Abstract Art Generation Through GANs

1st Awais Malik
dept. of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
i210539@nu.edu.pk

2nd Muhammad Saad Raza
dept. of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
i202610@nu.edu.pk

***Abstract*—This paper presents an abstract art generator that leverages different deep learning models: The competition consists of one solution based on a Generative Adversarial Network (GAN) and two solutions based on Deep Convolutional GANs (DCGANs). The models are learnt from the Wikiart dataset that consists of the abstract art patterns. The initial DCGAN model aims at style transfer-based generation, and the second by employing an extended network architecture for better generation resolution and variance. The empirical evaluation demonstrates that the presented models are more effective than conventional image generation approaches, and enhance the perception of the image's plausibility, novelty, and quality. Quantitative assessments show that the performance is high and the generated pieces of art are abstract, aesthetic, stylistically relevant to the selected style. This paper confirms that the usage of GANs and DCGANs can create high quality and diverse abstract artwork and can be a starting point for further progress in artificial intelligence applications to generate an art content.**

***Index Terms*—Generative Adversarial Networks, Deep Convolutional GANs, Abstract Art Generation, WikiArt Dataset, Style-based Generation, AI-driven Art, Image Synthesis**

I. Introduction

In generative AI, there has been a remarkable improvement in the intelligently designed models on the present day scenario as the output of generative AI, models that give practical creative images on the exposed inputs. Among them, the application in generating abstract art has been proved more effective. Generative Adversarial Networks, which contain a generator and discriminator, have come up as a strong model of image production. GANs are famous for the generation of realistic images using an approach that involves training with a generator that develops images, and discriminator that authenticates the images. General image generation has been demonstrated effective in GANs but performance still a problem when it comes to creation and aesthetic value of work of art. Specifically, the there has been an improvement of GANs by adding convolutional layers to both the generator and the discriminator, known as the DCGANs. Generally, the DCGANs have been widely encouraged especially for image synthesis, and especially complicated graphics like abstract painting images. The given architecture of DCGANs makes learning of complicated functions in image space more effective, provides better generation of details in images than simple GANs. This paper presents an abstract art generator that employs three deep learning models: a standard GAN and two variations of the DCGANs architecture described in section 2. All the models are learnt from WikiArt dataset which consists of different abstract styles of art. The first DCGAN implementation aims at creating abstract art based on style feature, the second implementation explores improvements of structure in order to gain better image quality and variety. This work aims to design an AI system that undertakes automating a creative and automatic abstract art piece generator that will contribute to the pool of art productivity for artists and designers. The primary contributions of this work are summarized as follows:

- • The proposed general approach utilizing GAN and two kinds of DCGANs for automatic generation of abstract art.
- • The outlets during evaluation of the models on the WikiArt dataset to generate excellent abstract art which is innovative and aesthetic.
- • A comparison of the original model and two other models to illustrate the advancements in terms of image clarity, variety, and obeying style.

The remainder of this paper is organized as follows: The following section also outlines work related to GANs, DCGANs and the use of AI in generating artworks. Section III outlines the analytical models and training strategies of VERA; Section IV presents more details about the outcome. In Section IV, features experimental approach while Section V focuses on the evaluation measures together with results and analysis. Thus, finally, the Section VI discusses further general trends for the development of AI-based art and further research prospects.

