J Inno Sci Eng, 2024, 8(2):213-225
https://doi.org/ 10.38088/jise.1497968

Research Article

Investigating the Effect of Loss Functions on Single-Image GAN Performance

Eyyup Yildiz [1*], Mehmet Erkan Yuksel [2], Selcuk Sevgen [3]

[1*] *Department of Computer Engineering, Erzincan Binali Yıldırım University, 24002, Erzincan, Türkiye*

[2] *Department of Computer Engineering, Burdur Mehmet Akif University, 15200, Burdur, Türkiye*

[3] *Department of Computer Engineering, İstanbul University-Cerrahpaşa, 34320, İstanbul, Türkiye*

Cite this paper as: Yildiz, E., Yuksel, M.E., Sevgen, S. (2024). *Investigating the Effect of Loss Functions on Single-Image GAN Performance.* Journal of Innovative Science and Engineering. 8(2): 213-225

*Corresponding author: Eyyup Yildiz
E-mail: eyyup.yildiz@erzincan.edu.tr

Received Date: 08/06/2024
Accepted Date: 09/08/2024

Abstract

Loss functions are crucial in training generative adversarial networks (GANs) and shaping the resulting outputs. These functions, specifically designed for GANs, optimize generator and discriminator networks together but in opposite directions. GAN models, which typically handle large datasets, have been successful in the field of deep learning. However, exploring the factors that influence the success of GAN models developed for limited data problems is an important area of research. In this study, we conducted a comprehensive investigation into the loss functions commonly used in GAN literature, such as binary cross entropy (BCE), Wasserstein generative adversarial network (WGAN), least squares generative adversarial network (LSGAN), and hinge loss. Our research focused on examining the impact of these loss functions on improving output quality and ensuring training convergence in single-image GANs. Specifically, we evaluated the performance of a single-image GAN model, SinGAN, using these loss functions in terms of image quality and diversity. Our experimental results demonstrated that loss functions successfully produce high-quality, diverse images from a single training image. Additionally, we found that the WGAN-GP and LSGAN-GP loss functions are more effective for single-image GAN models.

Keywords: Generative adversarial networks, Low data regime, Single-image GAN, Loss functions, Image diversity.

1. Introduction

Generative models are a key component of machine learning and aim to generate new data that follow certain distributions. Deep generative models excel at modelling complex data distributions using deep neural networks. These models are capable of learning complex patterns and correlations in training data and can produce data that resemble real-world examples [1, 2]. GANs, which are among the deep generative modelling approaches, have demonstrated very successful results in studies in the field of artificial intelligence as an innovative and dynamic method for generative modelling [3]. The success of GAN architecture relies on the complex interaction between two neural networks known as generator and discriminator. These neural networks are trained, using the same loss function in the GAN framework [3, 4]. Appropriate and effective loss functions are critical to unravelling the complexities of GAN training, addressing inherent challenges, and pushing the boundaries of generative modelling.

GANs are capable of generating a variety of purpose-specific datasets. The potentials of GAN models on low data regimes (e.g., a single natural image) such as data augmentation and synthetic data generation have been widely studied in the literature. Generating high-resolution, realistic, diverse image samples from a single training image (as limited data) has various challenges include capturing complex details, maintaining consistency, and mod collapse [1, 2, 4-8]. These challenges have been overcome with single-image GAN models that combine specialized deep neural network architectures with different GAN structures. Additionally, single-image GAN models enable many applications such as image inpainting, paint-to-image, super-resolution, retargeting, artistic style transfer, image-to-image translation, and animation. These models have an important place in literature in terms of enabling the generation of high-quality, diverse images from a single natural image [9-11].

In GANs, loss functions play a vital role in guiding the training process and shaping the generated outputs. These loss functions are used in a competitive manner between the generator and discriminator networks. The joint loss function helps the generator produce data that are indistinguishable from real samples, while the discriminator correctly classifies real and generated (fake) data [3-6]. The choice of loss functions in GANs significantly impacts the training stability, convergence, and the quality of the generated samples, with each loss function offering a different trade-off in terms of stability, convergence, and ease of optimization.

In our research, we examined how different loss functions impact the performance of a GAN model that generates high-quality, diverse image samples from a single natural image. Specifically, we tested various loss functions commonly used in the literature on the SinGAN model, which served as the basis for our study. Our findings revealed that loss functions have varying levels of effectiveness on single-image GANs. The paper is organized as follows: Section 2 provides a detailed explanation of the GAN loss functions used. Section 3 presents a quantitative and qualitative analysis of the results obtained by using different loss functions with the SinGAN model. The final section evaluates the findings and discusses their implications for future study.

