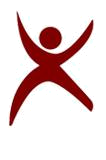
**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE** **TECHNOLOGIES**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**RGUKT-Nuzvid, Eluru Dist – 521202**

**IMAGE GENERATION USING GENERATIVE ADVERSARIAL NETWORKS (GAN’S)**

*Report submitted to*

*Rajiv Gandhi University of Knowledge Technologies,*

*Nuzvid for the fulfillment of Mini Project*

*of*

Bachelor of Technology

in Computer Science and Engineering

*by*

**N180114 (K.Sindhuja)**

**N180117(A.Pushpavathi)**

**N180509 (P.Naga lakshmi)**

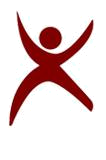
**N180257 (Sk.khaja.Hussain)**

**N180086(M.Thrisali)**

*Under the Esteem Guidance of*

**Dr. S.Chiranjeevi**

Head of the Department

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE** **TECHNOLOGIES**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**RGUKT-Nuzvid, Eluru Dist – 521202**

**Declaration**

We certify that

1. The work contained in this report is original and has been done by us under the guidance of my supervisor(s).
2. The work has not been submitted to any other Institute for any degree or diploma.
3. We have followed the guidelines provided by the Institute in preparing the report.
4. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, We have given due credit to them by citing them in the text of the report and giving their details in the references. Further, We have taken permission from the copyright owners of the sources, whenever necessary.

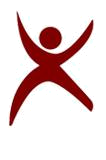
N180114 (K.Sindhuja)

N180117(A.Pushpavathi)

N180509 (P.Naga lakshmi)

N180257 (Sk.khaja.Hussain)

N180086(M.Thrisali)

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE** **TECHNOLOGIES**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**RGUKT-Nuzvid, Eluru Dist – 521202**

**Certificate**

This is to certify that the Dissertation Report entitled, **“Image Generation using Generative Adversarial Network (GAN’S)”** submitted by **K.Sindhuja, A.Pushpavathi, P.Naga lakshmi, Sh.Hussain, M.Thrisali** to Rajiv Gandhi university of Knowledge Technologies, Nuzvid, India, is a record of bonafide Project work carried out by us under my/our supervision and guidance and is worthy of consideration for the fulfillment of mini-project of Bachelor of Technology in computer Science and Engineering of the Institute.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr.S.Chiranjeevi Sir Examiner**

Project Supervisor Examiner Project Examiner

Faculty Dept. of CSE Faculty Dept. of CSE  **RGUKT IIIT Nuzvid RGUKT IIIT Nuzvid**

**ACKNOWLEDGEMENT**

We would like to express our profound gratitude and deep regards to our guide **Dr.S.Chiranjeevi *Sir*** for his exemplary guidance, monitoring and constant encouragement to usthroughout this semester. We shall always cherish the time spent with him during the course of this work due to the invaluable knowledge gained in the field of Machine Learning.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **“Image Generation using Generative Adversarial Networks”.**

We express gratitude to our HOD sir (Dept. of CSE) and other faculty members and our beloved Seniors for being source of inspiration and constant encouragement which helped us in completing the project successfully.

N180114(K.Sindhuja)

N180117(A.Pushpavathi)

N180509 (P.Naga lakshmi)

N180257 (Sk.khaja.Hussain)

N180086(M.Thrisali)

**CONTENTS**

|  |  |
| --- | --- |
| Title Page | 1 |
| Declaration | 2 |
| Certificate by Supervisor | 3 |
| Acknowledgement | 4 |
| Contents | 5 |
| Abstract:  1.What is the problem  2.Applications  3.Limitations for existing approach  4.What is your approach  5.Which dataset and it’s size  6.Results | 6 |
| Chapter 1 Introduction | 7 |
| Chapter 2 Background and Related Works | 9 |
| Chapter 3 Implementation  4.1 Import libraries  4.2 Data preparation  4.3 Define generator network  4.4 Define discriminator network  4.5 Define loss function  4.6 Training loop  4.7 Evaluation and Generation  4.8 Fine turning | 13 |
| Chapter 5 Internal Processes | 15 |
| Chapter 6 Result and Analysis | 16 |
| Chapter 7 Conclusion | 19 |
| References |

**ABSTRACT**

**1.What is the problem ?**

One of the main problem with GANs is that they can be difficult to train. Because the generator and discriminator networks are trained together, it can be difficult to get them to converge to a stable solution. This means that training a GAN can sometimes take a lot of time and resources.

