**CHAPTER 1**

**INTRODUCTION**

Generative Adversarial Networks (GANs) have emerged as one of the most powerful and innovative architectures in the realm of generative modelling. Introduced by Ian Goodfellow et al. in 2014, GANs consist of two neural networks—the generator and the discriminator—that are trained simultaneously through adversarial learning. The generator attempts to produce data indistinguishable from real samples, while the discriminator aims to differentiate between real and fake inputs. This minimax game leads to the generator progressively improving its synthetic output until it can effectively fool the discriminator.

The significance of generative models, especially GANs, lies in their ability to learn complex data distributions without explicit supervision. This has transformative implications in fields such as medical imaging, synthetic data generation, anomaly detection, and the creative arts. For this report, the assignment is divided into two parts: Part 1 focuses on building and understanding GANs from scratch using simple 2D distributions, while Part 2 explores a real-world application of GANs using the PathMNIST dataset from the MedMNIST suite. This medical imaging dataset provides a practical challenge for evaluating how well GANs can learn to generate realistic pathology images.

The project not only aims to demonstrate an understanding of the underlying theory and implementation of GANs but also encourages critical analysis of model performance, training challenges, and architectural decisions. Through hands-on experimentation and methodical analysis, this case study sheds light on the practical strengths and limitations of GANs across both synthetic and real-world data distributions.

**CHAPTER 2**

**PART 1: GANS ON SYNTHETIC 2D DATA**

**2.1 Objective**

The objective of this section is to build foundational understanding by implementing a GAN from scratch on synthetic 2D data using PyTorch. This involves reproducing the tutorial GAN trained on a sine-wave distribution and then experimenting with a new 2D distribution—specifically, a spiral distribution. Additionally, architectural variations were introduced to explore their impact on the generator’s output quality.

**2.2 Reproducing the Sine Wave GAN**

To begin, a simple GAN was constructed to model a 2D sine wave distribution defined by the function:



A dataset of 1,000 points was sampled from this function with minor Gaussian noise. The generator network received 2D Gaussian noise as input and mapped it to 2D coordinates attempting to replicate the sine wave, while the discriminator classified whether the points were real (from the sine function) or fake (from the generator).

**Generator Architecture:**

* Input: 2D noise vector
* Fully connected layers: [2 → 64 → 64 → 2]
* Activation: ReLU in hidden layers, no activation on output

**Discriminator Architecture:**

* Input: 2D coordinates
* Fully connected layers: [2 → 64 → 64 → 1]
* Activation: LeakyReLU (α=0.2), Sigmoid on output

Both networks were optimized using the Adam optimizer with a learning rate of 0.001 and β₁ = 0.5. Binary Cross Entropy (BCE) loss was used as the adversarial loss.

**Results:**  
Over 2,000 epochs, the generator successfully learned to approximate the sine curve. Generated samples closely aligned with the real sine points in 2D space. The loss functions stabilized after 1,500 epochs, with the discriminator maintaining around 0.5 accuracy—an indication that it could no longer reliably distinguish fake from real samples.

**2.3 Modelling a Spiral Distribution**

To add complexity, the target distribution was changed to a **2D spiral**, defined parametrically as:



A total of 1,500 spiral data points were generated with small Gaussian noise added for variation.

**Model Changes:**

* Increased generator and discriminator depth to [2 → 128 → 128 → 2]
* Replaced ReLU with LeakyReLU in the generator for better gradient flow
* Introduced dropout in the discriminator to improve generalization

**Results:**  
Initial training produced scattered samples, but by epoch 3,000 the generator began forming a clear spiral pattern. Compared to the sine wave task, the spiral required more epochs to converge due to its nonlinear complexity.

A comparison between the original and modified GAN architectures revealed that deeper networks with LeakyReLU activations generated more consistent and continuous spiral patterns. Loss curves also showed smoother convergence, and the discriminator accuracy hovered around 50%, indicating balanced learning.

