DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Alternate Assessment Tool Report submitted for the subject

Machine Learning – 22EC554

Submitted by

Krushik JP (1DS22EC107)
Shyam Sundar B (1DS22EC212)
Kartik Vijayasinha Desai (1DS22EC097)
Gagan M Kakol (1DS23EC408)

Under the Guidance of

Dr. Deepa N P
Associate Professor
Department of Electronics & Communication Engineering
Dayananda Sagar College of Engineering, Bengaluru

External Mentor

Shivaranjini Mithun
Senior Technical Architect,
Elektrobit India Pvt. Ltd., Bengaluru

Evaluation

USN	Name		Simulation & Analysis -10 Marks	Presentation & Report -10 Marks
1DS22EC107	Krushik jp			
1DS22EC212	Shyam Sundar B			
1DS22EC097	Kartik Vijayasinha desai			
1DS23EC408	Gagan M Kakol			

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590111, KARNATAKA, INDIA
2024-25

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Alternate Assessment Tool (AAT) entitled **"Text-to-Image generation for creating synthetic data Using GAN"** as part of **Machine Learning (22EC554)** is a bonafide work carried out by Krushik jp (1DS22EC107), Shyam Sundar B (1DS22EC212), Kartik Vijayasinha desai (1DS22EC097) & Gagan M Kakol (1DS23EC408) as 20-mark component in partial fulfillment for the 5th semester of Bachelor of Engineering in **Electronics & Communication Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2024-2025. The AAT report has been approved as it satisfies the academic requirements prescribed for the Bachelor of Engineering degree.

Signature of Faculty
[Dr. Deepa N P]

Signature of HOD
[Dr. Shobha K R]

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DECLARATION

We declare that we abide by the ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. The work submitted in this report of **Machine Learning (22EC554)**, 5th Semester BE, ECE has been compiled by referring to the relevant online and offline resources to the best of our understanding and in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in **Electronics & Communication Engineering**, at Dayananda Sagar College of Engineering, an autonomous institution affiliated to VTU, Belagavi during the academic year 2024-2025.

We hereby declare that the same has not been submitted in part or full for other academic purposes.

1DS22EC107-Krushik jp

1DS22EC212-Shyam Sundar B

1DS22EC097-Kartik Vijayasinha desai

1DS23EC408-Gagan M Kakol

Place:Bengaluru

Date:

ACKNOWLEDGEMENT

It is with great pleasure and gratitude that we acknowledge the support, encouragement, and guidance of numerous individuals who made the successful completion of this project possible.

We extend our sincere thanks to **Dayananda Sagar College of Engineering** for providing us with the platform and resources to undertake this project as part of our academic journey.

We are particularly grateful to **Dr. B. G. Prasad**, Principal of Dayananda Sagar College of Engineering, for his constant encouragement and visionary leadership that inspires excellence among students.

We express our heartfelt gratitude to **Dr. Shobha K. R.**, Professor and Head of the Department of Electronics & Communication Engineering, for her invaluable support, constructive feedback, and motivation during every stage of this project.

We extend our sincere thanks to **Dr. Deepa N. P.**, our course faculty, for her insightful guidance, encouragement, and expertise, which were instrumental in successfully completing this project.

Our heartfelt appreciation goes to the **ElectroBit Mentors**, whose guidance and valuable inputs helped us navigate challenges throughout the project.

We also take this opportunity to thank our **friends and peers**, whose collaboration and constructive suggestions added value to our work. Additionally, we acknowledge the indirect support and encouragement provided by **other faculty members**, whose guidance contributed to our learning.

Lastly, we are deeply grateful to our **families** for their unwavering support, patience, and motivation, which gave us the strength to persevere and excel.

We sincerely thank all these individuals for their contributions, making this project a successful and enriching experience.

1DS22EC107-Krushik jp

1DS22EC212-Shyam Sundar B

1DS22EC097-Kartik Vijayasinha desai

1DS23EC408-Gagan M Kakol

ABSTRACT

This project explores the use of Generative Adversarial Networks (GANs) for generating synthetic bird images from textual descriptions. Utilizing datasets of birds and their textual embeddings, models like StackGAN and AttnGAN were employed to translate text into realistic, high-quality images. This approach addresses challenges of limited labelled data by creating diverse synthetic datasets, enabling applications in data augmentation and AI model training.

