Ground-based weather radar can observe a wide range with a high spatial and temporal resolution. They are beneficial to meteorological research and services by providing valuable information. Recent weather radar data related research has focused on applying machine learning and deep learning to solve complicated problems. It is a well-known fact that an adequate amount of data is a positively necessary condition in machine learning and deep learning. Generative adversarial networks (GANs) have received extensive attention for their remarkable data generation capacity, with a fascinating competitive structure having been proposed since. Consequently, a massive number of variants have been proposed; which model is adequate to solve the given problem is an inevitable concern. In this paper, we propose exploring the problem of radar image synthesis and evaluating different GANs with authentic radar observation results. The experimental results showed that the improved Wasserstein GAN is more capable of generating similar radar images while achieving higher structural similarity results.

Keywords: weather radar; data augmentation; generative adversarial networks; structural similarity

1. Introduction

Weather radar, which uses electromagnetic waves to identify precipitations in the atmosphere, is one of the frequently used remote sensing devices. It can provide observation results with remarkably high spatiotemporal resolution in a wide observation area, which is beneficial to meteorological research and services. There are a large number of applications utilizing the observation results, such as quantitative precipitation estimation [1], convective storm nowcasting [2], downburst prediction [3], and post-event analysis [4]. Unfortunately, undesirable echoes, which are also called non-meteorological echoes, might contaminate the observation results when weather radar determines the location and the intensity of the precipitation. Sea clutter [5], ground clutter [6], and anomalous propagation [7] are typical examples of the non-meteorological echoes and make a quality control process [8] complicated by their similar characteristics to meteorological echoes. Therefore, a large number of domestic and foreign scholars have paid attention to implementing precise prediction methods or automated quality control processes to alleviate the experts’ tasks based on statistical [9] and machine learning methods, such as fuzzy logic [10], Bayesian [11], random forest [12], and clustering [13]. Furthermore, lots of meteorological researchers and forecasters have recently focused on deep learning-based approaches, which have shown their superior capability to solve real-world problems in various fields [14].

An adequate amount of data and equally distributed data are positively necessary to implement an effective and appropriately working machine learning method. However, it is nearly impossible to obtain ideal datasets when solving real-world problems: most of them tend to be imbalanced and insufficient. In the downburst prediction using machine learning techniques [15], for example, it requires enormous exertion to gather a sufficient amount of information on each specific class, even
though weather radar data are for free and plentiful in many countries. Data augmentation [16] is a popular selection method for solving the problems mentioned above by utilizing the given dataset. It can increase the quantity of the entire dataset to make it sufficient. By utilizing the data augmentation, it allows saving time and resources by generating realistic data based on a relatively small number of observed cases annotated by experts. Additionally, it can adjust the balance between majority and minority classes by increasing or decreasing in quantity of the minor or major class data. There are several traditional approaches to obtain augmented data: cropping; brightness control; and simple geometric transformations such as translation, rotation, and resizing. While traditional approaches can relieve some of those problems, they fundamentally generate data highly correlated with the data provided.

In this case, GAN (generative adversarial network) [17] can be the best alternative solution. The GAN consists of two models, the generator and the discriminator. They intensely compete and cooperate while training—the generator endeavors to produce elaborate counterfeit data that thoroughly fools the discriminator, and the discriminator exerts itself to distinguish between the counterfeit data and the original data. By using this intriguing structure, it has been providing a novel form of data augmentation and proving its potential in various fields, including natural language processing [18], hyperspectral image classification [19], and medical imaging [20]. Consequently, the active studies of the GAN yield a massive number of fascinating methods. However, which model is appropriate to solve a problem in a given situation remains a concern.

In this paper, we propose exploring the problem of data augmentation for radar data analysis and evaluate different models with actual radar observation results. We select six models of the generative adversarial networks: GAN, conditional GAN (CGAN) [21], deep convolutional GAN (DCGAN) [22], InfoGAN [23], least square GAN (LSGAN) [24], and improved Wasserstein GAN (WGAN-GP) [25,26].

