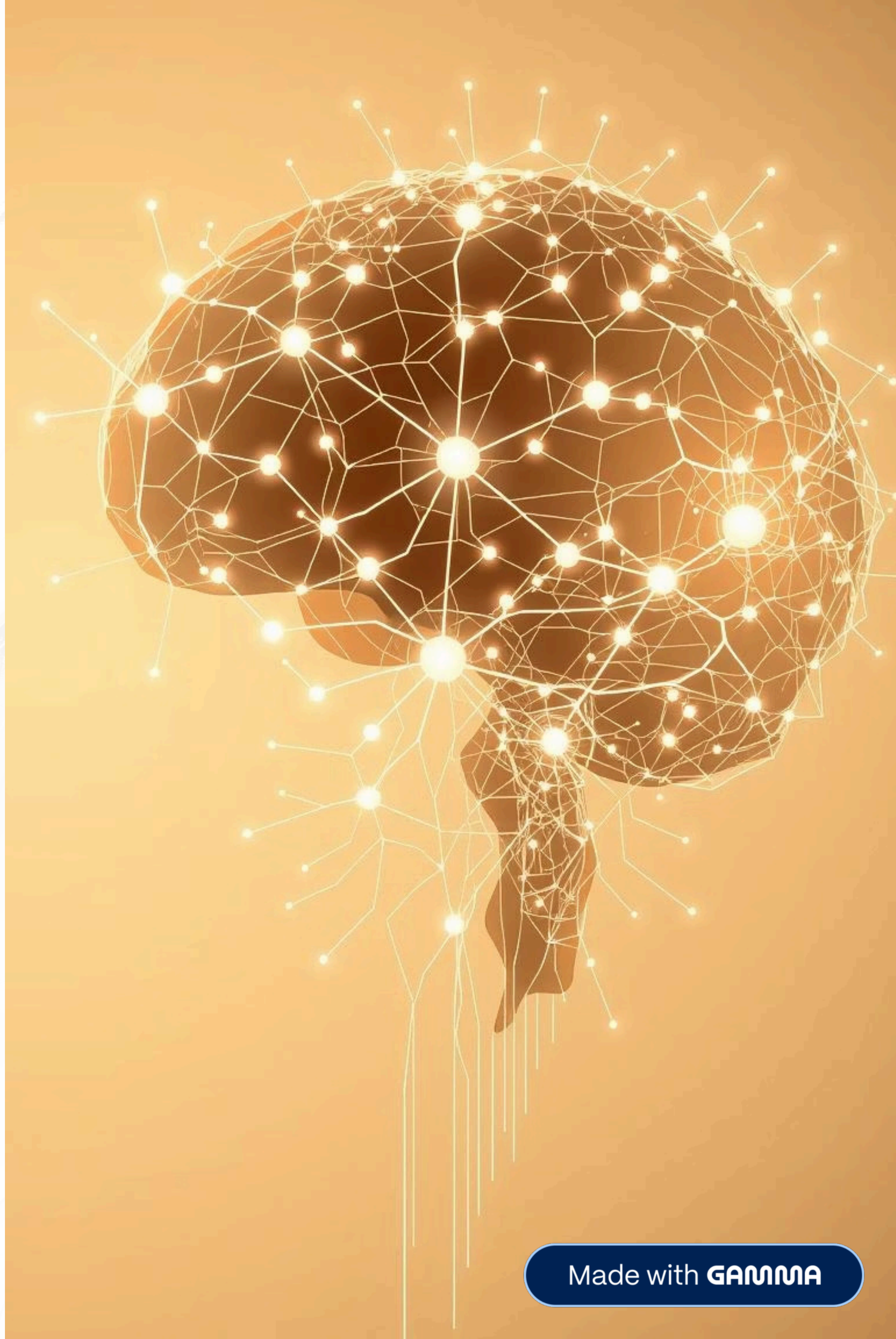
Generative Models for Data Augmentation – Internship Report

Presented by: Oumaima Bourmech

Company: CRNS

Supervisor: Madam Mouna Zouari

Date: July 1 to July 31



Plan Overview

1. Introduction & Problem Statement
2. Exploring GANs and DDPMs
3. Theoretical Models & Architectures
4. Dataset Utilization
5. Evaluation & Performance
6. Summarize & Conclude

1. Introduction & Problem Statement

1.Introduction & Problem Statement

Challenge: Class Imbalance & Small Data

Our datasets suffer from **class imbalance**, where some categories have very few samples. This makes models less accurate. Combined with **small data**, it leads to poor generalization, unstable training, and low-quality results.

The Solution: Data Augmentation with Generative Models

To address this, we explore **creating** new, representative data. Our goal is to use advanced generative models to rebalance datasets, thereby improving overall model performance.

Our Focus: GANs & DDPMs

This report delves into the implementation and application of **Generative Adversarial Networks (GANs)** and **Denoising Diffusion Probabilistic Models (DDPMs)** to create synthetic data effectively.

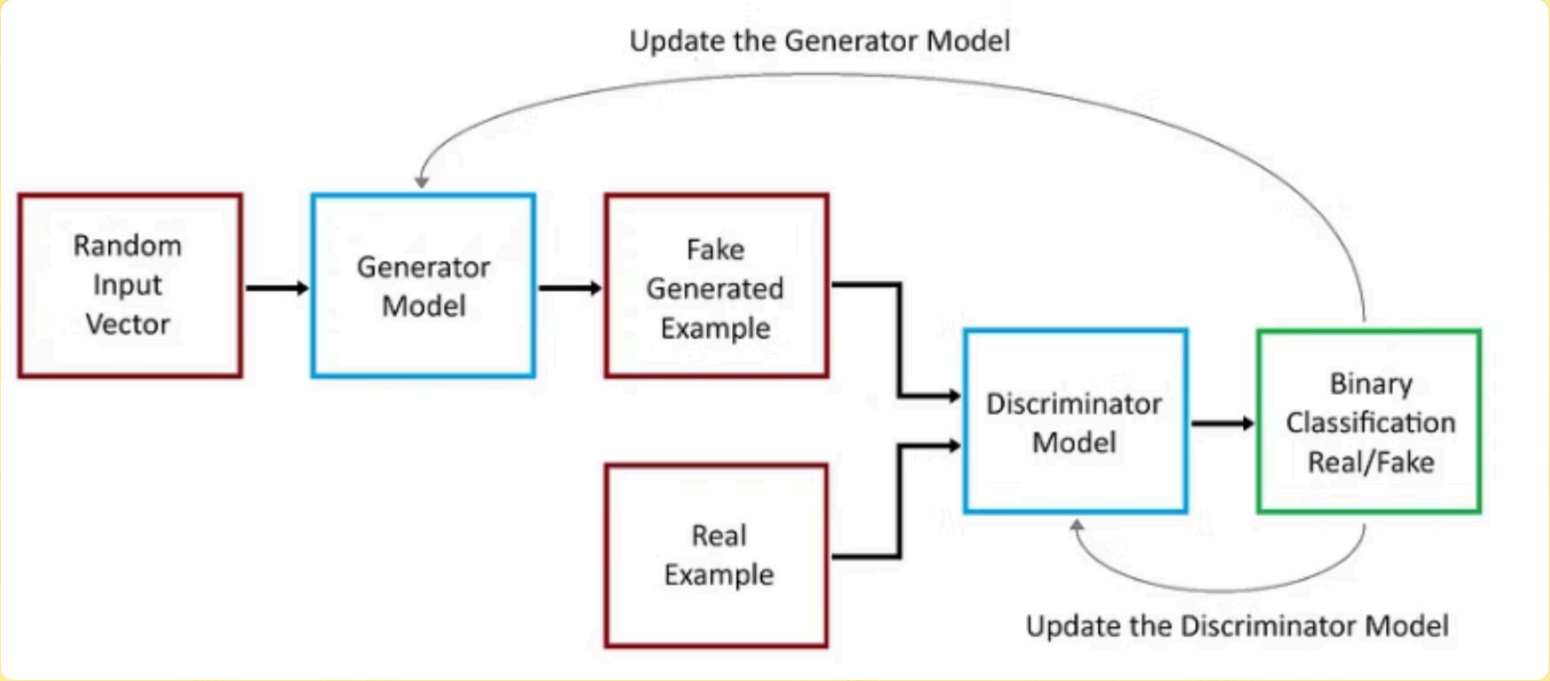
2.Exploring GANs and DDPMs

2.1.Generative Adversarial Networks (GANs)

A **Generative Adversarial Network** (GAN) operates on a principle of game theory, pitting two neural networks against each other in a dynamic competition:

- **Generator (G):** Tasked with creating synthetic data that mimics the distribution of real data.
- **Discriminator (D):** Aims to accurately tell the difference between real data samples and the synthetic data generated by G.

