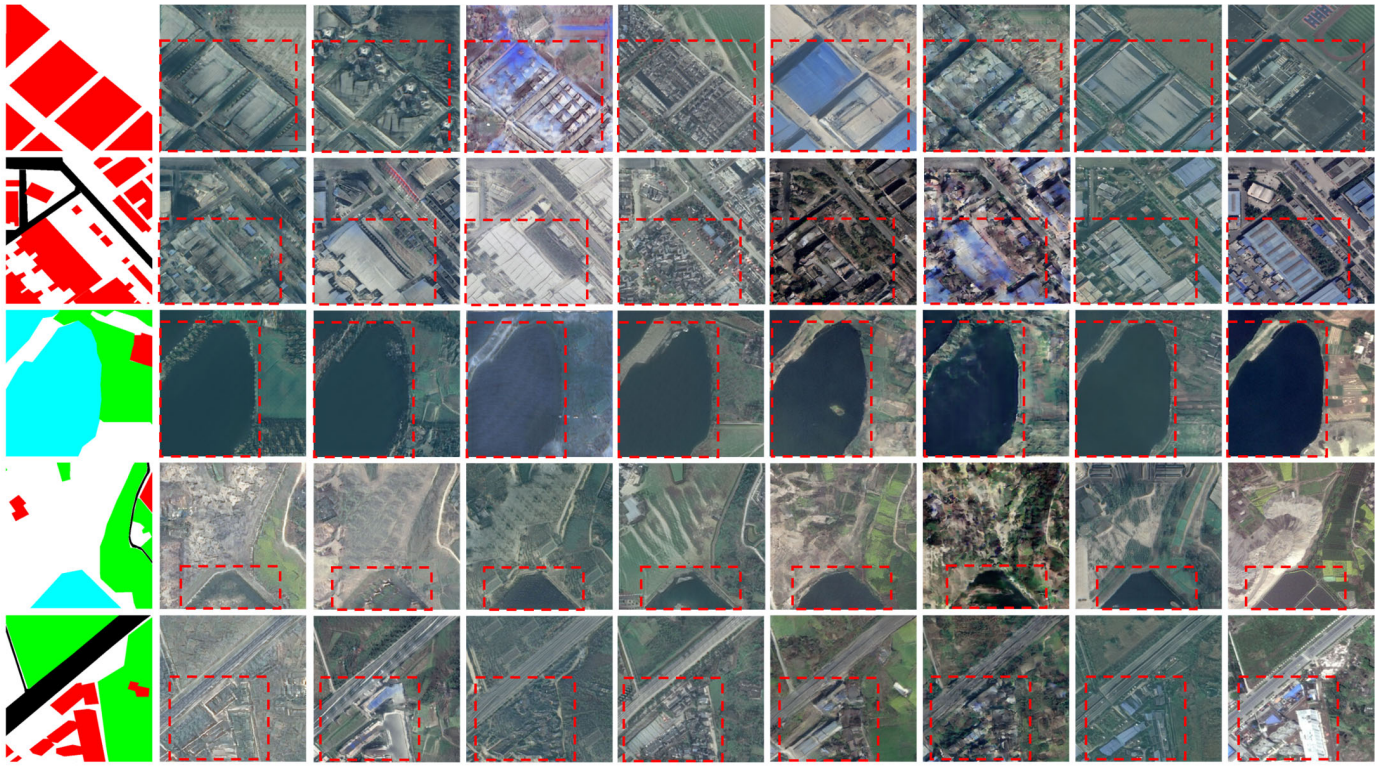



**Figure 10.** The generated results for the Chongzhou dataset.


**Figure 11.** The generated results for the Wuzhen dataset.

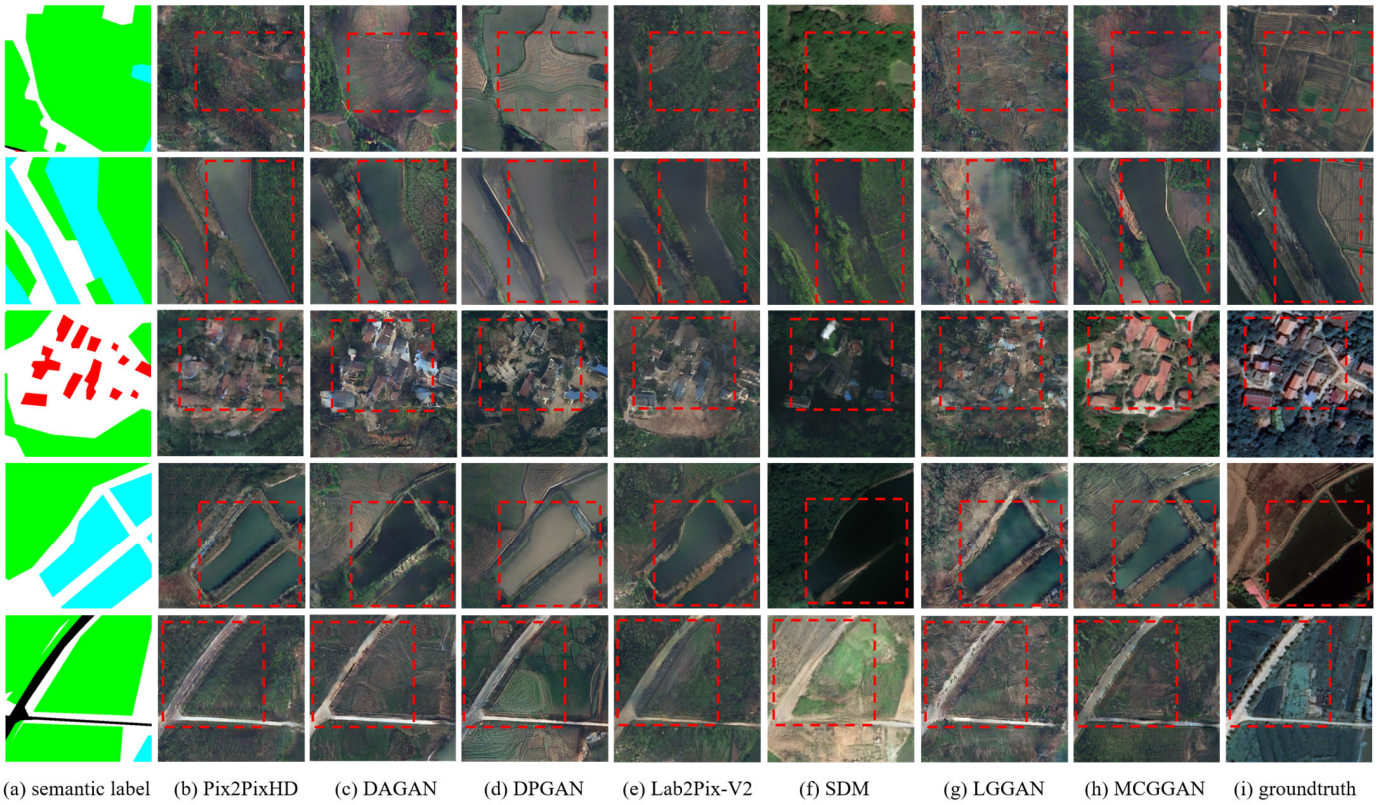



**Figure 12.** The generated results for the LoveDA dataset.

The experimental results demonstrate that MCGGAN achieves the highest accuracy on the Chongzhou dataset, with FWIoU and OA metrics consistently outperforming those of existing models. This suggests that MCGGAN's generated images are more realistic and reliable, contributing to enhanced segmentation performance. Additionally, the FID, LPIPS and SSIM scores for MCGGAN-generated images show significant improvements, indicating that these images align more closely with real remote-sensing data in both overall distribution and individual characteristics.

As shown in Figure 10, for the complex buildings in Chongzhou, MCGGAN generates images with clearer, more defined contours compared to other models, while maintaining better consistency in intra-class information for the same semantic label. Furthermore, MCGGAN excels in generating realistic textures for less common features, such as water bodies and roads. In terms of background and vegetation, MCGGAN offers richer color and texture details, resulting in generated images that are both more realistic and trustworthy.

In the Wuzhen dataset, land-cover types like water and vegetation, which have high semantic similarity, are often confused by other models. However, MCCGAN excels at accurately distinguishing between them. Table 6 indicates that images generated by MCGGAN significantly surpass existing methods across various metrics. The generation metrics reveal that MCGGAN-generated images closely mimic real data in terms of style and distribution. Moreover, the superior FWIoU and OA metrics suggest that the generated images offer more relevant information for the U-Net segmentation network. MCGGAN's advantage lies in its ability to produce vegetation with rich color and texture details, while also excelling in generating features with smaller sample sizes, such as buildings (6.50% of the sample) and roads (2.51% of the sample).

