RangoliGAN: Generating Realistic Rangoli Images using GAN

Sainath S[1], Sridhar S[2]

[1]Undergraduate Student, Department of Computer Science, SRM Institute of Science and Technology, India
[2]Assistant Professor, Department of Computer Science, SRM Institute of Science and Technology, India

---***---

Abstract - *The success of generating photorealistic images lately has been widely attributed to the invention of GAN (Generative Adversarial Networks) in 2014 and since then there have been multiple interesting applications of GANs. This paper shows that DCGANs can be used for synthesizing photo-realistic rangoli Images with considerable diversity. The specific domain application of generating rangoli Images poses a considerable challenge since rangoli Images are mostly symmetric in nature with a varied range of textures and hue involved. Rangolis are no less than a mathematical art. The RangoliGAN introduced in this paper is aimed towards understanding and applying the process of image generation on rangoli images. Many recent advances in GAN training have been taken into consideration in the experimentation process. The experiments show that this method can effectively generate realistic rangoli images. In simple terms one can say that the Generator is The Artist and Discriminator is The Art-Critic.This experiment can be considered as a base for exploring more on Creative AI - to demonstrate the creative capabilities of machine learning models.*

Key Words: **convolutional generative adversarial networks, Rangoli, image synthesis, neural networks, creative AI, GAN**

1. INTRODUCTION

Rangoli Images are fascinating and an integral part of Indian culture since early ages. Over the generations it has gone through many localization and adaptations. Their symmetric nature is well known and they appear to be a combination of several simpler designs superimposed over each other. This symmetricity in rangoli can be viewed from an angle of Cymatics - the study between vibrations and geometric patterns concerning them.

It is well known that the brain actually responds to visual patterns and depending on the shapes & patterns, it can have different effects on the mind, and this paper tries to understand and further build on how creativity is perceived by a machine. It is the GANs (Generative adversarial Networks) which can be used to generate realistic rangoli images.

GAN (Generative Adversarial Networks) [1] was first introduced by Goodfellow et al. drawing inspiration from generative models. Yann LeCun called GANs the most interesting idea of the decade.

Goodfellow et al. in their 2014 paper [1] propose an adversarial framework where they simultaneously train two models (Generator G and Discriminator D) without the need of Markov Chain. The training procedure is for G to maximize the mistake made by D. The MNIST, Toronto Face Dataset and CIFAR-10 dataset were used in the training process. This paved way for more adversarial process frameworks in the future. The results from this paper have been improved a lot with the various advancement in training techniques and architectures.

GANs initially evolved from zero-sum game borrowed from Game Theory. Two neural networks compete with each other to generate convincing fakes. They have been applied to generate images, text, sound, art, videos and also Multi-Modal Cross-domain applications.

Though GANs were originally classified under supervised learning, they have been adopted in supervised, semi-supervised and also in reinforcement learning. Putting in layman's terms GAN consist of two-part Discriminator(D) and Generator(G), the G generates images and the job of D is to identify whether the generated image is real or fake. This process of adversarial learning is very important in the GAN training, and through this contest, the GANs are able to generate realistic-looking fake data. The generator learns the input distribution of data with the aid of discriminator and thus over time and iterations it is able to generate images

Again in GAN there are multiple architectures that are used depending upon the application. Since we are going to work with images the DCGAN(Deep Convolutional Generative Adversarial Networks) [2] is of our interest. Although other architectures like WGAN (Wasserstein GAN) [3] does not stop us from generating fake images, the DCGAN architecture introduced by Radford et al. have been proven to give better results when applied to the task of working on images thus becoming the de-facto while working on images. The paper [17] tries to address the challenge of how AI can be used to evaluate creative designs. This paper aims to form a foundation for Creative AI and Explainable AI to understand GANs' creativity and interpretability.