2. Related Work

Generating diverse images from single natural training image is one of the complex problems being studied today Traditional generative models generally work on large datasets. However, Ulyanov et al. (2019) showed that with the Deep Image Prior (DIP) model that starts training with randomly valued weights and could learn the necessary and

sufficient information from a single training image [13]. Shocher et al. (2019) introduced InGAN, an unsupervised conditional GAN trained on a single image that captures the internal statistics of that image. InGAN can synthesize image samples of different sizes, shapes, and aspect ratios that have the same internal patch distribution as the training image [14].

Shaham et al. (2019) proposed the SinGAN model that learns image patches in different dimensions, rather than treating the entire image as a whole [12]. SinGAN consists of fully convolutional GANs connected in a pyramid structure. It functions with scaled training, where each GAN scale accepts a scaled version of the training image as input. The generator at each scale takes input from a lower scale, adds random noise, and produces a new image. Only the lowest scale's generator, which creates the smallest-sized image, uses solely noise input. A patch discriminator [15] is used to differentiate between the images produced at each scale and the corresponding real images. Hinz et al. (2021) introduced ConSinGAN, a model with a pyramid structure similar to SinGAN, but with differences in training methods and the internal structure of the generator [16]. ConSinGAN uses feature maps enriched with more information, rather than generating images at each scale. Additionally, ConSinGAN trains distinct scales with varying learning rates, unlike ProGAN [17], which trains only the final layer. Granot et al. (2021) proposed Generative Patch Nearest-Neighbor (GPNN), a model that creates new images from a single training image without employing GANs. GPNN uses a nearest-neighbor search to measure patch similarities within the training image [18].

Loss functions in GANs are vital for enhancing training stability, sample quality, and convergence. Goodfellow et al. (2014) adopted the Binary Cross Entropy (BCE) loss function in their GAN model to optimize the probability of correctly distinguishing real from generated samples [3]. Although BCE is successful, it may face problems such as convergence limitation and mode collapse. To address these challenges, Arjovsky et al. (2017) introduced the Wasserstein Generative Adversarial Network (WGAN) model, which employs the Wasserstein distance to measure data distribution similarity [19]. Gulrajani et al. (2017) further tackled the weight restriction issue in WGAN by proposing the Wasserstein GAN with Gradient Penalty (WGAN-GP). The Lipschitz continuity introduced by WGAN-GP has notably enhanced the stability and efficiency of the GAN training process [20]. Mao et al. (2017) developed the Least Squares Generative Adversarial Network (LSGAN) model, which employs least squares loss in the form of Mean Squared Error (MSE) to improve training stability and model performance by minimizing sensitivity to noise [21]. Lim et al. (2017) utilized hinge loss, typically used in support vector machines, to improve GAN training. Hinge loss seeks to make the discriminator more discriminative by enforcing a margin around the real/fake data decision boundary and ensures a more consistent and robust gradient flow during backpropagation [22].

3. Methodology

Figure 1 illustrates the multi-scale architecture of the SinGAN model. This model aims to generate realistic, high-quality image samples based on a single training image while preserving the training image's global structure. It employs hierarchically trained PatchGANs (Markovian discriminators). The training process begins by scaling the training image ($Real = \{Real_0, Real_1, \cdots, Real_n\}$) ($n$ is the scale number), which is then used as the training data for each scale of the GAN model ($\{G_0, D_0\}, \{G_1, D_1\}, \ldots, \{G_n, D_n\}$). At each scale, only $\{G_n, D_n\}$ pairs that work at that scale are trained. The Generator (G_0) at the smallest scale ($n = 0$) learns to generate new images from only noise map. Generators at

larger scales ($n \geq 0$) use scaled versions of the image generated from previous scales as inputs, along with noise sampled from a normal distribution. Training is conducted sequentially from the smallest scale to the largest, with each scale of the GAN model trained one at a time. The generator and discriminator networks across all scales share identical structures, each comprising 5 convolution layers. Each layer is composed of 3x3 kernels, batch normalization, and LeakyReLU activation functions (exceptionally, activation function of last layer of generators is Tanh for all scales). Additionally, the generator networks include residual connections that combine input data with output data. Weighted WGAN-GP and L2 summation are utilized as loss functions [12].

