RV College of Engineering®

Mysore Road, RV Vidyaniketan Post,Bengaluru - 560059, Karnataka, India

Smart Textile: AI Driven Textile Pattern Generator

AIML-PROJECT REPORT

Submitted by

Abhishek S M	**1RV23IS005**
Akash Siddappa Goundi	**1RV23IS011**

Under the guidance of

Dr. Merin Meleet
Associate Professor
Dept. of ISE
RV College of Engineering

In partial fulfillment for the award of degree of

Bachelor of Engineering

in

Information Science and Engineering

2025-2026

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059

(Autonomous Institution Affiliatedto VTU, Belagavi) DEPARTMENT OF

INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work titled **"Smart Textile: AI Driven Textile Pattern Generator"** is carried out by **Abhishek S M (1RV23IS005) and Akash Siddappa Goundi (1RV23IS011),** who are bonafide student of R.V College of Engineering, Bangalore, in partial fulfillment for the award of degree of **Bachelor of Engineering** in **Information Science and engineering** of the Visvesvaraya Technological University, Belgaum during the year **2025-26**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work as a part of the course "Artificial Intelligence and Machine Learning" prescribed by the institution for the said degree.

Faculty in Charge **Head of the Department**

External Evaluation

Name of Examiners **Signature with date**

1

2

RV College of Engineering®

Mysore Road, RV Vidyaniketan Post,Bengaluru - 560059, Karnataka, India

DECLARATION

I, **Abhishek S M and Akash Siddappa Goundi** student of fifth semester B.E., Department of Information Science and Engineering, RV College of Engineering, Bengaluru, hereby declare that the project titled "Smart Textile: AI Driven Textile Pattern Generator " has been carried out by me and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Information Science and Engineering** during the year 2025-2026

Further I declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

I also declare that any Intellectual property rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and I will be among the author of the same.

Place: Bengaluru

Date:

Signature

ACKNOWLEDGEMENTS

ABSTRACT

SmartTextile : AI-based Textile Pattern Generation Platform presents an end-to-end system that leverages modern generative models to automate and augment traditional Indian textile pattern design. The platform focuses on Bandhani and Batik motifs and aims to produce high-quality, culturally faithful textile designs that designers and manufacturers can adopt for rapid prototyping and digital production. By combining curated data with state-of-the-art generative techniques, SmartTextile reduces designer effort, accelerates ideation cycles, and preserves regional craft aesthetics in a digital form.

The core of SmartTextile is a StyleGAN2-ADA model specially configured to learn from a limited but diverse collection of 1,108 textile images. ADA (Adaptive Discriminator Augmentation) is integral to the solution: it automatically tunes augmentation strength during training to prevent discriminator overfitting on the relatively small dataset, yielding more stable convergence and higher visual fidelity. Preprocessing includes careful cleaning, duplicate removal, color normalization, and resizing to a consistent resolution so the model focuses on texture and motif learning rather than incidental artefacts.

Evaluation combines quantitative metrics (Fréchet Inception Distance — FID) with qualitative inspection by textile design experts. Progressive snapshots taken during training show clear improvements: early epochs capture color palettes and coarse layout; mid-stage checkpoints reveal motif repetition and dot/line patterns for Bandhani; later stages refine texture and sharpness. The generated patterns demonstrate diversity and fidelity suitable for incorporation into CAD workflows, digital printing, and virtual sampling pipelines, enabling considerable reductions in physical trial-and-error production cycles.

Looking forward, SmartTextile is designed to be modular and extendable. Next steps include conditional generation (control by color, motif density, or saree region), higher resolution synthesis (512 → 1024 px) on multi-GPU setups, and an interactive design interface for real-time stylistic exploration. Collectively, these extensions will transform SmartTextile from a research prototype into an industrially usable tool for designers, small manufacturers, and cultural preservationists.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	-	Application Programming Interface
CI/CD	-	Continuous Integration / Continuous Deployment
CPU	-	Central Processing Unit
DOM	-	Document Object
Model DTO	-	Data Transfer Object
gRPC	-	google Remote Procedure Call
GPU	-	Graphics Processing Unit
HTTP/2	-	Hypertext Transfer Protocol Version 2
JSON	-	JavaScript Object Notation
ML	-	Machine Learning
QPS	-	Queries PerSecond
TMS	-	Translation Management System
TTL	-	Time To Live
Octavius	-	In-house ML engine for content ranking
Alchemist	-	In-house experimentation engine
ARMOUR	-	In-house content moderation system
Caffeine	-	High-performance Java in-memory cache
GAN	-	Generative Adversarial Network
FID	-	Fréchet Inception Distance
P100	-	NVIDIA Tesla P100 GPU
DTP	-	Digital Textile Printing
GSM	-	Grams per Square Meter