2. RELATED WORK

DCGANs [2] have seen many interesting extensions on categorical images like in art for generating anime characters [4], in medical imaging – MelanoGANs: High Resolution Skin Lesion Synthesis with GANs [5] , a report on bird image generation using DCGAN [6], even augmentation of breast mass in X-ray Mammography [7] have all used DCGAN as their underlying structure. It might be also interesting to note that DCGANs are not restricted to only image generation, there have been many successful attempts to apply DCGAN to other domains like sound domain(speech, bird vocalizations, drums, piano) by introducing WaveGAN [8] an adaptation of the existing DCGAN structure.

Few of the other works by authors taken into consideration while experimentation are [10] , a great paper which lists the latest advancement in GAN with respect to Computer Vision Paper [9] by Creswell et al. reviews various architectures of GANs and techniques in order to get stable training.

The Two Time-Scale Update Rule [11] proposed by Heusel, Martin, et al. which uses individual learning rate both for discriminator and generator hence making it faster to converge to a local Nash equilibrium thus enhancing the learning capabilities of DCGAN.

According to Game Theory, Nash equilibrium is the state solution in which when two or more players are involved although each player knows the strategies of other players, no one gains by bringing a change in their own strategy. Furthermore John Forbes Nash Jr. showed that there is Nash equilibrium for every finite game.

In the context of GANs it can be put as Discriminator and Generator are said to be in Nash equilibrium if Discriminator is making the best decision it can, taking into account Generator's decision while Discriminator's decision remains unchanged, and Generator is making the best decision it can, taking into account the Discriminator's decision while Generator's decision remains unchanged.

3. RANGOLI

Rangoli is an ancient Art form which is also followed today in most India households. The word rangoli means *row of colors*. The paper [15] trace backs the history of the rangoli and its mentions in the Indian text and literature. They are a piece of *Mathematical Art* [16].They come in a variety of symmetry groups, spirals, mirror curves, fractal self-similarity, and processes of iteration. The paper further mentions about these designs being able to transfer as *Kinesthetic intelligence* in children. It might also be interesting how this Kinesthetic is perceived as part of machine intelligence. This paper is more interested in understanding how symmetry is learned. Symmetry can be rotational, reflective, cyclic symmetries. Almost all rangoli exhibit some kind of symmetry.

4. DCGAN

DCGAN (Deep convolutional generative adversarial networks) [2] introduced by Radford et al. was aimed at unsupervised learning. When GANs were initially proposed they did not draw much attention because they were very challenging to train and the initial results were unsatisfactory. Since then researchers have worked on enhancing the overall GAN architecture and bringing in more stability while training a GAN.

DCGAN is inferior to both BigGAN [13] and StyleGAN [14] with former being able to produce superior image quality and latter having styling control over latent space
In designing the DCGAN architecture Radford el al. mainly emphasis the core to their approach being

- Replace Pooling functions with strided Convolutions - thus allowing networks to learn its own spatial up sampling in generator
- Remove Fully Connected Layers after Convolutions - for a middle ground between faster convergence and increased stability
- Batch Normalization - with zero mean and unit variance to make up for poor initialization thus helping gradients flow deeper in the model

4.1 Architecture

The GAN architecture used in this paper has one Generator and one Discriminator, there has been an attempt to use multiple generators by proposing an MGAN (Mixture Generative Adversarial Nets) [12].

As discussed earlier DCGAN in our case consists of a Discriminator and Generator.

The architecture over the years has been carefully evolved to bring it stability and enhanced learning through many techniques like using ReLU activations to avoid sparse gradients and make computation faster, custom weight initialization, Two Time-Scale Update Rule, etc.

4.1.1 Generator

The Artist tries to generate images that look as real as possible. It captures the joint probability p(X, Y), or just p(X) if there are no labels. Generator contains the learned vector.

The learning process happens when Discriminator gives feedback to the Generator on whether the generated data is real or fake from the eyes of the Discriminator. A block in our generator network consists of Transposed Convolutional layers, BatchNorm and ReLU. There are 4 such blocks arranged sequentially in our generator network ending with another layer of Transposed Convolution with a Tanh function at the very end.