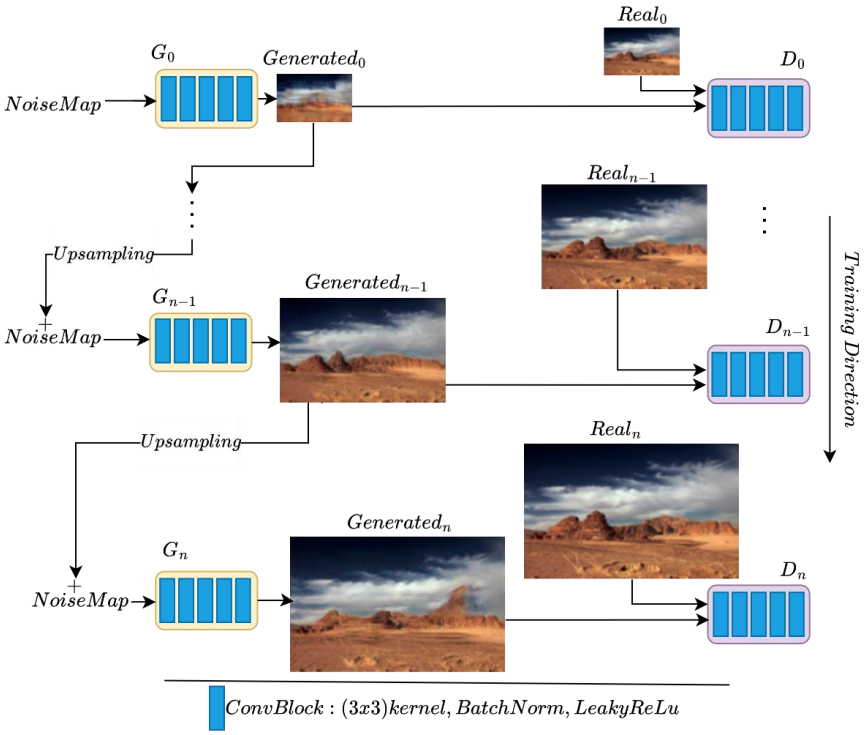

Figure 1: Multi-scale architecture of SinGAN [12].

In our study, we examined four different loss functions for the single-image GAN model (SinGAN): the vanilla GAN loss function (Binary Cross Entropy, BCE), the Least Squares GAN (LSGAN) loss function, the HingeGAN loss function, and the Wasserstein GAN with Gradient Penalty (WGAN-GP) loss function used by SinGAN. We compared the results of these loss functions with those obtained by using the WGAN-GP loss function in SinGAN. The overall loss function for SinGAN models is defined as the weighted sum of adversarial and reconstruction functions, as shown in Equation (1) and (2). $\mathcal{L}_{adv}$ represents the adversarial loss function for the generator and discriminator networks and is analyzed by using BCE, LSGAN, HingeGAN, and WGAN-GP. $\mathcal{L}_{rec}$ ensures that the generator can accurately produce the training image at any scale.

$$\boldsymbol{\mathcal{L} = \mathcal{L}_{adv} + 10 \times \mathcal{L}_{rec}} \quad (1)$$

$$\boldsymbol{\mathcal{L}_{rec} = \|G(z) - x\|_2^2} \quad (2)$$

BCE is designed to address the adversarial aspects of GANs by enhancing the discriminator's classification performance. Its primary goal is to push the generator, trained against the discriminator, to produce data that closely resemble real examples. However, the standard BCE used in traditional GANs has several limitations. For instance, when working with multimodal data, the generator may concentrate excessively on certain modes, resulting in a lack of diversity in the

generated samples, a phenomenon known as mode collapse. Additionally, BCE does not guarantee Nash equilibrium, which is critical for stable convergence. Nonetheless, when the gradient flow is maintained, BCE supports effective training of both generator and discriminator networks. In the context of single-image GANs, where the training data exhibit low mode sparsity and are easily classifiable, it is important to evaluate the performance of BCE [3, 23, 24]. Table 1 presents the vanilla BCE loss function as applied to the patch discriminator, which aims to capture the distribution of trained patches in single-image GAN models.