**CHAPTER 3**

**PART 2: MEDICAL IMAGING WITH PATHMNIST**

#### ****3.1 Dataset Exploration****

For the real-world application, the PathMNIST subset of the MedMNIST collection was chosen. This dataset consists of 107,180 RGB images of size 28×28 pixels across 9 classes. Each image represents a histopathology tile from H&E-stained human colon tissue, and each class corresponds to a specific tissue label ranging from normal to various disease states.

The dataset was split as follows:

* **Training set**: 89,996 images
* **Validation set**: 10,004 images
* **Test set**: 7,180 images

Each image has 3 channels (RGB), and the label distribution was observed to be slightly imbalanced. Class 4 and Class 6 appeared more frequently than others, which could potentially influence GAN training dynamics.

**Exploratory Visuals:**

* A grid of real sample images was generated, showing the diverse appearance of pathology tiles—some with dense cellular regions, others more uniform.
* A bar plot of class distribution confirmed mild imbalance, which was taken into account when training the conditional GAN (discussed later).

#### ****3.2 DCGAN Architecture Design****

The GAN architecture for this section was adapted from the **DCGAN (Deep Convolutional GAN)** framework, known for its effectiveness in image generation tasks. The model was implemented using PyTorch.

**Generator (G):**

* **Input**: Noise vector z ∈ ℝ¹⁰⁰
* **Layers**:
  + Linear layer → reshape to 512×4×4
  + ConvTranspose2d layers: [512 → 256 → 128 → 3]
  + BatchNorm2d after each layer (except last)
  + Activation: ReLU (intermediate), Tanh (final output)
* **Output**: 3×28×28 image with pixel values scaled to [-1, 1]

**Discriminator (D):**

* **Input**: 3×28×28 image
* **Layers**:
  + Conv2d layers: [3 → 64 → 128 → 256 → 1]
  + LeakyReLU activations (α = 0.2)
  + Sigmoid at the final layer
* **Output**: Probability of being a real image

**Training Setup:**

* Optimizer: Adam (lr=0.0002, β₁=0.5, β₂=0.999)
* Loss: Binary Cross Entropy (BCE)
* Epochs: 50
* Batch size: 128
* Normalization: Pixel values scaled to [-1, 1] for Tanh output

#### ****3.3 Training and Results****

Both generator and discriminator were trained alternately per batch. Loss curves were plotted throughout training.

**Loss Curves Analysis:**

* Discriminator loss initially decreased rapidly but later oscillated around 0.6, indicating a dynamic balance.
* Generator loss stabilized near 1.0, suggesting that the model was learning meaningful representations, although not perfectly indistinguishable from real data.

**Generated Image Samples:**

* At epoch 10, generated images appeared noisy and lacked structure.
* By epoch 30, structural features such as cellular clusters and stain patterns began to emerge.
* By epoch 50, many generated tiles showed class-like features (e.g., dense nuclei or lumen structures), although some images had artifacts or blurred regions.

A side-by-side comparison of real vs. generated images showed promising overlap in visual texture and color distribution, though sharpness and anatomical fidelity were not fully matched.

#### ****3.4 Quality and Interpretation****

While the model did not perfectly replicate all details, it succeeded in producing synthetic images that visually resembled real pathology slides in terms of:

* Color tone and staining patterns
* Cellular structures (in some cases)
* General composition and background texture

However, there were clear limitations:

* Fine-grained details (e.g., cell borders) were sometimes smoothed out
* A few generated samples exhibited checkerboard artifacts
* Diversity was not perfect—some classes were over-represented

These challenges are typical in GAN training and stem from the complexity of histopathological patterns, the relatively small image size (28×28), and class imbalance.

**3.5 Extension: Conditional GAN (cGAN) for Class-Conditional Pathology Image Generation**

To enhance control over the generation process and address class-specific variation, a Conditional GAN (cGAN) was implemented. Unlike the vanilla GAN where the generator samples from a noise prior alone, a cGAN conditions both the generator and discriminator on auxiliary information—in this case, the target class label—enabling the generator to produce class-specific synthetic pathology images on demand.