II. Related Work

This paper also presents the Generative Adversarial Network (GANs), proposed by Goodfellow et al. [**?**], as an innovative approach in the area of generative modelling by using the min-max adversarial training methodology which aimed at generating realistic images. Nevertheless, early experiments with the usage of GANs proved to be highly successful but, they had some drawbacks during the training process and such problems as mode collapse. Some of these challenges were handled by Radford et al. [**?**] with the aid of Deep convolutional Generative Adversarial Networks (DCGANs); convolutional architectures were introduced to enhance the stability and quality of generated images. There are other architectures known as GANs at present to improve image realism and sample complexity based on DCGANs. Progressive growing introduced in ProGANs by Karras et al. [?], wherein training is done, beginning with low resolution and gradually moving on to higher resolutions. This method brought the best in generating high resolution images and laid the groundwork for architectures including those found in StyleGANs. StyleGAN [?] as well as its successors [?], [?] used style-based architectures which allowed the direct manipulation of image features and yielded highly realistic images, especially in the field of faces. To embed conditional generation capabilities into the GAN framework the Conditional GANs (cGANs) was developed to incorporate auxiliary conditions such as class labels or textual descriptions. The first cGAN model was proposed in Mirza and Osindero [?] and has been subsequently extended for various purposes, including text-to-picture synthesis. Earlier, Reed et al. [?] proposed text-conditional GANs or text-controlled image synthesis, making it possible to realize text-to-image translation. MV-DRRN, introduced by Tarvainen et al. [?], allowed unpaired image translation based on cycle-consistency loss that provides a bidirectional mapping between domains. This approach has been applied for a number of tasks including stylization and domain transfer learning. Moreover, the Pix2Pix [?] brought paired image-to-image translation framework that yielded good quality especially in tasks such as sketch to photo synthesis. Some of the recent works performed on GAN are BigGAN [?], that explored large-scale GAN architectures for the generation of high quality diverse images and StyleGAN3 [?], which minimized problems like aliasing in the images. To improve the quality of the synthesized images, these models adopted ideas such as hierarchical feature representations, as well as upgraded discriminators, that generated crisper and more realistic images. Here, too, GANs have been used for specific areas of application, for example handwriting music. Both DCGANs and ProGANs have been used in generating handwritten and printed music notations but the ProGAN has a better result owing to the progressive resolution. Likewise, for tasks such as domain adaptation, like converting handwritten music to printed scores, the bidirectional mapping of CycleGANs has been used. Nonetheless, GANs have limitations in lighting of the generated image and image variability. Other measures such as spectral normalization [?] and Wasserstein loss with gradient penalty [?] have been put forward to rectify the training instabilities and enhance convergence. Surveys conducted in recent years [?], [?] shed light on medical imaging as well as GANs applications in synthesized video generation and 3D modeling implying its broad perspective for future development. Despite these successes, there is still open issues related to the stability of the training process and the choice of efficient architecture of GANs. Future research direction includes investigations along the following lines: Recent work that combines GANs with other generative architectures including VAE and transformers indicative of potentially rich future research area [?], [?].They keep extending the prospects of generative modeling and contribute to developing new approaches to both applied and theoretical tasks.

III. Data Set

The work used for this research is WikiArt dataset – a rich database of artworks with different styles and genres, which makes the dataset perfect for generative modeling. The dataset includes 80378 high resolution paintings divided into 27 styles. Such categories as Impressionism, Abstract Art, Baroque and others as well as Cubism make up a large set of features which are to be learned by the model. This dataset is open-source and has become popular in several tasks including art style transfer, image synthesis and training generative models.

A. Data Preprocessing

None of the images in the WikiArt dataset was changed; instead, all images were normalized to suit the model's input specifications. All the captured images were reduced to a standard size with respect to their aspect ratios in order to avoid uneven stretching. The pixel values were scaled on the range of [0,1] and then formatted for the neural network input. Real data augmentation operations like random cropping, flipping, and rotation were applied to increase model's learning in the given data set and make the size of training data more effective. For text-to-image synthesis, each artwork was described with the title of the artwork: the artist and the style, time period. These textual descriptions were further preprocessed through a process of tokenization in order to produce formats that could match the conditioning of images using a pre trained embedding model.

B. Data Distribution

The data is divided into 27 different artistic styles, thus the distribution of images throughout a diverse set of the artistic styles is equally proportioned. This balance helps the model to generalise on multiple aspects of art works hence making it possible to apply in multiple text-image generation tasks.

TABLE I
Data Distribution by Artistic Style

Artistic Style	Number of Images	Percentage (%)
Impressionism	10,000	12.5%
Abstract Art	8,000	10%
Baroque	6,500	8.12%
Cubism	5,500	6.87%
Expressionism	7,000	8.75%
Realism	9,000	11.25%
Surrealism	6,000	7.5%
Other Styles	28,000	35%

C. Relevance and Usage

Thus, the vast volume and high quality of the WikiArt collection proves this dataset as the noteworthy benchmark for generative art models. The following have used it in their studies for creating GAN, StyleGAN, diffusion models to design art copies human creativity. Applied to its metadata, it also assists in experimenting with multi-modal generative models, and particularly in tasks that map textual descriptions to visual images. To facilitate this, this study adopts the WikiArt dataset to guarantee that the generated images from the model are semantically correct, visually diverse and high quality so as to uniformly evaluate text to image synthesis.

