Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Mini-Project Report

on

"GenerAI: Photorealistic Landscape Generation"

[COMP 488: Neural Network and Deep Learning]

(For partial fulfilment of 4th Year 1st Semester in Computer Science)

Submitted by:

Sanskar Singh Dangol (CS)

Siddhartha Pradhan (CS)

Submitted to:

Dr. Bal Krishna Bal

Department of Computer Science and Engineering

Submission Date: December 25, 2024

Abstract

GenerAI aims to develop an AI model that generates photorealistic landscapes using Generative Adversarial Networks (GAN). These networks have proved successful throughout the years in generating synthesized, high-quality images and this project will explore its potential further in creating visually appealing landscapes that mimic natural landscapes such as mountains, forests, rivers, etc. By leveraging the training process of GANs, the model will learn to produce realistic outputs.

The goal is to train a deep learning model, and learn about the workings of a GAN. The GAN will be trained on a dataset of landscape images, allowing it to capture natural features of the images, and through iterative adversarial training, the system will learn to create photorealistic images.

GenerAI will explore the complexities of deep learning architectures, and focus on optimizing the training process to ensure high-quality outputs. This will require careful tuning of hyperparameters and training techniques. The expected outcome is an AI capable of generating high-res images of realistic landscapes, demonstrating the power of deep learning in creative image generation.

Keywords: deep learning, landscapes, image generation

Table of Contents

List of Figures

Acronyms/Abbreviations

GAN: Generative Adversarial Networks

DL: Deep Learning

ML: Machine Learning

VR: Virtual Reality

FID: Frechet Inception Distance

PIL: Python Imaging Library

CNN : Convolutional Neural Network

DCGAN: Deep Convolutional Neural Network

NN: Neural Network

GenerAI: Photorealistic Landscape Generation

1.1 Background

Deep learning, in today's perspective, is a very popular term, as it has effectively revolutionised many fields. Aside from complex computations, AI today is creative, dwelling into fields of image and video generation, enabling machines to 'create' realistic visual content. By the use of neural networks termed GANs, machines today can create lifelike images through adversarial training. GAN is a relatively new framework for estimating generative models via an adversarial process, where two models are simultaneously trained: a generative model (G) that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game (Goodfellow, 2014). GANs are used in a wide variety of fields today, including art, fashion, video game development, and even VR. Generative Adversarial Networks (GAN) can also yield realistic 3D environments based on the distribution of remotely sensed images of landscapes, captured by satellites or drones (Panagiotou & Charou, 2020).

Automated content generation is the new wave of social media, and its demand is only growing. Traditionally, creation of high-quality, dynamic environments is often deemed labour-intensive and costly. GenerAI seeks to hence explore the potential of GANs to develop a system capable of generating diverse and realistic landscapes. By automating the creation process, the project plans to streamline a form of content generation.

A GAN model trained on datasets of a wide variety of landscapes can learn to synthesise images that can accurately capture textures, lighting and structures seen in nature. GenerAI also aligns with the growing trend of implementing creativity using AI. The AI community has, in recent years, been pushing boundaries of generative models, and landscape generation is still an area with untapped potential. As technology continues to advance, the demand for realistic and immersive virtual experiences grows. Whether it's a sprawling fantasy landscape in a video game or a meticulously replicated real-world location in a training simulator, the terrain sets the stage for the user's interaction and engagement (Mayonaka, 2023). By

building on advances made on the technology, GenerAI will explore new possibilities in AI-powered content creation.

1.2 Objectives

The main objective of GenerAI is to develop a GAN-based machine learning model that generates high-res, photorealistic landscapes. This project will primarily focus on designing and training the model to generate diverse landscapes (mountains, forests, beaches). Other key objectives include:

- Building a generator network capable of producing realistic landscapes from random noise vectors
- Train a model on comprehensive landscape dataset by optimizing the GAN architecture
- Evaluate the model's performance using subjective evaluation and metrics such as the FID.