#### ****Model Design****

**Conditioning Mechanism:**

* **Class Input**: Each image belongs to one of 9 classes (labels 0–8).
* **Embedding**: Class labels were passed through an embedding layer to produce dense vectors of size 50.
* **Generator Input**: The noise vector z ∈ ℝ¹⁰⁰ was concatenated with the embedded label vector.
* **Discriminator Input**: The label was embedded and spatially expanded to match the image size, then concatenated with the image as an additional channel (making it 4×28×28).

**Generator Changes:**

* Input: [z + class\_embedding] → Linear layer
* Modified ConvTranspose2d layers retained from DCGAN
* Output: 3×28×28 image

**Discriminator Changes:**

* Input: [image + label\_map] → Convolutional layers
* Output: Real/fake score conditioned on class

#### ****Training Procedure****

* All 9 class labels were used during training, and batches were constructed to ensure each mini-batch had a diverse class distribution.
* The training loss used was BCE loss, unchanged from vanilla DCGAN.
* Generator was trained to maximize the probability that the discriminator classifies its output as real for the given class.
* Training ran for 60 epochs, slightly longer than vanilla GAN due to increased complexity.

#### ****Results****

**Class-Conditional Generation:**

* Images were generated per class by fixing the class label and sampling random noise.
* Visual grids were created to showcase synthetic outputs for all 9 classes.

**Visual Comparison:**

* Classes such as 0, 2, and 6 produced the most realistic and distinguishable features compared to their real counterparts.
* The generator successfully captured the global color tone and structural texture for each class.
* Lower-performing classes (e.g., Class 8) had less distinctive outputs, likely due to lower training representation or higher intra-class variability.

**Observations:**

* Conditioning improved stability and control in generation.
* Compared to vanilla GAN, the cGAN produced more coherent images for underrepresented classes.
* A visual T-SNE plot (optional) showed clustering of generated samples aligning with real class clusters, validating conditional consistency.

#### ****Limitations****

* The spatial label conditioning in the discriminator increased model complexity, which occasionally led to training instability.
* Label embeddings can introduce additional hyperparameters and require careful tuning for effective representation.
* While images showed improved class fidelity, some lacked high-resolution features needed for clinical realism.

#### ****Conclusion of Extension****

The implementation of a Conditional GAN significantly improved the generator’s ability to produce pathology images tailored to specific classes. It provided both a qualitative and interpretive advantage by enabling controlled image synthesis—an essential feature in practical applications like medical image augmentation and privacy-preserving synthetic data pipelines.

**CHAPTER 4**

**DISCUSSION AND FUTURE WORK**

**4.1 Summary of Results**

This project explored the construction and application of GANs, both in controlled 2D synthetic settings and in a real-world medical imaging scenario. In **Part 1**, the GAN effectively learned the sine wave distribution and demonstrated adaptability in modeling a complex 2D spiral after architectural refinements. In **Part 2**, a DCGAN was successfully trained on the PathMNIST dataset, generating pathology image tiles that visually resembled real examples. The implementation of a **Conditional GAN (cGAN)** further allowed for targeted image generation by class, improving visual fidelity and control.

The results revealed that:

* GANs are capable of learning and reproducing non-trivial 2D structures, though sensitive to architecture and training parameters.
* The PathMNIST-trained DCGAN generated realistic tissue patterns, especially for dominant classes.
* Conditioning the GAN on class labels provided additional control and boosted the coherence of class-specific outputs.

**4.2 Challenges Encountered**

Throughout the project, several challenges were observed:

1. **Training Stability**: GAN training is inherently unstable. Discriminator overpowering led to generator stagnation in early epochs.
2. **Checkerboard Artifacts**: Use of ConvTranspose2d layers in the generator resulted in periodic artifacts, especially in low-resolution outputs.
3. **Class Imbalance**: Less-represented classes in the dataset (e.g., Class 8) led to weaker conditional outputs, revealing the sensitivity of cGANs to label distribution.
4. **Mode Collapse**: Occasional mode collapse was observed when the generator produced similar images repeatedly across training epochs.