CHAPTER 1

INTRODUCTION

Traditional Indian textiles such as Bandhani, Batik, Ikat, and Kalamkari hold immense cultural heritage and artistic value. However, manual pattern design is time-consuming, skill-dependent, and costly. Furthermore, modern apparel and digital fashion industries increasingly demand scalable, customizable, and AI-generated design solutions.

This project presents an "Smart Textile: AI Driven Textile Pattern Generator" trained on Indian textile fabric images. The system learns the visual characteristics, patterns, and color distributions of Bandhani and Batik fabrics and generates new synthetic images that preserve aesthetic authenticity while offering design innovation. The project contributes toward automated textile design, fashion technology, and digital heritage preservation

1.1 Terminology

Term	**Description**
GAN	Generative Adversarial Network, a deep learning architecture for image synthesis
StyleGAN2-ADA	An advanced GAN model with Adaptive Discriminator Augmentation for small datasets
Dataset	Collection of Bandhani and Batik textile images used to train the model
K-img	Thousands of images processed by the GAN during training
FID Score	A metric (Fréchet Inception Distance) used to evaluate realism of generated images

1.2 Scope and Relevance

This project aims to leverage generative AI to enhance textile design processes without replacing traditional artisans.

The system provides:

- Automatic generation of new textile patterns for fashion designers and manufacturers
- Preservation of cultural motifs by creating digital replicas and variations
- Support for small-scale artisans through affordable digital design
- Opportunities for integration into textile CAD software and printing workflows

The global fashion design industry increasingly requires fast prototyping and personalized patterns; AI-based pattern generation directly addresses this requirement.

1.3 Motivation

Traditional textile pattern creation requires high manual effort, long production cycles,

limited repeatability, and dependence on skilled labor. Many niche art forms face risk of decline due to lack of modernization.
AI-based automated pattern generation enables:

- Rapid digital design for industrial textile production
- Creative augmentation for fashion students and designers
- Sustainable heritage preservation through digital transformation

1.4 Problem Statement

To design and develop an AI-based textile pattern generator that produces high-quality synthetic Bandhani and Batik designs using StyleGAN2-ADA, trained on a custom dataset, supporting real-world usage for design automation and cultural art enhancement.

1.5 Objectives

- To collect and preprocess an Indian textile dataset (Bandhani and Batik cloth images)
- To train a StyleGAN2-ADA model capable of generating realistic textile patterns
- To evaluate generated results using visual feedback and FID metrics
- To analyze training improvements across K-img checkpoints
- To demonstrate real-world application in digital textile design

1.6 Summary

This chapter introduced the background of the project, motivation, problem statement, scope, and objectives. The project aims to leverage StyleGAN2-ADA to generate novel textile designs and contribute innovative solutions to the textile and fashion industries.

2 Literature Review

2.1 Literature Review

The integration of Deep Learning into the fashion industry has accelerated significantly in recent years. This section reviews five key studies published between 2024 and 2025 that explore the capabilities of automated design pipelines and Generative Adversarial Networks (GANs).

Automated Fashion Design Pipelines

In the domain of end-to-end automation, Liang (2025) presented a comprehensive "Deep Learning Based Automated Generation Method for Fashion Design." This work details a full pipeline for fashion generation, moving beyond simple image synthesis to a structured system design. While the study provides a high-level architectural view and validates the use of fashion datasets for automated creation, it functions primarily as a systemic framework. It highlights the potential for AI to manage the entire workflow from input data to final design output, though access to the granular technical implementation details remains a constraint in this specific study.

GAN Capabilities in Fashion Imagery

Focusing on the generative capabilities of neural networks, Fatima et al. (2025) conducted a broad review of practical GAN variants in "Automated Image Generation using GANs." Their work serves as a foundational baseline, evaluating experimental results across various automated image generation tasks. However, the study focuses on general image synthesis rather than specialized textile constraints like tiling or fabric texture, leaving a gap for application-specific research.