The use of Tanh, in the end, is explained in [12] and attributes to the model's ability to learn more quickly to saturate and cover the color space of the training distribution. Also Tanh [-1,1] being symmetric in nature also helps it to better capture images features like treating dark and light colors equally.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

4.1.2 Discriminator

The Art Critic which acts as a binary classifier labelling whether the generated image is Real or Fake. It captures the conditional probability p(Y | X).

A block in our discriminator networks consists of the Convolution layer, BatchNorm, and LeakyReLU. Again there are 4 such blocks in our discriminator network ending with another 2D Convolutional with sigmoid activation function at the very end. The use of LeakyReLU helps in avoiding the "*Dying ReLU*" problem of getting stuck in zero gradient forever. The sigmoid activation function for getting the output as the probability of the image being real or fake.

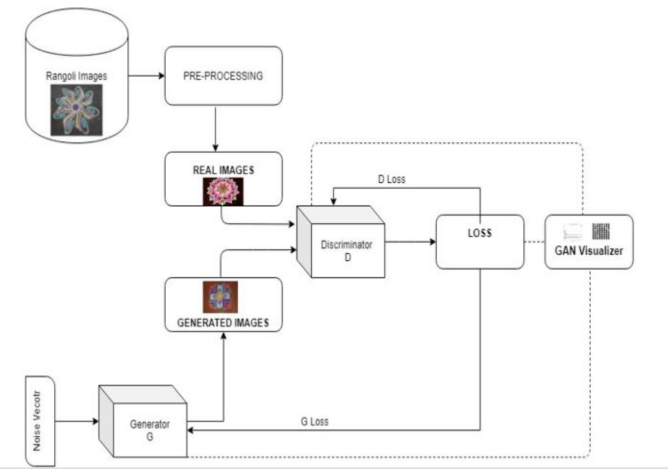

Fig-1: RangoliGAN architecture

4.2 Loss

GANs try to learn from a probability distribution. A distance metric is necessary to understand how far is the generated distribution from the real data distribution.

4.2.1 Minimax Loss

The loss used in our paper is Minimax Loss which is expressed as

$$\min_{G}\max_{D}\mathcal{L}(D,G)=\mathbb{E}_{x\sim P_{data}}[\log D(x)]+\mathbb{E}_{x\sim P_{noise}}[\log(1-D(G(z)))].$$

Where

D(x) - is the probability estimate that X is real, where X is a single data instance.

x~P data - this denotes the value that is expected drawn over all real data points.

G(z) - this denotes the output of the generator for a certain amount of noise denoted by 'z'.

D(G(z)) - denotes the probability estimate by the discriminator for a fake instance to be real.

X~P noise - denotes the expected output value drawn over all inputs of the generator

The generator has no control over the log(D(x)) term , so it indirectly minimizes the loss by minimizing log(1 - D(G(z))) term. In a nutshell, the generator tries to minimize the above equation whereas discriminator tries to maximize it.

4.2.2 Modified Minimax Loss

Studies also found that there is a likelihood that Minimax loss can cause the network to get stuck during its early learning process as little information about gradient is made available to the generator. So making the generator maximize log D(G(z)) would help to overcome this situation when the discriminator is powerful in the initial stages. This trick is sometimes called *log D trick* [1] for training GANs

4.3 Training

We use the Discriminator to train the Generator. The generator loss penalizes the generator for producing a sample that the discriminator network classifies as fake.
It is harder to train generator models because they have to model more thus learning from complicated distribution, whereas discriminator easily tries to classify between real or fake once it learns the pattern.

Initially in the early stages of GAN training, the discriminator are more powerful than the generator and the discriminator is good at rejecting fake images with high confidence, hence the Minimax loss saturates. Usage of *log D* trick helps prevent this

4.4 Challenges

Training a GAN is difficult and results are quiet unpredictable. Most of the GANs impose the following challenges which we might need to consider while training

- Mode Collapse - The generator collapses leading to noise data generation or repeating.
- Non Convergence - Happens because the discriminator feedback becomes less significant and hence random over time leading to unstable parameters
- Vanishing Gradients - Discriminator gets too successful thus generator ends up learning nothing
- Powerful Discriminator in initial stages.
- Highly sensitive to initial hyper parameters

5. DATA COLLECTION

A total of close to 2000 rangoli images were collected and each image was manually examined. Many images had to be rejected because of them being distorted, incomplete, had texts on them or had interfering objects(like hands) in it. After careful inspection the cleaned dataset resulted in a total of 1079 images which is very less when compared to similar projects done which involved huge datasets.
To compensate for this, appropriate image augmentation techniques were used to give an illusion of increased training data diversity.