Despite the significant land-cover style differences and sensor inconsistencies in the LoveDA dataset, MCGGAN still achieves superior performance. This is due to the dual-branch architecture, where the multi-class generator focuses on targeted generation for

individual categories, compensating for the global generator's limited learning capability on complex datasets. As shown in Figure 12, MCGGAN excels at generating vegetation, background, and other extensive land-cover types. Its advantages are particularly evident in generating buildings. The LoveDA dataset includes both urban and rural scenarios, featuring diverse architectural styles ranging from low-rise houses in rural areas to high-rise buildings in urban settings. Other models fail to effectively generate buildings, with building contours blending into the background and internal details appearing chaotic. In contrast, MCGGAN successfully captures the structure and style of buildings, accurately delineating contours and preserving internal features.

Comparison experimental results across three datasets demonstrate that MCGGAN exhibits significant advantages over other GAN networks. Compared to the advanced dual-branch network LGGAN, MCGGAN improves the FID metric by 28% and the SSIM metric by 19%. In terms of generated image quality, the multi-class generator equipped in MCGGAN not only produces high-quality images for underrepresented classes (e.g., buildings and roads) but also effectively distinguishes land-cover types with high semantic similarity (e.g., vegetation and water bodies).

Compared to diffusion models, MCGGAN also shows strong performance in scenarios with limited sample sizes. For instance, MCGGAN outperforms the stable diffusion model with a 19% improvement in FID and a 15% improvement in SSIM. Diffusion models, constrained by their complex noise addition and removal processes, face increased computational costs during training and inference, making it difficult to fully leverage their strengths when only a few hundred training samples are available.