1.3 Motivation and Significance

The primary motivation for this project stems from the ever growing role of AI in daily human life. Today, AI is used as a tool for creative processes, and it is only getting better and better. The ability to generate photorealistic images autonomously can have profound impacts on industries reliant on content creation and idea generation. By creating GenerAI, the project aims to reduce manual workload in creating complex landscapes, enable prototyping of worlds in games and simulations, and also explore the artistic capabilities of AI.

The significance of GenerAI lies in its potential to innovate fields of AI generated art and content creation. It also offers practical benefits in generating landscapes, and the project can make meaningful contributions in game development, virtual reality, film, animation, art and design.

Overall, GenerAI will not only show the potential of GANs in effectively generating photorealistic landscapes, but will also highlight the transformative impact AI is currently having on the creative and entertainment industries.

Chapter 2: Related / Existing Works

2.1 NVIDIA GauGAN

NVIDIA's GauGAN is an AI powered software that creates photorealistic landscapes from simple sketches. It takes a rough sketch from users and turns it into high-res photo realistic images using a form of generative adversarial network known as SPADE (Spatially-Adaptive Normalization). GauGAN generates images by interpreting the drawn shapes as elements like mountains, beaches, rivers, etc. GauGAN is popular in gaming, VR environments and even content creation. GauGAN uses semantic segmentation to define different regions in an image. The segments are then interpreted by the model, which then fills in the appropriate textures and colors for the segments.

2.2 Artbreeder

Artbreeder is an online AI tool that makes use of GANs to create and modify images through genetic-style blending and mutation. It allows users to generate images by breeding different images together. With the help of this tool, users can generate a variety of image types, such as faces, landscapes, and even anime characters, enabling them to produce a wide range of visual content. Artbreeder also lets users merge multiple images together, generating a new image that inherits features from both. So simply put, users can blend two different images, creating a unique hybrid of the two. Users can control various aspects of the generated image, such as color, structure, style, and other visual characteristics using sliders provided in the site.

2.3 Google DeepDream

Google DeepDream is a neural network-based image gen tool created by Google. DeepDream was first created as a tool to visualize and comprehend how CNNs interpret images, but its capacity to produce bizarre, dreamlike images quickly made it well-known. It also uses neural networks' latent patterns—despite not being a GAN—to manipulate and produce artistic imagery. It uses deep learning to identify patterns to magnify them into highly abstract images. The tool enhances existing patterns in an image, often producing visuals that resemble dreams, fractals, or otherworldly scenes. DeepDream was initially

developed to visualize the internal workings of CNNs. By selecting different layers of the network, users can explore how the network sees and enhances certain features in an image.

DeepDream showed that NNs can now be used for creative as well as practical tasks. DeepDream made it easier to use machines as tools for artistic expression, erasing the distinction between AI and art. Despite the fact that its pictures are frequently abstract and surreal, they show how closely technology and creativity are related.

Chapter 3: Methodology

3.1 Generative Adversarial Networks

Generative Adversarial Networks are types of neural networks useful for generating new data that resembles a given dataset. GANs are in the forefront of generative AI, and consist of two neural networks, the generator and discriminator, and one principal, the two networks train against each other until optimal results are found.

A generator is a neural network that learns to create fake data similar to the input data. It takes a random noise vector as input and transforms it into data. The goal of a generator is to fool the discriminator by generating data that looks very realistic. A discriminator is a neural network that serves as a binary classifier, and whose goal is to correctly classify the input as real or fake.

So, during GAN training, the generator tries to minimize the discriminator's ability to detect fake data, and the discriminator tries to maximize its ability to correctly classify data. The discriminator's objective is to maximise **log(D(x))**, the log probability of correctly identifying samples, and minimize **log(1-D(G(x)))**, the log probability of misidentifying fake samples The generator's objective is to maximize **log D(G(x))**, so that it can fool the discriminator. The opposing models act as adversaries, and play a minimax game with each other for training.