The training goal is a back-and-forth process: G tries to make fake data that looks real enough to trick D, while D keeps getting better at spotting fake data. This ongoing competition helps the Generator learn to create high-quality, realistic images.

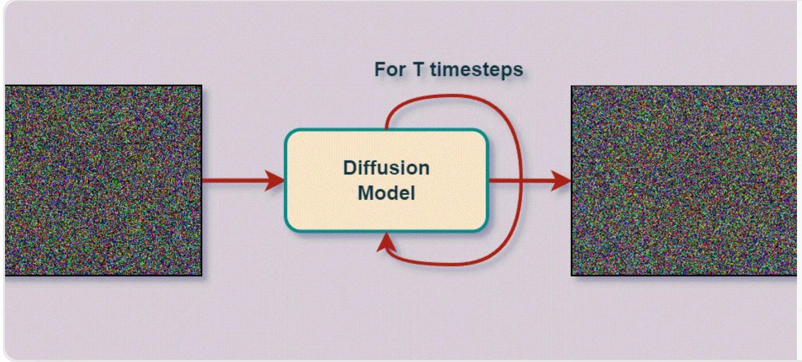


2.2.Denoising Diffusion Probabilistic Models (DDPMs)

DDPMs represent a distinct class of generative models that operate through a two-step process:

- **Forward Process:** Gradually adds Gaussian noise to an image over multiple steps, transforming it into pure noise.
- **Reverse Process:** Learns to systematically reverse this noise addition, effectively denoising the image step by step to reconstruct the original.

This methodology enables DDPMs to generate **high-fidelity, diverse images**. However, their primary limitations include **very slow training times** and a **high sensitivity to the quantity and quality of training data**.



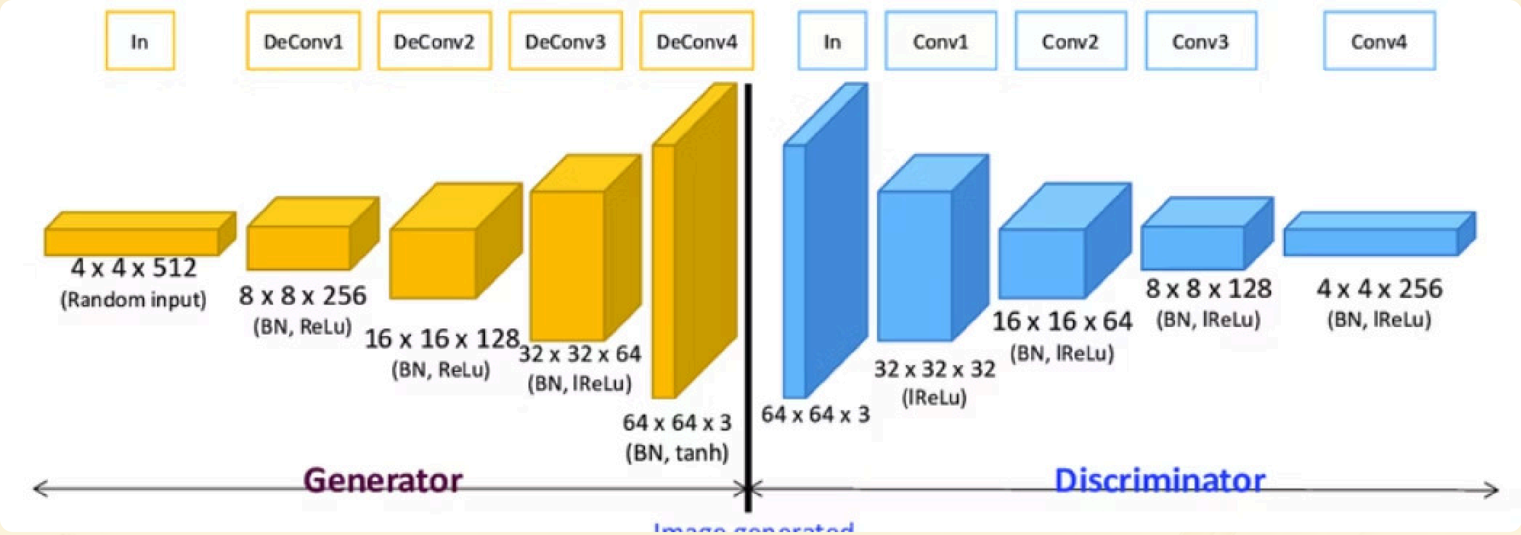
learnopencv.com

3.Theoretical Models & Architectures

3.1.GAN Variants & Architectures

1.DCGAN (Deep Convolutional GAN)

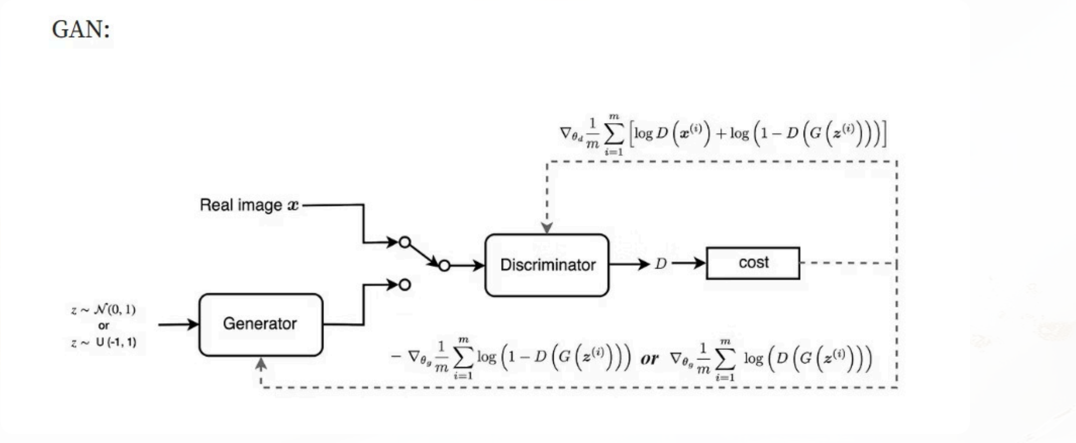
DCGAN is a GAN architecture that uses convolutional layers (especially ConvTranspose2D in the generator) and Batch Normalization to generate images. It was one of the first GANs to produce stable and visually coherent results, especially on simple datasets.

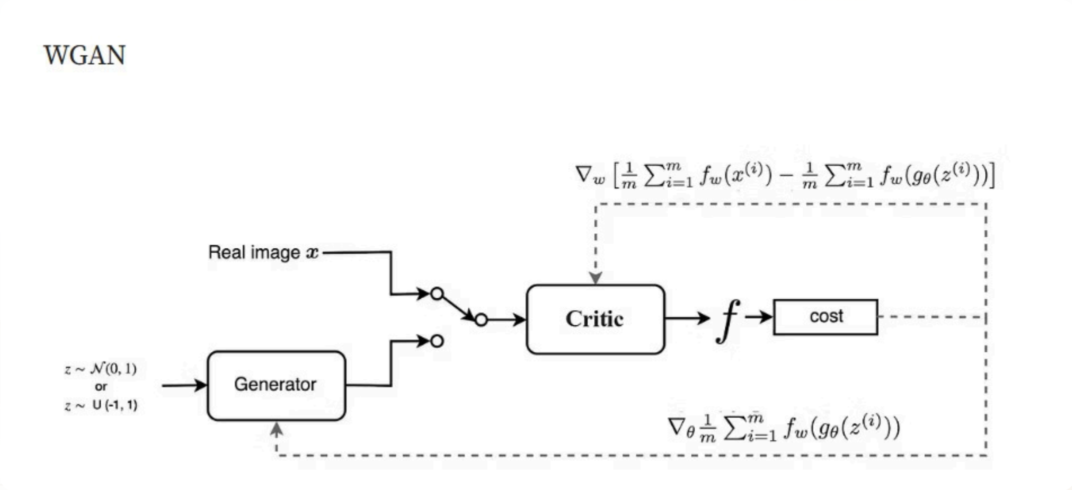


Made with GAMMA

2.WGAN (Wasserstein GAN)

Instead of the usual loss function, WGAN uses something called Wasserstein distance. This helps it train better and prevents it from getting stuck on just a few types of images. It needs a special step called "weight clipping" to work correctly.