The rest of this paper is organized as follows: Section 2 introduces GANs’ theory and architecture, including principles, variants, and learning schemes. Section 3 presents and discusses the experimental results of GANs. Finally, the concluding remarks are made in Section 4.

2. Methods

GANs were inspired by game theory and offered a fascinating substitute to maximum likelihood techniques: the generator and discriminator are the key components, which compete with each other to achieve the Nash equilibrium in the training process. The adversarial composition is the shared architecture of GAN and its variants. In this section, we represent the theory and architectures of the basic and variants of GANs: GAN, CGAN, DCGAN, InfoGAN, LSGAN, and WGAN-GP.

2.1. GAN

The architecture of the GAN is shown in Figure 1a. The learning scheme of the GAN trains a generator $G$ and a discriminator $D$ simultaneously: The generator captures the data distribution and generates exquisite counterfeit data from a random noise vector $z$, usually a uniform ($z \sim U(0, 1)$) or normal distribution ($z \sim N(0, 1)$). The discriminator evaluates data for authenticity. In other words, the discriminator does its best to distinguish real data from fake data. It reaches the optimal state when the discriminator cannot determine whether the data comes from the real dataset or the counterfeit. The generator and the discriminator have their loss functions set to achieve the global optimal solution, as shown in Equations (1) and (2).

$$L_G^{GAN} = \mathbb{E}_{z \sim P_z(z)} [\log (D(G(z)))]$$

(1)

$$L_D^{GAN} = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log (1 - D(G(z)))]$$

(2)

The basic idea of the GAN is outstanding, but it also has deficiencies that originate from various reasons, such as vanishing or exploding gradients, mode collapse, and failure to converge. Therefore, various derived generative models were proposed.
2.2. CGAN

The intuitive approach to solve the mode collapse problem is by improving the randomness property of the input noise vector \( z \). The conditional GAN (CGAN), as shown in Figure 1b, introduces the conditional variable \( c \) in both the generator and the discriminator to add conditional information, which can directly influence the data generation process. The input of the generator is both the random noise vector \( z \) and the conditional variable \( c \). The inputs of the discriminator are two-fold: the real data under the control of the same conditional variable \( c \), and the generated data \( G(z, c) \).

The generator and the discriminator have their loss functions, including the conditional variable, as shown in Equations (3) and (4).

\[
L_{CGAN}^G = \mathbb{E}_{z \sim P_z(z)} \left[ \log(D(G(z, c))) \right] \\
L_{CGAN}^D = \mathbb{E}_{x \sim P_{data}(x)} \left[ \log D(x, c) \right] + \mathbb{E}_{z \sim P_z(z)} \left[ \log (1 - D(G(z, c))) \right]
\]

2.3. DCGAN

The DCGAN is one of the successful network architectures for GANs. The main contribution of the DCGAN is to suggest a convolution filter based GAN model which provides stable training in most cases. The underlying architecture is the same as the GAN, as shown in Figure 1a. However, it primarily composes convolution layers that can downsample and upsample by utilizing convolutional stride and transposed convolution instead of max pooling and fully connected layers. In the generator, it uses the rectified linear unit (ReLU) except for in the output layer, which uses a hyperbolic tangent function. On the other hand, it utilizes the LeakyReLU in the discriminator. The generator and the discriminator have their loss functions, as shown in Equations (5) and (6).

\[
L_{DCGAN}^G = \mathbb{E}_{z \sim P_z(z)} \left[ \log(D(G(z))) \right] \\
L_{DCGAN}^D = \mathbb{E}_{x \sim P_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim P_z(z)} \left[ \log (1 - D(G(z))) \right]
\]