## 5. Discussion

### *5.1. Analysis of the Interpretability of Generated Images in Segmentation Models*

Class Activation Mapping (CAM) [56] is a visualization method that generates a heatmap to visualize the contribution distribution of the original image to the predicted output by applying linear weighting to the feature maps. The CAM method proposed by Zhou et al. involves performing Global Average Pooling (GAP) on convolutional feature maps before the final output layer (softmax layer), using the pooled features as input to a fully connected layer to obtain classification results. By projecting the weights of the output layer back onto the convolutional feature maps, the importance of different image regions can be determined. Selvaraju et al. introduced Grad-CAM [57] as an improvement to the CAM method, using the gradient information of the class output at the last convolutional layer as an importance assessment of the activation units, facilitating understandable visualizations of the model output. Grad-CAM can be applied to multi-class classification problems, allowing each class to have independent explanations, while traditional CAM is limited to binary classification. Additionally, SEG-GRAD-CAM [58], proposed by Vinogradova et al., extends Grad-CAM for use in the semantic segmentation domain.

We utilize the principles of SEG-GRAD-CAM to visualize the CAM for five land-cover types, namely vegetation, water, buildings, roads, and background, across multiple feature layers of the U-Net segmentation network. The feature layers examined include the output features of the second downsampling module, the bottleneck layer between the encoder and decoder, and the output features of the last upsampling module. The second downsampling module of the U-Net segmentation network is the initial convolutional layer, which primarily extracts low-level features such as edges and textures. Consequently, the generated heatmap highlights some edge-like structures in the image as shown in Figure 13. As the convolutional network deepens, it learns higher-level features, such as the shapes of objects, their components, and more abstract and complex spatial relationships.

Thus, the heatmaps from the bottleneck layer and the final layer increasingly resemble the output segmentation masks.