Another problem with GANs is that the generated images can sometimes be blurry or distorted. This can happen if the generator network is not able to capture all of the details of the real data, or if the discriminator network is not able to provide good feedback to the generator. This can make the generated images less useful for certain applications, such as medical imaging or self-driving cars, where high-quality images are important.

1. **Applications**

**Image Generation:** GANs are commonly used for generating realistic images. They can learn from a training dataset and generate new images that resemble the training data.

**Image Editing and Manipulation:** GANs can be employed for image editing tasks such as image inpainting (filling in missing parts of an image), super-resolution (increasing image resolution), style transfer (transforming the style of an image), and image-to-image translation (e.g., converting a day scene to night)

**Text-to-Image Synthesis:** GANs can generate images from textual descriptions. By combining natural language processing and GAN architectures, it is possible to generate images based on textual prompts, enabling applications such as text-based image retrieval and content creation.

**Video Synthesis and Prediction:** GANs can generate and predict videos based on existing video sequences. This has applications in video editing, special effects generation, and video game development.

**Medical Image Analysis:** GANs can assist in medical image analysis tasks, such as image segmentation, anomaly detection, and disease diagnosis. They can generate synthetic medical images that help in training and evaluation of medical imaging algorithms.

**Fashion and Product Design:** GANs can generate new fashion designs, product prototypes, or furniture designs based on existing examples. This can aid in the creative process, prototyping, and generating novel designs.

**Game Development:** GANs can generate game assets like characters, environments, and textures. They can also be used to enhance the realism and variety of computer-generated worlds in video games.

1. **Limitations of existing approach**

**Training Complexity:** GANs require careful training and tuning of hyperparameters to achieve satisfactory results. Configuring and optimizing GAN models can be a time-consuming and challenging process. The training process often involves balancing the generator and discriminator networks, selecting appropriate loss functions, and handling issues like mode collapse or training instability.

**Computational Resources:** Training GANs can be computationally intensive and resource-demanding. GANs typically require powerful hardware, such as high-performance GPUs, and substantial memory capacity.

**Unpredictable Results:** The output of GANs can be unpredictable and difficult to control. While GANs can generate visually appealing images, there is no guarantee that the generated samples will consistently meet specific requirements or match the desired style.

**Mode Collapse and Lack of Diversity:** Mode collapse is a common issue in GAN training where the generator fails to capture the full diversity of the training data. Instead, it generates a limited range of similar images, leading to a lack of variation and potential repetition in the generated samples. Achieving diverse and high-quality image generation remains a challenge.

**Ethical Concerns:** GANs can generate realistic fake images, which raises ethical concerns. The misuse of GAN-generated images for deceptive purposes, such as creating deepfakes or fake identities, is a significant concern.

**Limited Interpretability:** GANs operate as black-box models, making it challenging to interpret or understand the internal workings or decision-making processes. Interpreting how and why a GAN generates specific images can be difficult, hindering their use in applications where interpretability is essential.

1. **What is your approach ?**

Our approach for image generation is using GANs. GANs are widely used for image generation. They consist of two neural networks: a generator and a discriminator. The generator generates images from random noise, while the discriminator tries to distinguish between real and generated images. The networks are trained simultaneously, with the generator aiming to fool the discriminator. GANs have been successful in generating realistic and diverse images.