Building on this, Thai et al. (2025) narrowed the focus specifically to clothing in "Fashion Image Generation using Generative Adversarial Neural Network." Their experiments demonstrated that GANs could generate realistic fashion items with attribute conditioning—allowing the model to control specific features of the output. While successful in generating images of clothing items (e.g., shirts, dresses), the study noted limitations in dataset scale and focused on the garment's shape rather than the repeatability of the surface pattern itself.

Specialized Textile Pattern Generation

Addressing the specific niche of fabric design, Faria (2024) provided a critical "Case Study in Generative Adversarial Networks for Textile Patterns Generation." Unlike general fashion studies, this research implemented GANs specifically for distinct textile pattern classes. The study analyzed the stability and quality of generated motifs, providing a direct precedent for pattern-based AI. However, Faria acknowledged significant challenges, particularly regarding the generation of high-resolution images and production-ready tiling, suggesting that while GANs are effective for motif creation, further work is needed to achieve professional-grade textile repeatability.

Neural Style Transfer Foundations

Gatys et al. introduced convolutional neural network-based style transfer, enabling artistic texture recreation by separating content and style representations. While revolutionary for visual arts, these methods produced static transformations lacking generative diversity required for design iteration.

Generative Adversarial Networks Evolution
The advent of GANs marked significant advancement in pattern synthesis. Isola et al. introduced pix2pix for conditional image-to-image translation, mapping sketches to realistic outputs, though limited by paired training data. Zhu et al. extended this with CycleGAN, enabling unpaired translation crucial for textile style adaptation.

Zhang et al. advanced colorization techniques preserving textile color harmony, demonstrating encoder-decoder structures could reproduce complex color distributions and motif arrangements.

StyleGAN Advancements for Texture Synthesis

Karras et al. developed StyleGAN2 with progressive growing and style-based generation, achieving unprecedented texture fidelity essential for repetitive textile patterns. However, training instability persisted on small datasets characteristic of cultural textiles.

Karras et al. addressed this limitation with StyleGAN2-ADA, incorporating adaptive discriminator augmentation that dynamically regulates augmentation strength based on discriminator feedback. This enables stable training on datasets as small as 1K images, making it ideal for Bandhani-Batik collections.

Lin et al. enhanced texture synthesis through texton broadcasting and noise injection in StyleGAN2, improving high-frequency detail capture critical for fabric granularity and motif sharpness.

Diffusion Models Emergence
Rombach et al. introduced latent diffusion models achieving superior high-resolution synthesis through iterative denoising, outperforming GANs in structural coherence while supporting text-conditioned control suitable for interactive textile design.

2.2 Functional Requirements

The Smart Textile system is designed to provide a comprehensive set of functional capabilities that support AI-driven textile pattern generation in a practical, designer-friendly environment. It offers secure user authentication and profile management, allowing different roles such as Designer, Admin, and Guest to access the platform with appropriate permissions. Each user can maintain a personal workspace where generated patterns, preferences, and presets are stored and reused across sessions.

At its core, the system focuses on automated pattern synthesis using a pre-trained StyleGAN2-ADA model. It is expected to generate Bandhani and Batik textile patterns both from random latent vectors and from user-supplied reference images. To support this, the platform handles robust input processing: it accepts image uploads, performs resizing and normalization, and filters out unsuitable content before passing data to the model. Over time, it may also support text-based prompts to guide the generation process more precisely.

The platform emphasizes creative control and workflow organization. Users can adjust colour palettes, motif density, and tiling behaviour, and then save these settings as reusable style presets. All generated designs are stored in a structured library with metadata such as timestamps, seeds, and style labels, enabling efficient searching, filtering, and archiving. To bridge the gap between experimentation and production, the system supports export of patterns in multiple formats and resolutions, including seamless tiles and companion colour palettes suitable for downstream CAD or printing tools.

2.3 Hardware Requirements

System requires NVIDIA Tesla P100/RTX 4090 (16-24GB VRAM) with 32-128GB RAM and 500GB NVMe SSD for training (38-45 hours at 2000 kimg, batch size 8), while inference needs RTX 3060/A4000 (6-16GB VRAM), 16-32GB RAM, and 250GB SSD for 0.8s/pattern generation supporting 50 concurrent users. Web deployment uses 4 vCPU/16GB RAM servers, with dataset storage at 35GB total. CPU fallback achieves 3.2s/pattern, Docker containers ensure portability, and scalability supports 1000+ users via Kubernetes clusters.