2.4. InfoGAN

The InfoGAN is another popular form of conditional GAN, which works by introducing mutual information. The mutual information, which represents the correction between the generated data \( G(z, c) \) and the latent code \( c \) that is unknown different to the conditional variable \( c \) in the CGAN, makes the generation process more controllable and makes the generated results more easily interpretable. The essential point of the InfoGAN is that the information in the latent code \( c \) should not be lost in the generation process by changing the loss functions. As shown in the Figure 1c, the InfoGAN has an additional neural network \( Q(c|x) \) with the latent code \( c \). The generator and the discriminator have
their loss functions that include constraint function \( I(c, G(z, c)) \) with a hyper-parameter \( \lambda \), as shown in Equations (7)–(9).

\[
L^{\text{InfoGAN}}_G = \mathbb{E}_{z \sim p_z(z)} [\log(D(G(z)))] - \lambda I(c, G(z, c))
\]

\[
L^{\text{InfoGAN}}_{D,Q} = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] - \lambda I(c, G(z, c))
\]

\[
I(c, G(z, c)) = \mathbb{E}_{c \sim p(c), x \sim G(z, c)} = [\log Q(c|x)] + H(c)
\]

where \( H(c) \) indicates the individual entropy of the latent code \( c \). Maximizing the constraint function \( I(c, G(z, c)) \) makes the latent code \( c \) more reasonable for the generated data quality.

### 2.5. LSGAN

The GAN uses sigmoid cross-entropy loss function in the discriminator, which occurs the vanishing gradient problem when updating the generator. The underlying idea of the LSGAN starts here. Instead of the sigmoid cross-entropy, the LSGAN suggests using least square loss in the discriminator for giving a vanishing gradient problem when updating the generator. The underlying idea of the LSGAN starts here. Instead of using the sigmoid cross-entropy, the LSGAN suggests using least square loss in the discriminator for giving a vanishing gradient problem. Therefore, a novel approach was suggested, solving the problem by utilizing a gradient penalty to enforce the 1-Lipschitz constraint on the discriminator. The improved WGAN (WGAN-GP) allows the generated data to be closer to real data. Equations (10) and (11) are different from the others and intuitive.

\[
L^{\text{LSGAN}}_G = \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - c)^2]
\]

\[
L^{\text{LSGAN}}_D = \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}(x)} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - a)^2]
\]

where \( a \) and \( b \) are the labels for fake data and real data, respectively. \( c \) denotes the value that the generator wants the discriminator to believe for fake data.

### 2.6. WGAN-GP

As mentioned before, the loss functions of the GAN are susceptible to hyperparameter choice and random initialization, which are not desirable. The WGAN remedies the problem by redefining the loss function with Wasserstein distance, which makes the training process stable and less sensitive to hyperparameter selection. The WGAN attempts to minimize an approximation of the intractable Wasserstein-1 distance, also called the earth mover distance, between the distribution of real and generated images. The approximation has requirements, including that the weights of a discriminator must be a \( K \)-Lipschitz function. In other words, the first derivative of the function is bounded everywhere to be less than a constant. The early version of the WGAN introduces a weight clipping within a closed interval \([-c, c]\), where \( c \) is a constant. But the constant \( c \) for weight clipping causes a vanishing gradient problem. Therefore, a novel approach was suggested, solving the problem by utilizing a gradient penalty to enforce the 1-Lipschitz constraint on the discriminator. The improved WGAN (WGAN-GP) allows the discriminator to learn more complex functions and reduces the vanishing and exploding gradient problem. Equations (12) and (13) represent the loss functions of the discriminator and generator, respectively.

\[
L^{\text{WGAN-GP}}_G = -\mathbb{E}_{z \sim p(z)} [D(G(z))]
\]

\[
L^{\text{WGAN-GP}}_D = \mathbb{E}_{x \sim p_{\text{data}}(x), z \sim p(z)} \left[ D(\tilde{x}) - D(x) + \lambda (\|\nabla_x D(\tilde{x})\|_2 - 1)^2 \right]
\]

where \( \epsilon \sim U[0, 1], \tilde{x} = G(z), \tilde{x} = \epsilon x + (1 - \epsilon) \hat{x} \).

