BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE-DATA SCIENCE

A Mini-Project Report On

"GENERATION OF APPAREL IMAGES USING CNN-DRIVEN DEEP CONVOLUTION GAN"

A report submitted in partial fulfillment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

Submitted By

MOOLI SRUTHI **USN: 3BR22CD038**

Under the Guidance of

Mr. Azhar Biag

Asst. Professor

Dept of CSE (DATA SCIENCE), BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka 2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NBA and NAAC Accredited Institution

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi) "Jnana Gangotri" Campus, No.873/2,

Ballari-Hosapete Road, Allipur, Ballari-583 104 (Karnataka) (India) Ph:08392 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF

CSE-DATA SCIENCE

CERTIFICATE

Certified that the mini project work entitled **"Generation of Apparel Images Using CNN-Driven Deep Convolutional GAN Estimation"** carried out by **MOOLI SRITHI** bearing **3BR22CD038** bonafide students of Ballari Institute of Technology and Management, of CSE-DATA SCIENCE Department during the year ~~2024- 20~~25, Have submitted the following report for the fulfillment of the requirements Neural Network and Deep Learning Laboratory. The work reported here is original and is carried out under our supervision.

Signature of [redacted] Co-Ordinator's

Mr. AZHAR BAIG

Ms. CHAITHRA B M

Signature of HOD

Dr. ARADHANAN D

ABSTRACT

This Project aims to develop a system capable of generating new, realistic fashion images by utilizing a CNN-based Deep Convolutional Generative Adversarial Network (DCGAN) trained on the Fashion-MNIST dataset. The network consists of two core components: a Generator, which creates synthetic images from random noise vectors, and a Discriminator, which evaluates these images to determine whether they are real or artificially produced. Both networks are constructed using Convolutional Neural Network (CNN) layers to capture spatial structures, textures, and visual patterns that define various fashion apparel categories such as shirts, trousers, dresses, footwear, and accessories. Through the adversarial training process, the Generator gradually improves its ability to produce visually convincing images, while the Discriminator becomes increasingly precise at identifying subtle distinctions between real and generated samples. By the end of training, the model can create synthetic Fashion-MNIST images that closely resemble real dataset examples, showcasing the ability of GANs to learn data distributions without explicit labeling. This project highlights the effectiveness of CNN -based DCGAN frameworks in image synthesis tasks and suggests potential applications in computational fashion design, automated visual content generation, dataset expansion for training deep learning models, and creative design prototyping in industries related to fashion and digital art.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of project work on "**Generation of Apparel Images Using CNN-Driven Deep Convolutional GAN"** would be incomplete without mentioning those who made it possible. Their noble gestures, affection, guidance, encouragement, and support crowned our efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of this project.

We are extremely grateful to our Lab Coordinators, **Mr. AZHAR BAIG** Teaching Assistants, AIML, BITM, Ballari and **Dr. ARADHANA D,** Professor and Head of the Department of CSE-DS, BITM, Ballari, for their noble gestures, support, coordination, and valuable suggestions in completing the project work. We also thank the Principal, Management, and non-teaching staff of BITM, Ballari, for their coordination and valuable suggestions in completing the project work.

Name	**USN**
MOOLI SRUTHI	3BR22CD038

TABLE OF CONTENTS

Sl No.	Content	Page No
	Abstract	I
	Acknowledgement	II
	Table of Contents	III
	List of Figures	IV
1	Introduction	1
2	Objectives	2
3	Problem Statement	3
4	Methodology	4
5	Requirement Analysis	5
6	Design	6
7	Implementation	9
8	Results and Discussion	11
	Conclusion	12
	Reference	13

LIST OF FIGURES

Sl No.	Content	Page No
1	Flow Chart	6
2	Use case Diagrams	7
3	Sequence Diagram	8
4	Epoch 1	11
5	Epoch 5	11
6	Epoch 10	11
7	Epoch 15	11
8	Epoch 20	11

CHAPTER 1

INTRODUCTION

Generative Adversarial Networks (GANs) have emerged as powerful deep learning models capable of generating new data samples that closely resemble real data. Introduced by Ian Goodfellow in 2014, GANs consist of two neural networks, the Generator and the Discriminator, which compete with each other in a zero-sum learning process. The Generator attempts to create synthetic outputs that appear realistic, while the Discriminator evaluates these outputs and determines whether they are real or generated. Through this adversarial training process, both networks improve simultaneously, resulting in the production of highly realistic synthetic data. Among the different variants of GANs, the Deep Convolutional Generative Adversarial Network (DCGAN) is particularly effective for image-related tasks, as it makes extensive use of Convolutional Neural Network (CNN) layers to capture spatial and structural patterns in images.