The framework was validated using bird dataset, with evaluation focusing on image fidelity, diversity, and generation of image based on the embedding text that model has chosen. By leveraging GANs, the project successfully demonstrates the potential of text-to-image generation in producing rich and descriptive datasets for advancing machine learning applications in ecology, biology, and related domains. This project demonstrates the potential of GAN-based text-to-image systems in creating rich and varied datasets, driving advancements in AI model training and testing.

Table of Contents

Figures

1. Introduction:

Text-to-image generation represents a fascinating frontier in artificial intelligence, where models create realistic images based on textual descriptions. This project specifically focuses on generating synthetic bird images using a Generative Adversarial Network (GAN). The CUB-200-2011 birds' dataset, comprising bird images and their corresponding textual descriptions, serves as the foundation for training and validating the model.

The project employs a Stage-I GAN to translate text embeddings into images with a resolution of 64x64 pixels. The primary objective is to explore the effectiveness of GANs in linking textual inputs with visual outputs, with potential applications in data augmentation, content creation, and AI model training.

This report outlines the design, implementation, and results of the text-to-image generation system. It examines how GANs can synthesize high-quality images and demonstrates their capability to produce diverse datasets, showcasing their utility in addressing challenges in data-limited environments.

Through this study, we aim to contribute to the growing field of text-to-image generation, providing insights into the potential of GANs to revolutionize artificial intelligence and machine learning applications.

2. Use case description:

ProblemStatement:

Generating realistic images from textual descriptions is a challenging task in artificial intelligence. Despite advancements in image synthesis, traditional methods struggle to bridge the gap between textual semantics and visual representation effectively. The **text-to-image generation** task using **Generative Adversarial Networks (GANs)** addresses this challenge by enabling the transformation of descriptive text into corresponding images.

This capability is particularly significant in fields where acquiring large, labeled datasets is costly or impractical, such as ecology, biology, and environmental science. For example, creating datasets of rare bird species often involves time-intensive manual labeling or fieldwork. This project highlights the need for **GAN-based text-to-image models** to provide an efficient solution by generating synthetic, high-quality images based on textual descriptions, thereby augmenting data availability and reducing dependence on real-world data.

Methodology:

The primary objective of this project is to explore the use of Generative Adversarial Networks (GANs) to generate synthetic bird images based on textual descriptions. In many fields, especially in scientific research and environmental studies, acquiring large and labeled datasets is both costly and time-consuming. The use of GANs in generating images from textual inputs presents a powerful solution to this problem, allowing researchers and machine learning practitioners to augment existing datasets with synthetic images that reflect a wide range of visual variations based on textual descriptions. This approach has the potential to make training machine learning models more efficient and accessible, especially in domains with limited labeled data.

The system in this project takes textual descriptions of birds, including their physical characteristics and habitat information, and uses a trained GAN model to generate corresponding images. For example, a description such as "A small bird with a red head and yellow wings" would result in the creation of a synthetic image that matches these specified features.

This use case has several important applications:

- Data Augmentation: In many machine learning tasks, especially in image classification or object detection, having a sufficient amount of training data is crucial for the performance of the models. By generating realistic images from textual descriptions, this system can create additional training data when real data is limited or expensive to gather.

- Content Creation: Automatically generating images for use in educational materials, wildlife documentaries, or virtual environments..

- Generative AI Development: Text-to-image generation using GANs is pivotal in Generative AI, which focuses on creating new content (e.g., images, audio, or text) rather than analyzing existing data.

- Machine Learning: Enhancing models by providing more diverse, labelled visual data for training, thus improving the performance and generalization ability of AI systems.

By using GANs for text-to-image generation, this project provides a valuable tool for data augmentation, content creation, and the advancement of machine learning technologies, all while reducing the need for time-intensive manual data collection.