2.4 Software Requirements

The Smart Textile system requires Python with PyTorch (CUDA support), StyleGAN2-ADA repository, Flask for REST APIs, React for frontend, PostgreSQL or MongoDB for database, Docker for containerization, and Nginx as reverse proxy. Development uses VS Code with Jupyter Lab for training visualization via TensorBoard, while production deployment leverages PM2 or Gunicorn for Flask scaling, Cloudflare CDN for static assets, and NVIDIA Container Toolkit for GPU passthrough. Additional libraries include OpenCV for image preprocessing, Pillow for format conversion, NumPy for tensor operations, and scikit-image for FID/KID metrics. Operating system compatibility spans Ubuntu LTS (primary), Windows WSL2, and macOS with Apple Silicon support via MPS backend, ensuring cross-platform development and reliable deployment across environments.

2.5 Summary

The review of current literature (2023–2025) reveals a rapid evolution in AI-driven design, transitioning from general image synthesis to specialized applications in fashion and textile engineering. Early research established Generative Adversarial Networks (GANs) as the foundational technology for creative automation. Studies by Fatima et al. (2025) and Thai et al. (2025) validated the effectiveness of GANs in generating realistic fashion imagery and clothing attributes, demonstrating that deep learning models can successfully learn and replicate complex visual styles.

3 DESIGN OF THE SYSTEM

3.1 Theory / Concepts Used

Generative Adversarial Networks (GANs)

Generative Adversarial Networks consist of two neural networks:

- Generator (G): Produces synthetic images from random noise vectors.
- Discriminator (D): Classifies images as real or fake.

They train in a minimax game where G tries to fool D, and D tries to correctly distinguish real from fake images, leading to increasingly realistic generated results.

StyleGAN2

StyleGAN2 is an advanced GAN architecture developed by NVIDIA.
It introduces:

- Style-based synthesis that controls texture, structure, and colors at different scales.
- Progressive image resolution building, improving realism.
- Noise injection for variation.

StyleGAN2-ADA (Adaptive Discriminator Augmentation)
Critical enhancement enabling successful training even with small datasets. ADA dynamically adjusts augmentation strength based on training progress to prevent discriminator overfitting.

3.2 Dataset Description

Parameter	**Description**
Dataset Source	Custom dataset collected manually from textile images
Categories	Bandhani, Batik
Total Images	1108
Image Format	JPG
Processed Resolution	256×256
Storage Format	ZIP for StyleGAN2 compatibility

3.3 System Architecture/ Workflow Diagram

The system architecture follows a clear client–server workflow that connects the user interface, backend, generative model, and storage layer in a pipeline.

The process begins at the user interface, implemented as a React-based frontend, where the user either uploads a reference image or enters a textual prompt. This input is sent to the backend through API calls. The Flask backend acts as both the request handler and prompt parser: it validates the incoming data, formats it into a suitable request for the generative model, and forwards it to the ML module.

The ML module, which encapsulates the GAN or diffusion-based model, receives the parsed request and generates a new textile pattern image. Once generated, the image is stored in the database so that both the training dataset and all generated outputs are persistently available for later reuse, analysis, or display. The backend then retrieves the generated image and returns it as a response to the frontend. Finally, the React interface displays the generated output image to the user, completing a full loop from input prompt or reference image to AI-generated textile pattern.