1. **Which dataset and it’s size ?**
2. **Results ?**

**INTRODUCTION**

Generative Adversarial Networks (GAN’s) have gained significant attention in recent years for their ability to generate realistic images. GAN’s consist of two neural networks, a generator and a discriminator, that compete against each other to produce high-quality images. The generator creates images that are then evaluated by the discriminator, which tries to distinguish between real and fake images. The generator is trained to create images that fool the discriminator, resulting in highly realistic images. Our project focuses on exploring the potential of GAN’s for image generation, and investigates the challenges of training GAN’s to generate high-quality images. We also explore ways to improve the diversity and realism of generated images, and investigate the impact of different datasets on GAN performance. Our results demonstrate the potential of GAN’s for image generation, and provide insights into the challenges and opportunities of using GAN’s for real-world applications such as video game development, product design, and art.

**BACKGROUND AND RELATED WORKS**

As a new generation model framework, the Generative Adversarial Network proposed in 2014 can generate a composite image that is better than the previous generation model, and has since become one of the most popular research fields.

GANs have been used for a variety of image generation tasks, including generating photorealistic images of faces, landscapes, and even furniture. They have also been used for image-to-image translation, such as converting a daytime image to a nighttime image or a sketch to a photorealistic image. GANs have the potential to revolutionize many fields, including art, design, and entertainment. However, there are still many challenges to overcome, such as training instability and mode collapse.

Recent research in GANs has focused on improving their stability and robustness. One approach is to use different types of loss functions, such as Wasserstein distance or hinge loss, which can lead to more stable training. Another approach is to use techniques such as spectral normalization or weight normalization to improve the stability of the network. Additionally, researchers have explored the use of GANs in combination with other techniques, such as reinforcement learning, to generate more complex and diverse images.

**IMPLEMENTATION**

The proposed system was implemented as follows:

**4.1 Import libraries**:

Begin by importing the necessary libraries such as TensorFlow, PyTorch, or other deep learning frameworks.

**4.2 Data Preparation:**

Obtain a dataset of real images that will serve as the training data for the GAN. Preprocess and normalize the images as necessary. Ensure that the data is in a format suitable for training the GAN.

**4.3 Generator network:**

Design and define the architecture of the generator network. It typically takes random noise as input and generates synthetic samples.The size of the input vector can vary depending on the complexity of the images.The generator generates synthetic images as the output. The output shape of the generator should match the shape of the real images.

**4.4 Discriminator network**

Design and define the architecture of the discriminator network. It takes either real or synthetic samples as input and tries to classify them correctly.The discriminator produces a probability score indicating the likelihood that the input image is real.

**4.5 Loss Functions:**

Generator Loss: The generator aims to generate synthetic images that can fool the discriminator. The generator loss is typically computed based on the discriminator's output when fed with the generator's synthetic images. The goal is to minimize this loss.

Discriminator Loss: The discriminator's objective is to correctly classify real and synthetic images. The discriminator loss is calculated based on its ability to distinguish between the two. The goal is to minimize this loss.

**4.6 Training loop:**

Alternately train the generator and discriminator networks. In each iteration of the training loop, perform the following steps:

1. Generate synthetic samples using the generator network.

2. Combine real and synthetic samples into a single batch.

3. Train the discriminator on the combined batch, with appropriate labels indicating the source (real or synthetic) of each sample.

4.Train the generator using the output of the discriminator. The generator aims to generate samples that are classified as real by the discriminator.

5.Repeat steps 1-4 for a specified number of iterations.

**4.7 Evaluation and Generation:**

After training, evaluate the performance of the generator by generating synthetic images and assessing their quality.Use the trained generator to generate new images by sampling random noise vectors from the latent space and passing them through the generator.

**4.8 Fine turning:**

Fine-tune the GAN by adjusting hyperparameters, network architectures, or training procedures to improve the quality of generated images.

**INTERNAL PROCESSES**

Generating images using Generative Adversarial Networks (GANs) involves a multi-step process that includes data preparation, model architecture design, training, and image generation.