### 2.7. Image Quality Assessment

It is widely recognized that the GANs lack an objective function, which makes it difficult to compare the performances of different models. There is no absolute standard to evaluate their performance yet,
notwithstanding that several measures have been introduced [27]. Therefore, it is essential to select an adequate performance measure carefully. In this paper, we chose the structural similarity (SSIM) [28] index, which is a well-known image quality measure to derive the similarity between two images. It evaluates the similarity by predicting human perception similarity judgment by considering three independent quantities to yield an overall similarity measure between two non-negative images \(a\) and \(b\), luminance \(l(a, b)\), contrast \(c(a, b)\), and structure \(s(a, b)\), as shown in Equation (14). Each quantity has a value within \([0, 1]\) and has a high similarity when close to 1.

\[
\text{SSIM}(a, b) = \left( \frac{2\mu_a\mu_b + c_1}{\mu_a^2 + \mu_b^2 + c_1} \right)^\alpha \left( \frac{2\sigma_a\sigma_b + c_2}{\sigma_a^2 + \sigma_b^2 + c_2} \right)^\beta \left( \frac{\sigma_{ab} + c_3}{\sigma_a\sigma_b + c_2} \right)^\gamma
\] (14)

where \(\mu_a, \mu_b\) are the averages, \(\sigma_a^2, \sigma_b^2\) are the variances, and \(\sigma_{ab}\) is the covariance of \(a\) and \(b\). The constants \(c_1, c_2\) are to avoid instability when \(\mu_a^2 + \mu_b^2\) and \(\sigma_a^2 + \sigma_b^2\) are very close to zero, respectively. Additionally, \(\alpha > 0, \beta > 0,\) and \(\gamma > 0\) can adjust the relative importance of the three components. Generally, \(\alpha, \beta,\) and \(\gamma\) are set to 1 and \(c_3 = c_2/2\) to simplify the expression as shown in Equation (15).

\[
\text{SSIM}(a, b) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)}
\] (15)

The SSIM index takes values between 0 and 1, where 1 corresponds to having identical images, and 0 corresponds to a loss of all structural similarity.

3. Results and Discussion

3.1. Data Description

In this paper, we conduct experiments by utilizing the provided data from the Korea Meteorological Administration (KMA), which uses dual-polarization Doppler radars for weather prediction. The dual-polarization Doppler radar can provide useful information, including corrected reflectivity (CZ), differential reflectivity (DR), specific differential phase shift (KD), and cross-correlation (RH). Among them, we chose CZ feature data, which are essential for observing atmospheric scatters, from a radar installed in a specific place that can cover a 240 km radius observation area, including both onshore and offshore.

Additionally, we chose 5000 radar images 480 × 480 in size and 0 to 80 range in dBZ, and categorized three classes according to the ratio of radar echo existing in the observation range, as shown in Figure 2: small amounts (class 0, 206 cases similar to Figure 2a,d), medium amounts (class 1, 1963 cases similar to Figure 2b,e), and large amounts (class 2, 2831 cases similar to Figure 2c,f). By doing this, we redefined and simplified the given problem as a black and white image generative model.

3.2. Experiments

The primary objectives for the GANs in this paper are as follows: First, the generated patterns should appear inside the circle. Second, the generated patterns should be different between classes. Third, the generated patterns should not be identical to the training images. Further, the common settings of GANs are as follows: Batch normalization layers are excluded. The dimension of the input vector of the generator is 200, which follows a normal distribution, \(N(0, 1)\). The training is finished at 300 epochs, and the results are derived in specific epochs: 0 for the beginning, 150 for the intermediate, and 300 for the last step.
Figure 2 shows the graphical results of the GAN, and Figure 4 describes the architectures of the generators and discriminators of the GAN. It contains dense layers, activation layers using a rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent functions. It seems there were no improvements via training until at least 150 epochs. At 300 epochs, the graph shows that the GAN changes significantly. However, Figure 3g,h,i share almost identical outlines. That might indicate the mode collapse problem. The only difference is the level of green area. Therefore, the GAN seems inappropriate to generate synthetic radar images, because it does not satisfy the first and second objectives.