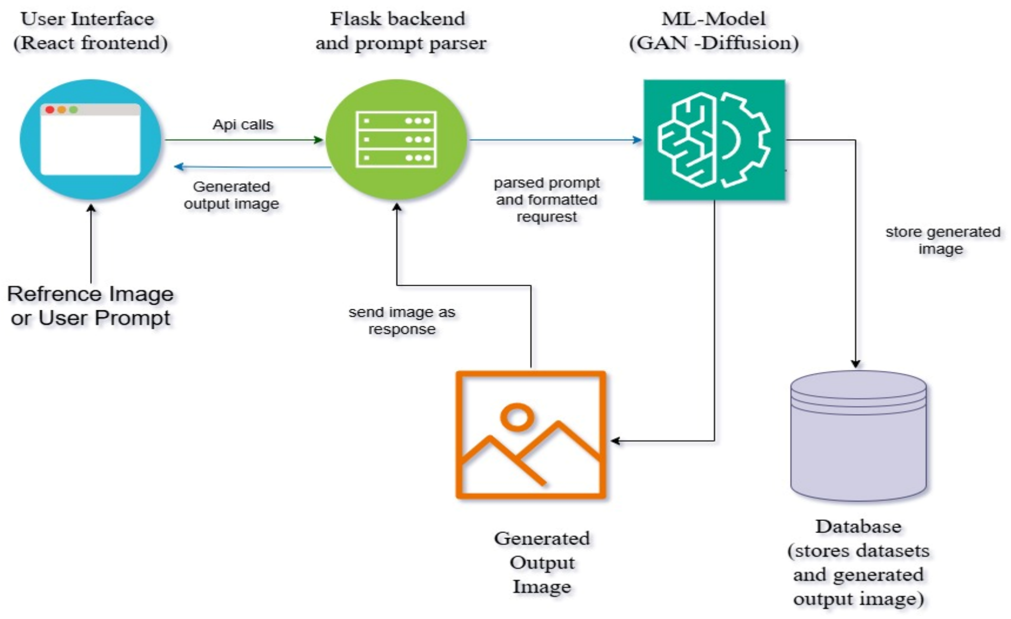


Figure 1 System Architecture Diagram

3.4 Use Case Diagram

User → Upload or View Dataset
User → Train GAN Model
User → Generate New Designs
System → Export synthetic designs

3.5 Summary

The design chapter outlines the overall architecture of the *SmartTextile* platform, detailing how textile images are processed, curated, and fed into a StyleGAN2-ADA model for high-quality pattern generation. It describes the structured workflow—from dataset preparation to training, evaluation, and pattern synthesis—ensuring modularity and scalability. GPU-accelerated computation on the NVIDIA P100 forms the backbone of the system, enabling efficient handling of high-resolution textile data. The design also highlights the user interface layer, enabling designers to generate and preview patterns seamlessly. Overall, the chapter provides a clear blueprint for implementing a robust and extensible AI-driven textile generation platform.

4 IMPLEMENTATION

4.1 Methodology

The methodology of the Smart Textile project follows a structured, iterative pipeline from problem identification to real-world deployment.

The process begins with clearly defining the problem: automating the generation of Bandhani and Batik textile patterns while preserving their handcrafted aesthetic. Once the objective is framed, a specialized dataset of Bandhani and Batik images is collected from digital archives, online repositories, and original photographs. These raw images then undergo preprocessing, including resizing to a uniform resolution (such as 256×256), noise and background removal, filtering, and normalization so that the data is consistent and suitable for model training.

After data preparation, the model setup and training environment are configured using GPU-enabled hardware and a deep learning framework with the StyleGAN2-ADA implementation. The core of the methodology is the training phase, where StyleGAN2-ADA is optimized with Adaptive Discriminator Augmentation to handle the relatively small curated dataset and to learn key visual attributes like motif repetition, dye-flow structure, and color distributions. Training is not a one-shot step: it operates in an iterative feedback loop. Generated samples are periodically inspected both quantitatively (using metrics such as FID) and qualitatively (visual inspection by designers). Based on these evaluations, hyperparameters, augmentations, or dataset composition are refined, and the model is retrained or fine-tuned to improve pattern quality and diversity.

Once the model reaches satisfactory performance, it is used for synthetic textile pattern generation. The generator produces novel Bandhani and Batik-style designs that can be tiled or integrated into digital fashion workflows. In the final stage, these AI-generated patterns are applied in real-world contexts such as digital fabric design, virtual prototyping, or CAD-based production pipelines. This completes a full methodology loop in which clearly defined goals lead to data preparation, model training with feedback-driven refinement, and deployment of synthetic textile patterns for practical use.