WGAN-GP employs a loss function commonly used in GAN models to provide a more stable training process. As shown in Table 1, WGAN redefines traditional GAN loss by using a more reliable metric to measure the disparity between the true and generated distributions. The objective is to minimize this gap, thereby enhancing their similarity. By replacing the Jensen-Shannon divergence with the Wasserstein distance, WGAN effectively mitigates issues such as mode collapse and training instability [19, 23]. WGAN-GP enhances the original WGAN by incorporating a gradient penalty - equation (3) - that enforces a Lipschitz constraint on the discriminator network. This ensures that the generator maintains Lipschitz continuity with fewer parameters and operates more efficiently. Though the gradient penalty calculation is computation-intensive and may slow down WGAN-GP, it improves the convergence properties of the conventional GAN training process and enables the generation of high-quality, diverse samples [20, 24].

The LSGAN, as outlined in Table 1, uses mean squared error (MSE) loss to compare real and generated data distributions. Its primary objective is to penalize synthetic samples that deviate significantly from real data while still being on the correct side of the decision boundary. LSGAN aims to generate more gradients and penalize instances far from the decision boundary [21]. The MSE loss modifies the generator's objective to minimize the mean squared error between real and generated data. This approach fosters a more consistent and stable training process, enhancing the quality and variety of the generated samples.

Table 1: Loss functions in generator and discriminator networks.

Loss Function	Generator	Discriminator
BCE	$\mathcal{L}_G = -\mathbb{E}\left[log\left(sigmoid\left(D(G(z))\right)\right)\right]$	$\mathcal{L}_D = -\mathbb{E}\left[log\left(sigmoid(D(x))\right)\right] - \mathbb{E}\left[log\left(1 - sigmoid\left(D(G(z))\right)\right)\right]$
WGAN	$\mathcal{L}_G = -\mathbb{E}[D(G(z))]$	$\mathcal{L}_D = \mathbb{E}[D(x)] - \mathbb{E}[D(G(z))]$
LSGAN	$\mathcal{L}_G = \mathbb{E}\left[\left(D(G(z)) - 1\right)^2\right]$	$\mathcal{L}_D = \mathbb{E}[(D(x) - 1)^2] + \mathbb{E}\left[D(G(z))^2\right]$
Hinge	$\mathcal{L}_G = = -\mathbb{E}[D(G(z))]$	$\mathcal{L}_D = \mathbb{E}[max(0,1 - D(x))] + \mathbb{E}\left[max\left(0,1 + D(G(z))\right)\right]$

The hinge loss function uses a margin around the decision boundary to improve discrimination between training image (real) and generated samples (fake). This enhances the discriminator network's ability. This margin-based approach supports a robust training process and prevents the generator from easily fooling the discriminator. As shown in Table 1, the hinge loss function is similar to the Support Vector Machines (SVM) formulation [22]. It stabilizes the training process, even with noisy or low-quality data. Therefore, the hinge loss function is preferred for low-quality or inconsistent datasets. However, multiple training sessions may be necessary to properly determine the margin hyperparameter [7, 23].

$$GP = \lambda E\left[\left(\left|\left|\nabla D\left(\alpha x + (1-\alpha)G(z)\right)\right|\right|_2 - 1\right)^2\right] \quad (3)$$

Equation (3) defines the gradient penalty (GP). Our experiments demonstrated that omitting GP in the loss functions significantly reduces model convergence. Thus, we applied the GP penalty to all loss functions in this study.

4. Findings

4.1. Unconditional Generation

Our study employs SinGAN as a training model. During the training process, we used the default values of SinGAN. The generator and discriminator networks at each scale were trained for 2000 iterations, with parameters updated at each iteration. LeakyReLU activation was used in all layers except the final layer of the generator network, which used Tanh activation. Convolution blocks at the coarsest scale contained 32 cores, doubling every 4 scales. The learning rate was set to $lr = 5e-4$, with a weight reduction of 0.1 after every 1600 iterations. The coefficients for adversarial and reconstruction loss functions were set to $\alpha_1 = 1$ and $\alpha_2 = 10$, respectively [12]. All experiments were conducted on an Intel i7-10700KF CPU and an NVIDIA TITAN X Pascal GPU.