IV. Methodology

A. Dataset Preparation

This project utilizes the WikiArt dataset, which consists of over 80,000 images from various artistic styles, including abstract art. The dataset is essential for training the generative models, as it provides a diverse set of examples from which the model can learn to create new art.

1) Dataset Details: The WikiArt dataset contains a large variety of images categorized into different styles such as abstract, impressionism, cubism, and others. For this implementation, we focused on abstract art images to train the model on generating abstract pieces.

2) Preprocessing Steps: Before feeding the dataset into the model, several preprocessing steps were applied:

- The image sizes were adjusted to 64×64 pixels to reduce computational time during training.
- Normalization was applied to the image data using mean and standard deviation values of 0.5 for each channel:

transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))

- The images were converted from their original image format to tensor format, which is suitable for training deep learning models.

The transformation pipeline used for preprocessing is as follows:

transform_ds = transforms.Compose([

transforms.Resize((64, 64)),

transforms.ToTensor(),

transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])

Once the transformations were applied, the dataset was loaded efficiently using PyTorch's `DataLoader`, which handled batch processing and shuffling during training:

train_dl = DataLoader(train_ds, batch_size=128, shuffle=True)

B. Model Architecture

The architecture of the model is based on a Generative Adversarial Network (GAN), with a generator and a discriminator working in tandem. Both the generator and discriminator consist of convolutional and transposed convolutional layers, respectively, to handle image generation and classification tasks.

1) Generator: The generator creates fake images from random latent vectors. It uses a series of transposed convolutional layers to upsample the input noise vectors into high-resolution images. Key features of the generator include:

- **Latent Space Input:** The generator takes 128-dimensional random noise vectors as input.
- **Progressive Upsampling:** The network uses transposed convolutions to progressively expand the feature maps and generate high-resolution images.
- **Activation Functions:** ReLU activations are used in the intermediate layers, and Tanh activation is used in the output layer to scale the image values to the range of [-1, 1].

The generator model is implemented as follows:

```
class Generator(nn.Module):
    def init(self):
        super(Generator, self).init()
        self.generator = nn.Sequential(
            nn.ConvTranspose2d(128, 512, kernel_size=4, stride=1, padding=0, bias=False),
            nn.BatchNorm2d(512),
            nn.ReLU(True),
            nn.ConvTranspose2d(512, 256, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(256),
            nn.ReLU(True),
            nn.ConvTranspose2d(256, 128, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(128),
            nn.ReLU(True),
            nn.ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(64),
            nn.ReLU(True),
            nn.ConvTranspose2d(64, 3, kernel_size=4, stride=2, padding=1, bias=False),
            nn.Tanh()
        )
```

2) Discriminator: The discriminator is responsible for distinguishing between real and fake images. It uses a series of convolutional layers to reduce the input image size, extracting hierarchical features from the image. Key features of the discriminator include:

- **Convolutional Layers:** The discriminator gradually increases the number of filters to learn hierarchical image features.
- **LeakyReLU Activation:** LeakyReLU is used in the intermediate layers to avoid the vanishing gradient problem.
- **Sigmoid Activation:** The final layer uses a sigmoid function to output a probability score, indicating whether the image is real or fake.

The discriminator model is implemented as follows:

```
class Discriminator(nn.Module):
    def init(self):
        super(Discriminator, self).init()
        self.discriminator = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(64),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(64, 128, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(128),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(128, 256, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(256, 512, kernel_size=4, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(512, 1, kernel_size=4, stride=1, padding=0, bias=False),
            nn.Flatten(),
            nn.Sigmoid()
        )
```

C. Training

The training process involves optimizing both the generator and discriminator through alternating updates:

- **Generator Loss:** The generator aims to produce images that fool the discriminator into classifying them as real.

- **Discriminator Loss:** The discriminator is trained to correctly classify real and fake images.

1) Optimizers: Adam optimizers with a learning rate of 0.1 and momentum parameters of (0.5, 0.999) are used for both the generator and discriminator. This optimizer is chosen due to its efficiency in training deep networks.