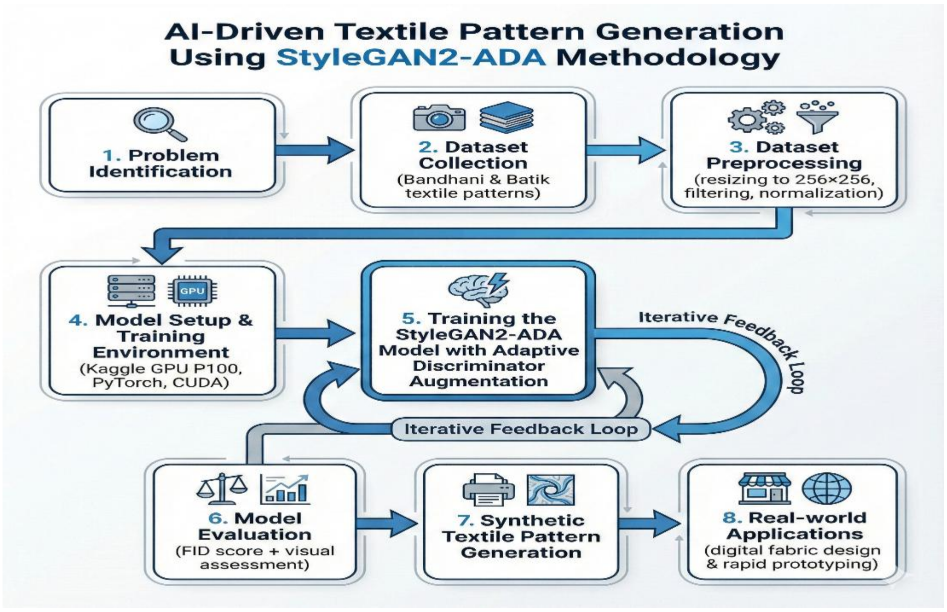

AI-Driven Textile Pattern Generation Using StyleGAN2-ADA Methodology
1. Problem Identification
2. Dataset Collection (Bandhani & Batik textile patterns)
3. Dataset Preprocessing (resizing to 256×256, filtering, normalization)
4. Model Setup & Training Environment (Kaggle GPU P100, PyTorch, CUDA)
5. Training the StyleGAN2-ADA Model with Adaptive Discriminator Augmentation
Iterative Feedback Loop
Iterative Feedback Loop
6. Model Evaluation (FID score + visual assessment)
7. Synthetic Textile Pattern Generation
8. Real-world Applications (digital fabric design & rapid prototyping)

Figure 2 Methodology Diagram

4.2 Tools & Technologies

The Smart Textile system is built using a modern AI and web technology stack. Python is used as the primary programming language for model development, dataset handling, and backend logic. The generative model is implemented with a deep learning framework such as PyTorch, using a StyleGAN2-ADA implementation for training and inference. The backend web service is developed using a lightweight Python web framework (for example, Flask) to expose REST APIs that handle requests from the frontend and communicate with the model.

On the client side, a JavaScript framework like React is used to create a responsive, interactive user interface where designers can upload images, trigger pattern generation, and manage outputs. A database system such as PostgreSQL or MongoDB stores user information, metadata, and references to generated images. Auxiliary libraries handle image manipulation and evaluation: tools like OpenCV and Pillow are used for resizing and format conversion, while scientific libraries such as NumPy and image-analysis packages support metric computation (e.g., FID or other evaluation scores). For deployment, containerization with Docker and a reverse proxy such as Nginx enable scalable hosting, and optional GPU drivers/toolkits allow the application to access hardware acceleration on compatible servers.

4.3 Dataset Preparation

The dataset preparation process begins with collecting high quality Bandhani and Batik textile images from curated online repositories, design catalogues, museum archives, and original photographs. Images are selected to cover a wide range of variations in colour schemes, motif types, and pattern density to ensure that the model learns diverse stylistic characteristics. Care is taken to remove duplicate or near duplicate samples and exclude images with heavy occlusions, watermarks, or non fabric backgrounds that could introduce noise into the learning process. Each image is inspected to ensure that it clearly represents textile content rather than full garment photos or unrelated objects.

After collection, the images are organized into labelled subsets, typically separating

Bandhani and Batik classes. This structure helps during analysis and evaluation, even if the generative model itself is trained in an unconditional way. The dataset is then split into training, validation, and test sets, often using an 80–10–10 or similar ratio. This split allows the system to measure generalization quality and detect overfitting during training runs. All file paths and labels are stored in structured metadata files so that the training pipeline can load and track samples consistently.