Fourteen images with distinct patterns were selected from the web to compare loss functions using a single training image. Most of these images contain repetitive patches, which facilitate modeling patch distributions. For each training image, 100 new images were generated post-training, and measurements were conducted on these images. Performance evaluation was conducted by using the Single Image Fréchet Inception Distance (SIFID) metric [12]. The diversity of the generated images was assessed by using the Multi-Scale Structural Similarity Index Measure (MS-SSIM) [25] and Learned Perceptual Image Patch Similarity (LPIPS) [26] metrics. Fréchet Inception Distance (FID) [6] is widely used to evaluate the quality and diversity of generated images. It measures the similarity between real and fake images based on the extracted feature maps. SIFID, a modified version of FID, evaluates single-image GANs by analyzing internal feature map distributions from the Inception network's convolution layer before the second pooling layer. A low SIFID value indicates a high similarity between real and fake images. MS-SSIM, an advanced version of the single-scale Structural Similarity Index (SSIM), is used for image quality assessment. It incorporates changes in image resolution and viewing conditions, providing more flexibility and better performance compared to single-scale methods. It extracts three key features from an image: brightness, contrast, and structure, to compare two images. LPIPS, a deep network-based image similarity metric, is developed to mimic image patch similarity based on human perception. It evaluates the similarity between two images by measuring the distance between image patches. Low or high LPIPS values indicate whether the image patches are perceptually similar or dissimilar.

Figure 2 presents a visual comparison of the loss functions for images randomly selected from the 14 training images. These images indicate that all models successfully learn the patch statistics of the training image. The capability of GANs to learn patch statistics is fundamental to their ability to generate visually convincing images. This learning process is quantitatively evaluated, as shown in Figure 2, which presents a detailed comparison of the performance of various models based on different loss functions. These quantitative results corroborate the visual observations from Figure 2, highlighting that all models exhibit a commendable ability to learn and reproduce the fine-grained details

present in the training images despite variations in the loss functions employed. This visual comparison provides an insightful overview of the effectiveness of different loss functions in guiding the GANs during training.

Figure 2: Random image samples generated by using loss functions.

