drones

MDPI

Article

Improved Image Synthesis with Attention Mechanism for Virtual Scenes via UAV Imagery

Lufeng Mo [1,2], Yanbin Zhu [1], Guoying Wang [1,*], Xiaomei Yi [1], Xiaoping Wu [3] and Peng Wu [1]

[1] College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China
[2] Information and Education Technology Center, Zhejiang A&F University, Hangzhou 311300, China
[3] School of Information Engineering, Huzhou University, Huzhou 313000, China
* Correspondence: wgy@zafu.edu.cn

Abstract: Benefiting from the development of unmanned aerial vehicles (UAVs), the types and number of datasets available for image synthesis have greatly increased. Based on such abundant datasets, many types of virtual scenes can be created and visualized using image synthesis technology before they are implemented in the real world, which can then be used in different applications. To achieve a convenient and fast image synthesis model, there are some common issues such as the blurred semantic information in the normalized layer and the local spatial information of the feature map used only in the generation of images. To solve such problems, an improved image synthesis model, SYGAN, is proposed in this paper, which imports a spatial adaptive normalization module (SPADE) and a sparse attention mechanism YLG on the basis of generative adversarial network (GAN). In the proposed model SYGAN, the utilization of the normalization module SPADE can improve the imaging quality by adjusting the normalization layer with spatially adaptively learned transformations, while the sparsified attention mechanism YLG improves the receptive field of the model and has less computational complexity which saves training time. The experimental results show that the Fréchet Inception Distance (FID) of SYGAN for natural scenes and street scenes are 22.1, 31.2; the Mean Intersection over Union (MIoU) for them are 56.6, 51.4; and the Pixel Accuracy (PA) for them are 86.1, 81.3, respectively. Compared with other models such as CRN, SIMS, pix2pixHD and GauGAN, the proposed image synthesis model SYGAN has better performance and improves computational efficiency.

Keywords: deep learning; unmanned aerial vehicle; image synthesis; generative adversarial network; attention mechanism

Citation: Mo, L.; Zhu, Y.; Wang, G.; Yi, X.; Wu, X.; Wu, P. Improved Image Synthesis with Attention Mechanism for Virtual Scenes via UAV Imagery. *Drones* **2023**, *7*, 160. https://doi.org/10.3390/drones7030160

Academic Editor: Seokwon Yeom

Received: 7 February 2023
Revised: 22 February 2023
Accepted: 23 February 2023
Published: 25 February 2023

1. Introduction

The simulation of image scenes has developed rapidly and is one of the current research hotspots [1]. More and more places need to use image synthesis technology, such as interior design, street design, park landscape preview maps, and so on. A real and reasonable image can improve people's impression of the project, and can also make people feel more intuitively about how the project will look on completion. However, there are few angles available for manual image acquisition and it is more time consuming and laborious. The popularity of unmanned aerial vehicles (UAV) makes the collection of remote sensing image data simpler and more convenient. UAV can obtain images from a wider range with more angles, which greatly expands the source of image synthesis datasets. Compared with artificial image acquisition, that derived from UAV has lower costs and a broad application prospect. Similarly, image synthesis based on deep learning is better than artificial image synthesis [2].

At present, image synthesis methods based on deep learning are mainly based on Generative Adversarial Networks (GAN) [3]. Pix2pixHD is one of the most widely used models at present, and it is a supervised learning model. By inputting the semantic

labels and the ground truth, realistic composite images can be generated in the model [4]. Chen et al. [5] proposed a Cascaded Refinement Network (CRN), which can repeatedly refine the output from low resolution to high resolution, resulting in high-quality images. Qi et al. [6] proposed SIMS, which first divides semantic labels into each plate, identifies patterns similar to the plate in the material library to supplement, and then refines the connections of each plate.

Although deep learning has made some progress in the field of image synthesis in recent years, some aspects need improvement [7]. For example, part of the structure can be optimized, and the receptive field of the model is inadequate [8]. Park et al. [9] showed that the traditional network architecture, which is a superposition of convolution, normalization, and nonlinear layers, is not optimal because their normalization layers tend to reduce the information contained in the input semantic mask. Transposed convolutional layers are a type of basic constituent layer that can capture the spatial properties of natural images, which are important for generating high-quality images. However, it has a major limitation in that it cannot model complex geometries and long-distance dependencies [10]. To compensate for this limitation and expand the receptive field of the model, some have introduced an attention mechanism into the model. This method was first proposed by SAGAN [11]. However, this mechanism also has the following limitations: first, the calculation cost of the standard dense attention mechanism is relatively high; second, when the attention mechanism is calculated, the spatial characteristics of the image are lost in the step of expanding the two-dimensional spatial structure into a one-dimensional vector [12].

To solve the above problems, an image synthesis model SYGAN is proposed in this paper. It is based on adjusted GAN and a spatially adaptive normalization module SPADE [9] and a sparsified attention mechanism YLG [13] which are imported. Using the SPADE module, both the normalization function and the initial semantic information are well retained. The attention mechanism YLG not only effectively improves the reading of feature point information and expands the receptive field of the model, but also reduces the computational complexity, which decreases the requirements of hardware equipment and improves the computational speed of the model.

The main contributions of this paper are as follows: (1) A new image synthesis model SYGAN is proposed. Compared with other models, the model SYGAN adopts a spatially adaptive normalization module and a sparsified attention mechanism to achieve good performance and low complexity. (2) Image synthesis of two kinds of scenes – natural and street scenes – are examined, and the reasons for the difference between the performance for the two scenes are analyzed. (3) Experiments for the comparison of performance of SYGAN and other models such as CRN, SIMS, pix2pixHD, and GauGAN and ablation experiments are conducted to verify the performance of SYGAN.