In this project, a CNN-based DCGAN is implemented and trained on the Fashion-MNIST dataset, which consists of grayscale images of various fashion-related items such as shirts, shoes, trousers, bags, and dresses. The dataset provides a suitable platform to explore how generative models learn and recreate visual concepts without requiring explicit labels. The Generator in the DCGAN takes a random noise vector as input and produces a synthetic image, while the Discriminator evaluates both real and generated images to classify them as genuine or fake. Over successive training epochs, the Generator becomes more capable of producing visually coherent images that resemble real fashion items, while the Discriminator becomes more accurate at distinguishing between real and synthesized samples.

The significance of this project lies in demonstrating how GAN-based models can learn underlying data distributions and generate new, meaningful content. Such generative models have growing applications in fields like creative design, automated art generation, data augmentation, and digital content synthesis. By successfully training a DCGAN on the Fashion-MNIST dataset, this project highlights the potential of deep learning in creative and industrial applications, particularly in areas connected to fashion technology and visual content production.

CHAPTER 2

OBJECTIVES

1. To implement a CNN-based Deep Convolutional Generative Adversarial Network (DCGAN) capable of generating synthetic fashion images that resemble real samples from the Fashion-MNIST dataset.
2. To understand and analyse the adversarial training process between the Generator and Discriminator, and how this competition improves the quality of image synthesis over time.
3. To utilize Convolutional Neural Network (CNN) layers for learning spatial patterns, textures, and structural features present in various types of fashion apparel.
4. To train the model using Fashion-MNIST and evaluate its performance based on the visual quality and clarity of the generated output images.
5. To explore the role of noise vectors (latent space) in controlling the variability and diversity of generated fashion images.
6. To demonstrate the practical applications of GAN-based image generation, such as automated fashion design, data augmentation, and creative digital content production.
7. To analyse challenges and limitations in GAN training, such as mode collapse and training instability, and observe how hyperparameters affect the results.

CHAPTER 3

PROBLEM STATEMENT

To develop a CNN-based Deep Convolutional Generative Adversarial Network (DCGAN) capable of learning the visual patterns in Fashion-MNIST and generating new, realistic fashion item images that resemble the original dataset samples.

CHAPTER 4

METHODOLOGY

The methodology for this project involves the implementation and training of a CNN -based DCGAN to generate synthetic fashion images using the Fashion-MNIST dataset. The process can be divided into several key stages:

1. **Data Collection and Preprocessing**: The Fashion-MNIST dataset (28×28 grayscale clothing images) is loaded using PyTorch and normalized to the range [-1, 1]. A DataLoader batches and shuffles the data for efficient training.
2. **Model Architecture Design:** The DCGAN architecture consists of two CNN-based models:
   - **Generator**: Takes a random noise vector called the latent vector and uses transposed convolution layers to upscale it into a 28×28 synthetic image. Batch Normalization and ReLU activations are used to stabilize learning and produce smooth gradients.
   - **Discriminator**: Uses convolution layers to analyse input images and classify them as real or fake. LeakyReLU activations are used to improve gradient flow in early training stages.
3. **Loss Function and Optimization:** A Binary Cross-Entropy Loss with Logits is used for training, while both the Generator and Discriminator are optimized using the Adam optimizer (lr = 2e-4, $\beta1 = 0.5$, $\beta2 = 0.999$) to maintain stable adversarial learning.
4. **Adversarial Training Process**: Training occurs in iterative steps:
   - The Discriminator is trained using real images from the dataset and fake images generated by the Generator.
   - The Generator is trained to produce images that the Discriminator will classify as real.
   - This competitive training continues across multiple epochs, gradually improving the quality of generated images.
5. **Performance Monitoring and Image Generation:** During training, a fixed noise vector is used to generate sample images at the end of each epoch. These samples are saved to observe how the Generator improves over time. The training continues until the generated images become visually similar to real Fashion-MNIST samples.