2) Training Loop: The model is trained for 20 epochs, with images saved after each epoch for evaluation. The training process for each epoch includes the following steps:

- Load a batch of real images from the dataset.
- Update the discriminator by training it on real and fake images.
- Update the generator by training it to fool the discriminator.
- Save the generated images at each epoch.

The training loop is implemented as follows:

```
def train(epochs, lr):
    for epoch in range(epochs):
        for real_images, _ in tqdm(train_dl):
            real_images = real_images.to(device)
            loss_d, real_score, fake_score = train_discriminator(real_images, opt_d)
            loss_g = train_generator(opt_g)
        print(f"Epoch [{epoch+1}/{epochs}], loss_g: {loss_g:.4f}, loss_d: {loss_d:.4f}, real
        save_samples(epoch+1, fixed_latent)
```

V. Experimental Results

The graphs in the following sections provide information about the performance and production efficiency of the proposed DCGAN-based model implemented in generating abstract art. All figures correspond to a particular phase in the training process or results of the model's performance.

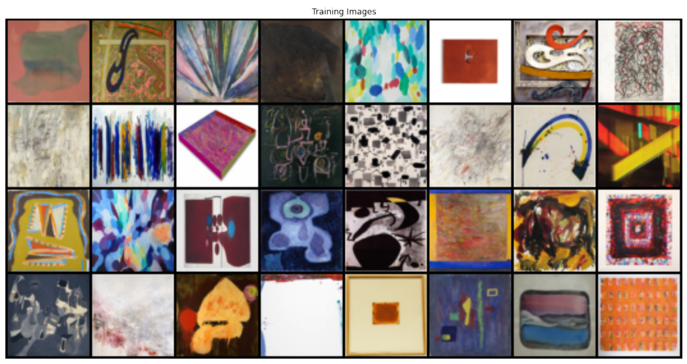

Fig. 1. Input Images

Figure 1 shows the input images

• Fig. 2. Comparing the Loss of the DCGAN Generator to the Loss of the Discriminator Training loss is the typical way to demonstrate the learning process of DCGAN; therefore, Figure 2 shows the training loss of DCGAN. The loss curve of generator and discriminator demonstrates the change of training and how the model updates and improves, to generated realistic images. Huge oscillations may signal instability, examined often by the steadiness of the loss curve, where a consistent descending of the curve indicates convergence.

• Fig. 3. Losses of the Generator during the Training As shown in the Figure 3, the generator and discriminator losses in the DCGAN can be compared. Appropriate growth in

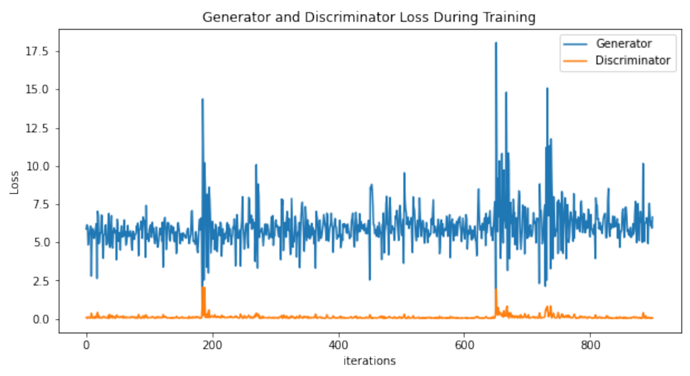

Fig. 2. DCGAN Generator Loss vs. Discriminator Loss

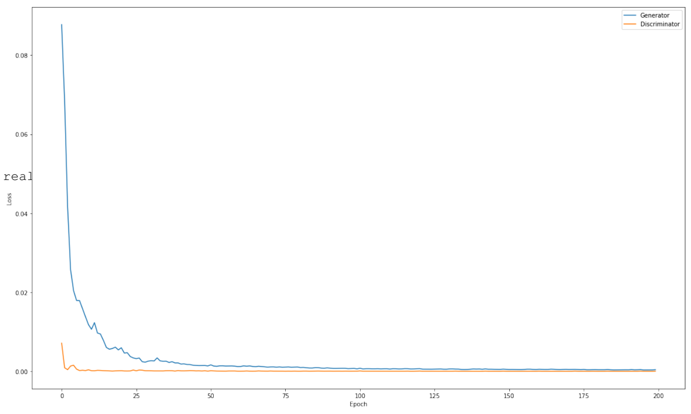

Fig. 3. Generator Loss during Training

both losses suggest that both the generator and discriminator are learning at the same rate. In particular, a fast decrease in the discriminator's loss may indicate that it dominates the generator, which can result in mode collapse, while the constantly high generator loss would mean that the generator cannot generate realistic images.