2. Materials and Methods

2.1. Main Idea

SYGAN, an image synthesis model based on deep learning whose overall structure is shown in Figure 1, is proposed in this paper.

In SYGAN, the image encoder first encodes the real images and then generates the mean and variance vectors, which are used for the noise input of the generator. In addition to these data, the generator also accepts the label images as input to the SPADE block, and then generates the output images. The output images and the real images are used as the input of the discriminator. Finally, the discriminator makes the judgment classification and outputs the attention map to the generator to help it focus on the regions with higher discrimination in the image.

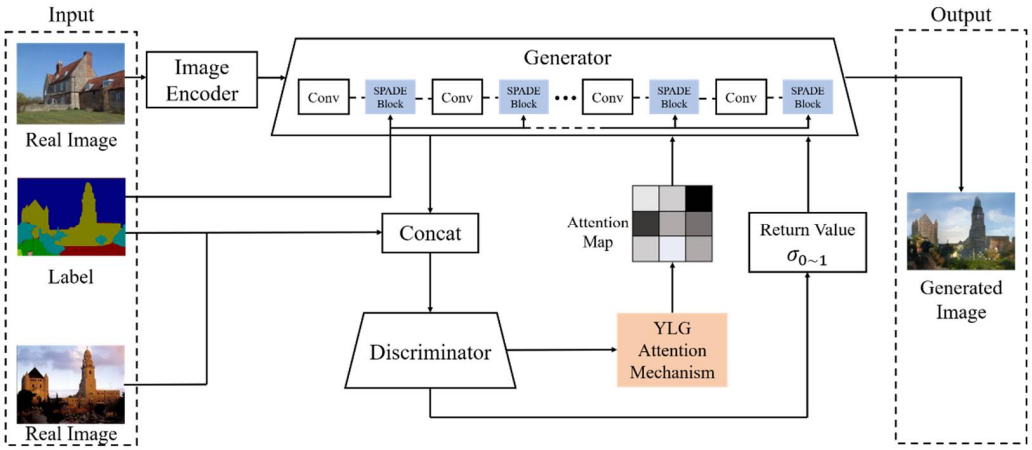

Figure 1. Overall structure of the SYGAN model.

As shown in Figure 1, the main idea of SYGAN includes the following aspects:

(1) Adjusting GAN as a main framework

The main framework of SYGAN is based on GAN which uses generators and discriminators against each other to obtain a reasonable output. As a generative model, it deals well with the problem of data generation. The neural network structure used in this model can fit the high-dimensional representation of various types of data. GAN uses two neural networks against each other and end-to-end optimization, which can effectively improve the training efficiency [14]. The image encoder is mainly composed of a convolutional layer and a linear layer. Real images are encoded as input to generate vector data as input to the generator. The discriminator adopted by SYGAN refers to the classical design of some other models and is mainly composed of convolutional layers. It takes the label image, the output of the generator, and the real image as inputs and judges them.

(2) Importing spatially adaptive normalization module SPADE into the generator

In the past, deep learning-based methods often sent semantic images directly to the neural network in the generator for learning. Although these methods have some impact, they are not conducive to generating high-quality images, because the normalization layer in ordinary neural networks will unconsciously reduce the semantic information. In order to solve this problem, in this study a spatially adaptive normalization module SPADE is imported to replace the ordinary normalization layer, use the layout of input semantic information to activate regulation through spatially adaptively learned transformations, and effectively propagate semantic information throughout the network.

(3) Adding attention mechanism YLG

By modeling the relationships between pixels, the attention mechanism can effectively handle complex geometric shapes and capture long-distance dependencies to further improve network performance [15]. However, common attention also has some of the limitations described above. In view of these, the sparsified attention mechanism YLG is added into SYGAN, which introduces the local sparse attention layer, reducing both the computational complexity and the loss of spatial characteristics when the two-dimensional spatial structure tensor is expanded into one-dimensional spatial structure, and can support good information flow. Compared with other attention mechanisms, the performance and training time have been optimized to a certain extent.

2.2. SYGAN Model

2.2.1. Adjusting GAN as Main Framework

The basic framework for GAN is shown in Figure 2. A set of random noise vectors z satisfying a specified distribution is given as input. The generator *G* will generate a sample

x, and then the discriminator D will make a binary classification decision, resulting in a value $\sigma_{0\sim1}$ (if $\sigma_{0\sim1}$ is 1, it means that the discriminator considers the sample to be a real sample; otherwise, it is a false sample, which means that the sample is generated). There are two types of inputs to discriminator D: generated sample x_f and real sample x_t. In the process of optimizing model parameters through adversarial training, the generator G fits the latent distribution of the real data, so that it is able to synthesize samples that approximate the latent distribution of the real data using the random noise vector z. Then the generated sample x_f and the real data x_t are sent to the discriminator D, which then tries to distinguish the real and fake input samples as much as possible. Meanwhile, the generator G tries to generate samples that are indistinguishable from the real data in order to make the discriminator D judge that the generated samples are true. In the process of confrontation between the generator and the discriminator, both are optimized, and their respective performances are also improved. When the discriminator cannot distinguish the source of the sample data, the optimization ends, and the mathematical expression of the optimization process is shown in Equation (1):

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}}[logD(x)] + E_{z\sim P_z}[log(1-D(G(z)))] \quad (1)$$

where z and x represent the random noise vector and the true sample, respectively. x can be generated by randomly sampling from the true data distribution P_{data}, and z can be generated by sampling from the specified prior distribution P_z. In the process of optimizing this adversarial generative model, the generator attempts to minimize $V(D,G)$, while the discriminator maximizes $V(D,G)$. In the optimization process, an alternate iterative updating method is adopted. First, the generator G is fixed to maximize $V(D,G)$ to solve D, and then the discriminator D is fixed to minimize $V(D,G)$ to solve G.