3.2 Wasserstein GAN

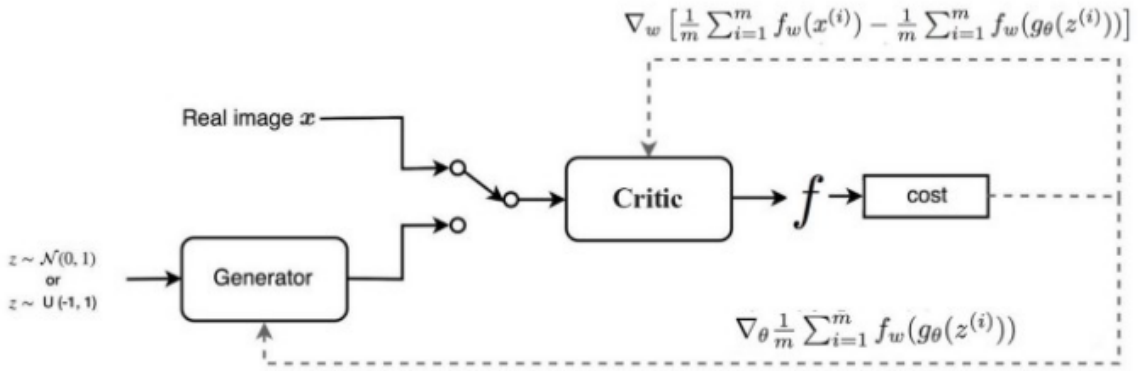

Fig 3.2.1 A WGAN architecture

The Wasserstein GAN is a novel approach to training generative adversarial networks. Training standard GANs are sometimes unstable and difficult, due to the usage of Jensen-Shannon divergence method, a method that measures the difference between two probability distributions.

The WGAN uses the Wasserstein distance to train a GAN, as it offers a continuous and differentiable loss function. As this distance does not saturate as the discriminator improves (discriminator can figure out generated images). Hence, the discriminator can further be trained to achieve optimal results from the generator, as it continuously tries to fool the discriminator by generating better 'fake' images.

During loss calculation, the Wasserstein distance offers a loss metric that directly correlates with the generated sample quality. The discriminator tries to maximize the Wassterstein distance(loss) by giving higher scores to real samples and lower scores to fake samples.The generator, on the other hand, tries to minimize the Wasserstein distance by generating samples that maximize the discriminator's output.

3.3 Data Extraction and Preprocessing

For our data, we used the Landscapes Dataset from Kaggle, which consisted of a diverse range of landscape images, from forests to beaches. The dataset contained about 4,000 images for different landscapes, and this was used to train our network to generate landscapes.

As for the preprocessing, the images were transformed into 64x64 pixels for faster training, converted into Pytorch Tensors to scale pixel values on a range of 0 to 1, and then normalized to have values ranging from -1 to 1. As our generator outputs images in the range of -1 to 1 due to having the tanh activation, the input image is normalized to such a range.

3.4 Model Description

The model used is a variant of GAN that employs deep convolutional layers for both the generator and the discriminator, known as a DCGAN (Deep convolutional GAN). It generates high quality 64x64 images based on a random noise vector as input.

The discriminator is a CNN that takes an input image (real or fake) and determines whether it is real(from the dataset) or fake(generated by the generator). In the context of WGAN, the

discriminator is referred to as critic, as it outputs a scalar representing how “real” the image is. The generator is a CNN that takes a random noise vector and transforms it to an image that resembles a real image.

The generator is a random noise vector which employs the convolutional layers to progressively increase spatial dimensions from 1x1 to 64x64 pixels. It uses batch normalization for stable learning and to avoid overfitting. The final image is output using a tanh activation function to scale pixel values to the range of -1 to 1.

The discriminator uses a sequence of convolutional layers with progressively increasing feature dimensions. It also uses a Leaky ReLU activation function for better gradient flow. It also utilizes normalization for improved performance. The discriminator reduces the spatial dimensions from 64x64 to a single value output of 1x1.

Weights are initialized from a normal distribution with mean 0 and standard deviation 0.02, and weights are initialized for the convolutional and batch normalization layers. The model also uses a gradient penalty, which is used to improve performance and stability in training. Instead of weight clipping to enforce the 1-Lipschitz constraint on the discriminator function (the gradient norm must be at most 1 everywhere), gradient penalty enforces the constraint by penalizing the discriminator whenever the norm exceeds 1, which further approximates the Wasserstein distance.