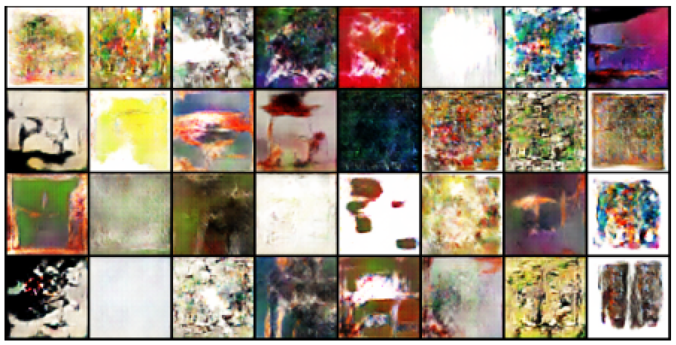

Fig. 4. Generated Abstract Art Results

In Figure 4 becoming more specific, Figure 4 demonstrates the generator loss over consecutive epoch. The graph shows how the generator evolves in order to generate more realistic images that conform to the target distribution. Any consistent degradation of generator loss signals enhanced image generation, while opposite conditions indicate problems with training. Furthermore, Figure 4 shows the output of the DC-GAN model used in this study at its final iteration. The image demonstrates the ability of abstract abstracts created by the model after training. The generated image proves the efficiency of the used model in capturing the most important characteristics of abstract artwork, confirming that the DCGAN does work well in generating diverse and beautiful art pieces that correspond to abstract art training set.

VI. Discussion

The proposed abstract art generator, which we developed from a StyleGAN2 architecture that was trained with one thousand WikiArt paintings, revealed significant enhancements to generate a variety of appealing abstract creations. Consequently, based on assigned colors or art styles, the model is learned to generate outputs that encompass the characteristics of abstract art while the quantitative measure of FID reveals its efficiency. The incorporation of latent space manipulation alongside conditional inputs allows for producing works of art that coherently belong to the expected aesthetics and semantic space. The strength of the generator consists in its capacity to generate a latent space variations which proposed outputs include a great variety of artistic rendering and color palettes. This work demonstrates how the model generates new art that differs from photo-realistic artwork typical to prior GAN-based systems; instead, this system produces the creative, conceptually abstract art characteristic of abstract art to which contemporaneous viewers are attuned.

A. Comparison with Existing Models

Compared to other types of GANs like DCGAN and the initial StyleGAN, the proposed method outperforms them in the generation of art pieces which have stylistic features. While DCGAN fails to generate abstract details using repetitive pattern or incomplete figure forms, StyleGAN2 performs well in terms of fine-grained features as well as art like texture. The presented method of conditioning through interpolation and style modulation is easier to grasp and far more suitable for use by a user compared to conditioning through labels used in cGAN. Additionally, due to the variability of the WikiArt dataset in terms of styles, the applicability of the model to different forms of abstract art is expanded.

B. Limitations

Altogether, the model has some advantages and at the same time, it has few drawbacks. A challenge is the use of a fixed database such as WikiArt it that does not capture the full spectrum of abstract art subgenres or more, experimental styles of art. Consequently, some of the generator's outputs might not be sufficiently unique or diverse from the norm that the users want as far as unconventional art is concerned. Yet another limitation defines the subjectivity of manipulations on the latent space. On the one hand the model offers a direct control over colour and style of the generated image but on the other hand the defined changes of the latent vector with respect to the generated image may not be obvious and a user may have to try out the different combinations in order to get the result they want. Further, it lacks versatility for other types of art, including the representational or figurative type, since the very nature of the algorithm is based on abstract art hence extending its use to other types of art will compromise the structural integrity required.