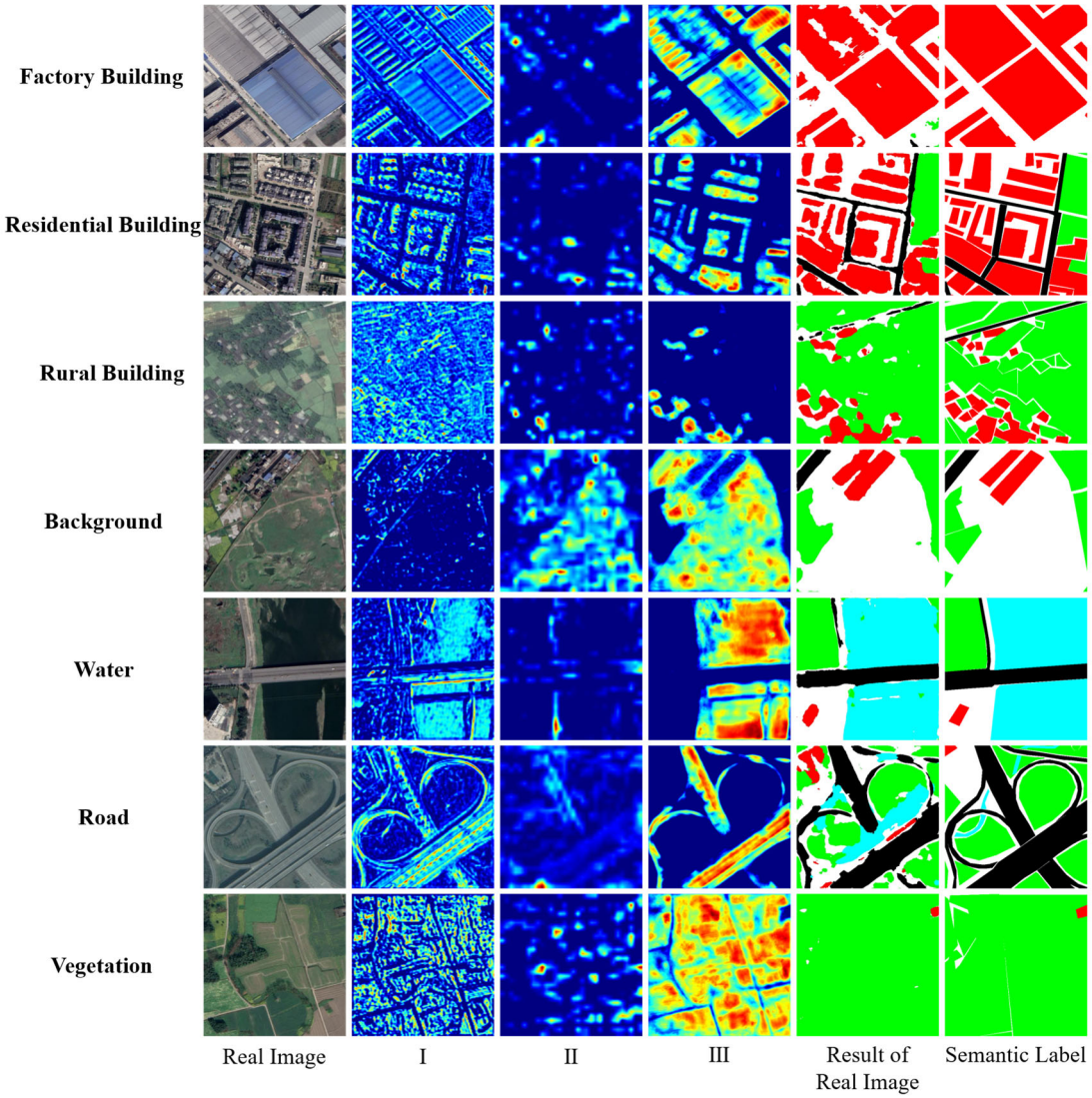



**Figure 13.** CAM visualization results of different U-Net layers in real remote-sensing images. I, II, and III, respectively, represent the visual results of the second downsampling module, the bottleneck layer between the encoder and decoder, and the output features of the last upsampling module.

We first input the real images into the segmentation network, as shown in Figure 13. Rows one to three of Figure 13 visualize the building class, with the first row representing factory buildings, the second row representing complex residential buildings, and the third row representing rural buildings. From the visualization results, it is evident that the segmentation contours of factory and residential buildings play a decisive role. Regardless of whether the heatmaps are generated from the initial convolutional features, the bottleneck layer, or the final layer, the contour areas of the buildings are highlighted. This indicates that the U-Net segmentation network focuses on the contour information of the buildings. Therefore, to improve the segmentation accuracy of the generated images, enhancing the generation of building contours is essential. For rural buildings, the contour information is relatively weak. The heatmaps generated from the second downsampling module and the bottleneck layer show that the segmentation of rural buildings is significantly influenced by the surrounding environment. Thus, the generated images should enhance the color

differentiation between buildings and vegetation to improve the segmentation accuracy of rural buildings.

The fourth row of Figure 13 presents the visualization results for the background, where the heatmaps generated from the bottleneck layer and the final layer primarily highlight pixels that differ significantly from vegetation. Based on the background distribution in the dataset, some background data exhibit similarities to vegetation features. Therefore, to ensure high segmentation accuracy for the background, the generated physical characteristics of the background should maintain a difference from vegetation, minimizing confusion.

Rows five and six of Figure 13 show the visualization results for water and roads, respectively. The results indicate that the textural features of these land-cover types are the main factors affecting their segmentation. These two classes have relatively low representation in the dataset, particularly water bodies, which are easily influenced by vegetation during the generation process, leading to artifacts that can impact the segmentation accuracy of U-Net. The seventh row displays the visualization results for vegetation, where the main factors affecting the segmentation accuracy are, similarly, the color and texture of the vegetation.

MCGGAN's generated results achieved high FWIoU and OA overall, primarily due to its network model design principle of assigning a corresponding class generator for each type of land cover. During the generation process, MCGGAN is able to focus on the details of each land-cover type while also considering global contextual information, thereby enhancing the generation quality for each category. Specifically, MCGGAN improved the contour details of buildings, reduced artifacts caused by interference from other samples during water-body generation, and enhanced the generation quality of roads, which occupy a smaller proportion of the samples, thus improving the overall quality of the generated remote-sensing images.

Figure 14 shows some CAM visualization results of the generated images and the final segmentation results. The first to third rows in the figure display the segmentation results represented by factory buildings, complex residential buildings, and rural buildings. Since MCGGAN can generate images with well-defined contours, the buildings exhibit good segmentation performance. The fourth row of Figure 14 shows the segmentation results of the generated image background, where the color and texture features of the background are more aligned with actual ground features, reducing confusion with vegetation characteristics. Furthermore, the segmentation results and visualization of water bodies, roads, and vegetation from the fifth to seventh rows of the generated images demonstrate that MCGGAN not only ensures the quality of generation for land covers with a large sample proportion but also enhances the generation quality of those with a smaller sample proportion.

### *5.2. The Effect of Adding Generated Samples on Segmentation Accuracy*

The primary objective of this paper is to augment the sample dataset to enhance semantic segmentation tasks in typical land-cover remote-sensing images. We propose the MCGGAN generative model, which demonstrates superior results compared to existing models.

To determine the optimal dataset size and the number of generated images for maximizing segmentation accuracy, we designed the following experiment. First, we created training datasets of different sizes by randomly selecting images from the original Chongzhou training dataset at 80%, 60%, and 40% sizes, and from the Wuzhen training dataset at 90%, 70%, and 50% sizes. This resulted in datasets of varying sizes, as detailed in Table 8. Next, synthetic samples were generated using the trained MCGGAN on these datasets of different sizes and added to the corresponding real datasets for training the

U-Net network. Segmentation accuracy, assessed by FWIoU and OA, was then measured to evaluate improvements in these metrics.