3.5 Model Parameters

GenerAI has several hyperparameters which help the model train properly. The lambda value for gradient penalty is set to 10, which controls the strength of the gradient penalty. The higher the value of lambda, the greater the emphasis on enforcing the 1-Lipschitz constraint.

There are 3 channel images as the images are RGB and not grayscale. Training is done with a batch size of 64, and feature extraction is done in both the networks in 64x64 pixels. The learning rate is set to 0.00004. The dimension for the noise vector(z-dim) is set as 100, and critic iterations are set to 5. In the model, critic is updated more frequently than the generator for better performance.

3.6 Training the WGAN model

The training process is divided into two main components, critic training and generator training. The critic is trained more frequently to approximate the Wasserstein distance more accurately.

In critic training, noise is sampled from z-dim and passed through the generator to create fake images. The critic evaluates both real and fake images, and produces scores for each. The Wasserstein loss is then calculated as:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]}_{\text{Our gradient penalty}}.$$

The expected values for the critic's score for real samples and fake samples are found, and the gradient penalty is found. The loss is backpropagated and the critic's parameters are updated using an Adam optimizer.

The generator is trained to minimize the critic's ability to distinguish fake from real. Noise is sampled and passed through the generator to produce images. The critic evaluates the images, and generator loss is calculated as the expected value from the critic. This loss is also backpropagated and parameters are updated. The images produced from the generator are saved as images in every 500 batches, and offers a visual checkpoint to evaluate the generator progress.

The gradient penalty function enforces the 1-Lipschitz continuity constraint on the critic. The penalty is calculated in three steps. First, the interpolation between real and fake samples is formed to create mixed samples. The gradient of the critic's output with respect to these interpolated samples is computed. Then, the norm of the gradient is found, and if it is anywhere from 1, the critic is penalized.

4. Performance Results

4.1 Loss of discriminator and generator

Our GenerAI model was first trained on landscapes, which was done on 64x64 pixels, mainly due to training times taking too long for higher resolution images. In a period of 100 epochs with a batch size of 64, the generator produced somewhat decent results from random noise vectors. The loss results for the generator(G) and the critic(D) are found as:

```
Epoch [49/50] Batch 30/68                   Loss D: -16.5165, loss G: 155.5403
Epoch [49/50] Batch 40/68                   Loss D: -14.6871, loss G: 138.6955
Epoch [49/50] Batch 50/68                   Loss D: -18.7397, loss G: 142.4044
Epoch [49/50] Batch 60/68                   Loss D: -16.6582, loss G: 139.3935
```

Fig 4.1 Loss results after training

As the input dataset contained a wide array of landscapes, with different features, we came to a conclusion that simply 100 epochs were not particularly enough to train a network to generate truly diverse landscape images. The loss for the generator was still not converging, but was getting closer to zero (convergence) slowly. Critic loss was increasing steadily, meaning the critic was finding it harder and harder to distinguish the generated images.

4.2 Image generation

As the models and training processes were done from scratch, checkpoints (saved models) were difficult to form. The generated images were hence saved during training itself. 25 images of 64x64 pixels were outputted from the batch, and the results were seen as:

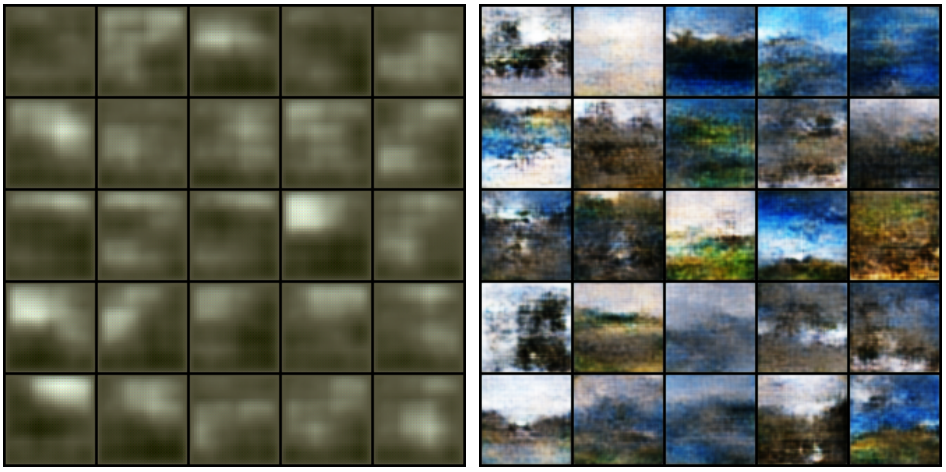

Fig 4.2.1 Generated results in epoch one(1) and epoch fifteen(15)