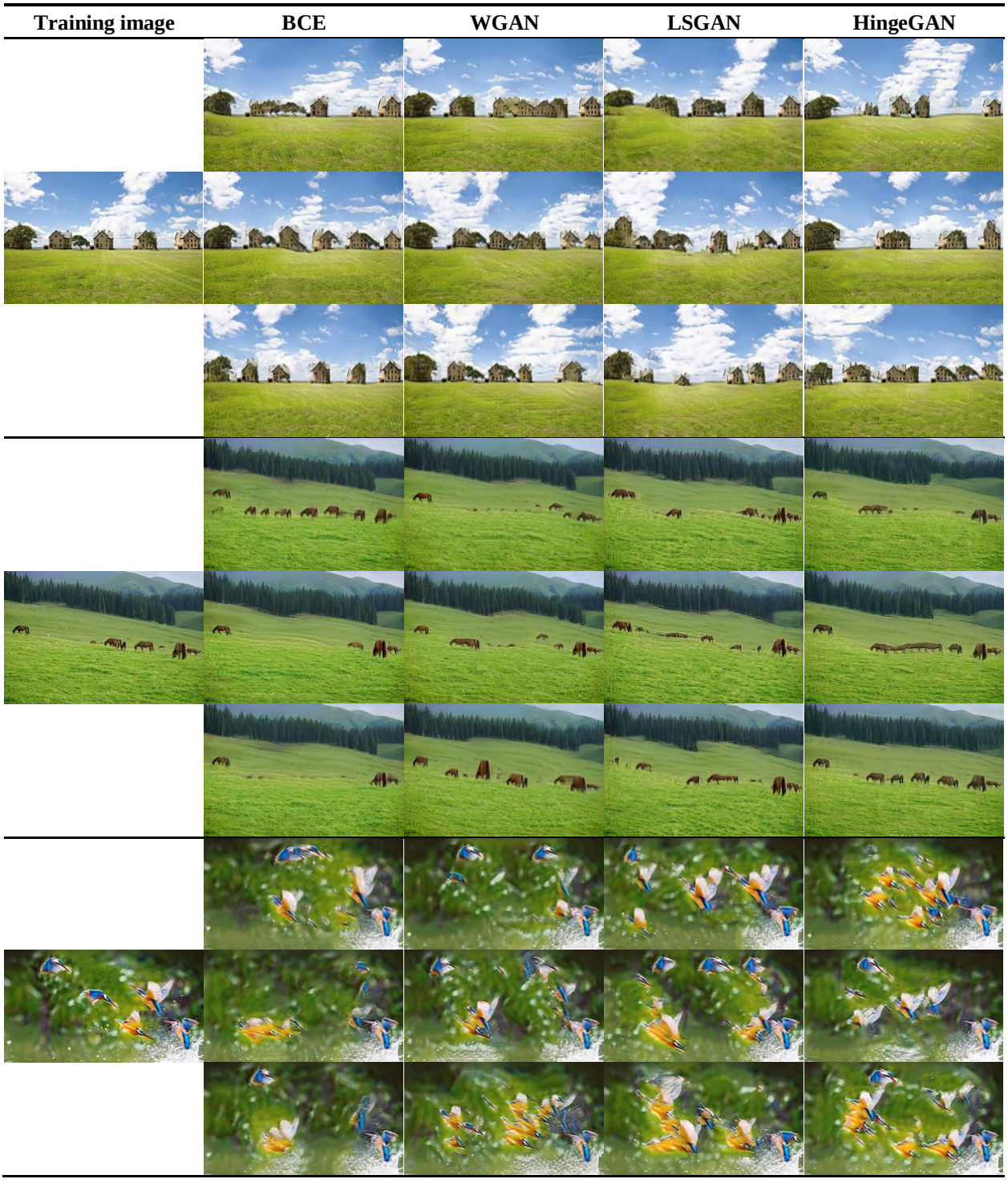

Table 2 lists the quantitative results of the loss functions. According to the SIFID values, WGAN-GP and LSGAN-GP achieved the best results, indicating that these loss functions better optimize the SinGAN model to produce realistic, high-quality images. For MS-SSIM, all models had close results, but WGAN-GP performed better. For LPIPS, the performance range was narrower, with LSGAN-GP outperforming the others.

Table 2: Quantitative results of Loss Functions.

Loss function	SIFID(↓)	MSSSIM(↓)	LPIPS(↑)
WGAN-GP	**0.05**	**0.52**	0.40
BCE-GP	0.08	0.55	0.40
LSGAN-GP	**0.05**	0.61	**0.43**
HingeGAN-GP	0.06	0.57	0.41

4.2. Applications

We examined the performance of single-image GAN models for different loss functions across various applications. In this context, we performed image harmonization and image editing applications, which are the primary applications where single-image GAN models are typically tested [12].

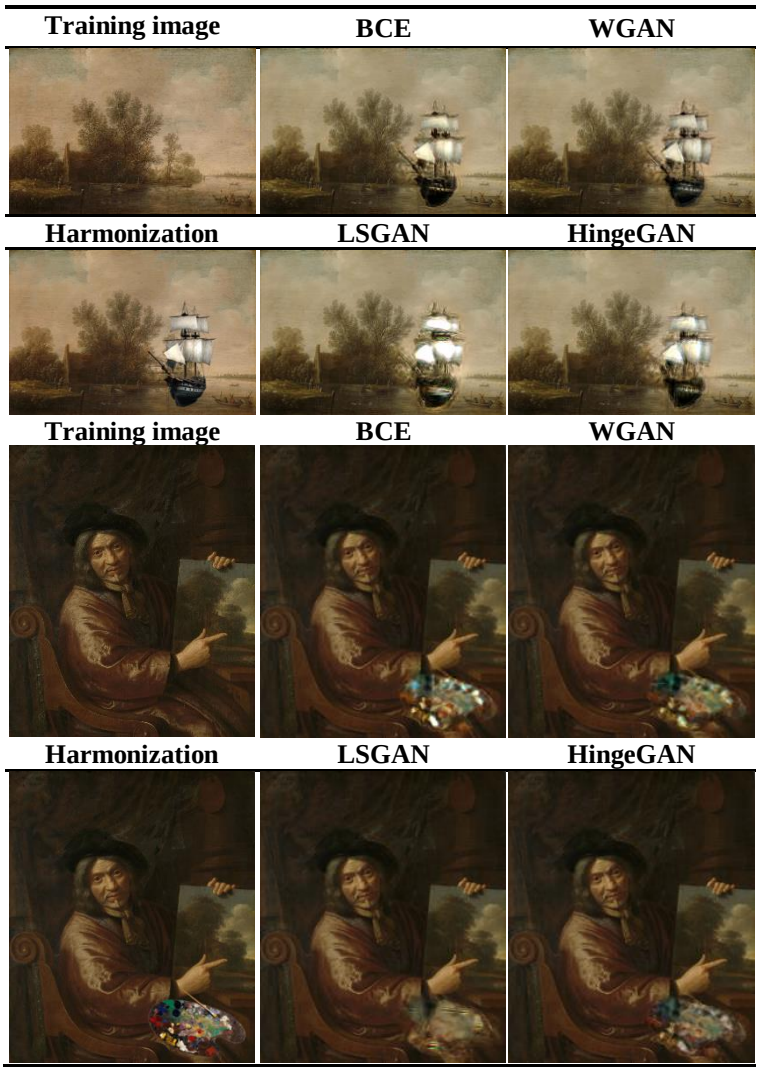

Figure 3: Harmonization results with different loss functions.