**Data Collection and Preparation:**

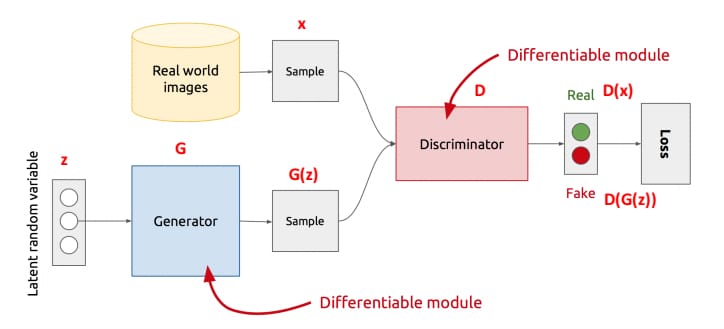
****1.Prepare the Data****

We will train the DCGAN with a dataset called CelebA from Kaggle, which is a collection of Celebrity faces scraped from **[www.getchu.com](http://www.getchu.com/" \t "https://pyimagesearch.com/2022/02/07/anime-faces-with-wgan-and-wgan-gp/_blank)**. There are 63,565 small color images to be resized to 64x64 for training.

**2.GAN Model Architecture Design:**

Determine the type of GAN architecture suitable for your task, such as Deep Convolutional GANs (DCGANs),or Conditional GANs (cGANs).Design the generator network, which takes random noise as input and transforms it into synthetic images. The generator should learn to map the noise vectors to meaningful image representations.Design the discriminator network, which acts as a binary classifier to distinguish between real and fake images. The discriminator should learn to differentiate between the generated images and real images from the dataset.

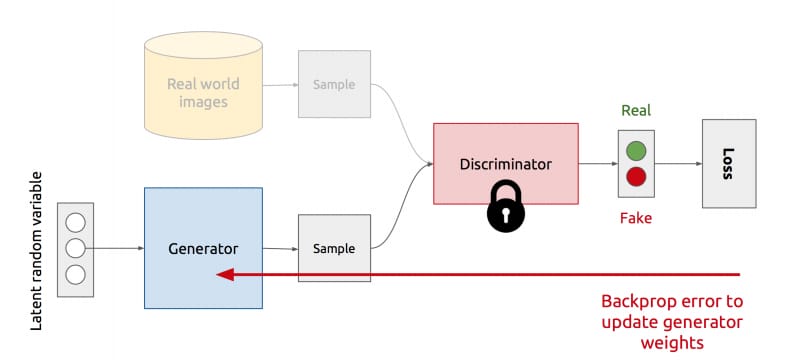
**GAN’s Architecture**

****

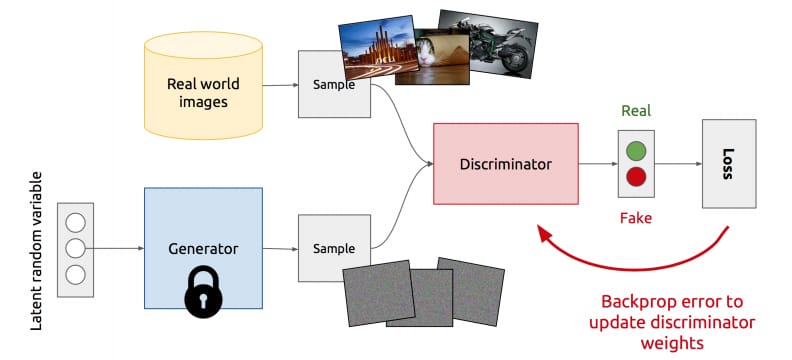
**3.Training Process:**

Initialize the generator and discriminator networks with random weights.Alternate between training the generator and discriminator in a series of mini-batch iterations.