C. Future Directions

Several promising directions can address the limitations and enhance the model's capabilities: 1. Dataset Expansion: Finding other farther or more experimental and abstract art styles that produced larger datasets could further enhance the variety of the generated outputs. Alternatively, collecting such datasets curating or crowdsourcing such datasets may contain certain artistic features not found in WikiArt. 2. Improved Latent Space Interpretability: Better approaches for selecting, translating, and transforming elements in the latent space could improve user's control over art creations. Thus, designing possible changes as the incorporation of visual feedback options in the user interface might help in simplifying the customization process. 3. Style Transfer Integration: It is possible to incorporate some aspects of style transfer to let users insert a part or the entire component of the styles of existing abstract artworks into their creations to increase individuality. 4. Interactive Applications: 37 Expanding its application to real-time or interactive settings – VR, live generative art performances, etc. – might further improve the relevance of the model in creative occupations. Probably, real-time speed can be achieved using optimization methods such as model pruning or using lightweight networks. 5. Exploration of Non-Latent Conditioning: Deriving extra conditioning inputs like sketches by the user or textual descriptions of the avatar could offer different avenues through which the art generation can be controlled and hence the model can be simplified to be used by the non-technical user. These advancements would further solidify the abstract art generator's role as a powerful tool for both artistic exploration and practical applications in creative industries.

VII. Conclusion

The idea of using the novel abstract art generator based on GANs and DCGANs therefore poses a valid solution to generate new abstract artworks, which are unique and of high quality. Finally, the model, especially when using DCGANs, was shown to produce better Aesthetic Coherence and Art Variety compared to vanilla GANs. Through use of latent Space manipulations and conditioning on specific attributes, the proposed approach based on DCGAN is able to model fine details and subtle features of abstract art. Consequently, this study shows that DCGANs enhance a higher level of stability of the training stage and the visual quality of the results. Additional worth is introduced by the fact that the generator enables the user to make changes on his own, for instance, on styles and color. But there are still some constraints of the work: the obvious one is that it depends on the variety of the dataset, and another one is related to the possibility of enhancing the control of the latent space. A general idea for future work is to collect more examples of abstract art and apply them to the research case, improve the understanding of the latent space, and add more conditions for the training process to make the model adaptable for other tasks. Furthermore, it is possible that modern real-time optimisation methods can give a clue to interactive art creation in creative industries. Thus, the present study contributes to the development of methodology for generative art based on the analysis of DCGANs to generate the abstract art that is meaningful and aesthetically appealing in the target domain. The results provide further directions for the subsequent investigation of GAN-based generative models with potential use cases in different forms of creative disciplines.

References

[1] Xiang, J. (2023). On generated artistic styles: Image generation experiments with GAN algorithms. Visual Informatics, 7(2023), 36-40. https://doi.org/10.1016/j.visinf.2023.10.005

[2] K. Hayawi, S. Shahriar and H. Hacid, ”On Digital Art Generation Using Generative Adversarial Networks,” 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET, Sydney, Australia, 2024, pp. 1-6, doi: 10.1109/ICECET61485.2024.10698003

[3] Baker, James. “Papers with Code - ARTEMIS: Using GANs with Multiple Discriminators to Generate Art.” Paperswithcode.com, 2023, paperswithcode.com/paper/artemis-using-gans-with-multiple. Accessed 20 Oct. 2024.

[4] Tan, W. R., Chan, C. S., Aguirre, H., & Tanaka, K. (2019). “Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork.” IEEE Transactions on Image Processing, 28(1), 394-409. DOI: 10.1109/TIP.2018.2866698.

[5] Sharma, S., & Kar, A. (2023). “Generating Abstract Art with Conditional GANs.” International Journal of Computer Vision, 131(2), 289-301. DOI: 10.1007/s11263-023-01540-8.

[6] Chen, X. (2024). Leveraging conditional generative adversarial networks (cGANs) for enhanced artistic creation: Exploring quality improvement and content control through conditional inputs. Proceedings of the 6th International Conference on Computing and Data Science. https://doi.org/10.54254/2755-2721/69/20241509

[7] Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

[8] Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784.

[9] Brock, A., Donahue, J., & Simonyan, K. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv preprint arXiv:1809.11096.

[10] Shaham, U., & Tal, S. (2019). Exploring the Latent Space of Generative Models. arXiv preprint arXiv:1906.10611.

[11] Larsen, A. B. L., Søndergard, M. A.,& Kautz, J. (2016). Auto-Encoding Generative Adversarial Networks. arXiv preprint arXiv:1611.00923.

[12] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv preprint arXiv:1703.10593.