3. WORKING PRINCIPLE:

The model used in this project is **Generative Adversarial Networks (GANs)** to generate realistic bird images based on textual descriptions. The process consists of several steps, including data preparation, training of two primary components (the Generator and the Discriminator), and adversarial training. Below steps explain how the model works:

1. **Dataset Preparation**:
    - Text descriptions of birds are converted into text embeddings using pre-trained models (e.g., CNN-RNN).
    - Each image in the CUB-200-2011 dataset is paired with a corresponding text description.
2. **Text Embedding**:
    - Text descriptions are processed into embeddings that represent their semantic meaning.
3. **Generator**:
    - Takes text embeddings and random noise as input.
    - Generates images by up sampling from low resolution (4x4) to the target size (64x64) as the epochs are generated.
    - Uses fully connected layers and convolutional layers for image generation.
4. **Discriminator**:
    - Evaluates whether the generated image is real or fake.
    - Verifies if the image aligns with the input text embedding.
    - Uses convolutional layers to analyse the image features.
5. **Adversarial Training**:
    - Generator improves by trying to fool the Discriminator.
    - Discriminator gets better at distinguishing real from fake images and verifying the alignment with the text.
6. **KL Divergence Loss**:
    - Used to minimize the difference between the generated and real distributions of latent variables.

- the KL Divergence loss is calculated between the mean and log variance of the latent variables. Here's the exact formula we have used for calculation of losses :

$$DKL(q(z)\|p(z)) = -0.5 \times \sum(1+\log(\sigma 2) - \mu 2 - \sigma 2)$$

Where:

μ: The mean of the latent variable distribution.

σ2: The variance (exponentiated from log variance) of the latent variable distribution.

The summation is over all dimensions of the latent variable vector.

- In the context of the code, the KL Divergence is used in the **CA_NET** module to regularize the latent space by encouraging the posterior distribution (the distribution of the latent variables conditioned on the text embeddings) to be close to the prior distribution (typically a standard normal distribution, N (0,1).

7. **Image Generation**:
   - After training, the Generator creates images based on new text descriptions.
   - The output is a synthetic bird image that matches the given description.
8. **Output**:
   - The generated images are displayed or saved at 64x64 resolution
9. **Epochs:**
   - The model is trained over several epochs (iterations) where the Generator and Discriminator are updated to improve their performance.
   - In each epoch, the Generator tries to produce better images while the Discriminator learns to distinguish between real and fake images more accurately.
   - The model's performance improves with each epoch, with the loss gradually decreasing as the Generator creates more realistic images aligned with the text.

- **Workflow of a GAN network**

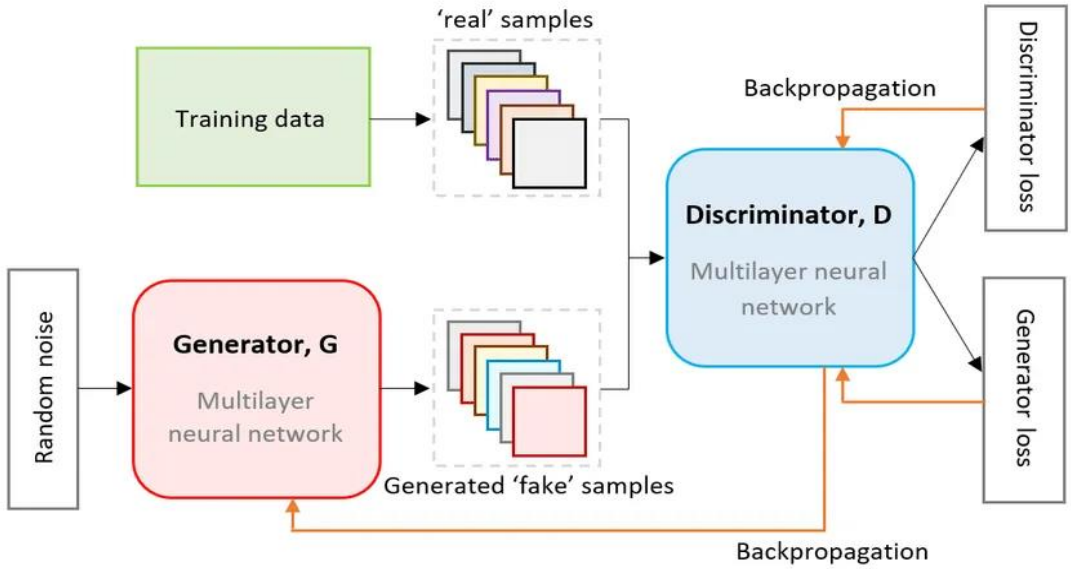

Figure 1: GAN block diagram

The process begins with an embedding text file that provides a description of the image to be generated. This text file is then preprocessed to clean and prepare the text for encoding. The cleaned text is converted into a numerical format using a text encoder, enabling the model to interpret the content. This encoded text is passed to a generative model, like Stack GAN, which generates an image based on the provided description. The generated image is then output, completing the process.