CHAPTER 5

REQUIREMENT ANALYSIS

1. FUNCTIONAL REQUIREMENTS

- Data Processing: The system should load and normalize the Fashion-MNIST dataset to the required input format for training.
- Model Functionality: The Generator should produce synthetic fashion images from random noise, and the Discriminator should evaluate images as real or fake.
- Training Procedure: The system should train both networks adversarial, updating the Discriminator and Generator alternately to improve output quality.
- Output Generation: The system should generate and save sample images after each training epoch to visualize improvements over time.
- Error Handling: The system should manage invalid inputs (e.g., incorrect tensor dimensions) and display informative debugging messages.
- Visualization and Analysis: The system should allow inspection of generated images, loss values, and training behaviour over epochs to assess performance.

2. NON-FUNCTIONAL REQUIREMENTS

- Performance: Training should be optimized to run efficiently, with GPU support significantly improving speed.
- Accuracy: The generated images should gradually improve in visual quality to resemble real Fashion-MNIST samples.
- Scalability: The model should be capable of adapting to other image datasets with minimal architectural changes.
- Reliability: The training process should consistently produce progressively better outputs across epochs.
- Maintainability: The code should be clear, modular, and easy to update for future model enhancements.

CHAPTER 6

DESIGN

1. FLOWCHART

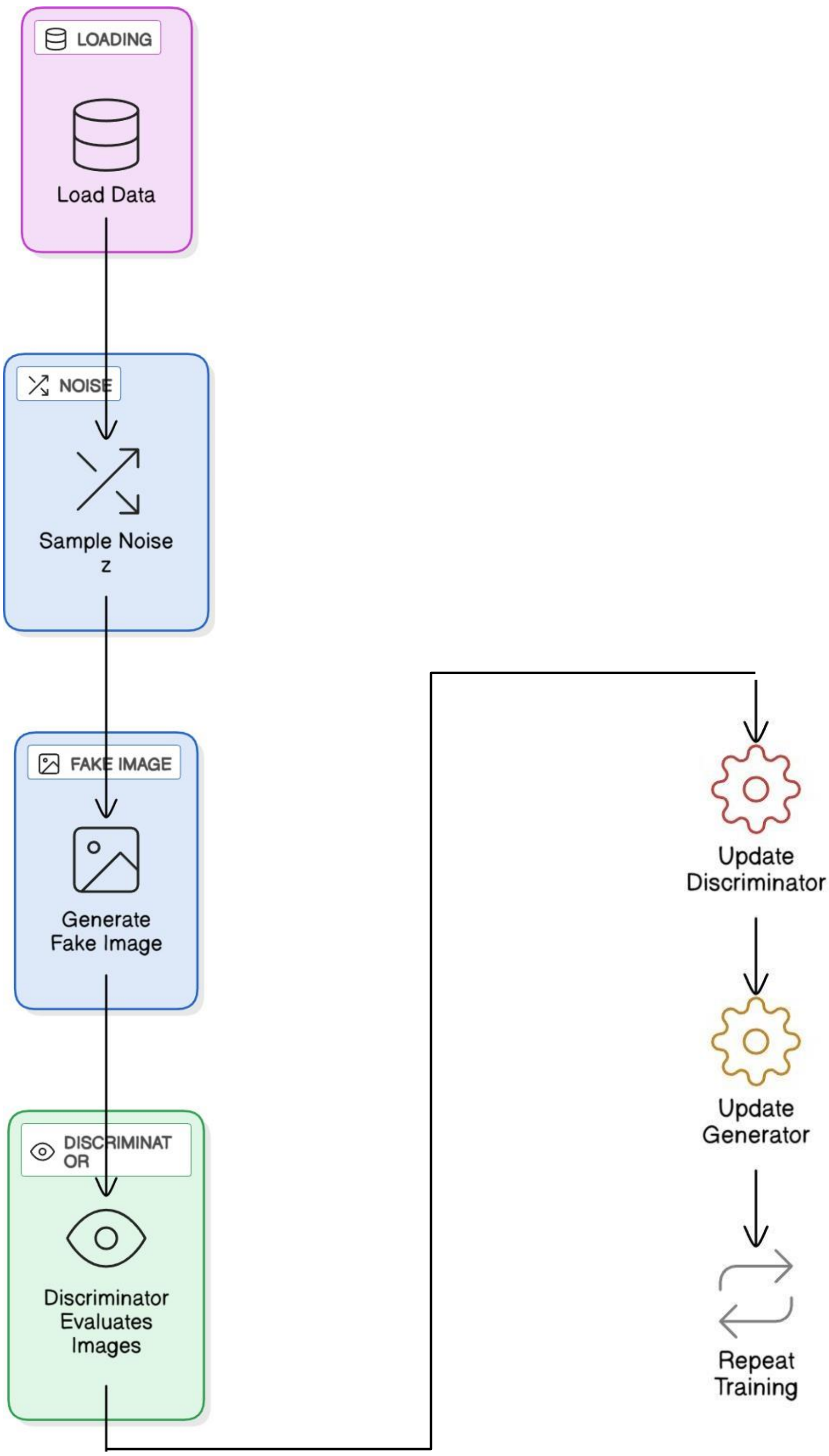

Fig 6.1 Flow Chart

2. USE CASE DIAGRAM

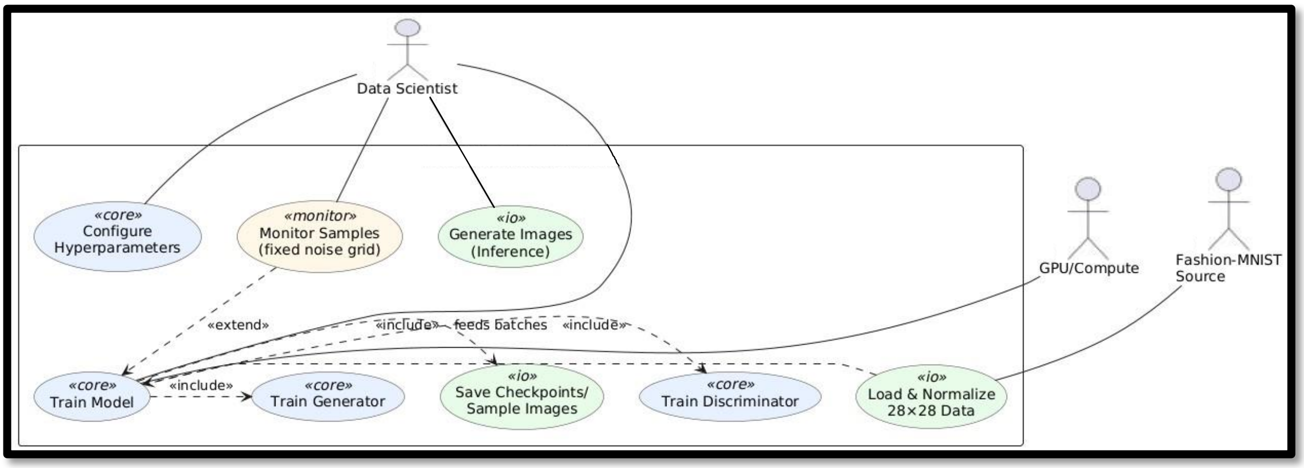

Fig 6.2 Use case Diagram

3. SEQUENCE DIAGRAM

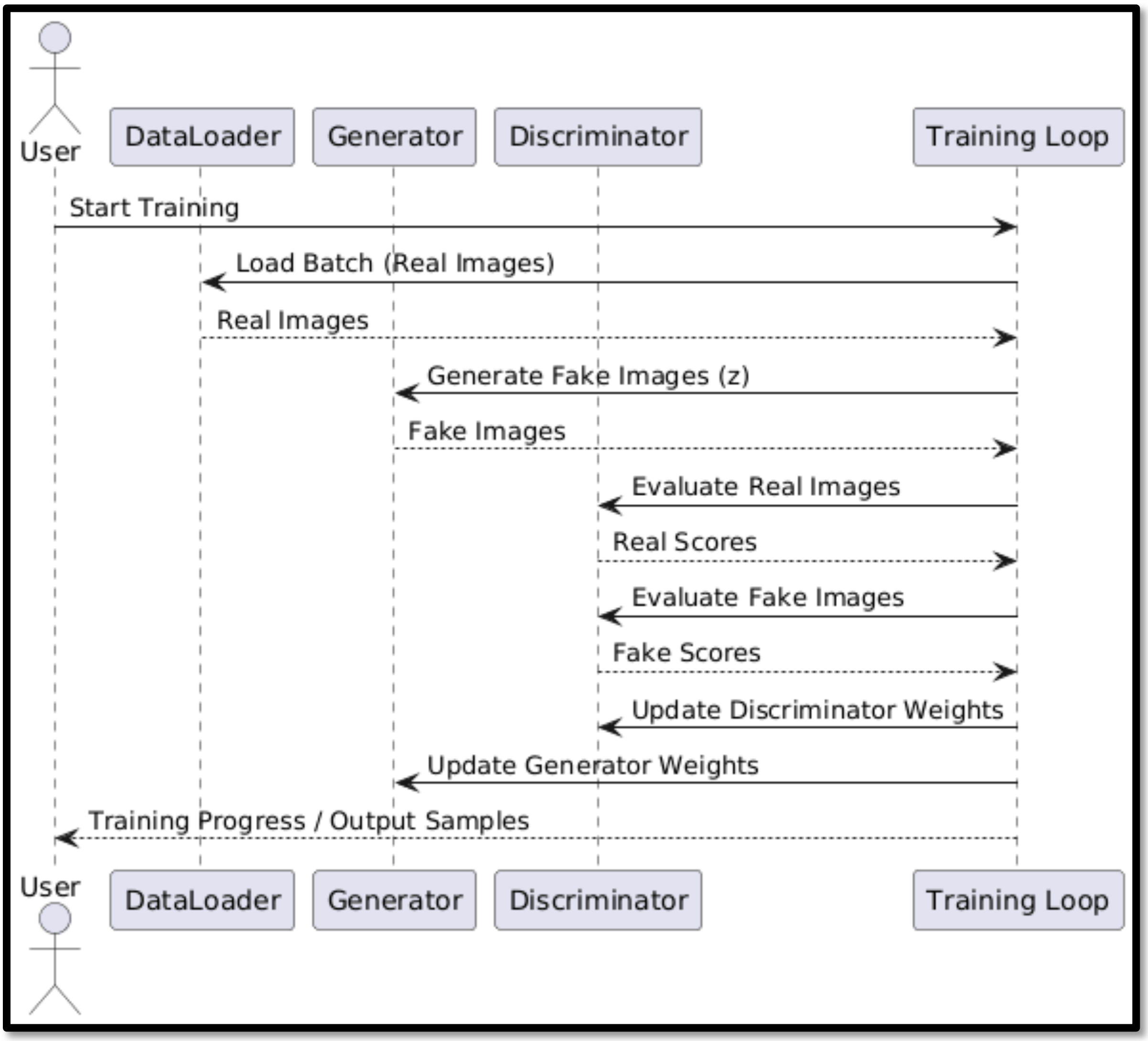

Fig 6.3 Sequence Diagram

CHAPTER 7

IMPLEMENTATION

1. **Environment & Setup:** The model is implemented in PyTorch. Training runs on GPU if available (torch.device("cuda" if torch.cuda.is_available() else "cpu")), otherwise CPU. Output images are stored in samples_fashion_mnist/.

2. **Dataset & Preprocessing.** The Fashion-MNIST training split is downloaded via torchvision.datasets.FashionMNIST(root="./data", train=True, download=True, transform=...). Images are converted to tensors and normalized to [-1, 1] using transforms.Normalize((0.5,), (0.5,)), matching the generator's Tanh output. A DataLoader feeds mini-batches (batch_size=128, shuffle=True, num_workers=2, pin_memory=True).

3. **Model Architecture:**

   a. **Generator (G):** Takes latent noise z ∈ ℝ^{100×1×1} and upsamples via transposed convolutions: 100→128@7×7 → 64@14×14 → 1@28×28. Each upsampling block uses ConvTranspose2d + BatchNorm2d + ReLU, with a final Tanh to emit a 1×28×28 grayscale image.

   b. **Discriminator (D):** A CNN binary classifier on 1×28×28 inputs: Conv2d(1→64, stride=2) + LeakyReLU(0.2) → Conv2d(64→128, stride=2) + BatchNorm2d + LeakyReLU → Conv2d(128→1, kernel=7) producing a single logit per image.