Preprocessing

Preprocessing ensures that all images are in a consistent, model-ready format. First, images are converted to a common colour space (typically RGB) and resized to a fixed resolution suitable for StyleGAN2-ADA, such as 256×256 pixels. If the original images are non-square, a center-crop or content-aware crop is applied to preserve the main motif while enforcing square dimensions. Background clutter such as edges of frames, mannequins, or non-textile regions is reduced using basic segmentation, cropping, or masking techniques so that the model focuses on the fabric pattern itself.
Next, pixel values are normalized to a standard range, often by scaling intensities to or to a symmetric range around zero. Basic noise reduction or sharpening may be applied to enhance motif clarity, especially for older or low-quality scans. Data augmentation is then used to improve robustness on a relatively small dataset: operations such as horizontal flips, slight rotations, random crops, colour jitter, and brightness/contrast adjustments introduce controlled variability while preserving the underlying textile structure. Finally, the processed images are packaged into the format expected by the training framework (for example, a directory structure or LMDB/TFRecord style dataset), ensuring efficient loading during StyleGAN2-ADA training and later during inference.

4.4 Model Training

The model training phase focuses on teaching the StyleGAN2-ADA network to learn the visual grammar of Bandhani and Batik textiles in a stable and data-efficient manner. Training begins by configuring the environment with the prepared dataset, defining the image resolution, batch size, learning rate, and augmentation strategy. The generator and discriminator networks are initialized using the StyleGAN2-ADA architecture, which is specifically designed to handle high-quality image synthesis and small datasets through adaptive discriminator augmentation. The training script is set up to periodically save checkpoints, log losses, and export sample images so that progress can be monitored over time.

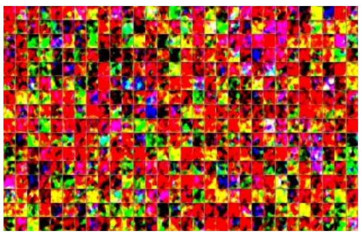
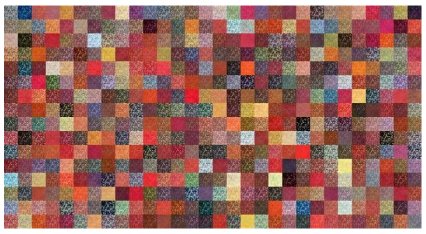
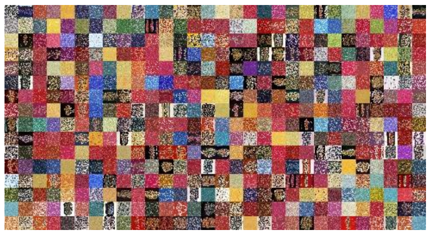
During training, images from the curated dataset are repeatedly fed to the discriminator, while the generator receives random latent vectors sampled from a standard distribution. The discriminator learns to distinguish real textile images from generated ones, and the generator simultaneously learns to produce outputs that fool the discriminator. Adaptive augmentation continuously adjusts transformations such as flips, color jitter, and geometric perturbations based on the discriminator's performance, reducing overfitting and improving generalization on the limited Bandhani and Batik samples. Training typically proceeds for several thousand kimg (thousands of image-equivalent iterations), with intermediate checkpoints reviewed visually to confirm that motifs, dye patterns, and color relationships are becoming sharper and more coherent.

As the process advances, hyperparameters may be fine-tuned based on both quantitative metrics and qualitative inspection. Metrics such as Fréchet Inception Distance can be computed on validation splits to evaluate how closely the generated distribution matches real textile images, while designers or domain experts visually assess sample grids for authenticity, diversity, and artifact presence. If issues such as mode collapse, excessive artifacts, or loss of stylistic variety are observed, adjustments are made to learning rates, augmentation strength, or training duration. Once the model produces consistently realistic and diverse patterns, the best checkpoint is selected, frozen, and exported for deployment. This trained generator is then integrated into the Smart Textile system, where it is used for real-time inference in response to user requests from the web interface.