Image harmonization involves integrating an external object added to the training image into the image based on the structural characteristics of the training image. For this, we input the harmonized image (with an object added to the training image) to the trained GAN model at different scales during the test phase. The final image is generated by the GAN operating at the last scale. As noted in SinGAN, the scale at which the harmonized image is input affects the realism of the final image [12]. Figure 3 shows the harmonization results obtained from the scale that produces the most realistic output image. According to this, it is observed that Single-image GAN models trained with different loss

functions produce outputs with different characteristics. The visual results support that all loss functions produce successful outputs.

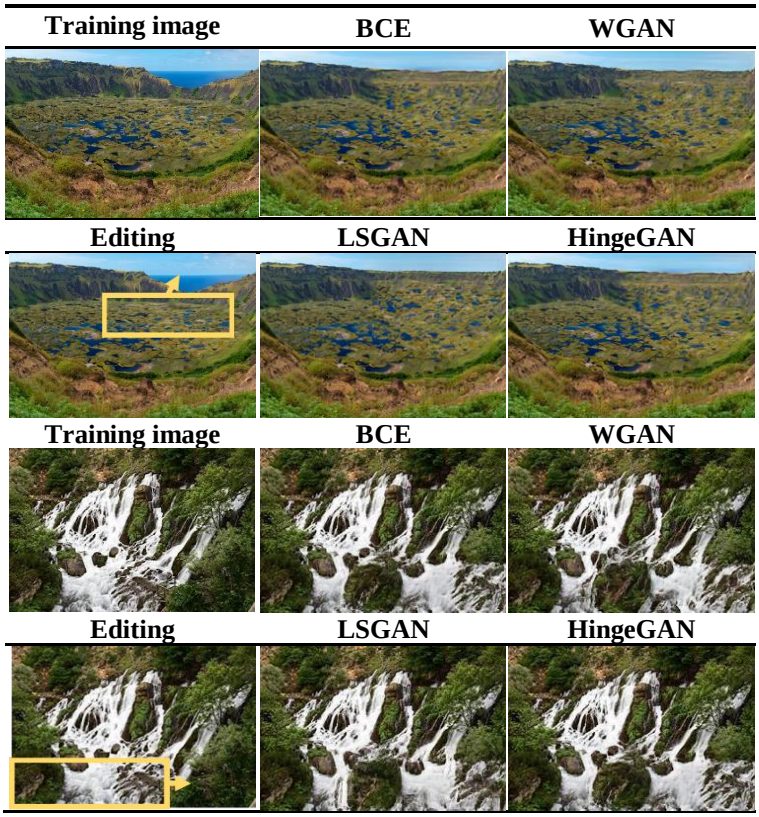

Figure 4: Editing results with different loss functions.

In the image editing application, a selected patch within the training image is placed in a different location within the image. It is expected from the trained GAN model to blend the patch into new location within the training image as seamlessly as possible. Accordingly, we trained a single-image GAN model with four different loss functions on two different images. During the test phase, we followed the same process as in the harmonization application and provided the edited images as input to the GAN models. Figure 4 presents the obtained visual results. The results indicate that the loss functions achieve visual success in this application.

4.3. Limitations

Single-image GAN models are suitable for applications in many fields. They can be used for various purposes in areas where collecting a large number of images is difficult (e.g., healthcare, military). For example, some features in image editing applications (i.e. photoshop) used by humans can also be accomplished by using single-image GAN models. Additionally, in the military, different versions of a rare map environment can be generated to create simulation environments, which is one of the potential application areas of single-image GAN models. However, a critical challenge persists in the domain of single-image GANs: the maintenance of semantic integrity in generated images. Semantic integrity refers to the preservation of meaningful and coherent structures within an image, which is crucial for generating images that are not only visually appealing but also contextually accurate. Despite the successful learning of patch statistics, single-image GANs often struggle to maintain this semantic integrity, leading to the generation of images that may appear visually plausible but lack coherent and meaningful content. User tests conducted in studies [11,12,16] support this observation. Therefore, there is a need to improve single-image GAN models to enhance the realism of the

generated images. This limitation restricts the use of single-image GAN models in sensitive applications. Despite all this, the ability to generate new images from a single training image for different purposes in various fields highlights the high potential of this research area.