Figure 5 describes the graphical results of the CGAN, and Figure 6 indicates the architectures of the generator and discriminator of the CGAN. It also contains dense layers, activation layers using a rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent function. The difference between it and the GAN is that class information is added when generating synthetic images. It seems that there are no meaningful improvements but noise in the whole epochs. Therefore, the CGAN also seems inadequate to generate synthetic radar images, because it does not satisfy the first and second objectives.

Figure 7 indicates the graphical results of the DCGAN, and Figure 8 presents the architectures of the generator and discriminator of the DCGAN. Unlike previous models, the DCGAN includes not only dense layers and activation layers using a rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent function, but convolution layers that are appropriate to analyze and extract features of given images. It seems that there are remarkable changes compared to previous models. As the epoch increases, the generated pattern seems to build a circular shape. Unfortunately, the outside of the circle also has noisy data, as shown in Figure 7g,h,i. Therefore, the DCGAN also seems improper to generate synthetic radar images, because it does not satisfy the first and second objectives. But it provided us helpful information that includes the convolution layer in the architectures of the generator, and the discriminator allows obtaining favorable results.
(a) Class 0 at epoch 0
(b) Class 1 at epoch 0
(c) Class 2 at epoch 0
(d) Class 0 at epoch 150
(e) Class 1 at epoch 150
(f) Class 2 at epoch 150
(g) Class 0 at epoch 300
(h) Class 1 at epoch 300
(i) Class 2 at epoch 300

Figure 3. Images generated by GAN.

(a) Generator
(b) Discriminator

Figure 4. Modules of GAN.
Figure 5. Images generated by CGAN.

Figure 6. Modules of CGAN.
Figure 7. Images generated by DCGAN.

Figure 8. Modules of DCGAN.
Figure 9 presents the graphical results of the InfoGAN, and Figure 10 shows the architectures of the generator and discriminator of the InfoGAN. The primary difference compared to the previous models, is that the discriminator has the Q network, which computes mutual information in the latent code \( c \) using the softmax layer. The other layers are the same as the DCGAN architecture: convolution layers, dense layers, and activation layers using the rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent function. In the early training stage, it was possible to expect positive results because the InfoGAN generated distinct and intense patterns, unlike previous models. However, as the learning progresses, the generated patterns seem not to converge to original images. It just generates exquisite patterns from given input vectors. Therefore, the InfoGAN also seems improper to generate synthetic radar images, because it does not satisfy the first and second objectives.

Figure 11 shows the graphical results of the LSGAN, and Figure 12 describes the architectures of the generator and discriminator of the LSGAN. The architectures are identical to the DCGAN, except the number of convolution filters: convolution layers, dense layers, and activation layers using the rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent function. It seems that the generated images form a circular pattern in all stages of the epochs. Additionally, it seems that the LSGAN learned features of individual original images in the training process because there is a slight difference between classes. Unfortunately, the generated images still include noise that is impossible to neglect outside the circle. Therefore, the LSGAN also seems improper to generate synthetic radar images, because it does not satisfy the first objective.

Finally, Figure 13 describes the graphical results of the WGAN-GP, and Figure 14 represents the architectures of the generator and discriminator of the WGAN-GP. The architectures are identical to the DCGAN: convolution layers, dense layers, and activation layers using the rectified linear unit (ReLU), LeakyReLU, and hyperbolic tangent function. It seems that the generated images form a circular pattern in the entire stages of epochs. Additionally, it seems that the WGAN-GP learned features of individual original images in the training process because there is a remarkable difference between classes, as shown in Figure 13g,h,i. The final results contain patterns inside the circle; the generated patterns are different between classes; and the generated images are dissimilar to the training images. Therefore, the WGAN-GP seems appropriate to generate synthetic radar images, because it satisfies all objectives.