Fig 4.2.2 Generated results in epoch thirty(30) and epoch sixty(60)

Fig 4.2.2 Generated results in epoch ninety-nine (99)

4.3 Using CelebA dataset to validate model

Due to the landscapes dataset having a high degree of variance in landscapes (forests, beaches, lakes, deserts, cities), the WGAN model was having a hard time converging in 100 epochs. However, we tried to check whether a dataset with similar features will show better convergence in losses for our adversarial networks.

We implemented a celebA dataset, which contained portraits of many people. As a human face is a perfect example to test image generation on, particularly because the feature vectors are common (features of a face), we tested our WGAN model for 100 epochs, and found the results.

Fig 4.3.1 Generated images (top) and Input images (bottom) in 100 epochs

5. Conclusion

For "GenerAI: Photorealistic Landscape Generation," the main objective was to create a generative adversarial network (GAN) that could generate diverse and realistic landscape images. Because of the high variability in the landscape images, the model faced a number of difficulties during the 100 training epochs, which complicated feature extraction and model convergence. Despite these difficulties, the model was able to produce photorealistic landscapes, highlighting WGANs' potential for this use case.

The observed high discriminator (D) and generator (G) losses can be attributed to the diverse nature of the landscape dataset. This variability, however, also illustrates how difficult it is to create photorealistic landscapes, and that the model may be able to generate even more varied and realistic results with additional improvements. The WGAN model was made from scratch, and posed several other difficulties, such as creating proper convolutional, normalization and activation layers for both D and G, and also the entire training process.

In summary, even though the project was successful in using WGANs to generate landscapes, more work will be required to improve the model's performance, lessen training instability, and improve the quality of the images that are produced. Future developments will help to improve the model and increase its capacity to generate photorealistic landscapes. These developments will include better evaluation metrics, enhanced GAN architectures, and sophisticated data preprocessing.

5.1 Future Enhancements

- Explore Data Augmentation methods to generate a more diverse dataset for training.
- Implement better training metrics and hyperparameters for better convergence of cost function.
- Develop an interactive system where users can generate landscapes based on various prompts.
- Optimize the model for high-res landscape generation (1920x1080) by leveraging more computational resources.

References

- *Mayonaka88.(2023).Mayonaka88/realistic-terrain-generation-using-generative-adversarial-networks: Using DCGAN and CGAN to generate hyper-realistic terrain.(GitHub) https://github.com/Mayonaka88/Realistic-Terrain-Generation-Using-Generative-Adversarial-Networks*

- *Panagiotou, E., & Charou, E. (2020, October 13). Procedural 3D terrain generation using generative adversarial networks. arXiv.org. https://arxiv.org/abs/2010.06411*

- *Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014, June 10). Generative Adversarial Networks. arXiv.org. https://arxiv.org/abs/1406.2661*

- *Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017, December 25). Improved training of Wasserstein Gans. arXiv.org. https://arxiv.org/abs/1704.00028*

- *Arjovsky, M., Chintala, S., & Bottou, L. (2017, December 6). Wasserstein Gan. arXiv.org. https://arxiv.org/abs/1701.07875*

APPENDIX

This section contains figures of the user interface of the related projects that we studied.