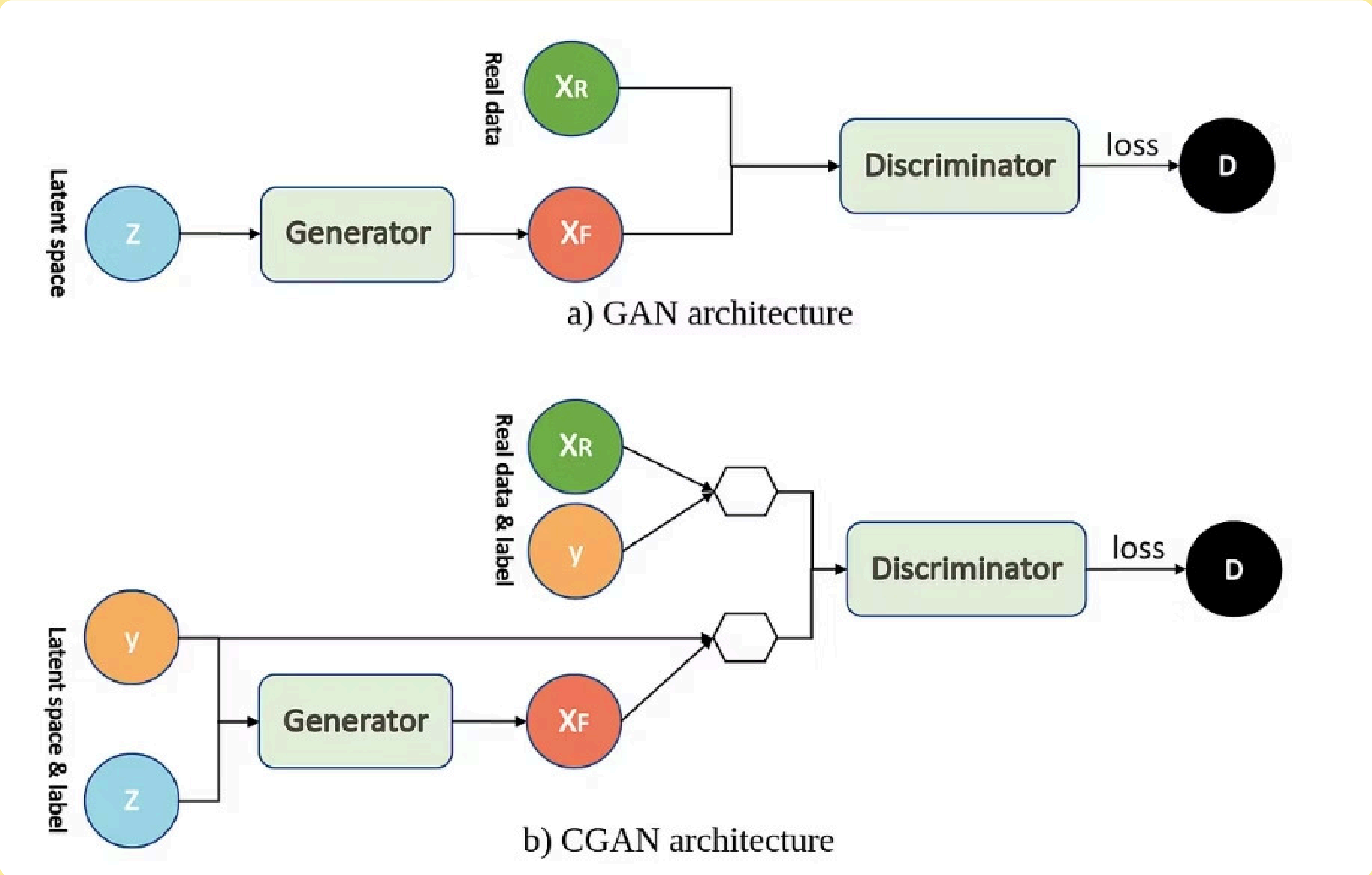
Made with GAMMA

3.WGAN-GP (Wasserstein GAN with Gradient Penalty)

This version improves WGAN by using a "gradient penalty" instead of weight clipping. This makes it more stable and less sensitive to how you set its training parameters.

4.CGAN (Conditional GAN)

CGANs let you control what kind of image they create. They do this by giving information (like class labels) to both the Generator and Discriminator, so you can tell them to generate specific things (e.g., only cats, or only dogs).



5.StyleGAN (Style-based Generator Architecture)

StyleGAN introduces a mapping network and adaptive instance normalization (AdaIN) to control different aspects of the image (like shape, texture, color). It also uses progressive training (starting from low to high resolution), which helps generate high-quality and detailed images.

3.2.Conditional DDPM for Controlled Generation

- **Conditional DDPM**

Conditional DDPM improves the denoising diffusion probabilistic model by including class labels directly in the image creation process. This is done by adding label information inside the UNet structure that controls the cleaning (denoising) steps.

The main advantage of this method is that it can create images that match specific classes with high quality, giving good control over the features of the generated images.

⚠ **Limitations:** Despite its capabilities, Conditional DDPM suffers from **slow convergence** and demands **massive, well-prepared datasets** for effective training. In our specific application to the Plastic dataset,Because it is quite small and unbalanced , the DDPM couldn’t learn properly.

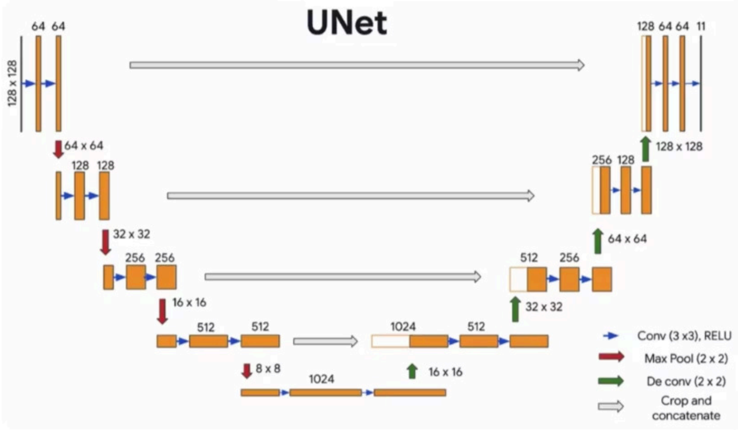


Figure: U-Net Architecture Overview

4.Dataset Utilization

Datasets Utilized for Benchmarking and Application

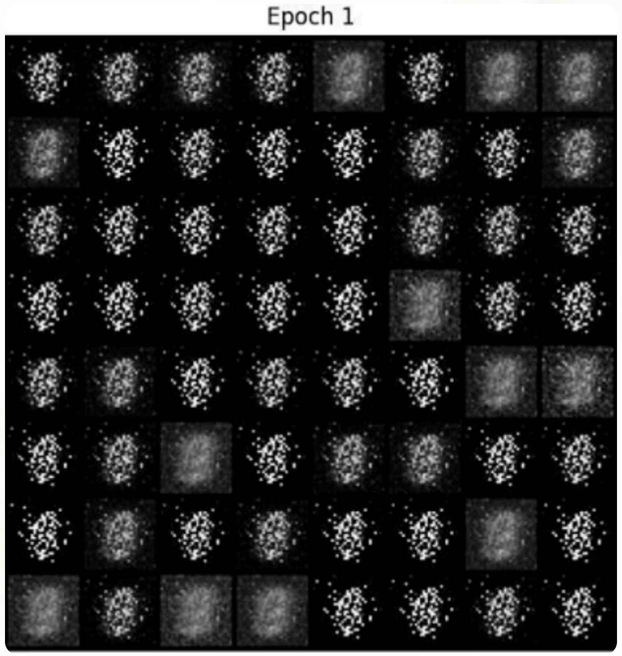
To carefully check how well our generative models work, we used several different**Benchmark** datasets for the first tests. This step-by-step process helped us confirm the model's performance in simple settings before using it for our real Dataset.

Dataset	Description
MNIST	28×28 grayscale images of handwritten digits.
Fashion-MNIST	28×28 grayscale images of various clothing items.
CIFAR-10	32×32 RGB images across 10 distinct categories (e.g., cars, animals).
Ours (Plastic)	Custom dataset of plastic-waste images, uniformly resized.