The SSIM index measures the perceptual difference between two similar images rather than determining which one is better. Therefore, we chose representative examples of each class of radar image in Figure 2 as references. Furthermore, we generated five images at 300 epochs from each GAN model to compare. Table 1 shows the derived SSIM index results. We separated the generated images at 300 epochs into three groups \( A, B, \) and \( C \) for convenience, which included the generated images of the GAN models of Class 0, Class 1, and Class 2, respectively: \( A = \{\text{Figure 3g, Figure 5g, Figure 7g, Figure 9g, Figure 11g, Figure 13g}\}; B = \{\text{Figure 3h, Figure 5h, Figure 7h, Figure 9h, Figure 11h, Figure 13h}\}; C = \{\text{Figure 3i, Figure 5i, Figure 7i, Figure 9i, Figure 11i, Figure 13i}\}. Every first row indicates experimental results between groups \( A, B, \) and \( C \) and original images are shown in the second column.

The WGAN-GP has the best average SSIM index, which is almost 5 to 2000 times better than other GAN models: GAN (0.1241, 0.0075, 0.01500), CGAN (0.0003, 0.0020, 0.0111), DCGAN (0.0018, 0.0073, 0.0707), InfoGAN (0.0009, 0.0061, 0.0082), LSGAN (0.0018, 0.0060, 0.0057), and WGAN-GP (0.6086, 0.4611, 0.3308), where the first, second, and third elements indicate the average SSIM index for Class 0, Class 1, and Class 2, respectively. Therefore, we can conclude that Table 1 numerically underpins the graphical results: that the WGAN-GP generally has better results than the others.
Figure 9. Images generated by InfoGAN.

Figure 10. Modules of InfoGAN.
Figure 11. Images generated by LSGAN.

Figure 12. Modules of LSGAN.
Figure 13. Images generated by WGAN-GP.

Figure 14. Modules of WGAN-GP
Table 1. SSIM (structural similarity) based numerical comparison.

| Class | Case | GAN  | CGAN | DCGAN | InfoGAN | LSGAN | WGAN-GP |
|-------|------|------|------|-------|---------|-------|---------|
| 0     | 1    | 0.1413 | 0.0002 | 0.0013 | 0.0007 | 0.0023 | 0.6245  |
|       |      | 0.1243 | 0.0003 | 0.0015 | 0.0006 | 0.0020 | 0.5687  |
|       |      | 0.1250 | 0.0002 | 0.0014 | 0.0006 | 0.0015 | 0.6022  |
|       |      | 0.1230 | 0.0002 | 0.0016 | 0.0007 | 0.0014 | 0.5455  |
|       |      | 0.1239 | 0.0002 | 0.0013 | 0.0007 | 0.0019 | 0.7570  |
| 1     | 2    | 0.1349 | 0.0004 | 0.0025 | 0.0011 | 0.0025 | 0.5987  |
|       |      | 0.1174 | 0.0003 | 0.0021 | 0.0010 | 0.0018 | 0.5436  |
|       |      | 0.1178 | 0.0005 | 0.0020 | 0.0010 | 0.0013 | 0.5792  |
|       |      | 0.1161 | 0.0004 | 0.0021 | 0.0012 | 0.0013 | 0.5262  |
|       |      | 0.1169 | 0.0004 | 0.0022 | 0.0011 | 0.0016 | 0.7401  |
| 2     | 1    | 0.0058 | 0.0024 | 0.0073 | 0.0050 | 0.0068 | 0.5306  |
|       |      | 0.0091 | 0.0027 | 0.0078 | 0.0063 | 0.0073 | 0.4792  |
|       |      | 0.0097 | 0.0019 | 0.0064 | 0.0068 | 0.0069 | 0.4848  |
|       |      | 0.0136 | 0.0023 | 0.0073 | 0.0059 | 0.0086 | 0.4778  |
|       |      | 0.0034 | 0.0008 | 0.0061 | 0.0062 | 0.0073 | 0.4471  |
| 1     | 2    | 0.0038 | 0.0016 | 0.0083 | 0.0061 | 0.0040 | 0.4424  |
|       |      | 0.0076 | 0.0024 | 0.0089 | 0.0061 | 0.0049 | 0.4803  |
|       |      | 0.0082 | 0.0012 | 0.0073 | 0.0065 | 0.0038 | 0.4379  |
|       |      | 0.0122 | 0.0016 | 0.0068 | 0.0062 | 0.0056 | 0.4258  |
|       |      | 0.0012 | 0.0027 | 0.0063 | 0.0058 | 0.0044 | 0.4054  |
| 2     | 1    | 0.0017 | 0.0151 | 0.0314 | 0.0468 | 0.0332 | 0.2584  |
|       |      | 0.0064 | 0.0116 | 0.0330 | 0.0453 | 0.0252 | 0.3878  |
|       |      | 0.0074 | 0.0077 | 0.0460 | 0.0450 | 0.0358 | 0.3425  |
|       |      | 0.0116 | 0.0086 | 0.0389 | 0.0429 | 0.0318 | 0.3322  |
|       |      | 0.0088 | 0.0095 | 0.0431 | 0.0426 | 0.0333 | 0.3127  |
| 2     | 2    | 0.0387 | 0.0045 | 0.0876 | 0.1255 | 0.0899 | 0.3138  |
|       |      | 0.0130 | 0.0136 | 0.0975 | 0.1197 | 0.0876 | 0.3962  |
|       |      | 0.0168 | 0.0158 | 0.1106 | 0.1164 | 0.0777 | 0.3422  |
|       |      | 0.0248 | 0.0048 | 0.1039 | 0.1215 | 0.0805 | 0.3118  |
|       |      | 0.0205 | 0.0194 | 0.1154 | 0.1129 | 0.0790 | 0.3106  |