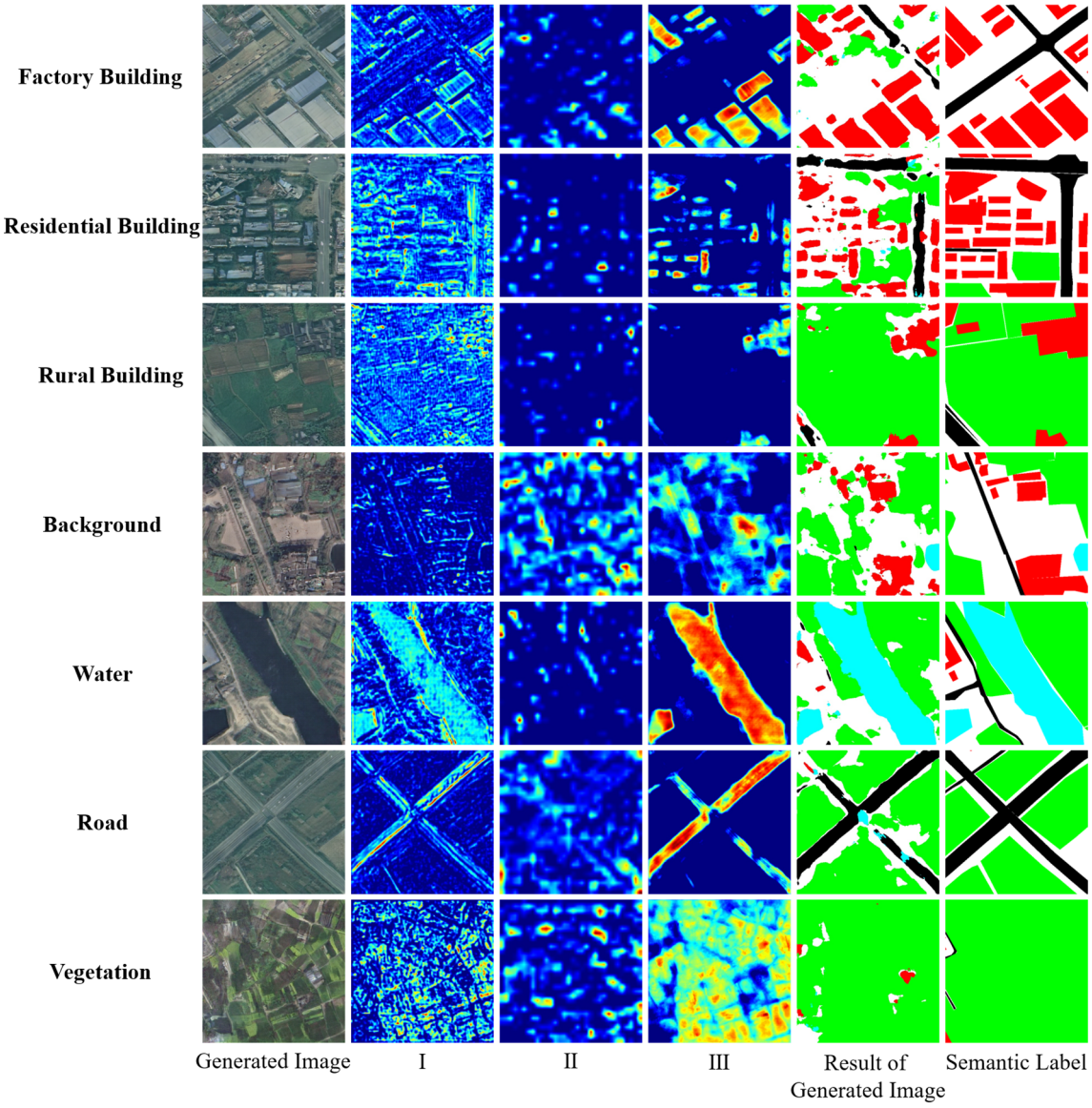



**Figure 14.** CAM visualization results of different U-Net layers in generated images. I, II, and III, respectively, represent the visual results of the second downsampling module, the bottleneck layer between the encoder and decoder, and the output features of the last upsampling module.

**Table 8.** The size of the experimental dataset divided proportionally.

| | Dataset | Training Set | Test Set |
|---|---|---|---|
| **Chongzhou** | Dataset 1.0 | 845 | 211 |
| | Dataset 0.8 | 676 | 211 |
| | Dataset 0.6 | 507 | 211 |
| | Dataset 0.4 | 338 | 211 |
| **Wuzhen** | Dataset 1.0 | 706 | 176 |
| | Dataset 0.9 | 634 | 176 |
| | Dataset 0.7 | 483 | 176 |
| | Dataset 0.5 | 352 | 176 |

For the Chongzhou dataset, 2000 semantic images were created. Samples generated by the MCGGAN model were progressively added to the original training sets, resulting in a total of 500, 1000, and 2000 generated samples. The U-Net segmentation network was then retrained with these augmented datasets. In the case of the Wuzhen dataset, a similar approach was applied. However, due to its simpler image distribution, only 500 samples were incrementally added, totaling 500, 1000, and 1500 generated samples in the training set. The U-Net segmentation network was retrained accordingly. In summary, the training processes were consistent across all datasets. After training, the segmentation performance (FWIoU and OA) of U-Net on the respective test sets was assessed to analyze the impact of the different amounts of generated samples on segmentation accuracy. Table 9 summarizes the FWIoU and OA metrics for the different-sized Chongzhou datasets, both before and after the incremental addition of generated samples.