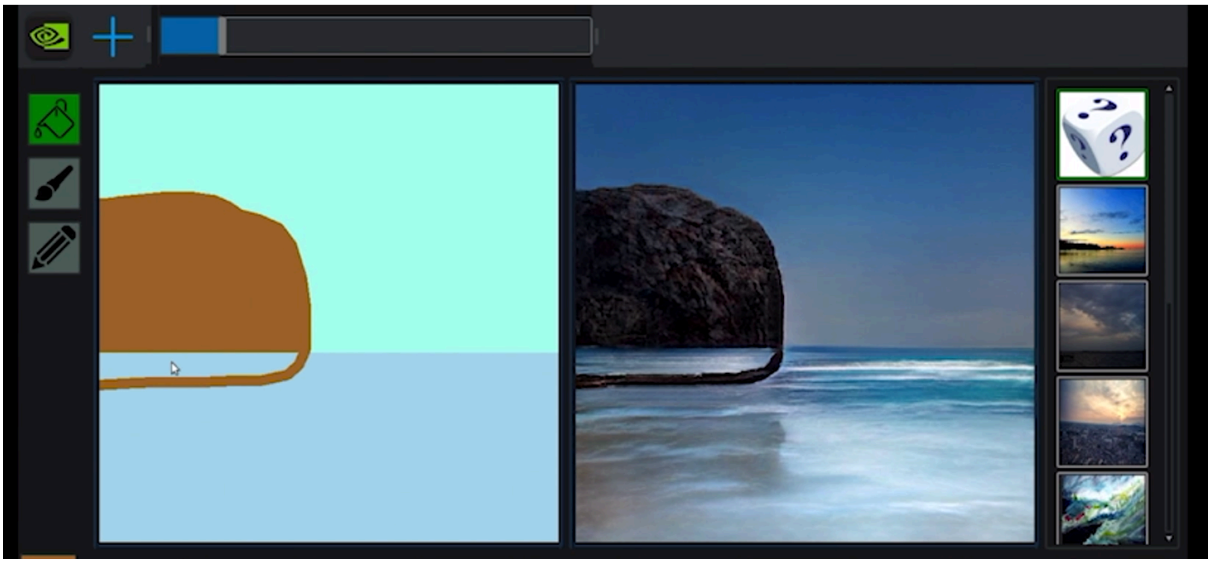

Fig A.1: NVIDIA GauGAN

Fig A.2: ArtBreeder

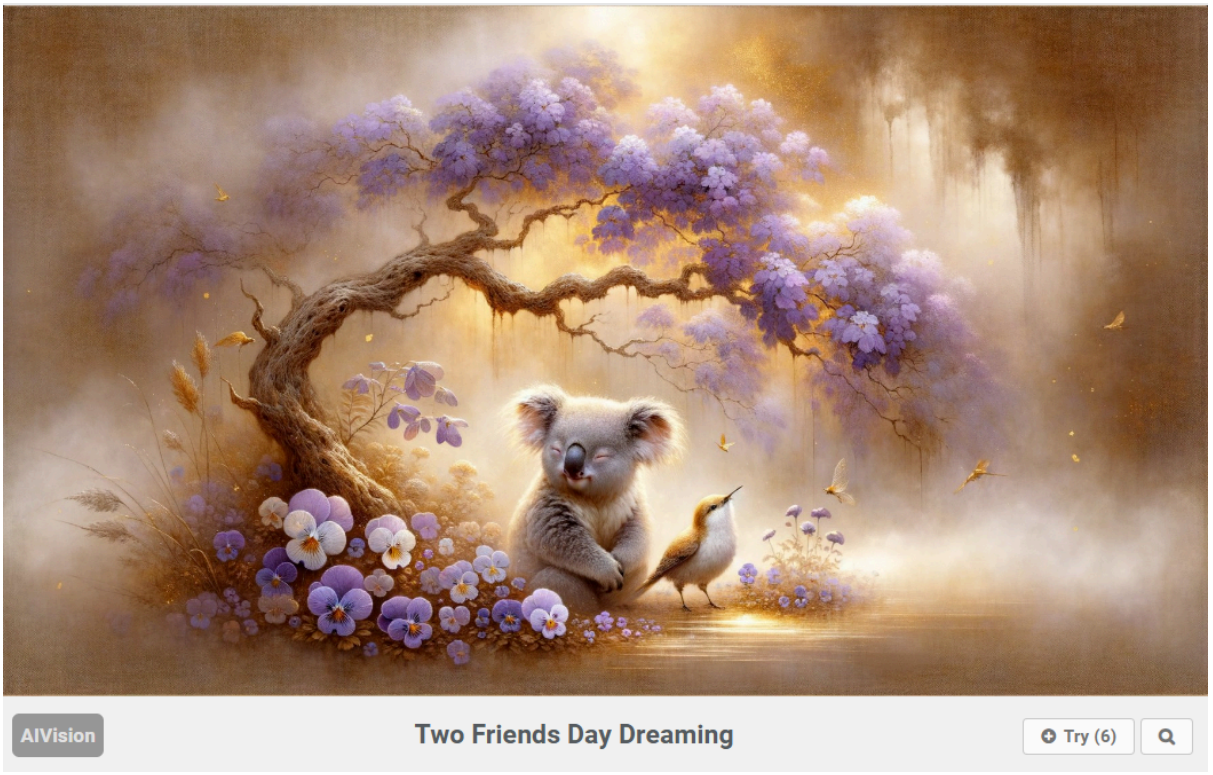