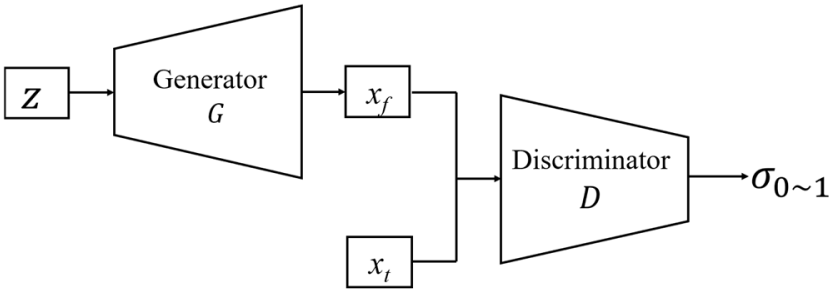

Figure 2. Structure of GAN.

In image synthesis applications, the function of generator G is to process vector data generated by image encoder as input to generate image x_f. The role of the discriminator D is to determine whether the received input samples are generated images x_f or real images x_t. The training goal of generator G is to make its output fool discriminator D, and the goal of discriminator D is to identify which image samples come from discriminator G.

As shown in Figure 3, the encoder consists of six convolutional layers with a step size of 2 and two linear layers to output a mean μ and a variance σ, which are used as the input of the generator. It uses the LReLU activation function [16] and Instance Norm (IN) [17]. LReLU is easy to compute, fast in convergence, and solves the problem of vanishing positive interval gradients. Compared with ReLU [18], it solves the problem that some neurons cannot be activated.

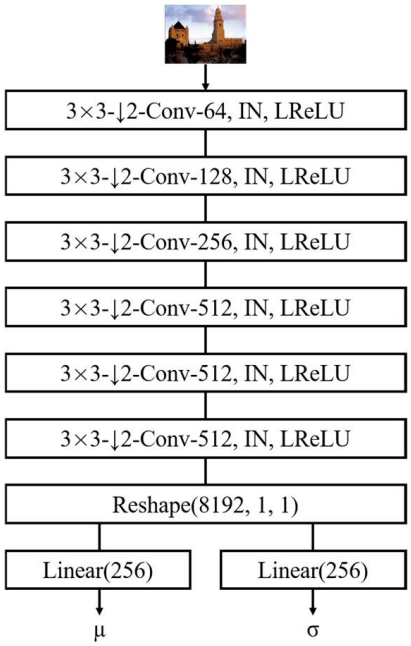

Figure 3. Image encoder.

The discriminator in SYGAN refers to the design of pix2pixHD and Patch-GAN to some extent [19] whose input is segmented images and the connection between the generator output and the real image, uses the LReLU activation function and IN, and takes the convolution layer as the last layer. The output of the discriminator will be received by the attention mechanism YLG (Section 2.2.3) to generate an attention map, which is then input to the generator to assist in its focusing on areas of higher discrimination in the image. Its structure is shown in Figure 4.

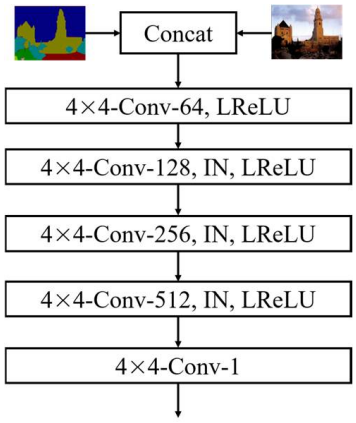

Figure 4. Structure of discriminator.

2.2.2. Importing Spatially Adaptive Normalization Module SPADE into Generator

The structure of the spatially adaptive normalization module SPADE is shown in Figure 5. The label image is first projected onto the embedding space and then convolved to produce the modulation parameters γ and β. Unlike the previous conditional normalization method, γ and β here are not vectors, but tensors with spatial dimensions. The generated γ and β are processed in the next step, similar to batch normalization (BN) [20] It is also regularized in the channel and modulated with the learned scale and bias. The input of SPADE is a segmented image with different colors representing different labels. First, a unified convolution is performed, and then two different convolutions are performed to

generate γ and β with the same number and size as the current number of channels. Next, γ is multiplied by the layer that has just been normalized, and β is added. It is equivalent in that each pixel point of each channel in a layer is normalized separately. In contrast to BN, it depends on the input label image and varies depending on the location. With SPADE, there is no need to input semantic images at the first level of the generator, because the learned modulation parameters already encode enough information about the label layout.

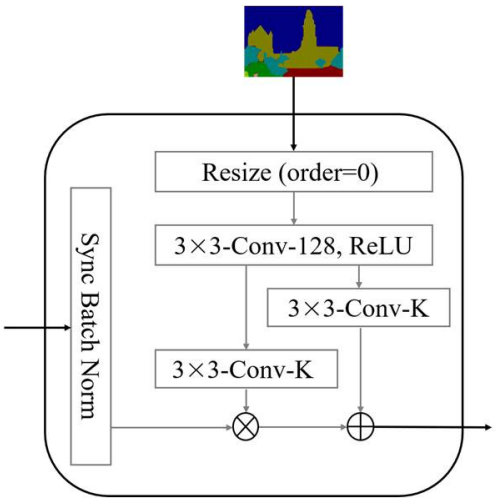

Figure 5. Structure of SPADE.