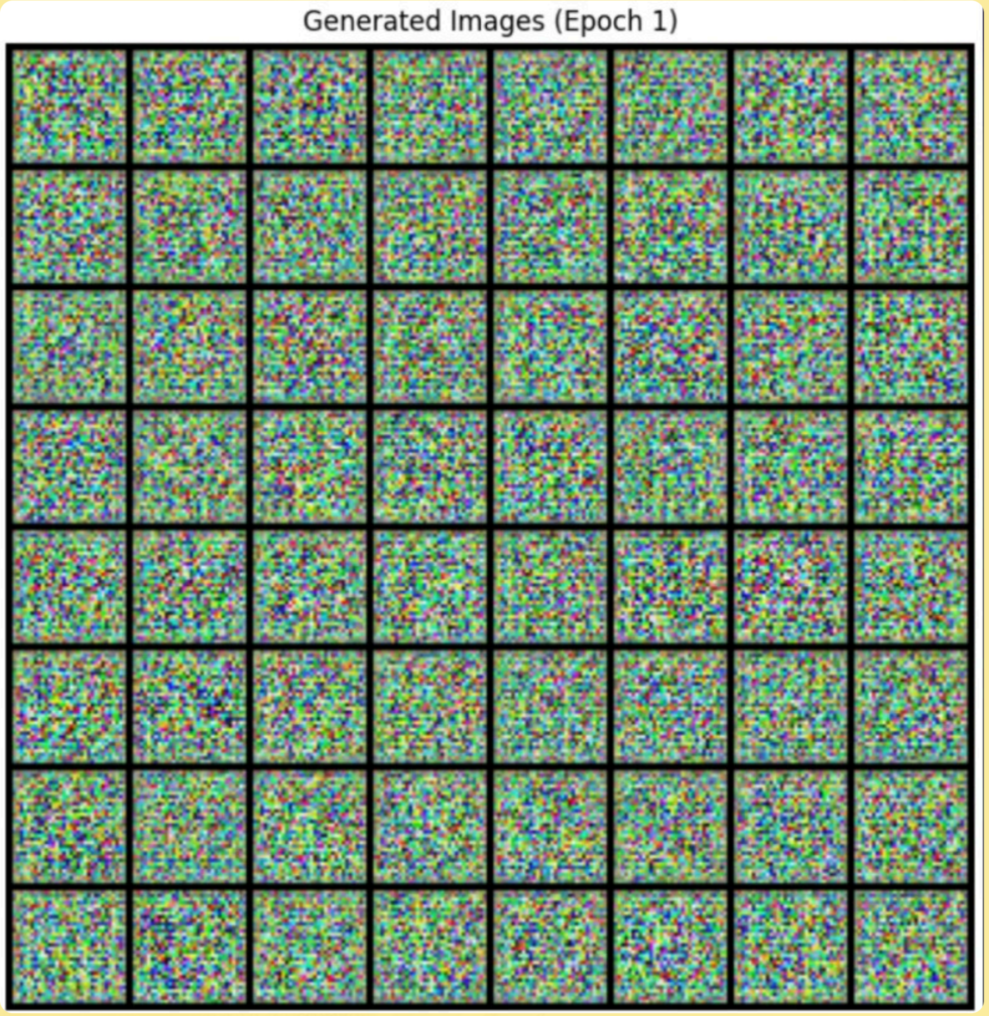
5. Evaluation & Performance

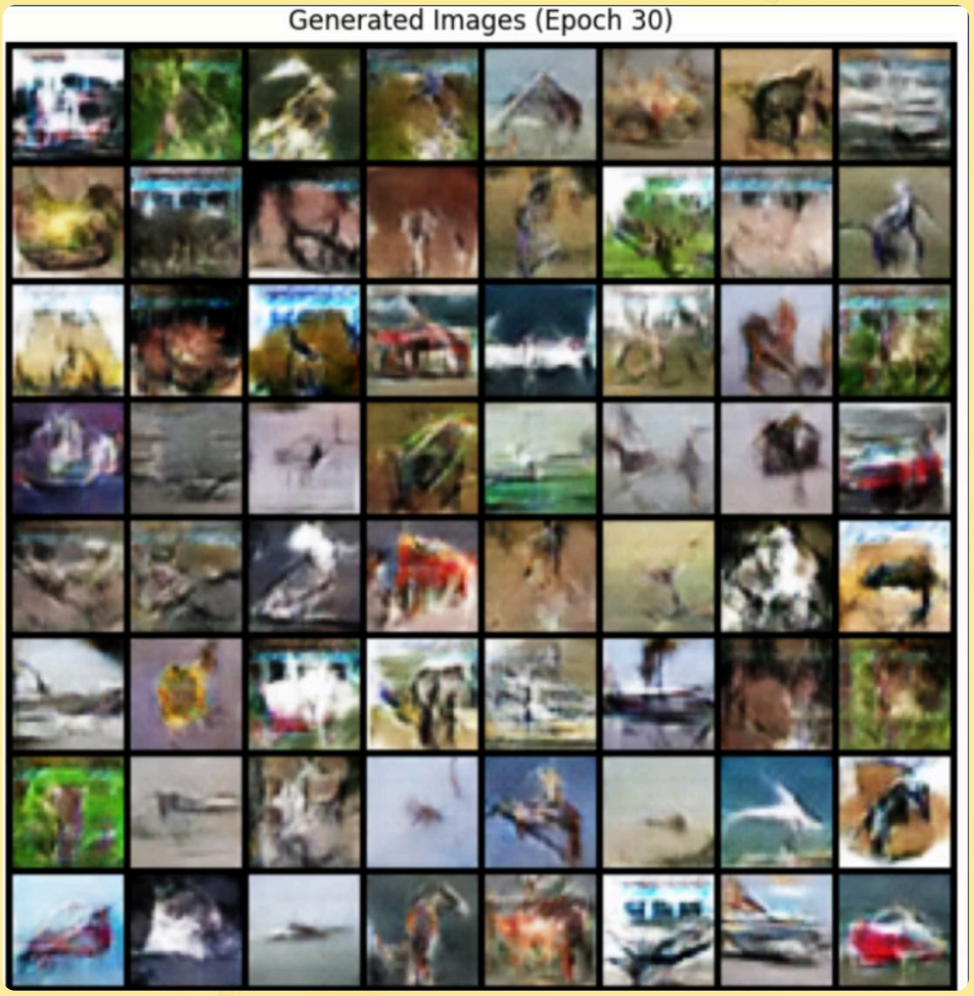
5.1.Evaluating on Benchmark Datasets

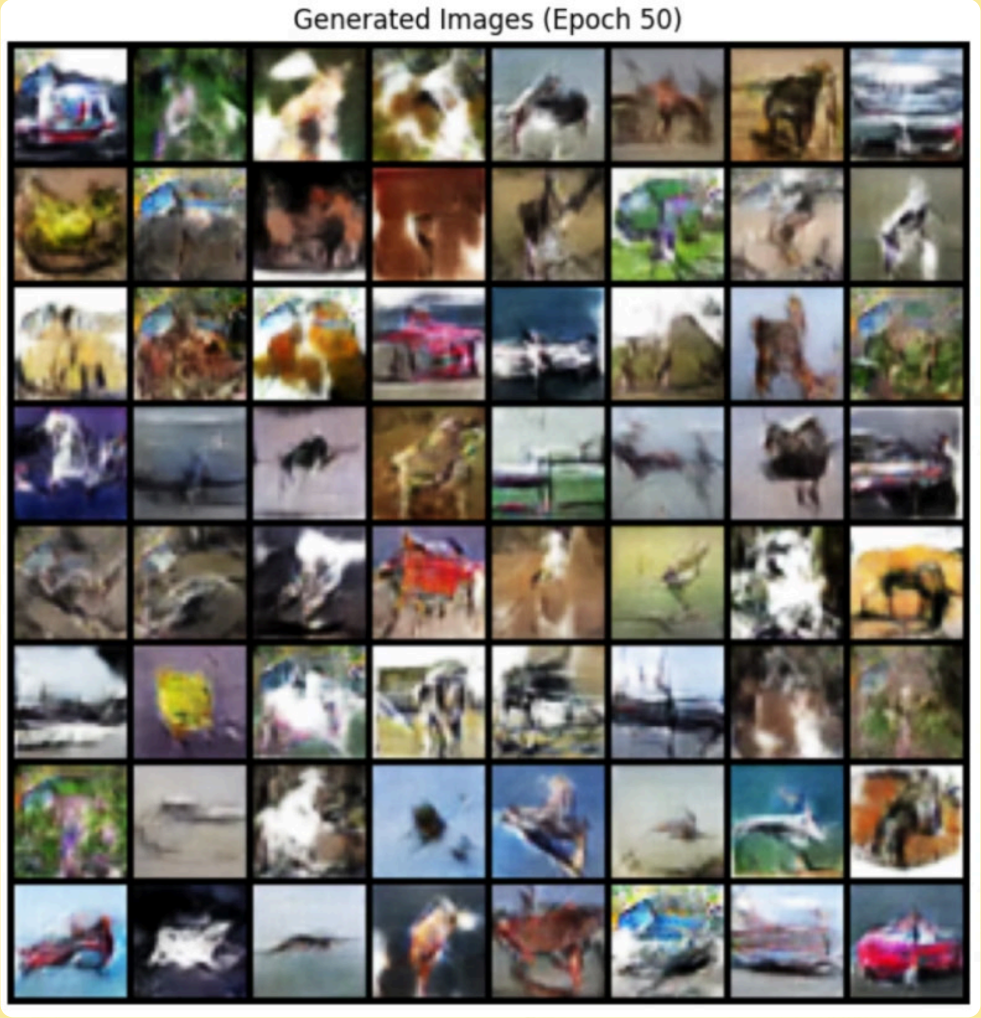
- **WGAN on MNIST (50 epochs)**: Generated digits with decent quality. Training was stable, but some images appeared blurry due to limited model complexity.