**Table 9.** The analysis of the impact of Chongzhou's generated images on the accuracy of the U-Net Network.

| Dataset | +0 | | +500 | | +1000 | | +2000 | |
|---|---|---|---|---|---|---|---|---|
| | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) |
| Dataset 1.0 | 66.99 | 78.30 | 68.76 | 79.29 | 69.32 | 79.65 | 68.16 | 78.63 |
| Dataset 0.8 | 67.15 | 78.04 | 69.43 | 79.66 | 69.32 | 79.55 | 67.52 | 78.22 |
| Dataset 0.6 | 63.66 | 75.07 | 67.44 | 78.20 | 67.77 | 78.17 | 67.28 | 77.63 |
| Dataset 0.4 | 64.68 | 76.24 | 65.20 | 76.22 | 66.21 | 77.13 | 65.43 | 76.44 |

Incorporating generated samples into the training set improves the metrics on the test sets across all Chongzhou datasets. The most significant enhancement in segmentation metrics occurs with the addition of 500 generated samples. However, as more samples are included, the improvement becomes less pronounced and may even decline in some cases. As shown in Table 9, a substantial increase in segmentation metrics is observed after adding 500 generated samples, but the improvement diminishes with the addition of 1000 samples, and the metrics decrease when 2000 samples are added.

The statistics of segmentation metrics, including FWIoU and OA, on the test set for different-sized Wuzhen datasets are summarized in Table 10. This table includes results both before and after the addition of 500, 1000, and 1500 generated samples.

**Table 10.** The analysis of the impact of Wuzhen's generated images on the accuracy of the U-Net Network.

| Dataset | +0 | | +500 | | +1000 | | +1500 | |
|---|---|---|---|---|---|---|---|---|
| | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) | FWIoU (%) | OA (%) |
| Dataset 1.0 | 70.76 | 80.84 | 73.57 | 83.05 | 73.60 | 83.23 | 73.36 | 82.91 |
| Dataset 0.9 | 70.21 | 79.82 | 72.37 | 82.25 | 72.53 | 82.21 | 72.08 | 82.13 |
| Dataset 0.7 | 68.38 | 78. 05 | 72.52 | 82.25 | 71.26 | 81.92 | 71.85 | 81.81 |
| Dataset 0.5 | 67.87 | 77.40 | 70.39 | 80.98 | 71.25 | 81.41 | 71.01 | 81.32 |

The trend observed with the different-sized Wuzhen datasets indicates that adding generated samples improves segmentation metrics on the test set. Specifically, the addition of 500 samples yields the most significant improvement, while adding larger numbers of samples does not result in further gains.

Overall, including MCGGAN-generated images enhances U-Net's effectiveness in analyzing remote-sensing images, improving segmentation metrics. However, excessive generated samples can introduce noise and reduce performance, highlighting the need for an

optimal balance. The results suggest that for U-Net training, adding around 500 generated samples achieves a good balance between enhancing segmentation accuracy and maintaining a manageable training process, especially when the training set contains approximately 500 samples.

## 6. Conclusions

We propose a Multi-Class Guided Generative Adversarial Network (MCGGAN) to generate high-fidelity remote-sensing images from semantic labels. MCGGAN uses a dual-branch architecture, with a global generator capturing the overall image structure and multi-class generator improving land-cover differentiation. To ensure consistent output from two branches, we introduce the shared-parameter encoder and spatial decoder. Additionally, the use of perceptual loss ($L_{VGG}$) and texture matching loss ($L_T$) enhances the texture details of the generated images, resulting in more realistic outputs.

To evaluate MCGGAN's image-generation quality, we conducted comparative and ablation experiments on three datasets: two custom datasets (Chongzhou and Wuzhen) and a public dataset (LoveDA). MCGGAN outperforms the comparative methods across all five metrics: FID, LPIPS, SSIM, FWIoU, and OA. With its unique multi-class generator design, MCGGAN outperforms existing methods by generating high-quality images for underrepresented land-cover types despite data imbalance. MCGGAN effectively distinguishes between land-cover types with high semantic similarity, producing images with greater clarity and recognizability. MCGGAN demonstrates better generative quality compared to diffusion models when trained with only a few hundred samples.

Additionally, to assess the impact of generated images on semantic segmentation, we performed segmentation comparisons on datasets of varying sizes from Chongzhou and Wuzhen. Our results show that incorporating generated images boosts UNet segmentation performance, with FWIoU and OA increases of 3.89% and 3.07% on Chongzhou, and 4.47% and 3.23% on Wuzhen, respectively. MCGGAN effectively enhances segmentation accuracy by generating high-quality remote-sensing images, thus achieving data augmentation.

In summary, to address the challenges of severe data imbalance and semantic similarity among land-cover classes in remote sensing, MCGGAN introduces several targeted improvements to the generative model, yielding promising results. However, there is still room for improvement in this paper. The current generative model can only generate a single style of image from a semantic label. Future work will focus on developing a multimodal generation network that enables a semantic label to produce images in multiple styles. By incorporating CLIP for text encoding, we can control the generated image styles based on textual descriptions, such as land-cover types, seasonal changes, or scenarios affected by noise and extreme weather.