The SPADE structure is shown in Equations (2)–(4), where h^i represents the activation of the ith layer of the deep convolutional network for a batch of N samples. c^i is the number of channels in the layer, H^i and w^i are the height and width of the activation map in the layer. $h^i_{n,c,y,x}$ denotes activation before normalization, μ^i_c and σ^i_c are the mean and variance in channel c. normally, N is set to 1.

$$\gamma^i_{c,\,y,\,x}(m)\frac{h^i_{n,c,y,x}-\mu^i_c}{\sigma^i_c}+\beta^i_{c,y,x}(m), \tag{2}$$

$$\mu^i_c=\frac{1}{NH^iW^i}\sum_{n,y,x}h^i_{n,c,y,x}, \tag{3}$$

$$\sigma^i_c=\sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}\left(\left(h^i_{n,c,y,x}\right)^2-\left(\mu^i_c\right)^2\right)} \tag{4}$$

The SPADE is combined with the activation function and convolution to form a SPADE block, refer Mescheder et al. [21] and Miyato et al. [22], the SPADE block replaces the commonly used "convolution → activation → normalization" module with "SPADE → activation → convolution". This module can be seen as using the image semantic information to guide the feature map for normalization processing. The structure is shown in Figure 6. In order to solve the problem that the number of channels before and after the residual block is different, a skip connection is added to the structure [23]. That is the portion within the dashed box in Figure 6.

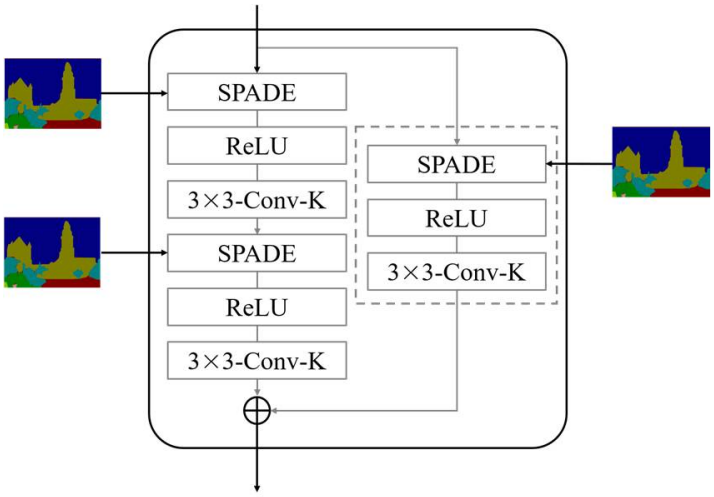

Figure 6. SPADE Block.

Since the learned modulation parameters already encode enough information about the label layout, there is no need to feed the segmented images back to the first layer of the generator, whereby the encoder part of the generator may not be used, which could make the network more lightweight. Figure 7 shows the structure of the generator of SYGAN, which is composed of a series of SPADE blocks and convolutions. The whole network structure is formed by learning the data distribution in a row, and then stacking the SPADE blocks layer by layer. The size of the feature map is from small to large, and the number of channels is from large to small to generate the final real image. In each layer of SPADE block, semantic segmentation images are continuously added to intervene, so that the network can learn multi-scale semantic information in each layer.

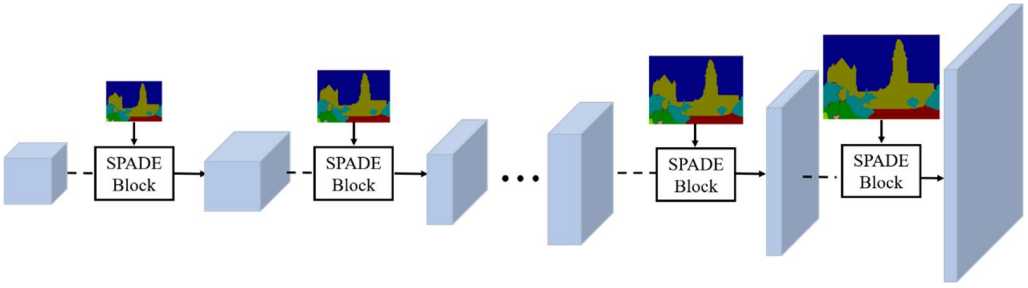

Figure 7. Importing the SPADE block into the SYGAN generator.

In Figure 7, the SPADE block takes the previously output low-resolution image and the different-sized label image of the input image as the input of the next block to generate a higher-resolution image. The growing blue squares are images of increasing size.

2.2.3. Adding Attention Mechanism YLG

The YLG attention mechanism is a sparse attention mechanism, which can improve the computational efficiency of the module. It divides the attention into multiple steps for computation instead of concentrating the computation together. The second-order complexity of the input attention can be expressed by a matrix $A_{X,Y} = X_Q \cdot Y_K^T$.

X, Y is an intermediate representation that associates several matrices with the input. At each step i, attention is directed to a subset of the input locations, which are determined by the binary mask M^i, as shown in Equation (5).

$$A^i_{X,Y}[a,b] = \begin{cases} A_{X,Y}[a,b], & M^i[a,b] = 1 \\ -\infty, & M^i[a,b] = 0 \end{cases} \tag{5}$$

$-\infty$ means that after the function is activated, the value of this position will be cleared, and the calculation will no longer be transferred, so it has no effect on it. Therefore, the design of mask M^i is very important, which is related to the complexity of the data involved in the calculation of attention. The mechanism is designed to solve this problem by using a kind of attention mask that specifies which points have a calculation relationship with points and which points are not settled. The mechanism also refers to the method of Rewon Child et al. [24], which allows individual attention heads to operate on different matrices in parallel, and then connect them in series along the feature dimension. This attention mask also has two modes, which are Left to Right (LTR) in Figure 8a and Right to Left (RTL) in Figure 8b. RTL is the transposed version of LTR. The related information flow diagram is shown in Figure 4. These two modes only allow attention to some areas, which can significantly reduce the quadratic complexity of attention. The mask is actually a superposition of the connected graphs of the two calculations, in which dark blue represents the position of both calculations, light blue represents the position of the first calculation, and green represents the location of the second calculation. The remaining yellow squares represent the positions that are not involved twice, from which the sparsity of the attention mechanism can be reflected.