**3.1Training Generator**

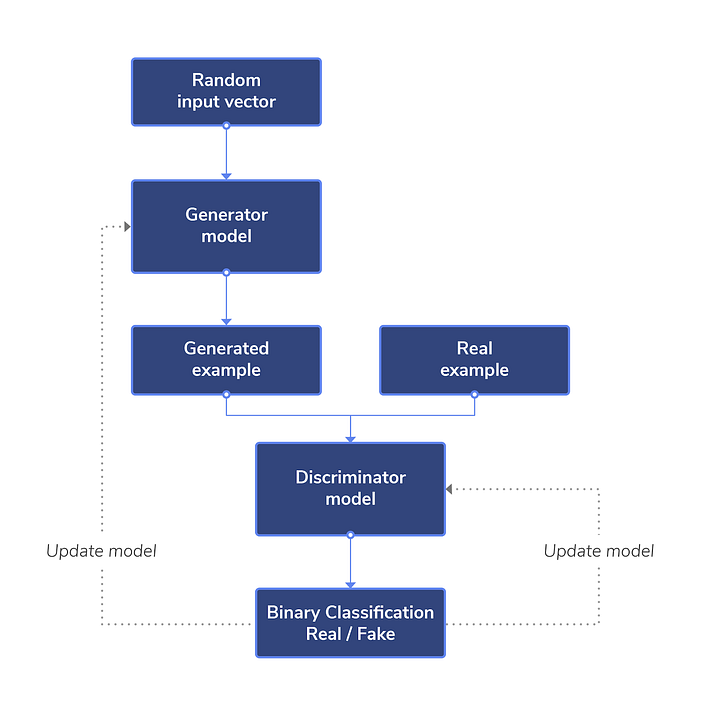


**3.2Training Discriminator**



**For each iteration:**

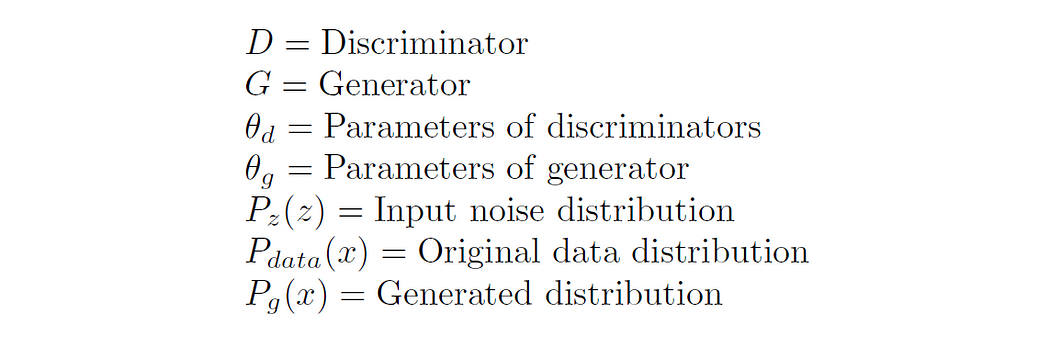
Generate a batch of synthetic images using the current generator.Randomly select a batch of real images from the dataset.Train the discriminator to correctly classify the real and fake images by updating its weights.Train the generator to fool the discriminator by producing more realistic images and updating its weights.Optimize the training process by adjusting hyperparameters (Gradient-Based Optimization)such as learning rate, batch size, and optimization algorithms (Stochastic Gradient Descent(SGD).



**3.3.Evaluation and Validation:**

Monitor the training progress by assessing key metrics such as discriminator and generator losses, image quality, and diversity.Adjust the model architecture or training parameters as needed based on the evaluation results to improve performance.