**Author Contributions:** Conceptualization, Y.G. and Y.L.; methodology, Z.N. and Y.G.; software, Y.G., Z.N. and M.T.; validation, Z.N., Y.G. and J.Z.; resources, L.H. and Y.H.; writing—review and editing, Z.N., Y.L. and Y.G.; visualization, J.Z., B.Z. and L.H.; supervision, Y.L. and B.Z.; project administration, L.H. and Y.L.; funding acquisition, L.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Key Projects of Global Change and Response of Ministry of Science and Technology of China under Grant 2020YFA0608203. In part by the Science and Technology Support Project of Sichuan Province under Grant 2023YFS0366 and 2024YFFK0414.

**Data Availability Statement:** For more information on the Chongzhou and Wuzhen datasets, please contact the authors.

**Acknowledgments:** The authors are grateful for the producers of the LoveDA dataset (https://github.com/Junjue-Wang/LoveDA (accessed on 11 November 2021)).

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

## References


1. Pan, X.; Zhao, J.; Xu, J. Conditional Generative Adversarial Network-Based Training Sample Set Improvement Model for the Semantic Segmentation of High-Resolution Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7854–7870. [CrossRef]
2. Sun, C.; Zhang, X.; Meng, H.; Cao, X.; Zhang, J. AC-WGAN-GP: Generating Labeled Samples for Improving Hyperspectral Image Classification with Small-Samples. *Remote Sens.* **2022**, *14*, 4910. [CrossRef]
3. Rui, X.; Cao, Y.; Yuan, X.; Kang, Y.; Song, W. DisasterGAN: Generative Adversarial Networks for Remote Sensing Disaster Image Generation. *Remote Sens.* **2021**, *13*, 4284. [CrossRef]
4. Zhao, C.; Ogawa, Y.; Chen, S.; Yang, Z.; Sekimoto, Y. Label Freedom: Stable Diffusion for Remote Sensing Image Semantic Segmentation Data Generation. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023; pp. 1022–1030.
5. Khanna, S.; Liu, P.; Zhou, L.; Meng, C.; Rombach, R.; Burke, M.; Ermon, S. DiffusionSat: A Generative Foundation Model for Satellite Imagery. *arXiv* **2023**, arXiv:2312.03606.
6. Dao, T.; Gu, A.; Ratner, A.; Smith, V.; DeSa, C.; Ré, C. A Kernel Theory of Modern Data Augmentation. *Int. Conf. Mach. Learn.* **2019**, *97*, 1528–1537.
7. Mumuni, A.; Mumuni, F. Data Augmentation: A Comprehensive Survey of Modern Approaches. *Array* **2022**, *16*, 100258. [CrossRef]
8. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification Using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
9. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. *arXiv* **2017**, arXiv:1710.09412.
10. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 26 November 2019; pp. 6023–6032.
11. Shi, J.; Ghazzai, H.; Massoud, Y. Differentiable Image Data Augmentation and Its Applications: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 1148–1164. [CrossRef]
12. Ma, D.; Tang, P.; Zhao, L.; Zhang, Z. A Review of Deep Learning Image Data Augmentation Methods. *J. Image Graph.* **2021**, *26*, 487–502. [CrossRef]
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *Stat* **2014**, *1050*, 10.
14. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 1 September 2022; pp. 10674–10685. [CrossRef]
15. Zou, X.; Li, K.; Xing, J.; Zhang, Y.; Wang, S.; Jin, L.; Tao, P. DiffCR: A Fast Conditional Diffusion Framework for Cloud Removal From Optical Satellite Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5612014. [CrossRef]
16. Xiao, Y.; Yuan, Q.; Jiang, K.; He, J.; Jin, X.; Zhang, L. EDiffSR: An Efficient Dif-fusion Probabilistic Model for Remote Sensing Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5601514. [CrossRef]
17. Bai, X.; Pu, X.; Xu, F. Conditional Diffusion for SAR to Optical Image Translation. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4000605. [CrossRef]
18. Caesar, H.; Uijlings, J.; Ferrari, V. COCO-Stuff: Thing and Stuff Classes in Context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Online, 1 July 2018; pp. 1209–1218.
19. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Online, 5 July 2016; pp. 3213–3223.
20. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing through ADE20K Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 633–641.
21. Wang, C.; Chen, B.; Zou, Z.; Shi, Z. Remote Sensing Image Synthesis via Semantic Embedding Generative Adversarial Networks. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4702811. [CrossRef]
22. Zhang, G.; Zhou, R.; Zheng, Y.; Li, B. Binary Noise Guidance Learning for Remote Sensing Image-to-Image Translation. *Remote Sens.* **2024**, *16*, 65. [CrossRef]
23. Kuang, Y.; Ma, F.; Li, F.; Liu, Y.; Zhang, F. Semantic-Layout-Guided Image Synthesis for High-Quality Synthetic-Aperature Radar Detection Sample Generation. *Remote Sens.* **2023**, *15*, 5654. [CrossRef]