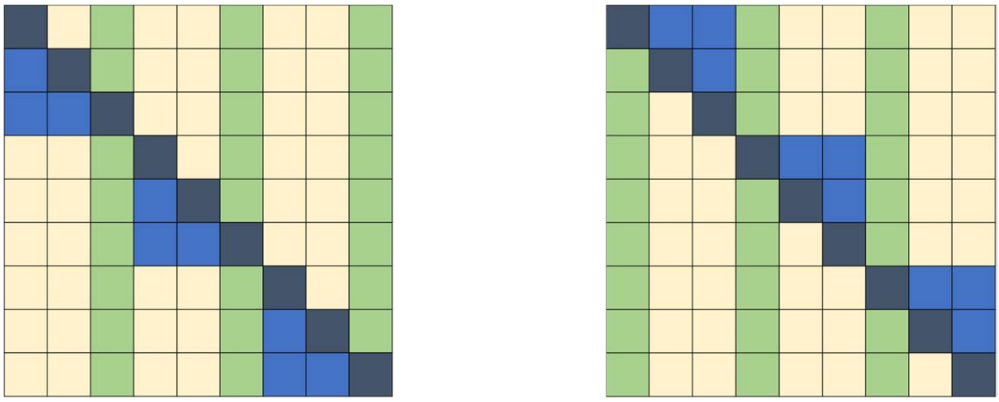

Figure 8. Attention masks. (**a**) Left to right (LTR). (**b**) Right to left (RTL).

2.3. Datasets

Experiments were conducted using the following three datasets:

COCO-Stuff [25]: From the COCO dataset. It has 118,000 training images and 5000 test images from different scenes, containing 182 semantic categories.

ADE20K [26]: Consists of 20,210 training images and 2000 test images. Similar to COCO-Stuff, the dataset contains 150 semantic categories.

UAVid [27]: An image segmentation dataset of urban scenes captured by UAVs, with a total of 3296 images containing 8 semantic categories.

2.4. Design of Experiments

2.4.1. Hardware and Software Configuration

The deep learning framework PyTorch was used to implement the SYGAN model and the experiments. The hardware and software configurations are shown in Table 1.

Table 1. Software and hardware configuration.

Item	Detail
CPU	AMD Ryzen 7 3900X 12-Core processor
GPU	NVIDIA GeForce RTX 3090
RAM	32GB
Operating system	64-bit Windows 11
CUDA	CUDA11.3
Data processing	Python 3.7

2.4.2. Evaluation Indicators

In order to evaluate the accuracy of the model, Pixel Accuracy (PA), Mean Intersection over Union (MIoU) and Fréchet Inception Distance (FID) [28] were used in this paper to measure the gap between the synthetic image distribution and the ground truth distribution.

Pixel Accuracy (PA) is an evaluation criterion for predicting the accuracy of pixels. PA = number of correctly predicted pixels/total number of predicted pixels, as shown in Equation (6):

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{6}$$

The definition of MIoU is given in Equation (7). Where $k+1$ is the number of classes (including null classes), i is the true value, j is the predicted value, and p_{ji} is the number of true values i and predicted values j.

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{7}$$

FID is an index commonly used to evaluate GAN. Its idea is to send the samples generated by the generator and those generated by the discriminator to the classifier respectively, extract the abstract features of the middle layer of the classifier, assume that the abstract features conform to the multivariate Gaussian distribution, and estimate the mean value of the Gaussian distribution of the generated samples' μ_g, variance $\sum g$, training samples μ_{data}, and variance $\sum data$ to calculate the Fréchet distance between two gaussian distributions. In addition, tr represents trace. This distance value is the FID, as shown in Equation (8).

$$FID = \|\mu_{data} - \mu_g\|^2 + tr\left(\sum data + \sum g - 2\left(\sum data \sum g\right)^{\frac{1}{2}}\right) \tag{8}$$

2.4.3. Parameters of Experiments

(1) Loss function.

The loss function is the combination of ordinary cross entropy loss (Cross Entropy Loss) and Dice Loss. Dice coefficient is an aggregate similarity measure function, which is used to calculate the similarity between two samples. The value is usually between 0 and 1, and the lower the loss value, the better the fitting effect and robustness of the synthetic model.

(2) Training parameters.

The learning rates of the generator and discriminator are set to 0.0001 and 0.0004 respectively, and the setting of the learning rates is referred to Heusel et al. [29] The first 200 epochs are performed, and the learning rate is linearly attenuated to 0.00005 over the course of 150 to 200 epochs. The test found that the loss value reached the lowest value of 0.15 after 110 times of training, and then there was almost no change, so the epoch = 120 was determined after comprehensive consideration. Due to the limitation of GPU memory,

when the batch size is greater than 16, it is likely to stop training due to insufficient memory, so it is determined as batch size = 16. The loss function diagram is shown in Figure 9 and the hyperparameter setting is shown in Table 2.

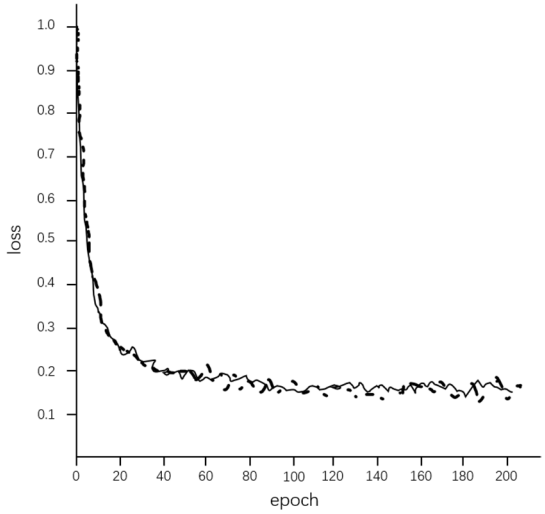

Figure 9. Loss function.