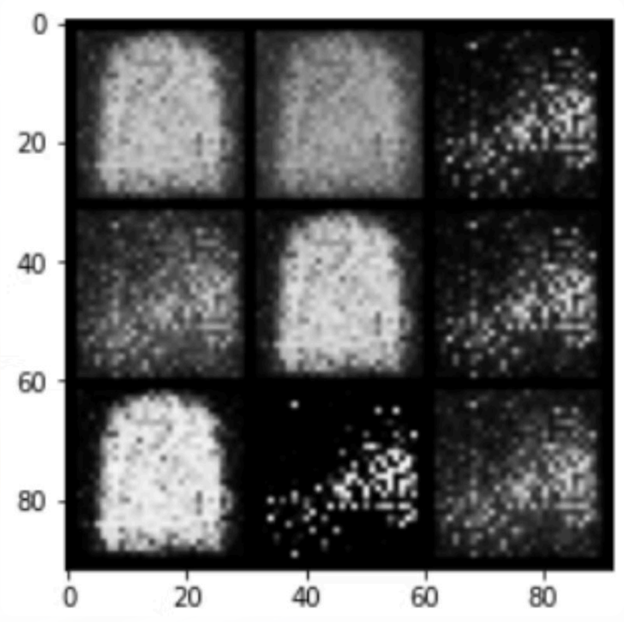
- **DCGAN on CIFAR-10 (50 epochs)**: Struggled with complex and colorful images. The training was unstable and outputs were low quality and incoherent.

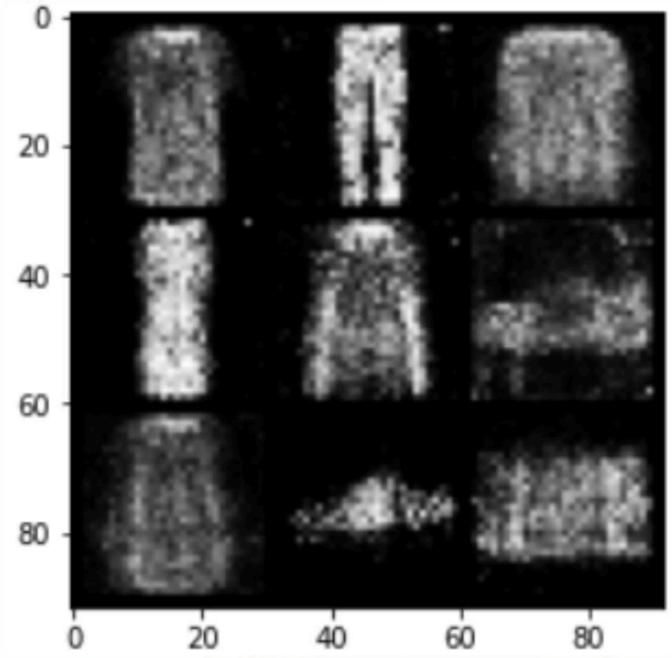


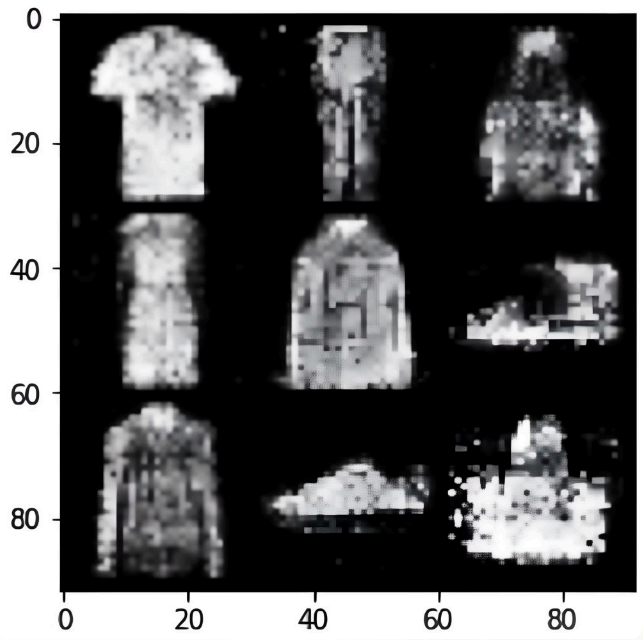




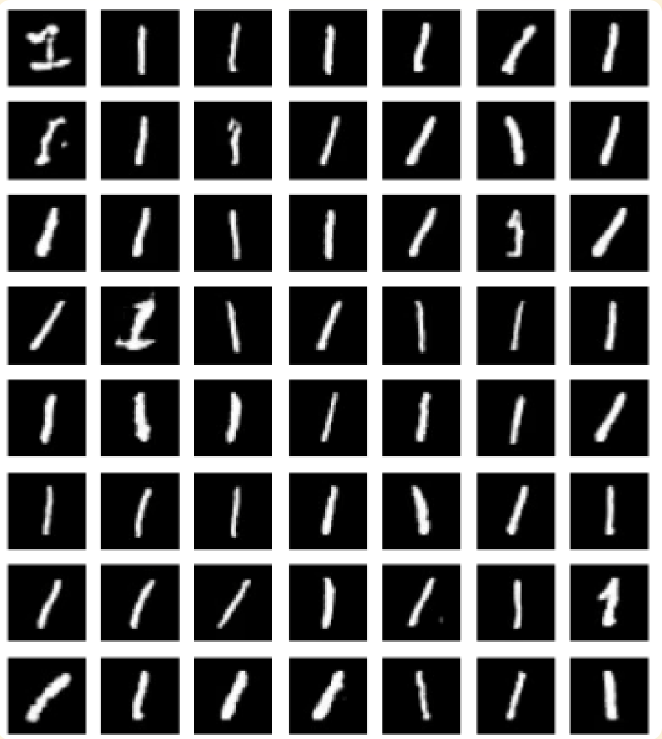
- **cGAN on Fashion-MNIST (50 epochs):** Generated labeled clothing items with reasonable accuracy. The images were recognizable but lacked sharpness.

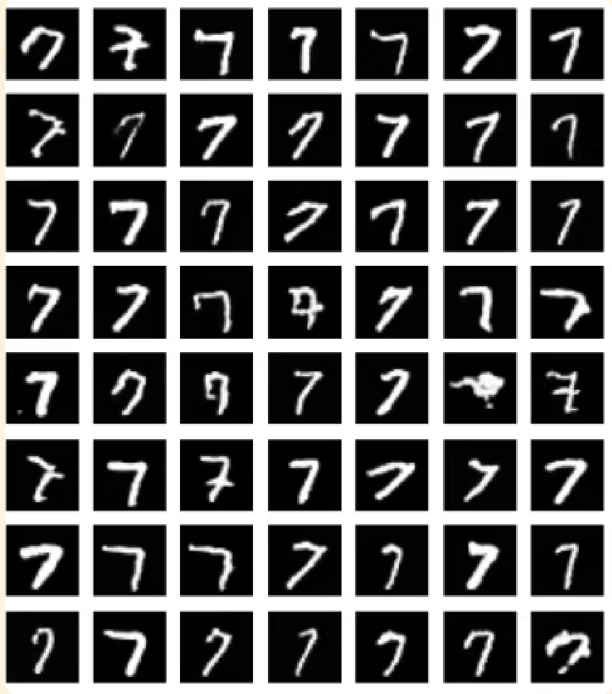


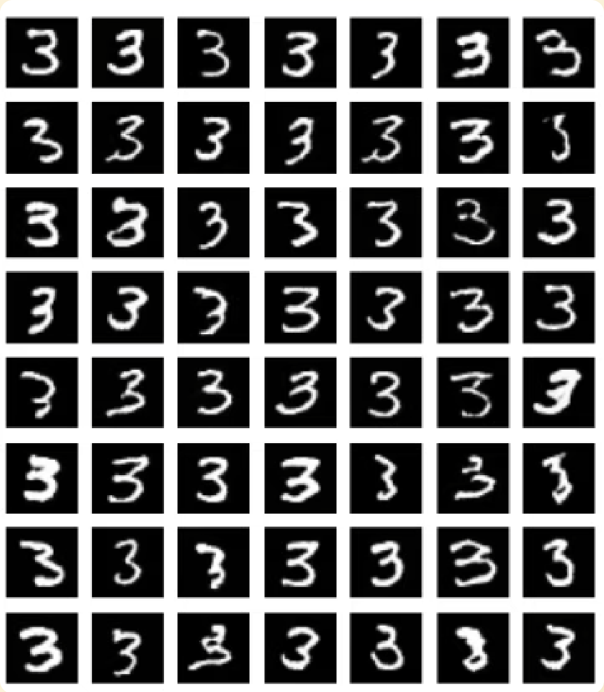




- **StyleGAN on MNIST (70 epochs):**Produced clear and realistic images of digits. The design is complicated, but the results looked very good.(exemple of final output for the classes "1" , "7" and "3")