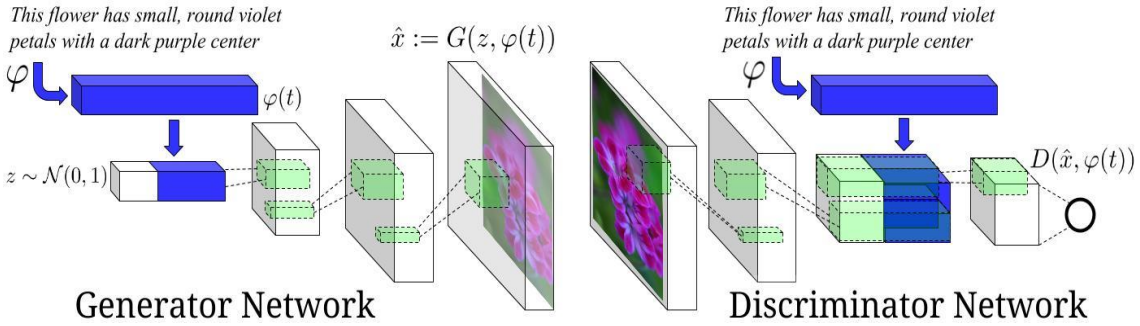

Figure 2: Detailed view of working mode

4. Alogrithms Used

Generative Adversarial Network (GAN)

Generator Pseudocode:

```
Function Generator(Z, Text_Embedding):
    # Z is random noise vector, Text_Embedding is the text processed into embeddings
    Combined_Input = Concatenate(Z, Text_Embedding)  # Combine noise and text embedding
    Image = Create_Image(Combined_Input)  # Use a neural network to generate the image
    Return Image
```

Discriminator Pseudocode:

```
Function Discriminator(Real_Image, Fake_Image, Text_Embedding):
    # Evaluate if images are real or fake based on the text condition
    Real_Decision = Classify(Real_Image, Text_Embedding)
    Fake_Decision = Classify(Fake_Image, Text_Embedding)
    Return Real_Decision, Fake_Decision
```

GAN Training Loop:

```
For each Epoch:
    # Train the discriminator on real and fake images
    D_loss_real = Train_Discriminator(Real_Image, Real_Labels)
    D_loss_fake = Train_Discriminator(Fake_Image, Fake_Labels)
    D_loss = D_loss_real + D_loss_fake

    # Train the generator to fool the discriminator
    G_loss = Train_Generator(Text_Embedding, Z)
```

Text Embedding Models (e.g., Word2Vec, BERT):

```
Function Text_Embedding(Text_Input):
    # Use a pre-trained model (e.g., BERT or Word2Vec) to generate text embeddings
    Embedding = Pretrained_Model(Text_Input)
    Return Embedding
```

Deep Convolutional Neural Networks (DCNN):

DCNNs are used for processing and generating images by learning hierarchical patterns in spatial data. They consist of layers that apply convolutions to detect features like edges, textures, and shapes in images.

```
Function Convolutional_Layer(Input, Filters, Stride):
    # Apply filter to input image and perform convolution
    Output = Convolve(Input, Filters, Stride)
    Return Output
```

Attention Mechanism:

```
Function Attention(Queries, Keys, Values):
    # Compute attention weights based on queries and keys
    Attention_Scores = Compute_Scores(Queries, Keys)
    Normalized_Attention = Softmax(Attention_Scores)
    Output = Weighted_Sum(Normalized_Attention, Values)
    Return Output
```

Perceptual Loss:

Perceptual loss measures the difference between high-level features of generated and real images, extracted using a pre-trained model like VGG.

```
Function Perceptual_Loss(Generated_Image, Real_Image, Pretrained_Model):
    # Extract features from a pre-trained model (e.g., VGG)
    Gen_Features = Pretrained_Model(Generated_Image)
    Real_Features = Pretrained_Model(Real_Image)
    # Calculate difference between generated and real features
    Loss = Mean_Squared_Error(Gen_Features, Real_Features)
    Return Loss
```

5. RESULTS AND DISCUSSION:

The text-to-image generation program successfully created images based on the provided embedding text descriptions, but in this model, the text embeddings were taken by the model itself, so we're desperate to create our own database for the embedding text and the prompt can be given by user and based on that prompt our model will provide the images.