Fig-2: Sample Data from Dataset

5. EXPERIMENT

Considering the fact that GANs are hard to train and involve a lot of hyperparameter. Many different experiments were conducted tweaking the hyper parameter settings. The most significant ones are considered for this paper.

5.1 Training Details

There are multiple ways of training a GAN. The author of the paper studied them and chose the best one considering the scenario. The below is well studied and widely used in training of GAN.

(1) Update Discriminator (D) network: maximize log(D(x)) + log(1 - D(G(z)))
Train with all-real batch
Forward pass real batch through D
Calculate gradients for D in backward pass
Train with all-fake batch
Generate batch of latent vectors
Calculate D's loss on the all-fake batch
Calculate the gradients for Fake batch
Add the gradients from the all-real and all-fake batches
Update D

(2) Update Generator (G) network: maximize log(D(G(z)))
Calculate G's loss on output based on forward pass of all-fake batch through D
Calculate gradients for G
Update G

Fig-3: Training Steps Followed

5.2 Model Architecture

The model architecture for Discriminator and Generator was identical to the one described in the Architecture subsection. *See Section 4.1 for more details.*

5.3 Hyperparameter

As discussed before, GAN are highly sensitive to hyper parameter selection. Taking into consideration our dataset size and available compute power (Tesla K80, P4 and T4) a range of hyperparameters were adjusted to give varied results. The main hyper parameters were epochs, batch size, optimizer settings, latent vector size and weight initialization.

The number of epochs chosen were 100, 200, 300 and 500. Batch size of 32, and 64 were used depending upon RAM. Optimizer used was Adam (learning rate = 0.0002, 0.0005 ,0.002, beta1 = 0.5).

5.4 Two Time-Scale Update Rule (TTUR)

This involves having different learning rates for Generator and Discriminator.

The use of TTUR was run on 100, 200 and 500 epochs and D_lr and G_lr of 0.0002 and 0.002 respectively were chosen. The experiment was rerun again on 100 epoch with D_lr and G_lr swapped to infer the results when the learning rate of Discriminator and Generator are switched.

Table -1: Conducted Experiments Details

S No	Epochs	TTUR Used	Learning Rate
1	100	No	D_lr = G_lr = 0.0002
2	100	No	D_lr = G_lr = 0.0005
2	200	No	D_lr = G_lr = 0.0002
3	300	No	D_lr = G_lr = 0.0002
4	500	No	D_lr = G_lr = 0.0002
5	100	Yes	D_lr = 0.0002, G_lr = 0.002
6	100	Yes	D_lr = 0.002, G_lr = 0.0002
7	200	Yes	D_lr = 0.0002, G_lr = 0.002
8	500	Yes	D_lr = 0.0002, G_lr = 0.002

6. RESULTS

The results were obtained from running many experiments under multiple configuration. Loss plots of different runs expose interesting insights into our experimentation process.