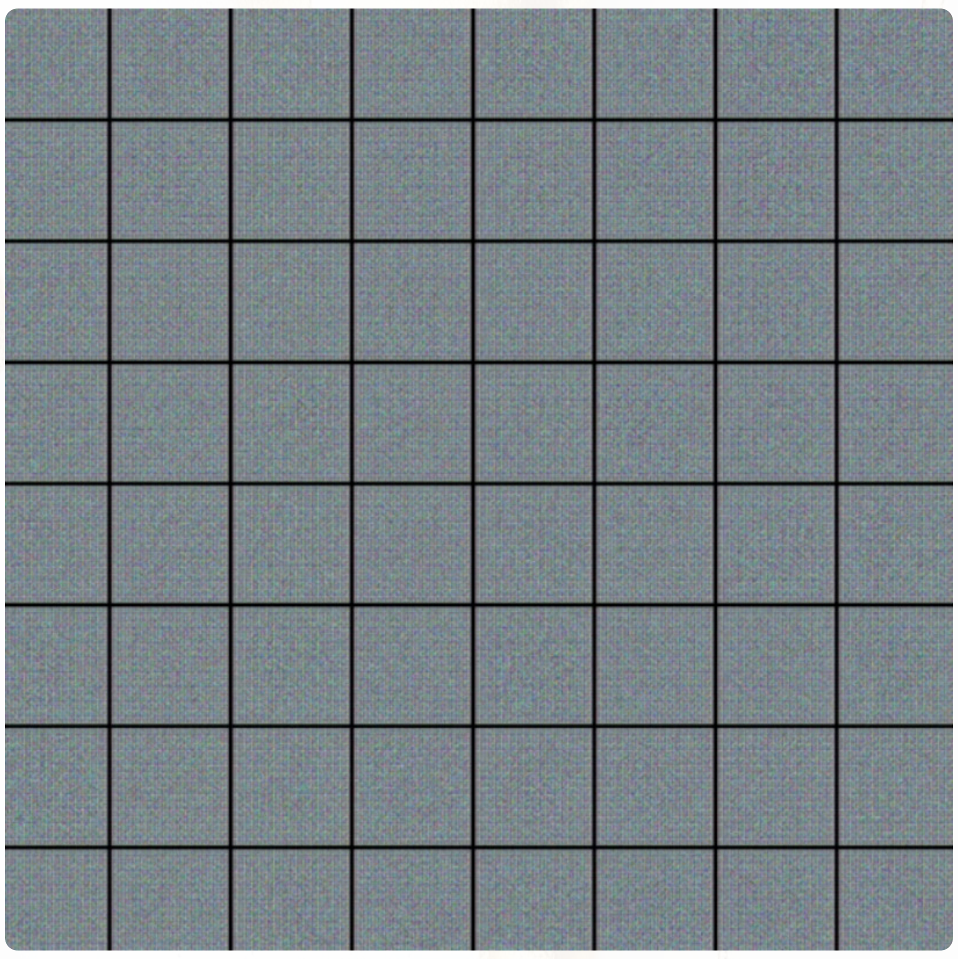


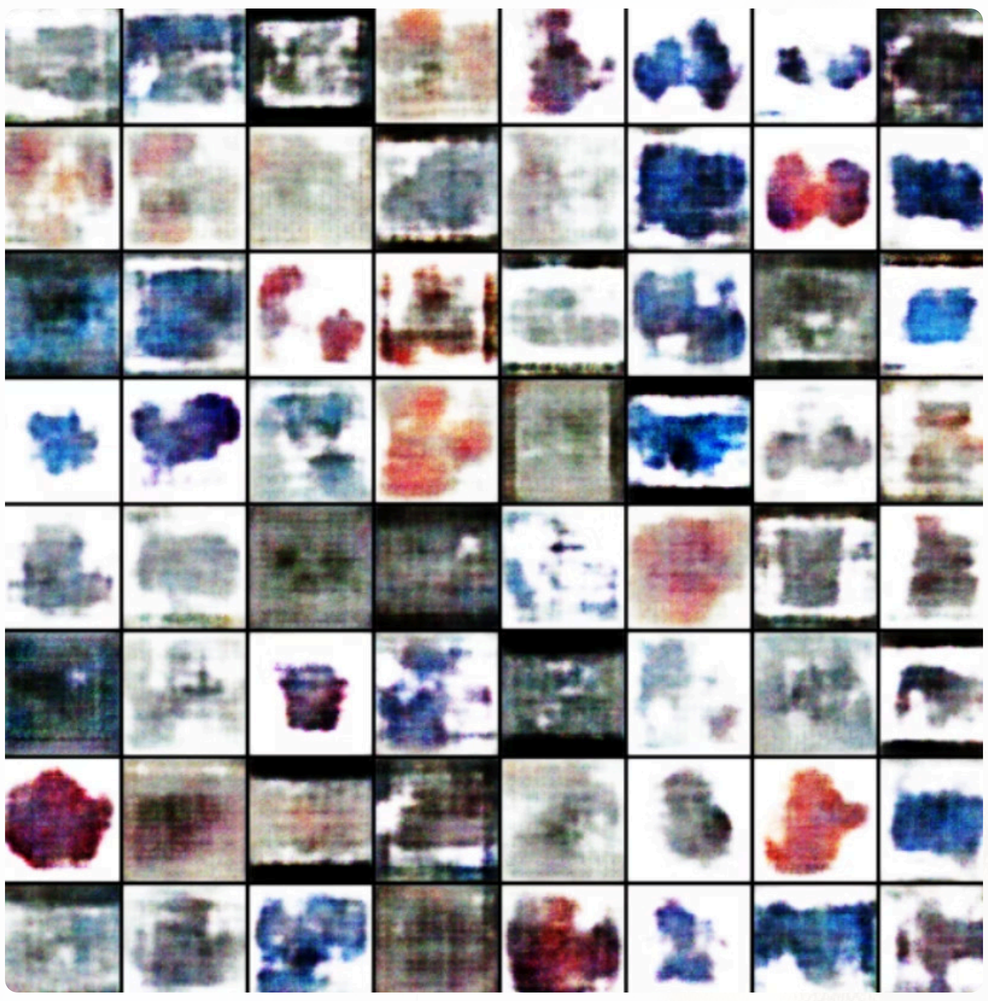




5.2.Evaluating on our Plastic Dataset

- **WGAN-GP on Plastic (PVC class) (4700 epochs)**: Stable training and good-quality outputs, but some images were still blurry.

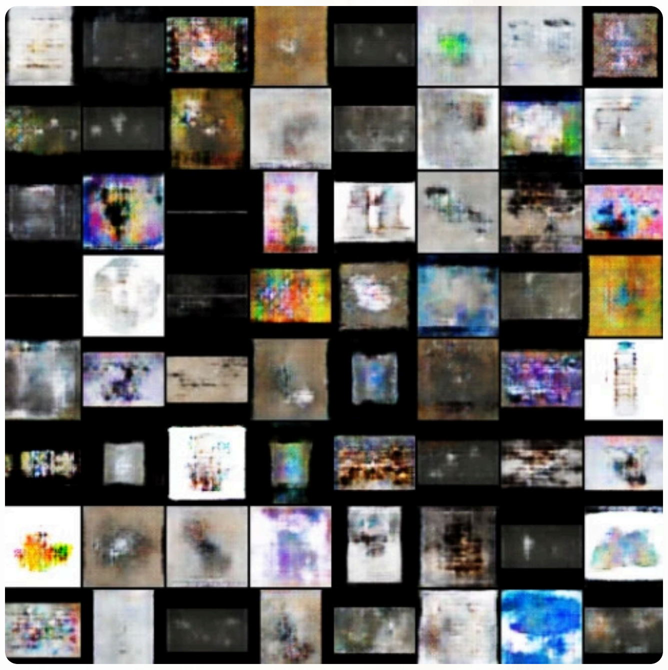




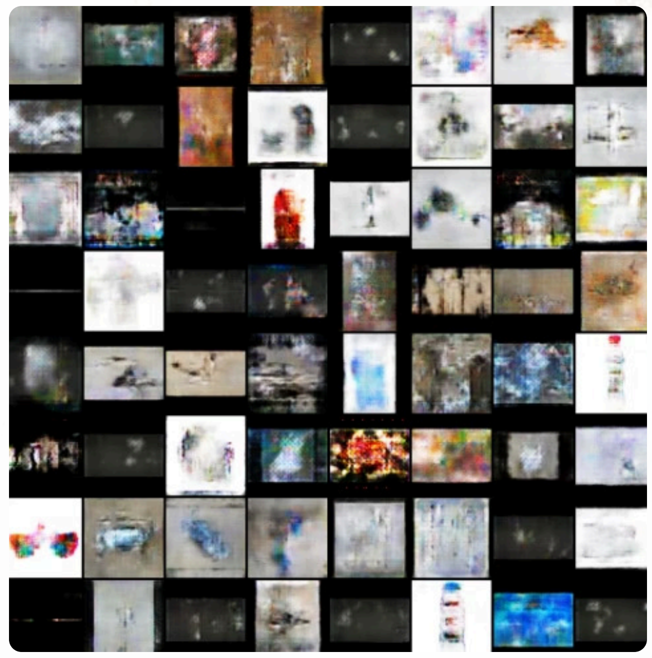
The following are selected examples of high-quality images produced by the WGAN_GP.