Table 2. Hyperparameter setting.

Item	Value
epoch	120
Batch size	16
Lr(G)	0.0001
Lr(D)	0.0004
Image size	512 × 512

2.4.4. Schemes of Experiments

(1) Comparative experiments.

This part includes two experimental subjects: natural scene and street scene. On the basis of the three datasets, COCO-Stuff, ADE20K, and UAVid, images were selected and classified, and then divided into two new datasets – natural scene and street scene – for training and testing. These are the two most commonly used image scenes, and they have different styles. The difficulty of model training is also different, so it is better to carry out comparative experiments. The training set for each of the two new datasets consists of 10,000 images. The test set for each of the two new datasets consists of 1000 images. The image size used is 512 × 512. Four other models, CRN, SIMS, pix2pixHD, and GauGAN, were used to conduct the comparison experiments.

(2) Computational complexity experiments.

COCO-Stuff were used in this experiment. We counted the number of epochs that reach the highest FID and the time it took each epoch. These data are used to calculate the total time required for training for comparison, so as to compare the complexity between SYGAN and SAGAN [11]. In contrast to SYGAN, other models such as CRN, SIMS, pix2pixHD, and GauGAN do not incorporate attention mechanics, so we don't compare the complexity of SYGAN with that of these models.

(3) Ablation experiments.

The ablation experiment is one of the key factors to assess the quality of the model. The three datasets, COCO-Stuff, ADE20K and UAVid, were used for the experiments to verify the necessity of the corresponding improvement features.

3. Results and Discussion

3.1. Comparative Experiments

In the experiments, the proposed model SYGAN was compared with several image synthesis models: CRN, SIMS, pix2pixHD, GauGAN. CRN uses a deep learning network to repeatedly refine the output from low resolution to high resolution; SIMS uses a semi-parametric method to synthesize real segments from the training set and refine the boundary; pix2pixHD is a conditional image synthesis model based on GAN. A higher value of MIoU and PA indicates better performance, while a lower value of FID indicates better performance. Because the generated image does not need to be completely consistent with the real image, such as vegetation and sky, the image synthesis only needs to be subjectively reasonable to the naked eye, and does not need every tree and cloud to be the same as ground truth, so the MioU index in the above experimental results will be relatively low. However, as it can reflect the coincidence of the generated image and the label image, it can also show the quality of the model to a certain extent.

3.1.1. Natural Scene

Experiments were conducted using natural scene images. MioU, PA, and FID were used as indicators, where the higher the values of MioU and PA, the better the performance, and the lower the value of FID, the better the performance. The results are shown in Figure 10 and Table 3.

Figure 10. Visual comparison of natural scene image synthesis results. (**a**) Label. (**b**) Ground truth. (**c**) CRN. (**d**) SIMS. (**e**) pix2pixHD. (**f**) GauGAN. (**g**) SYGAN (ours).

Table 3. Results of the comparison of natural scene images.

Model	PA (%)	MIoU (%)	FID
CRN	68.4	45.3	48.6
SIMS	63.6	38.6	43.6
pix2pixHD	73.9	46.3	39.8
GauGAN	83.9	54.8	22.6
SYGAN(ours)	86.1	56.6	22.1

It can be seen from Figure 10 that SYGAN, the model proposed in this paper, successfully synthesizes the real details of semantic labels, and the generated images are significantly improved compared with other models, making the generated images closer to human subjective feelings, more natural in various performances, smoother and more natural in the edges of different generated categories. Various indicators also show that the performance of SYGAN is better than the comparative methods.

As is shown in Table 3, the FID of SYGAN was 22.1, which was 26.5, 21.5, 17.7, and 0.5 lower than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively. The MioU of

SYGAN was 56.6%, which was 11.3%, 16%, 10.3%, and 1.8% higher than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively. The PA of SYGAN was 86.1%, which was 17.7%, 22.5%, 12.2%, and 2.2% higher than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively.

3.1.2. Street Scene

The results of experiments for street scenes using SYGAN and the four comparative models are shown in Figure 11 and Table 4.

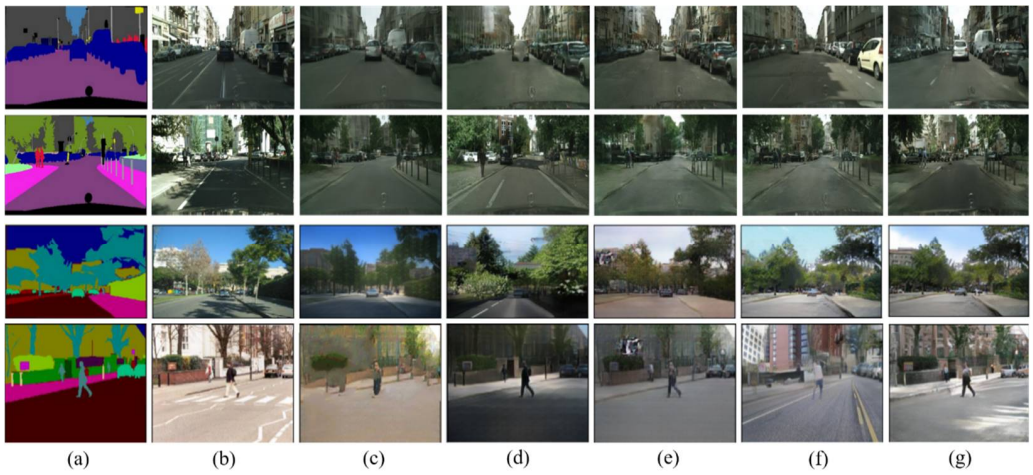

Figure 11. Visual comparison of street scene image synthesis results. (**a**) Label. (**b**) Ground truth. (**c**) CRN. (**d**) SIMS. (**e**) pix2pixHD. (**f**) GauGAN. (**g**) SYGAN (ours).