5. Conclusion

The effectiveness of GAN models on large datasets is well-established. However, developing successful GAN models for limited data has been challenging until the advent of SinGAN and its derivatives. Most existing GAN models have focused on improving performance through modifications to network architectures and training procedures. This study investigates the impact of loss functions on single-image GAN models rather than focusing on changes to network architectures or training methods. Understanding the pros and cons of different loss functions is crucial for optimizing GAN architectures. To this end, we employed four different loss functions during the training of a single-image GAN model. Our quantitative results showed that WGAN-GP and LSGAN-GP are particularly suitable for single-image GANs. Thus, this study is significant in evaluating the influence of loss functions on single-image GANs. We expect our findings to guide the selection of the most effective loss functions for future single-image GAN models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Shahriar, S. (2022). GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network. *Displays*, 73, 102237.

[2] Chakraborty, T., KS, U. R., Naik, S. M., Panja, M., & Manvitha, B. (2024). Ten years of generative adversarial nets (GANs): a survey of the state-of-the-art. *Machine Learning: Science and Technology*, 5(1), 011001.

[3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville A. & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.

[4] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.

[5] Zhang, Z., Li, M., & Yu, J. (2018). On the convergence and mode collapse of GAN. *SIGGRAPH Asia 2018 Technical Briefs*, 1-4.

[6] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

[7] Iglesias, G., Talavera, E. & Díaz-Álvarez, A. (2023). A survey on GANs for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48, 100553.

[8] Xia, W., Zhang, Y., Yang, Y., Xue, J. H., Zhou, B., & Yang, M. H. (2022). Gan inversion: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(3), 3121-3138.

[9] Wang, P., Li, Y., Singh, K. K., Lu, J. & Vasconcelos, N. (2021). Imagine: Image synthesis by image-guided model inversion. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3681-3690.

[10] Yildiz, E., Yuksel, M. E., & Sevgen, S. (2024). A Single-Image GAN Model Using Self-Attention Mechanism and DenseNets. Neurocomputing, 596, 127873.

[11] Zhang, Z., Han, C. & Guo, T. (2021). Exsingan: Learning an explainable generative model from a single image. *32nd British Machine Vision Conference*.

[12] Shaham, T. R., Dekel, T. & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. *IEEE/CVF international conference on computer vision*, 4570-4580.

[13] Ulyanov, D., Vedaldi, A. ve Lempitsky, V. (2018). Deep image prior. *IEEE conference on computer vision and pattern recognition*, 9446-9454.

[14] Shocher, A., Bagon, S., Isola, P., & Irani, M. (2019). Ingan: Capturing and retargeting the" dna" of a natural image. *IEEE/CVF international conference on computer vision*, 4492-4501.

[15] Isola, P., Zhu, J. Y., Zhou, T. & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *IEEE conference on computer vision and pattern recognition*, 1125-1134.

[16] Hinz, T., Fisher, M., Wang, O. & Wermter, S. (2021). Improved techniques for training single-image gans. *IEEE/CVF Winter Conference on Applications of Computer Vision*, 1300-1309.

[17] Karras, T., Aila, T., Laine, S. & Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations*.

[18] Granot, N., Feinstein, B., Shocher, A., Bagon, S. ve Irani, M. (2022). Drop the gan: In defense of patches nearest neighbors as single image generative models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13460-13469.

[19] Arjovsky M., Chintala S. & Bottou L. (2017). Wasserstein generative adversarial networks. *34th International Conference on Machine Learning, ICML*, 298–321.

[20] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

[21] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. *IEEE international conference on computer vision*, 2794-2802.

[22] Lim, J. H., & Ye, J. C. (2017). Geometric gan. *arXiv preprint arXiv:1705.02894*.

[23] Iglesias, G., Talavera, E. & Díaz-Álvarez, A. (2023). A survey on GANs for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48, 100553.

[24] Jabbar, A., Li, X., & Omar, B. (2021). A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8), 1-49.

[25] Wang, Z., Simoncelli, E. P. & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers IEEE*, 1398-1402.

[26] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *IEEE conference on computer vision and pattern recognition*, 586-595.

Appendix

Training images and generated images to evaluate the performance of loss functions