I continued training our pre-trained WGAN model on the entire dataset, and it produced these results. It still needs more epochs, but I couldn't continue due to time limits. However, I believe that with more epochs, it will generate better images, as it has already produced some good ones. starting with epoch (4701)

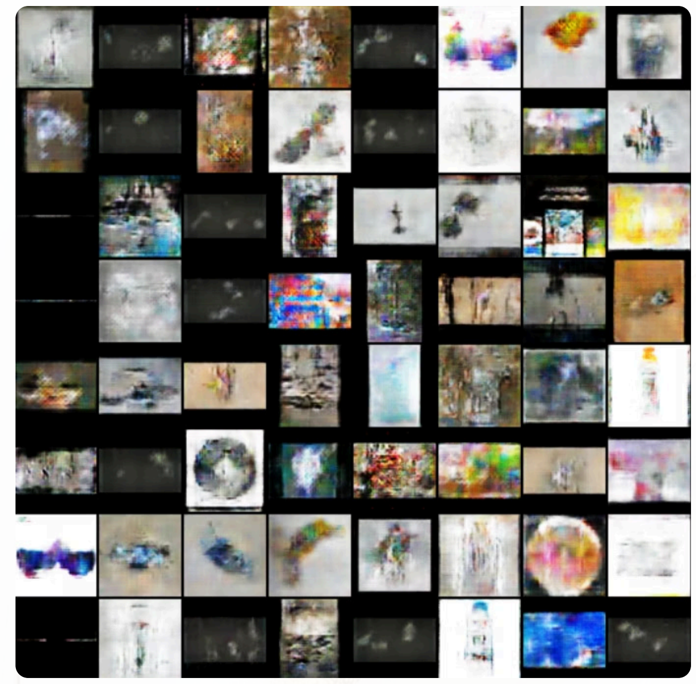
Epoch 4900



Epoch 5000



Epoch 5100



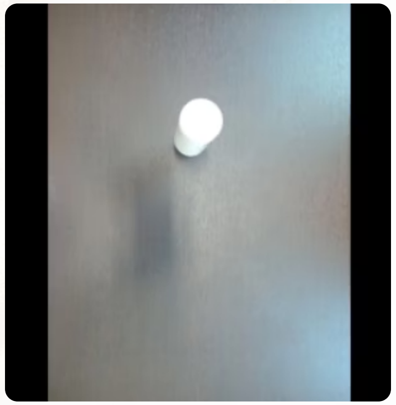
Here are some simple results that show the model's potential. It's already performing well.

also this is an exemple of data we had

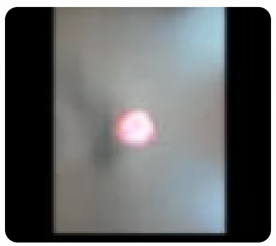
→ **thats why we are getting something like that**

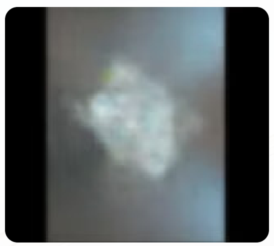
We also trained the model on **another** plastic dataset that we collected ourselves. It showed **good potential** and produced promising results. Below is an example of the original data and the corresponding output.