Key discussions:

1. **Image Quality**: The images were clear and aligned with the descriptions, capturing key attributes like colour and shape. However, complex or abstract descriptions occasionally resulted in minor inconsistencies.
2. **Text-Image Consistency**: The generated images generally reflected the text input well, though ambiguity in the description sometimes led to mismatches.
3. **Performance**: The model generated images within a reasonable time frame, with longer descriptions causing slight delays. The computation efficiency could be improved.
4. **Limitations**: The model struggled with complex or detailed descriptions, and images lacked fine details like textures, and it takes more time to generate the image on the CPU's which are not that high end.
5. **Applications**: This system shows potential in fields like game development, design, and education, Generative Ai development, providing an automatic way to generate visuals from text descriptions.

Obtained results:

Generated text description:

	image	train_text	vector_emb
0	[[[104, 131, 143], [121, 142, 154], [135, 151,...	this is a yellow bird with grey wings and a bl...	[0.30727547, 0.11746128, 0.05263393, 0.0447600...
1	[[[53, 63, 69], [74, 83, 86], [79, 87, 89], [7...	this is a yellow bird with grey wings and a bl...	[0.048059117, 0.090454206, 0.21026824, 0.13315...
2	[[[83, 103, 23], [80, 102, 26], [87, 111, 19],...	this is a yellow bird with grey wings and a bl...	[0.2544848, 0.112521246, 0.056623276, 0.097706...
3	[[[150, 148, 153], [153, 151, 156], [155, 153,...	this is a yellow bird with grey wings and a bl...	[-0.18242551, 0.1438286, 0.02397212, -0.130611...
4	[[[78, 69, 67], [83, 72, 71], [75, 63, 62], [5...	this is a yellow bird with grey wings and a bl...	[-0.02236132, -0.025680661, 0.47425216, 0.1033...

Generated image with losses based on the based on the text description:

- **Epoch 1**

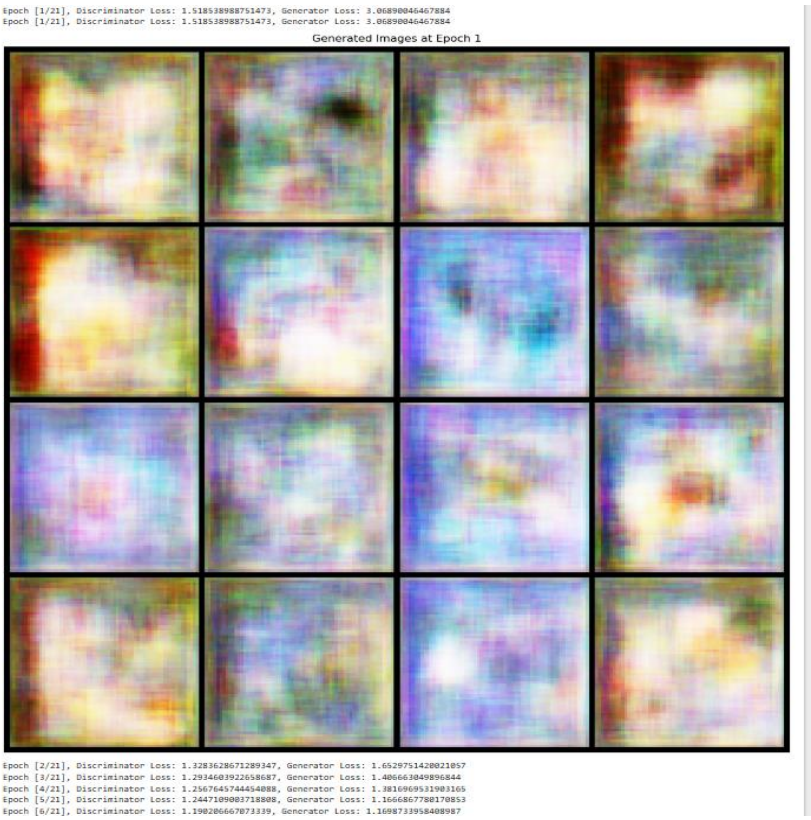

Figure 3: 1st Epoch Display

Discriminator Loss: 1.3283628671289347, Generator Loss: 1.6529751420021057

- **After 10 Epochs**

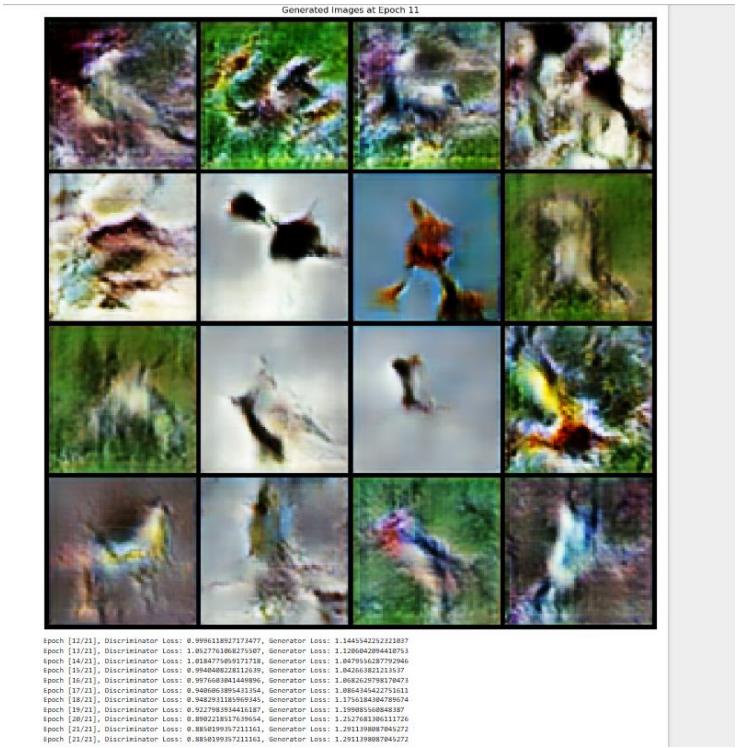

Figure 4: Display after 10 Epoch

Discriminator Loss: 0.9996118927173477, Generator Loss: 1.1445542252321037

- **After 20 Epochs**

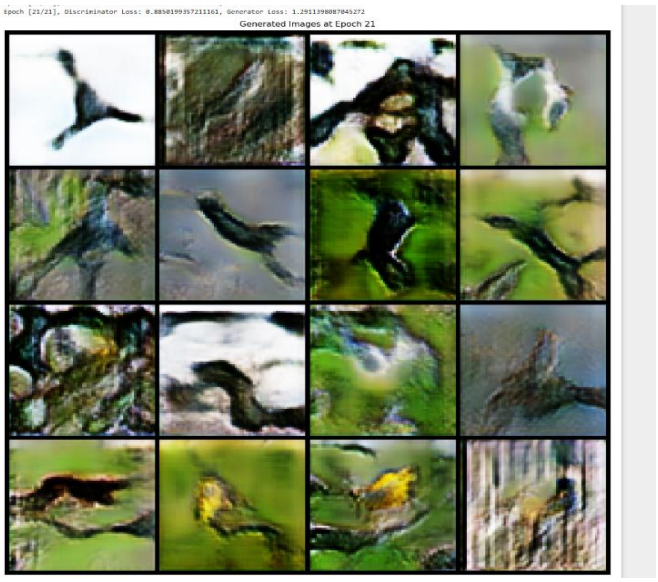

Figure 5: Display after 20 Epoch

Discriminator Loss: 0.8850199357211161, Generator Loss: 1.2911398087045272

- **Final images generated**

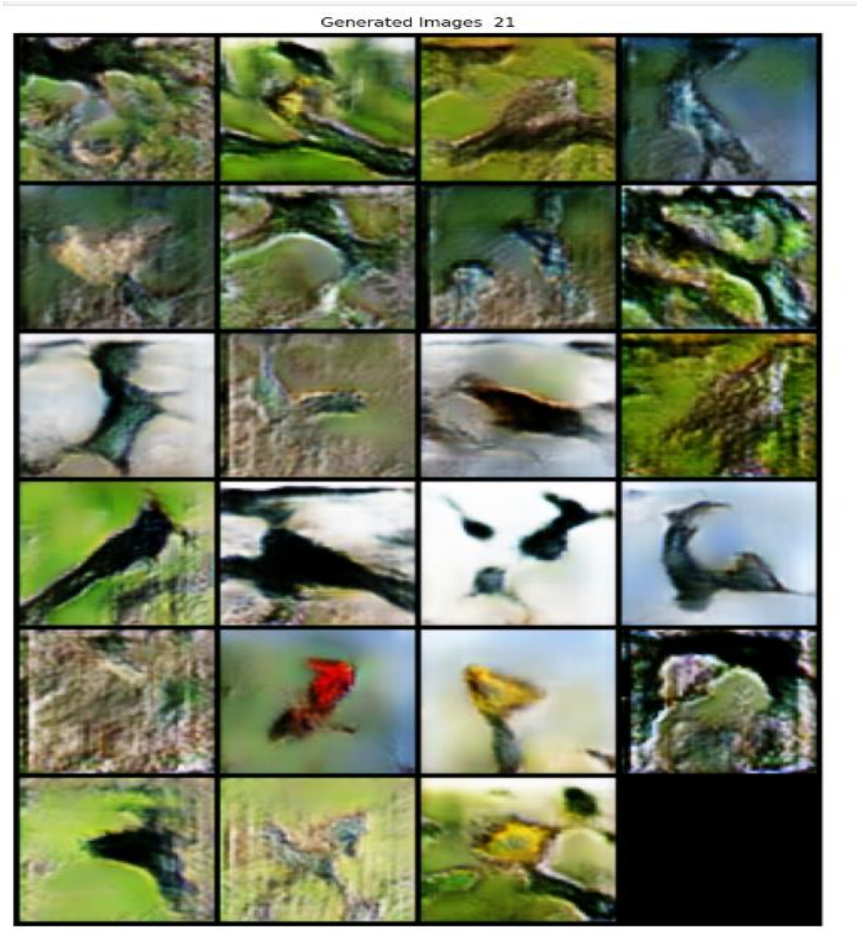

Figure 6: Final Display

"At the 21st epoch, with Discriminator Loss at 0.885 and Generator Loss at 1.291, the model is progressing well. The generator is refining its ability to produce more realistic images while the discriminator continues to effectively distinguish between real and fake images. This balance indicates stable training, and as the epochs progress, both losses are expected to further decrease, resulting in higher-quality, text-aligned images."

6. Conclusion

The text-to-image generation system demonstrated the ability to transform embedding text descriptions into relevant visual representations. While it performed well in creating basic images with accurate representations of colours and shapes, certain limitations were observed in handling complex or abstract descriptions. Despite these challenges, the model showed promise in automating visual content creation, with practical applications in various domains like design, education, and entertainment.

Future Work

1. **Improving Accuracy**: Enhancing the model's ability to handle more complex or detailed descriptions, ensuring better alignment between the text and generated image.
2. **Real-time Generation**: Optimizing the model for faster processing to generate images in real-time, making it more suitable for interactive applications.
3. **Higher Image Resolution**: Working towards generating higher-resolution images to capture finer details and improve overall image quality.
4. **Better Training Models**: Experimenting with larger and more diverse datasets to improve the model's robustness and versatility.
5. **Incorporating User Feedback**: Implementing a feedback mechanism to allow users to refine and adjust the generated images for better customization.
6. **User Prompt Input**: Developing the system to accept prompts directly from the user, allowing more dynamic and flexible image generation based on individual requests.

References:

- H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, ‘‘StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5908–5916.
- C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, ‘‘LAION-5B: An open large-scale dataset for training next generation image-text models,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 35, 2022.
- https://www.kaggle.com/datasets/sovitrath/cub-200-bird-species-xml-detection-dataset
- https://www.kaggle.com/datasets/klu2000030172/birds-image-dataset
- https://github.com/Rakshith-Manandi/text-to-image-using-GAN