Fig A.3: Google DeepDream

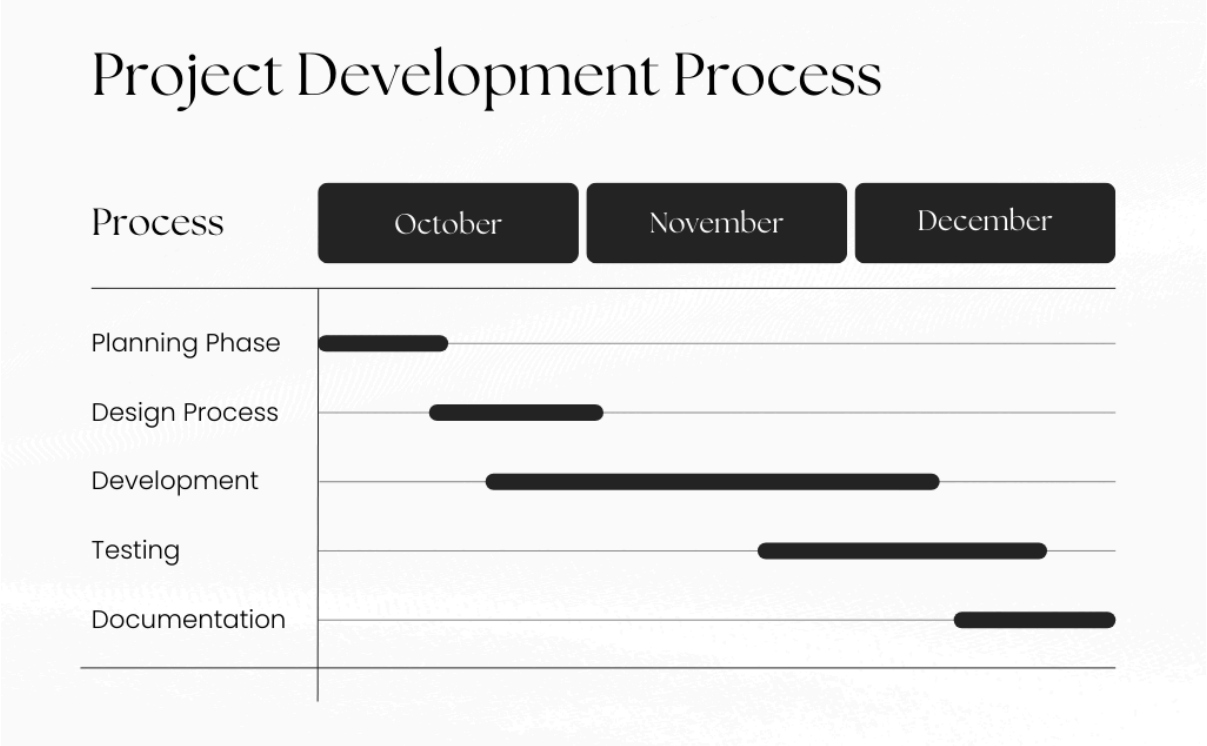

Fig A.4: GANTT chart