4. **Training Objective:** Binary cross-entropy with logits (nn.BCEWithLogitsLoss) is used for both networks. Real labels are smoothed to 0.9 to improve stability; fake labels are 0.0.

5. **Optimization & Hyperparameters:** Both G and D use Adam with lr = 2e-4, β1 = 0.5, β2 = 0.999. Training runs for 20 epochs with batch_size = 128. A fixed latent batch (fixed_z, 64 samples) tracks generator progress across epochs.

6. **Training Loop (per mini-batch):**

   a. **D step:**

      - Forward real images → loss against smoothed real labels (0.9).
      - Sample noise z, generate fake images with G, detach from graph, forward through D → loss against 0.0.
      - Sum losses, backprop, optD.step().

   b. **G step:**

      - Resample z, generate images, forward through D, compute loss against real labels (0.9) to "fool" D, backprop, optG.step().

7. **Monitoring & Output:** After each epoch, the model switches to eval mode and writes a grid of samples generated from fixed_z using torchvision.utils.save_image(..., normalize=True, value_range=(-1, 1), nrow=8) to samples_fashion_mnist/epoch_XXX.png. Console logs report final mini-batch losses for D and G per epoch.

8. **Inference (Sampling):** After training, sampling requires only the generator: draw z ~ N(0, I) with shape (N, 100, 1, 1), run G(z), optionally denormalize from [-1,1] to [0,1] for visualization.

9. **Reproducibility Notes:** To make runs repeatable, set seeds for Python, NumPy, and PyTorch, and record package versions. Keep hyperparameters (latent_dim=100, epochs=20, lr=2e-4, β1=0.5, β2=0.999) consistent with the code block above.

CHAPTER 8

RESULTS AND DISCUSSION

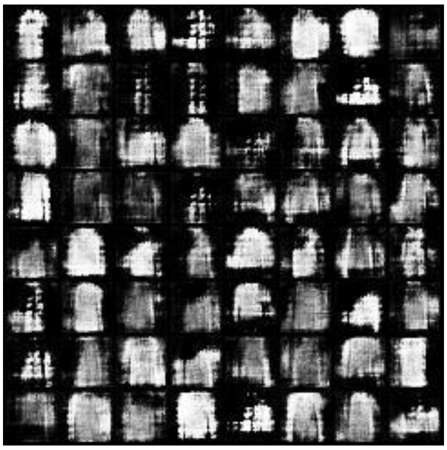

Fig 8.1 Epoch 1

Fig 8.2 Epoch 5

Fig 8.3 Epoch 10

Fig 8.4 Epoch 15

Fig 8.5 Epoch 20

CONCLUSION

This project successfully implemented a CNN-based Deep Convolutional Generative Adversarial Network (DCGAN) to generate synthetic fashion images using the Fashion-MNIST dataset. The Generator learned to create new images from random noise while the Discriminator learned to differentiate between real and generated samples through adversarial training. Over multiple epochs, both networks improved simultaneously, resulting in generated images that gradually resembled real clothing items. The project demonstrates how GANs can learn underlying data distributions without explicit supervision and produce realistic visual outputs. This implementation highlights the potential of DCGANs in creative design, data augmentation, and other visual generation applications, showing how deep learning models can contribute to fields like fashion technology, digital art, and automated content synthesis.

REFERENCES

1. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., “Generative Adversarial Networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

2. T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12104–12114, 2020.

3. D. Zheng, Z. Zhou, and X. Chen, “A DCGAN-Based Method for Image Generation and Data Augmentation,” *IEEE Access*, vol. 9, pp. 25612–25622, 2021.

4. H. H. Nguyen, T. T. Nguyen, and M. N. Nguyen, “Improving Generative Adversarial Networks for Fashion Image Synthesis,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 1454–1458, 2022.

5. Y. Wang and S. Chen, “Survey on Deep Generative Models for Image Synthesis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 256–275, 2023