4. Conclusions

The results in Section 3 demonstrate the versatility of the GANs for the synthetic data generation of weather radar images. Among selected variants of GANs, the WGAN-GP verifies its ability for the radar data augmentation qualitatively and quantitatively using the SSIM index. Additionally, the results showed that for generating the synthetic images under the same environment (Ubuntu 17.10, TITAN V, 16GB RAM), the WGAN-GP takes less time to converge than other GANs as shown in Table 2. The WGAN-GP finishes its learning process within 8 min, while the InfoGAN consumes more than 17 h. The other GANs spend 7 to 12 h, which is also considerably longer than the WGAN-GP. Therefore, the WGAN-GP can be applied to make a database within a reasonable time by using a small amount of annotated examples by experts to solve challenging real-world problems, including qualitative precipitation estimation, convective storm prediction, and non-meteorological echo elimination. Future work needs to concentrate on applying the WGAN-GP to other features of weather radar, such as DR, KD, and RH, and combining them altogether as a color image.

Table 2. Learning times for 300 epochs (seconds).

|       | GAN  | CGAN | DCGAN | InfoGAN | LSGAN | WGAN-GP |
|-------|------|------|-------|---------|-------|---------|
| Time  | 27,930 s | 28,364 s | 42,950 s | 62,843 s | 42,893 s | 464 s |
Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (number 2018R1A2B6005123).