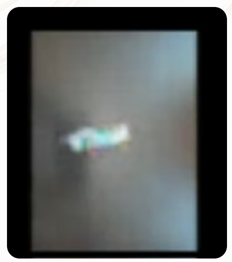
This is an exemple of the real data



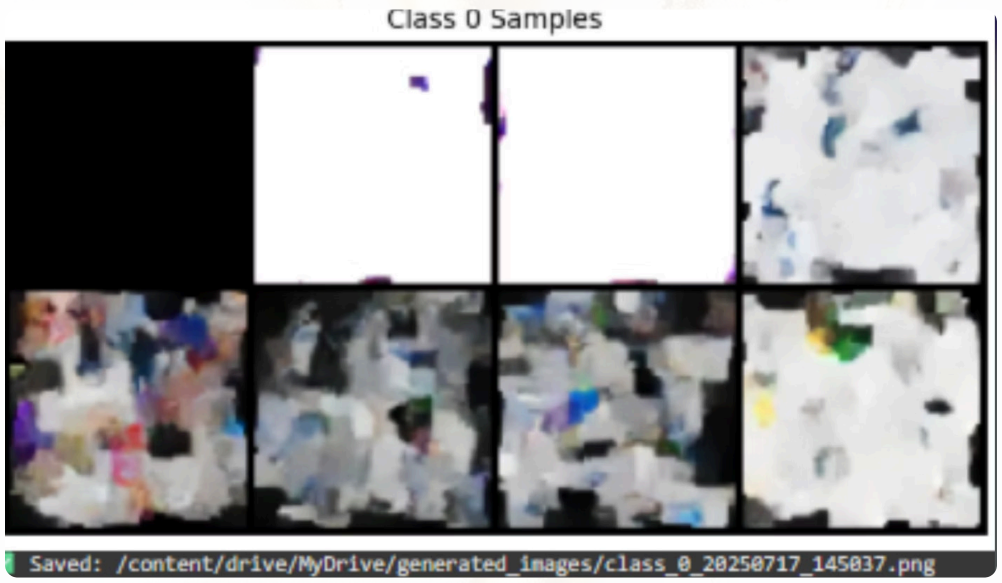
This was the output

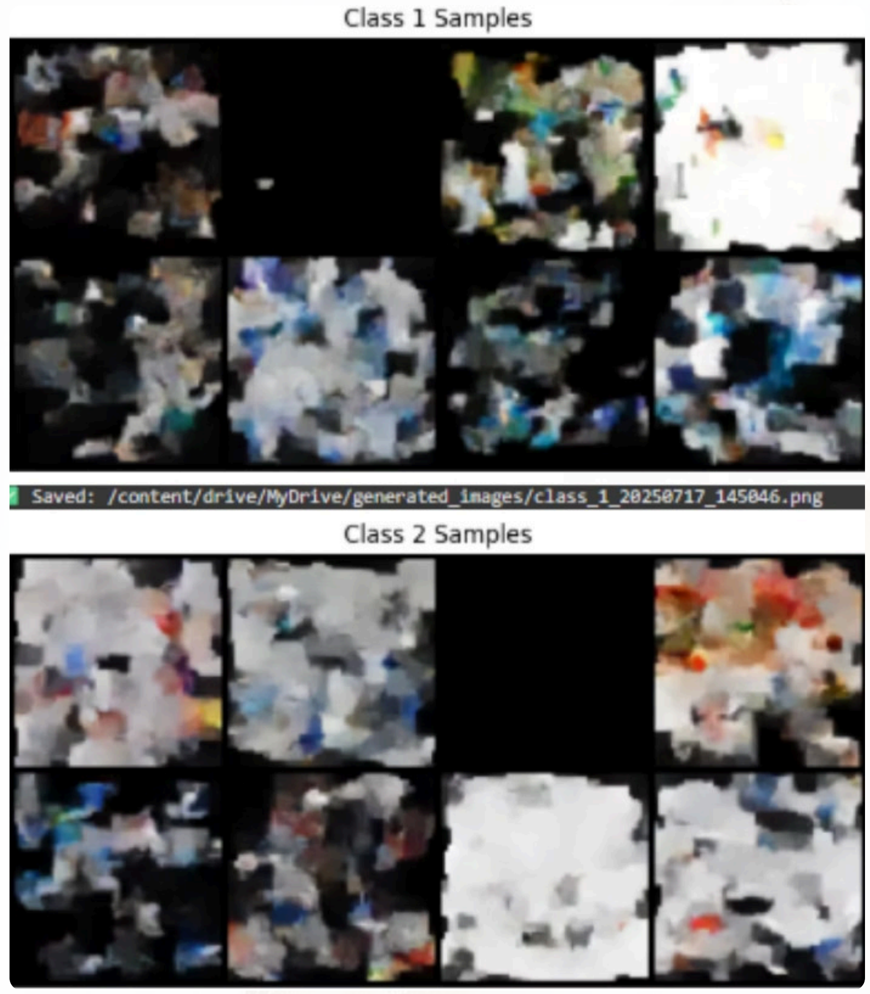


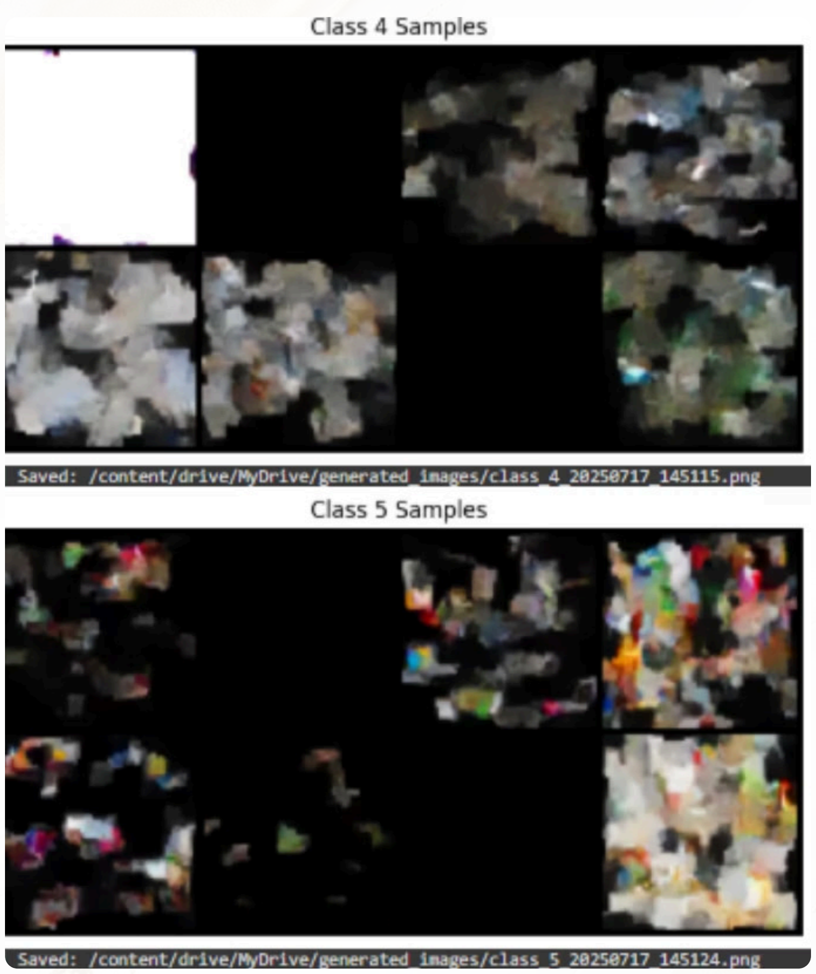


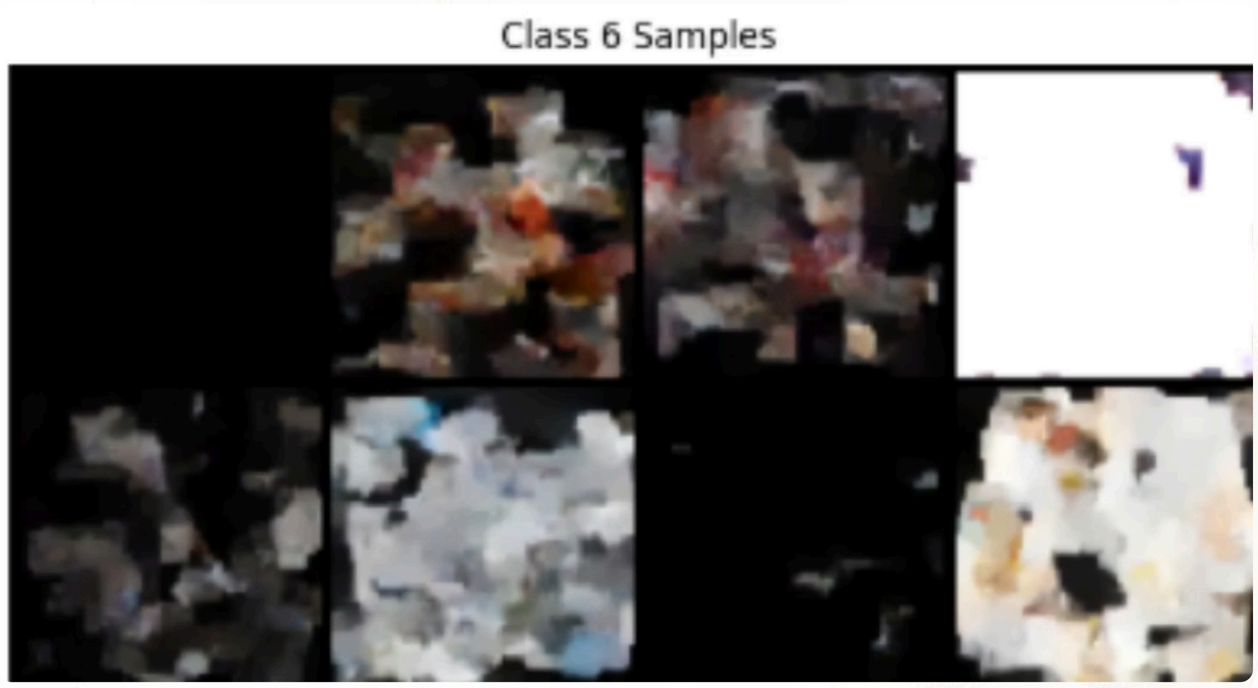


- **DDPM on our Plastic dataset (1200 epochs):** Performed poorly. The model failed to converge properly, likely due to limited and unbalanced training data.









Made with GAMMA

Summarize

Model	Dataset	Epoch	Visual Quality	Notes
WGAN	MNIST	50	Good	Stable but images could be blurry.
DCGAN	CIFAR-10	50	Poor	Performed poorly; unstable training and low-quality, incoherent images.
cGAN	Fashion-MINIST	50	Good	Goals realized, but images were less sharp.
StyleGAN	MNIST	70	Excellent	Complex architecture, but high fidelity.
WGAN-GP	A class from our plastic dataset (PVC)	5000	Good	also Stable but images could be a little bit blurry.
DDPM	Our Plastic data	1200	Poor	Did not converge effectively with limited data.

6.Conclusion & Future Directions

1

WGAN: Optimal Balance

WGAN demonstrated the most effective balance between generated image quality and computational training speed, making it highly practical for many applications.

2

Diffusion Models: High Potential

While Diffusion Models have great ability to create detailed images, they need much more training time and bigger, cleaner datasets to work well.

3

Benchmark vs. Custom Datasets

Our generative pipeline produced strong results on **benchmark, well-structured datasets** like MNIST and Fashion-MNIST. However, performance dropped on our **custom plastic dataset**, which was **quite** small, unstructured, and visually diverse (containing 8,365 varied images). This shows the challenge of training generative models when the data lacks visual consistency and domain **similarity**.

Note: WGAN and StyleGAN gave us the best-looking results and should get even better with more training time, better settings, and more data (especially for StyleGAN). DDPM still has some problems now, but it has a lot of potential and might do better than the other models if it gets enough training and data.

🧩 Final Reflection: But Wait... Isn't Our Goal Data Augmentation?

If we don't have much data to begin with...
How can we train models that usually need lots of it?

This paradox shaped our approach:

In our experiments, we saw that **diffusion models and StyleGAN**—two of the most powerful generative models—**require large, balanced datasets** to perform well.
But as stated in the introduction, **our goal was to augment and balance a small, imbalanced dataset** — not to assume one already exists.

👉 That's why i **chosed not to focus on those heavier models**, and instead prioritized:

- **WGAN-GP**, for its stability in low-data settings.
- **Class-wise training**, to reduce intra-class variation.

🚀 Future Work

DeiT (Data-efficient Image Transformer) – Facebook AI, 2021
A promising approach that performs well with fewer images using **strong augmentations** and **distillation**, making it ideal for low-resource scenarios like ours.

Thank You

For your attention and insights.

Questions?