To address these, the model architectures were tweaked, loss trends were monitored closely, and batch balancing was implemented during cGAN training.

**4.3 Implications and Applications**

The ability to synthesize high-quality pathology images has real-world relevance in:

* **Medical Data Augmentation**: Expanding training datasets for diagnostic classification models.
* **Anonymization**: Creating privacy-preserving synthetic datasets.
* **Class Balancing**: Generating minority-class samples to balance classification tasks.

By generating diverse, class-specific image samples, cGANs could assist in building robust and fairer diagnostic tools, especially in resource-constrained domains where labeled medical data is limited.

**4.4 Suggestions for Future Work**

Several improvements and future directions are proposed to extend this work:

1. **Use of Advanced GAN Variants**:
   * **Wasserstein GAN (WGAN-GP)** to improve training stability.
   * **StyleGAN2** or **Pix2Pix** for higher resolution image synthesis.
2. **Evaluation Metrics**:
   * Incorporate quantitative metrics such as **FID (Fréchet Inception Distance)** or **SSIM (Structural Similarity Index)** to objectively compare real vs. fake image distributions.
3. **Data Preprocessing**:
   * Use stain normalization techniques or contrast enhancement to improve data quality.
   * Upsample images to 64×64 or higher using super-resolution methods before generation.
4. **Domain Expert Evaluation**:
   * Involve medical professionals in assessing the visual and clinical quality of generated images.
5. **Latent Space Interpolation and Manipulation**:
   * Investigate how interpolating between latent vectors affects the generated pathology tiles to understand feature representation.
6. **Multi-class Conditioning or Attention**:
   * Incorporate attention mechanisms or class-conditional batch normalization to refine image features.

**CHAPTER 5**

**CONCLUSION**

This case study provided an in-depth exploration of generative adversarial networks (GANs), from foundational theory to real-world application. The initial phase using synthetic 2D data allowed for hands-on experimentation and structural introspection, establishing a clear understanding of GAN training dynamics. It demonstrated the sensitivity of GAN performance to architecture choices and the value of thoughtful activation and depth modifications.

The second phase, centered on medical imaging, validated the utility of GANs in high-impact domains. The DCGAN model trained on the PathMNIST dataset successfully produced visually plausible pathology image tiles. The introduction of a conditional GAN (cGAN) brought enhanced control over class-specific generation, addressing one of the key challenges in clinical machine learning—data imbalance and interpretability.

Despite the progress, the work also highlighted core limitations of GANs, such as training instability, limited diversity in underrepresented classes, and the challenges of evaluating synthetic data quality without domain expertise. Future directions such as architectural refinements, metric-driven evaluation, and expert validation will be critical in translating these capabilities into robust tools for medical data science.

Ultimately, this project has reaffirmed GANs as a powerful yet intricate tool within the generative modelling landscape, offering both creative and technical avenues for innovation across industries—from synthetic biology to digital arts to cybersecurity.

**CHAPTER 6**

**REFERENCES**

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). *Generative adversarial nets*. In Advances in neural information processing systems (pp. 2672–2680).
2. Radford, A., Metz, L., & Chintala, S. (2016). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.
3. Yang, J., Shi, R., Wei, D., & Sun, H. (2021). *MedMNIST v2: A large-scale lightweight benchmark for 2D and 3D biomedical image classification*. Scientific Data, 8, Article 111. <https://doi.org/10.1038/s41597-021-00977-z>
4. Isola, P., Zhu, J.Y., Zhou, T., & Efros, A.A. (2017). *Image-to-image translation with conditional adversarial networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125–1134).
5. Odena, A., Olah, C., & Shlens, J. (2017). *Conditional image synthesis with auxiliary classifier GANs*. In Proceedings of the 34th International Conference on Machine Learning.
6. MedMNIST Dataset. (2023). Retrieved from <https://medmnist.com/>
7. PyTorch Documentation. (2023). https://pytorch.org/docs/stable/index.html