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

References
1. Tang, L.; Zhang, J.; Langston, C.; Krause, J.; Howard, K.; Lakshmanan, V. A physically based precipitation–nonprecipitation radar echo classifier using polarimetric and environmental data in a real-time national system. *Weather. Forecast.* 2014, 29, 1106–1119. [CrossRef]
2. Han, L.; Fu, S.; Zhao, L.; Zheng, Y.; Wang, H.; Lin, Y. 3D convective storm identification, tracking, and forecasting—An enhanced TITAN algorithm. *J. Atmos. Ocean. Technol.* 2009, 26, 719–732. [CrossRef]
3. Smith, T.M.; Elmore, K.L.; Dulin, S.A. A damaging downburst prediction and detection algorithm for the WSR-88D. *Weather. Forecast.* 2004, 19, 240–250. [CrossRef]
4. Lakshmanan, V.; Herzog, B.; Kingfield, D. A method for extracting postevent storm tracks. *J. Appl. Meteorol. Climatol.* 2015, 54, 451–462. [CrossRef]
5. Karimian, A.; Yardim, C.; Gerstoft, P.; Hodgkiss, W.S.; Barrios, A.E. Refractivity estimation from sea clutter: An invited review. *Radio Sci.* 2011, 46, 1–16. [CrossRef]
6. Hubbert, J.; Dixon, M.; Ellis, S.; Meymaris, G. Weather radar ground clutter. Part I: Identification, modeling, and simulation. *J. Atmos. Ocean. Technol.* 2009, 26, 1165–1180. [CrossRef]
7. Rico-Ramirez, M.A.; Cluckie, I.D. Classification of ground clutter and anomalous propagation using dual-polarization weather radar. *IEEE Trans. Geosci. Remote. Sens.* 2008, 46, 1892–1904. [CrossRef]
8. Friedrich, K.; Hagen, M.; Einolf, T. A quality control concept for radar reflectivity, polarimetric parameters, and Doppler velocity. *J. Atmos. Ocean. Technol.* 2006, 23, 865–887. [CrossRef]
9. Lakshmanan, V.; Zhang, J.; Hondl, K.; Langston, C. A statistical approach to persistent clutter in radar reflectivity data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* 2012, 5, 652–662. [CrossRef]
10. Dufton, D.; Collier, C. Fuzzy logic filtering of radar reflectivity to remove non-meteorological echoes using dual polarization radar moments. *Atmos. Meas. Tech.* 2015, 8, 3985–4000. [CrossRef]
11. Rennie, S.; Curtis, M.; Peter, J.; Seed, A.; Steinle, P.; Wen, G. Bayesian echo classification for Australian single-polarization weather radar with application to assimilation of radial velocity observations. *J. Atmos. Ocean. Technol.* 2015, 32, 1341–1355. [CrossRef]
12. Medina, B.L.; Carey, L.D.; Amiot, C.G.; Medickalski, R.M.; Roeder, W.P.; McNamara, T.M.; Blakeslee, R.J. A Random Forest Method to Forecast Downbursts Based on Dual-Polarization Radar Signatures. *Remote. Sens.* 2019, 11, 826. [CrossRef]
13. Wen, G.; Protat, A.; Xiao, H. An objective prototype-based method for dual-polarization radar clutter identification. *Atmosphere* 2017, 8, 72. [CrossRef]
14. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, 521, 436. [CrossRef] [PubMed]
15. Lagerquist, R.; McGovern, A.; Smith, T. Machine learning for real-time prediction of damaging straight-line convective wind. *Weather. Forecast.* 2017, 32, 2175–2193. [CrossRef]
16. Van Dyk, D.A.; Meng, X.L. The art of data augmentation. *J. Comput. Graph. Stat.* 2001, 10, 1–50. [CrossRef]
17. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
18. Rajeswar, S.; Subramanian, S.; Dutil, F.; Pal, C.; Courville, A. Adversarial generation of natural language. *arXiv* 2017, arXiv:1705.10929.
19. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.* 2018, 56, 5046–5063. [CrossRef]
20. Yi, X.; Walia, E.; Babyn, P. Generative adversarial network in medical imaging: A review. *Med Image Anal.* 2019, 101552. [CrossRef] [PubMed]
21. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* 2014, arXiv:1411.1784.
22. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* 2015, arXiv:1511.06434.
23. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2172–2180.

24. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2794–2802.

25. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.

26. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, United States of America, 4–9 December 2017; pp. 5767–5777.

27. Borji, A. Pros and cons of gan evaluation measures. Comput. Vis. Image Underst. 2019, 179, 41–65. [CrossRef]

28. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 2004, 13, 600–612. [CrossRef] [PubMed]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).