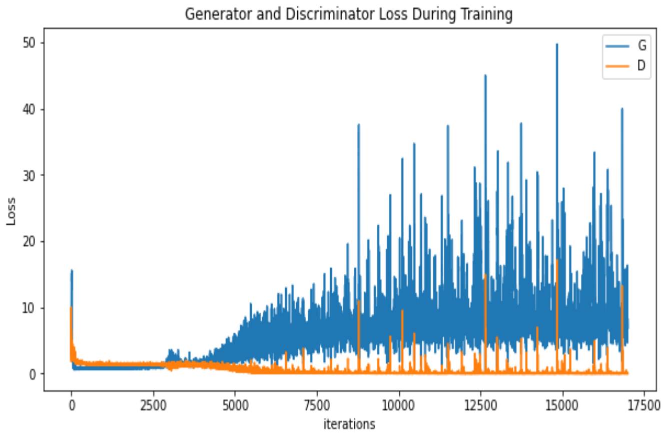

Fig 4: Running for 500 epochs TTUR *with D_lr = 0.0002 and G_lr = 0.002*

From Fig 4 and Fig 5 we see contrasting results which are quite interesting. The D_loss and G_loss seems to have converged at around 3500 iterations in the Fig 4 which uses TTUR but on running for the same number of epochs without the use of TTUR the losses seems to not converge. There seems to be a point of intersection of G_loss and D_loss in Fig 4 after which the curves branch out in opposite direction.

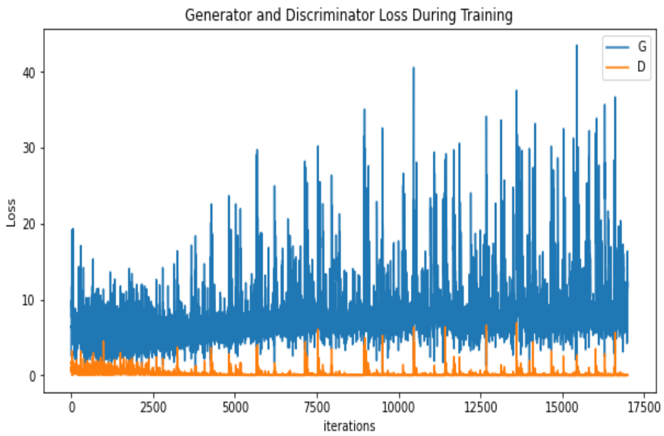

Fig 5: Running for 500 epochs without TTUR

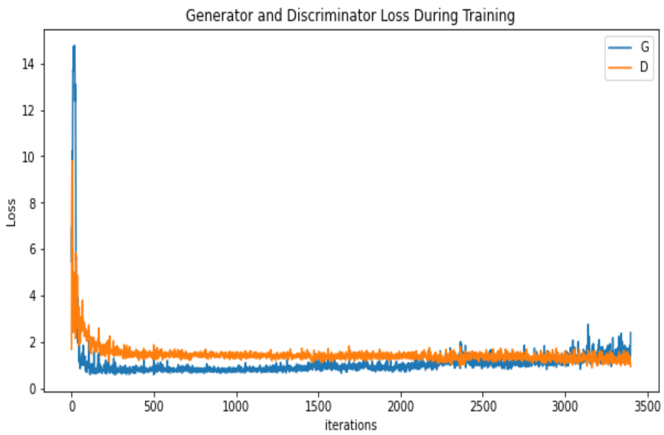

Fig 6: Running for 100 epochs TTUR with D_lr = 0.0002 and G_lr = 0.002

The plot obtained in Fig 6 and Fig 7 are run on 100 epochs each but the values of learning rate of D and G were swapped between the two trials. Again these results are quite surprising. On an average the G_loss in Fig 7 is decreasing. There is a point of convergence in Fig 6 unlike in Fig 7.

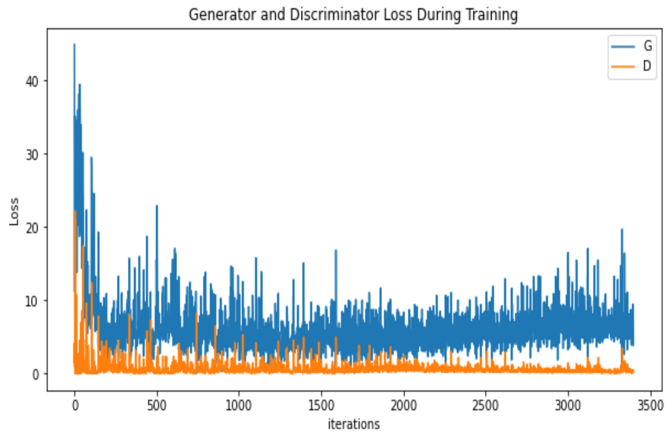

Fig 7: Running for 100 epochs TTUR with D_lr = 0.002 and G_lr = 0.0002

The generated images are shown in the Fig 8 and Fig 9.