Table 4. Results of the comparison of street scene images.

Model	PA (%)	MIoU (%)	FID
CRN	67.5	43.5	58.2
SIMS	73.1	34.2	61.3
pix2pixHD	68.9	41.4	47.6
GauGAN	78.8	49.6	33.8
SYGAN(ours)	81.3	51.4	31.2

It can be seen from Figure 11 that the effect of CRN is not good in complex street scenes. Although SIMS looks good, it often deviates from the input label image. Pix2pixHD also has the same problem; the output will be deviated. On the whole, the results of our model SYGAN can achieve more detail than others, which can better generate the semantic information contained in the tags, and the indicators also show that SYGAN has better performance.

As is shown in Table 4, the FID of SYGAN was 31.2, which was 27, 30.1, 16.4, and 2.6 lower than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively. The MIoU of SYGAN was 51.4%, which was 7.9%, 17.2%, 10%, and 1.8% higher than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively. The PA of SYGAN was 81.3%, which was 13.8%, 8.2%, 12.4%, and 2.5% higher than that of CRN, SIMS, pix2pixHD, and GauGAN, respectively.

3.1.3. Comparison of the Two Scenes

According to the indicators in Tables 3 and 4, the performance of all the mentioned methods for natural scenes is better than that for street scenes. As to SYGAN, its PA and MIoU for natural scenes are 86.1 and 56.6 which are 5.90% and 10.12% higher than those for street scenes, respectively, and its FID for natural scenes is 22.1 which is 29.17% lower than that for street scenes.

The reason for the above conclusion is that street scenes are usually more complex than natural scenes. Street scenes usually include many relatively small elements, and there are many complex boundaries between different elements. Conversely, natural scenes tend to have few and large elements, and the boundaries between different elements are relatively long and obvious.

In natural scenes, there are usually four or five elements, and the existence of sky is very frequent. This element often makes up a large proportion of the entire image, ranging from 10% to 70%. Other elements that appear in high proportions are mountains, trees, and water. The distribution of these elements is concentrated, and they usually have long, smooth boundaries. In a street scene, there are usually groups of seven or eight elements. The components are fixed, like buildings, cars, trees. The different elements are scattered and cover each other. Trees usually appear alone, so they have uneven boundaries.

3.2. Computational Complexity Experiments

Compared with CRN, SIMS, pix2pixHD, and GauGAN, SYGAN has a relatively high complexity due to the addition of attention mechanism, but it achieves better synthesis quality. Therefore, the complexity analysis in this paper does not consider the comparison with the above four methods, but only with SAGAN in terms of complexity. SAGAN also introduces the attention mechanism in the network, which solves the limitation of the receptive field size caused by the convolutional structure, and also enables the network to learn different areas that to which attention should be paid in the process of generating images. However, the dense attention mechanism also brings some problems, such as high computational cost. Compared with the comparison method, SYGAN uses the YLG attention mechanism. In terms of ensuring accuracy, it can also reduce the overhead brought by the attention mechanism. The experimental results are shown in Figure 12. It can be seen that the proposed model SYGAN has reached the best FID at about epoch = 110, with an average of 21 min each time, while SAGAN needs about 140 times to reach the best FID, with an average of 19 min each time. The overall time of SYGAN has an advantage over SAGAN and the FID performance is better.

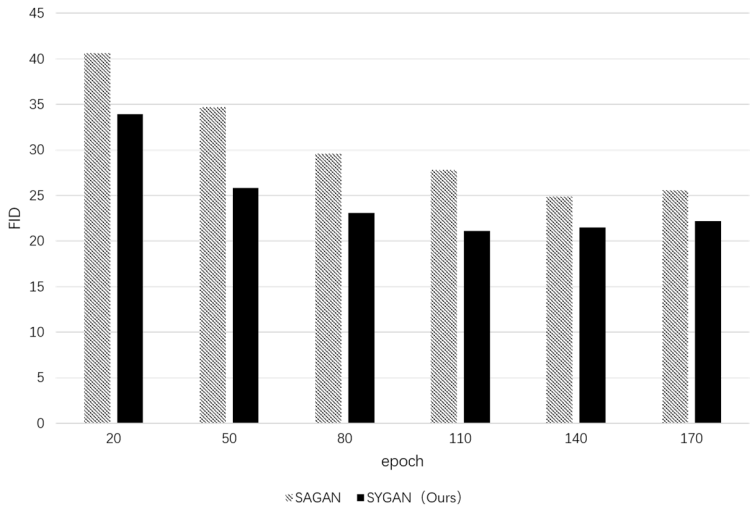

Figure 12. Relationship between epoch and FID.

3.3. Ablation Experiments

Ablation experiments were performed using public datasets with the same hyper-parameters. The results of ablation experiments are shown in Tables 5–7 where SGAN represents the model without YLG, YGAN represents the model without SPADE, GAN represents the models without SPADE and YLG. Higher MIoU and PA values in the table indicate better performance, and lower FID values indicate better performance.

Table 5. Results of ablation experiments on the COCO-stuff.

Model	PA (%)	MIoU (%)	FID
SYGAN	69.5	48.2	22.3
SGAN	66.3	46.1	25.3
YGAN	55.4	38.6	36.5
GAN	33.4	30.6	68.2

Table 6. Results of ablation experiments on the ADE20K.

Model	PA (%)	MIoU (%)	FID
SYGAN	81.4	51.3	37.8
SGAN	78.6	48.1	42.3
YGAN	68.2	41.8	51.2
GAN	44.3	25.6	71.5

Table 7. Results of ablation experiments on the UAVid.