24. Remusati, H.; Le Caillec, J.-M.; Schneider, J.-Y.; Petit-Frère, J.; Merlet, T. Generative Adversarial Networks for SAR Automatic Target Recognition and Classification Models Enhanced Explainability: Perspectives and Challenges. *Remote Sens.* **2024**, *16*, 2569. [CrossRef]
25. Fu, Q.; Xia, S.; Kang, Y.; Sun, M.; Tan, K. Satellite Remote Sensing Grayscale Image Colorization Based on Denoising Generative Adversarial Network. *Remote Sens.* **2024**, *16*, 3644. [CrossRef]
26. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. *Int. Conf. Mach. Learn.* **2017**, *70*, 214–223.
27. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
28. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134. [CrossRef]
29. Ouyang, X.; Cheng, Y.; Jiang, Y.; Li, C.L.; Zhou, P. Pedestrian-Synthesis-GAN: Generating Pedestrian Data in Real Scene and Beyond. *arXiv* **2018**, arXiv:1804.02047.
30. Kong, X.; Shen, Z.; Chen, S. A GAN-Based Algorithm for Generating Samples of Pedestrians in High-Speed Railway Perimeter Environment. *Railw. Perimeter Environ.* **2019**, *55*, 57–61. [CrossRef]
31. Yang, S. Research on Image Generation and Velocity Estimation Based on Generative Adversarial Networks. Master's Thesis, Zhejiang University of Technology, Hangzhou, China, 2019.
32. Wang, Y.; Wang, H.; Xu, T. Aircraft Recognition of Remote Sensing Image Based on Samples Generated by CGAN. *J. Image Graph.* **2021**, *26*, 663–673. [CrossRef]
33. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119. [CrossRef]
34. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
35. Jiang, Y.; Zhu, B. Data Augmentation for Remote Sensing Image Based on Generative Adversarial Networks under Condition of Few Samples. *Laser Optoelectron. Prog.* **2021**, *58*, 238–244.
36. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
37. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807. [CrossRef]
38. Park, T.; Liu, M.Y.; Wang, T.C.; Zhu, J.Y. Semantic Image Synthesis with Spatially-Adaptive Normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2337–2346.
39. Tang, H.; Bai, S.; Sebe, N. Dual Attention GANs for Semantic Image Synthesis. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1994–2002. [CrossRef]
40. Tang, H.; Sebe, N. Layout-to-Image Translation with Double Pooling Generative Adversarial Networks. *IEEE Trans. Image Process.* **2021**, *30*, 7903–7913. [CrossRef]
41. Zhu, P.; Abdal, R.; Qin, Y.; Wonka, P. SEAN: Image Synthesis with Semantic Region-Adaptive Normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5104–5113.
42. Tan, Z.; Chen, D.; Chu, Q.; Chai, M.; Liao, J.; He, M.; Yuan, L.; Hua, G.; Yu, N. Efficient Semantic Image Synthesis via Class-Adaptive Normalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4852–4866. [CrossRef] [PubMed]
43. Berrada, T.; Verbeek, J.; Couprie, C.; Alahari, K. Unlocking Pre-Trained Image Backbones for Semantic Image Synthesis. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 7840–7849. [CrossRef]
44. Wang, W.; Bao, J.; Zhou, W.; Chen, D.; Chen, D.; Yuan, L.; Li, H. Semantic image synthesis via diffusion models. *arXiv* **2022**, arXiv:2207.00050.
45. Li, B.; Xue, K.; Liu, B.; Lai, Y.K. Bbdm: Image-to-image translation with brownian bridge diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1952–1961.
46. Huang, R.; Zhang, S.; Li, T.; He, R. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
47. Gu, S.; Bao, J.; Yang, H.; Chen, D.; Wen, F.; Yuan, L. Mask-guided portrait editing with conditional gans. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019.
48. Li, P.; Hu, Y.; Li, Q.; He, R.; Sun, Z. Global and local consistent age generative adversarial networks. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018.

49. Li, Y.; Li, Y.; Lu, J.; Shechtman, E.; Lee, Y.J.; Singh, K.K. Collaging Class-Specific GANs for Semantic Image Synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 14418–14427. [CrossRef]
50. Wang, J.; Zheng, Z.; Ma, A.; Lu, X.; Zhong, Y. LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv* **2021**, arXiv:2110.08733.
51. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–12.
52. Tong, X.-Y.; Xia, G.S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-Cover Classification with High-Resolution Remote Sensing Images Using Transferable Deep Models. *Remote Sens. Environ.* **2020**, *237*, 111322. [CrossRef]
53. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
54. Tang, H.; Xu, D.; Yan, Y.; Torr, P.H.; Sebe, N. Local Class-Specific and Global Image-Level Generative Adversarial Networks for Semantic-Guided Scene Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7870–7879. [CrossRef]
55. Zhu, J.; Gao, L.; Song, J.; Li, Y.F.; Zheng, F.; Li, X.; Shen, H.T. Label-Guided Generative Adversarial Network for Realistic Image Synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3311–3328. [CrossRef] [PubMed]
56. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
57. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
58. Vinogradova, K.; Dibrov, A.; Myers, G. Towards Interpretable Semantic Segmentation via Gradient Weighted Class Activation Mapping (Student Abstract). *AAAI Conf. Artif. Intell.* **2020**, *34*, 13943–13944. [CrossRef]