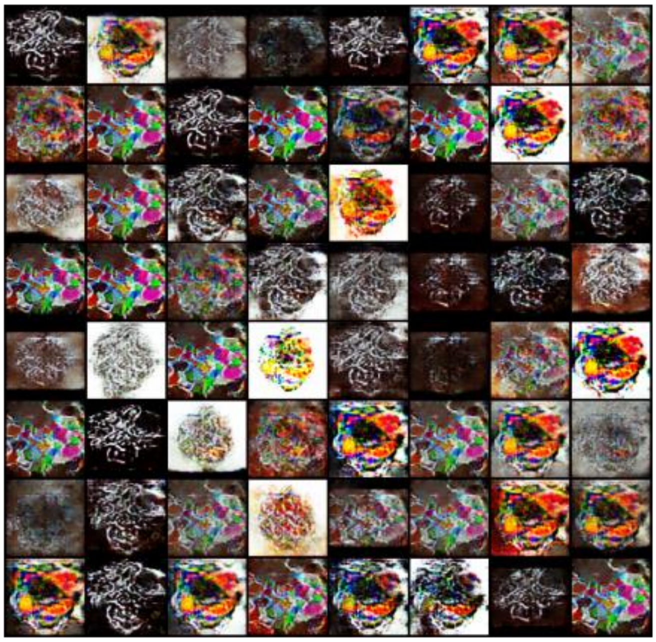

Fig 8: Fake Image after 500 epochs

Fig 8 we see that images are repeated which indicate maybe there is a possibility of Mode Collapse happening where the generator would have collapsed leading to noise or repeated set of image generation happening

In Fig 9, considerable fake Rangoli images have been generated which appear to be symmetry in nature although we have not enforced symmetry as part of our modelling.