Model	PA (%)	MIoU (%)	FID
SYGAN	86.3	57.1	32.3
SGAN	82.9	54.3	36.2
YGAN	71.5	46.3	46.1
GAN	49.6	29.8	70.3

It can be seen from Tables 5–7 that the FID of SGAN and YGAN has a high improvement compared with that of GAN, indicating that SPADE and YLG have a very good improvement on the performance of the model. The FID of SGAN is improved by about 10 compared with that of YGAN, indicating that the performance improvement of SPADE is greater than that of YLG. SYGAN, when combined with SPADE and YLG, has about 4 and 14 improvements, respectively, compared with SGAN and YGAN. The YLG attention mechanism combined with Figure 12 shows that compared with the usual intensive attention mechanism, it can significantly reduce the computational complexity and improve the training speed.

4. Conclusions

An image synthesis model SYGAN is proposed in this paper, which imports a spatial adaptive normalization module SPADE and an attention mechanism YLG on the basis of GAN. These improvements ensure the model has good performance, increases the accuracy of image synthesis, reduces the generation of false features, expands the receptive field of the model, and shortens the training time. The PA of the model SYGAN is 86.1% in the natural scene dataset, and 81.3% in the street scene dataset. The MIoU of the model SYGAN is 56.6% in the natural scene dataset, and 51.4% in the street scene dataset. The FID score of the model is 22.1 in the natural scene dataset, and 31.2 in the street scene dataset. SYGAN has a better performance in the natural than the street scene. Compared with other models in the experiment, the synthesis effect is better in both datasets. In the computational complexity experiments, the training time of SYGAN is shorter and the FID lower than that of SAGAN with the addition of typical attention mechanisms. From the experimental results, we can see that the model has a good performance as it generates a virtual image through the label image, which can easily preview engineering tasks. This has a very positive significance for the construction of smart cities.

Although SYGAN can complete the task of image synthesis well, it generates some problem images in complex environments, edge generation, and shadow display, which does not conform to the subjective impression of human beings. This will be studied and solved in our future study and work.

Author Contributions: Conceptualization, Y.Z. and L.M.; methodology, Y.Z., G.W., X.Y., X.W. and P.W.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z., X.Y., X.W. and P.W.; investigation, Y.Z., G.W., X.Y., X.W. and P.W.; data curation, Y.Z.; writing—original draft. Y.Z.; resources, G.W. and L.M.; writing—review & editing, G.W. and L.M.; visualization, G.W., X.Y. and X.W.; supervision, G.W. and L.M.; project administration. G.W.; funding acquisition, L.M. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (Grant number: U1809208) and the Key Research and Development Program of Zhejiang Province (Grant number: 2021C02005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Botín-Sanabria, D.M.; Mihaita, A.-S.; Peimbert-García, R.E.; Ramírez-Moreno, M.A.; Ramírez-Mendoza, R.A.; Lozoya-Santos, J.D.J. Digital twin technology challenges and applications: A comprehensive review. *Remote Sens.* **2022**, *14*, 1335. [CrossRef]
2. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 4401–4410.
3. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
4. 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, 18–22 June 2018; pp. 8798–8807.
5. Chen, Q.; Koltun, V. Photographic Image Synthesis with Cascaded Refinement Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1511–1520.
6. Qi, X.; Chen, Q.; Jia, J.; Koltun, V. Semi-Parametric Image Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8808–8816.
7. Bai, G.; Xi, W.; Hong, X.; Liu, X.; Yue, Y.; Zhao, S. Robust and Rotation-Equivariant Contrastive Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–14. [CrossRef]
8. Wang, H.; Zhang, Y.; Yu, X. An overview of image caption generation methods. *Comput. Intell. Neurosci.* **2020**, *2020*, 3062706. [CrossRef] [PubMed]
9. 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, 16–17 June 2019; pp. 2337–2346.
10. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a Convolutional Neural Network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
11. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-Attention Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
12. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
13. Daras, G.; Odena, A.; Zhang, H.; Dimakis, A.G. Your local GAN: Designing Two Dimensional Local Attention Mechanisms for Generative Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 14531–14539.
14. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
15. Fukui, H.; Hirakawa, T.; Yamashita, T.; Fujiyoshi, H. Attention Branch Network: Learning of Attention Mechanism for Visual Explanation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 10705–10714.
16. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex Made More Practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
17. Cai, T.; Luo, S.; Xu, K.; He, D.; Liu, T.-Y.; Wang, L. Graphnorm: A Principled Approach to Accelerating Graph Neural Network Training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 1204–1215. [CrossRef]
18. Hara, K.; Saito, D.; Shouno, H. Analysis of Function of Rectified Linear Unit Used in Deep Learning. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.
19. 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.

20. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
21. Mescheder, L.; Geiger, A.; Nowozin, S. Which Training Methods for GANs do Actually Converge? In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3481–3490.
22. Miyato, T.; Koyama, M. cGANs with Projection Discriminator. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
23. Mazaheri, G.; Mithun, N.C.; Bappy, J.H.; Roy-Chowdhury, A.K. A Skip Connection Architecture for Localization of Image Manipulations. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 119–129.
24. Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating long sequences with sparse transformers. *arXiv* **2019**, arXiv:1904.10509.
25. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1209–1218.
26. 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.
27. Pedamonti, D. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *arXiv* **2018**, arXiv:1804.02763.
28. Obukhov, A.; Krasnyanskiy, M. Quality Assessment Method for GAN Based on Modified Metrics Inception Score and Fréchet Inception Distance. In Proceedings of the Computational Methods in Systems and Software, Online, 14–17 October 2020; pp. 102–114. Available online: https://link.springer.com/chapter/10.1007/978-3-030-63322-6_8 (accessed on 6 February 2023).
29. 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.