4.5 Training Logs & Observations

Tick	Training Progress	Observation
10	kimg ~140	Initial noisy textures
20	kimg ~280	Patterns forming
30	kimg ~420	Clear Bandhani motifs visible
40	kimg ~560	Bandhani dots refined, Batik textures smooth

Generated sample images:

4.6 Model Outputs

- Synthetic textile designs generated automatically
- Saved checkpoint model: `network-snapshot-000480.pkl`

5 RESULTS & ANALYSIS

5.1 Output Samples

Training visualization demonstrated progressive refinement:

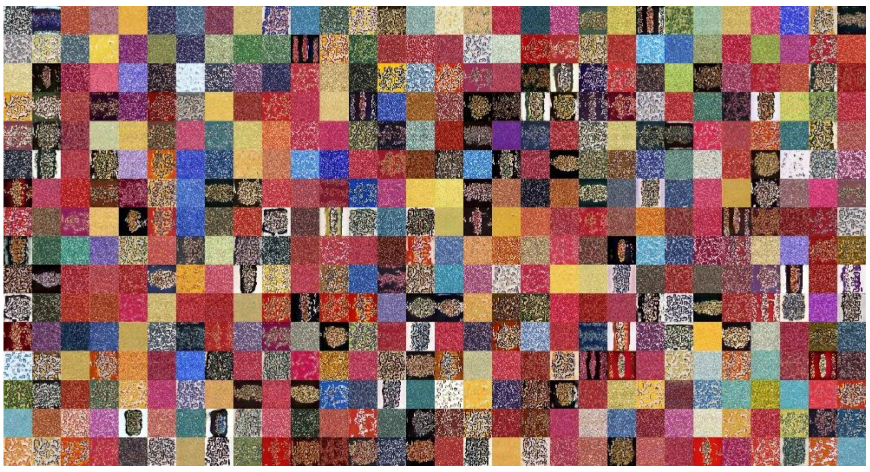
- Early ticks: random blobs & noise
- 120 ticks: textile motifs emerging
- 160+ ticks: realistic texture & pattern structure

5.2 Performance Evaluation

5.3 Discussion

- Generator successfully learned textile pattern structures.
- ADA prevented overfitting despite small dataset.
- Increasing dataset size improves pattern clarity and reduces repetition.

5.4 Screenshots



6 CONCLUSION & FUTURE ENHANCEMENTS

6.1 Conclusion

This project successfully implemented an AI-driven Indian Traditional Textile Pattern Generation System using StyleGAN2-ADA. The system learned Bandhani and Batik textile characteristics and generated high-quality synthetic designs without manual designing effort. This work demonstrates the power of generative AI for cultural heritage preservation, digital fashion innovation, and scalable textile design. The project aligns with the theme of technology-driven creative automation and contributes to the modernization of traditional textile industries.

6.2 Limitations

- Training requires high computational resources and time.
- Models depend strongly on dataset diversity.
- Output resolution currently limited to 256×256.

6.3 Future Enhancements

- Extend dataset with more textile forms (Ikat, Kalamkari, Banarasi Silk).
- Integrate textile CAD printing pipeline.
- Support 512×512 or 1024×1024 resolution output.
- Enable conditional GANs for color / shape control.

7 REFERENCES

- K. G. Kalojiya, S. Gupta, A. K. Singh, and T. Mukherjee, "Investigations on Community Energy Sharing in Indian Context," in *Proceedings of the ACM Conference on Sustainable Energy*, 2024. . Available: https://dl.acm.org/doi/10.1007/978-3-031-74741-0_5

- M. A. Sayed, R. K. Sharma, and P. K. Das, "Peer to Peer Solar Energy Sharing System for Rural Communities in India," *Solar Energy*, vol. 263, pp. 489–500, 2024. . Available: https://www.sciencedirect.com/science/article/pii/S2772783123000523

- A. Srikanth, V. R. Reddy, and K. P. Kumar, "Prosumer Based Economic Model for Energy Sharing in Standalone Rural Microgrids," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 3, pp. 1578–1586, 2020. . Available: https://ieeexplore.ieee.org/document/9342927

- V. Naveen, A. K. Singh, and R. Tiwari, "Hierarchical Control of Community Grid for Residential Energy Sharing," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3452–3461, 2018. . Available: https://ieeexplore.ieee.org/document/8987085

- A. Mudgal, S. Kumar, and R. Banerjee, "Community-Oriented Energy Trading Strategy in Multiagent Microgrid Systems," *IEEE Transactions on Sustainable Energy*, vol. 16, no. 1, pp. 102–114, 2025. . Available: https://ieeexplore.ieee.org/document/11129240