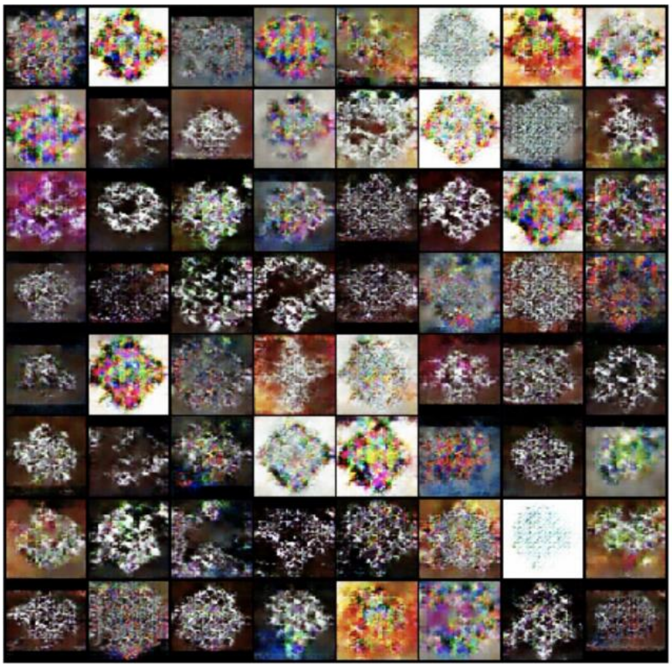

Fig 9: Fake Image after 100 epochs

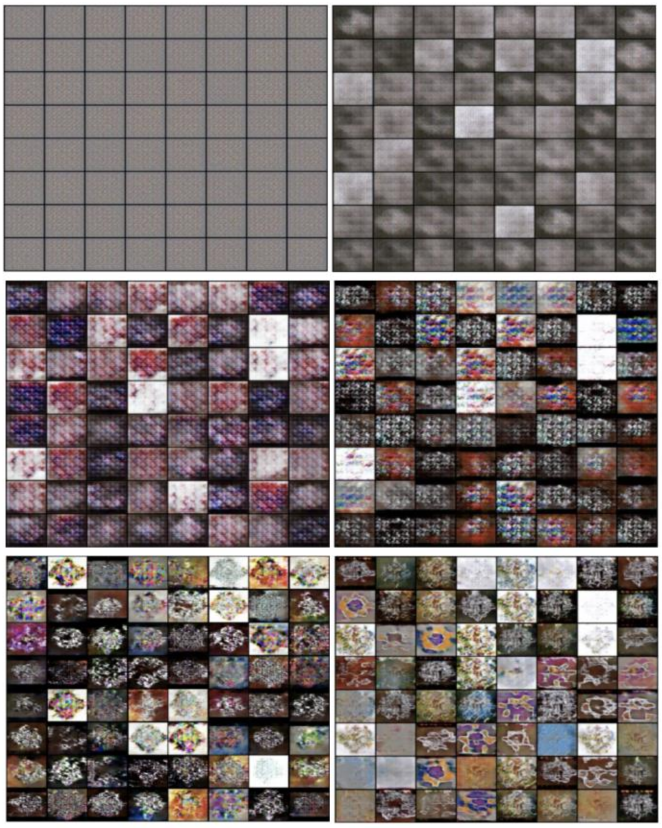

Fig 10: Learning Evolution across Epochs

7. CONCLUSIONS AND FUTURE WORK

The scope of this project and its future work will be to understand symmetricity and creativity is perceived by a machine and make it explainable AI. GANs are currently in the forefront of building up on Creative AI. The field is dynamic with new and advanced models coming out regularly. This paper demonstrated on how to generate rangoli images using a GAN. The paper used a small dataset with limited number of epochs, better results can be expected with larger dataset. The rangoli images are no lesser than a piece of mathematical art form with most of them being symmetry in nature. Currently this paper didn't enforce a symmetry on the generated images so a way to impose a conditional symmetry either implicitly or explicitly can be a future step of this work. Another possible extension would be to have a stacking of multiple GAN architecture so as to generate images at higher resolution. The Image to Image translation of coloring a rangoli image and an alternate implementation of DiscoGAN to have cross domain styling of entities based on rangoli designs seems to be within the scope of this project's future work. As art has no boundaries so does the GAN.

REFERENCES

[1] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu,David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

[2] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434, 2015.

[3] Arjovsky, Martin, Soumith Chintala, and Lon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875, 2017.

[4] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. arXiv preprint arXiv:1708.05509, 2017.

[5] Baur, C., Albarqouni, S., Navab, N., 2018b. Melanogans: high resolution skin lesion synthesis with GANs. arXiv:1510.07818v1.

[6] Huang, Zhiling. "Bird Generation with DCGAN.", unpublished

[7] Alyafi, Basel, Oliver Diaz, and Robert Martí. "DCGANs for realistic breast mass augmentation in x-ray mammography." Medical Imaging 2020: Computer-Aided Diagnosis. Vol. 11314. International Society for Optics and Photonics, 2020.

[8] Donahue, Chris, Julian McAuley, and Miller Puckette. "Synthesizing audio with GANs." (2018).

[9] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 35(1), 53-65.

[10] Y. Cao et al., "Recent Advances of Generative Adversarial Networks in Computer Vision," in IEEE Access, vol. 7, pp. 14985-15006, 2019.

[11] 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. In Advances in neural information processing systems (pp. 6626-6637).

[12] Hoang, Q., Nguyen, T. D., Le, T., & Phung, D. (2018). MGAN: Training generative adversarial nets with multiple generators.

[13] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

[14] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).

[15] Thakur, M. (2014). A Creative Expression of Celebrations: Rangoli.

[16] Bapat, M. (2008). Mathematics in RangoLee art from India. In Proceedings of Conference on Mathematical Art (pp. 429-432).

[17] Maher, Mary & Fisher, Doug. (2012). Using AI to Evaluate Creative Designs. ICDC 2012 - 2nd International Conference on Design Creativity, Proceedings. 1.