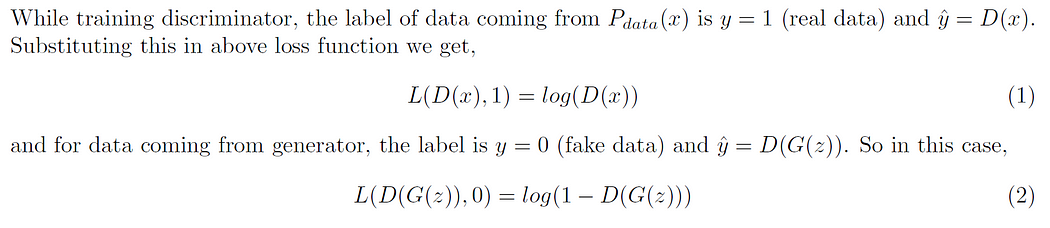
Before we go into the derivation, let’s describe some parameters and variables.



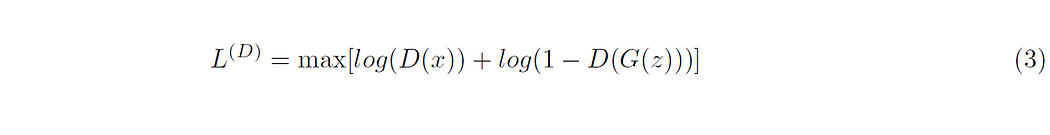
# **3.3.1 Derivation of the loss function**

The loss function described in the original paper by Ian Goodfellow et al. can be derived from the formula of binary cross-entropy loss. The binary cross-entropy loss can be written as,

## **3.3.2 Discriminator loss**

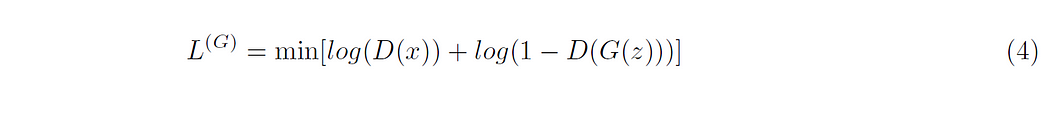


Now, the objective of the discriminator is to correctly classify the fake and real dataset. For this, equations (1) and (2) should be maximized and final loss function for the discriminator can be given as,



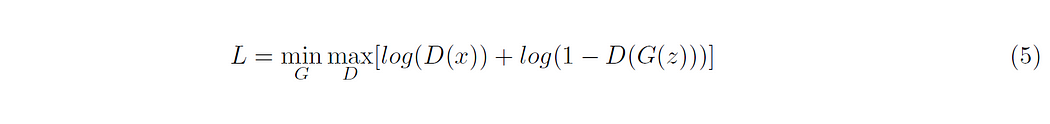
## **3.3.3 Generator loss**

Here, the generator is competing against discriminator. So, it will try to minimize the equation (3) and loss function is given as,

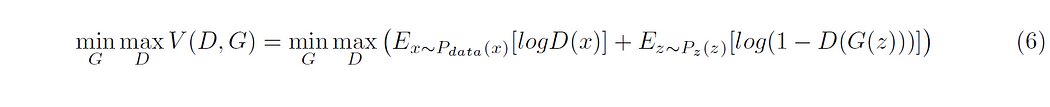


## **3.3.4 Combined loss function**

We can combine equations (3) and (4) and write as,



Remember that the above loss function is valid only for a single data point, to consider entire dataset we need to take the expectation of the above equation as



**4.Image Generation:**

After the training process, the generator is capable of generating new images.Sample random noise vectors from a chosen distribution (e.g., Gaussian or uniform) as input to the generator.Pass the noise vectors through the trained generator to obtain synthetic images.Post-process the generated images if necessary (e.g., denoising, resizing, color adjustments).

**5.Iterative Refinement:**

If the generated images do not meet the desired quality or exhibit certain issues, you may need to refine the process.Experiment with different model architectures, training techniques, loss functions, or regularization methods to improve the image generation results.Collect additional data or augment the existing dataset to increase the diversity of training examples.It's important to note that the specifics of each step may vary depending on the GAN architecture, problem domain, and available resources.

**RESULT AND ANALYSIS**

**CONCLUSION**

**REFERENCES**

https://realpython.com/python-sockets/

**<https://ieeexplore.ieee.org/document/10038584/>**