PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

This project aims to classify fabric pattern images using a deep learning model. It employs a custom Convolutional Neural Network (CNN) built with PyTorch to differentiate various fabric types using image datasets.

1.2 Purpose

The purpose is to automate pattern recognition in fabric for quality control and textile classification, reducing human error and increasing efficiency in manufacturing and textile analysis.

2. IDEATION PHASE

2.1 Problem Statement

Manually inspecting and classifying fabric patterns is time-consuming and error-prone. Automating this process can enhance accuracy and consistency in the textile industry.

2.2 Empathy Map Canvas

- **User**: Quality control personnel
- **Needs**: Accurate, quick classification of fabric patterns
- **Pain Points**: Manual sorting is slow, inconsistent
- **Gains**: Reduced workload, increased efficiency, better quality control

2.3 Brainstorming

- Use CNN for visual pattern recognition
- Augment data for robustness
- Evaluate multiple architectures (ResNet, Custom CNN)
- Create a prediction tool for real-time classification

3. REQUIREMENT ANALYSIS

Step	Action	Experience	Emotion
1	Upload fabric images	Simple interface	Curious
2	Model processes image	Fast prediction	Impressed
3	Result displayed	Clear result	Satisfied

3.2 Solution Requirement

- High accuracy fabric classification
- Scalable dataset support
- Efficient training loop and model evaluation

3.3 Data Flow Diagram

1. Load image dataset
2. Preprocess data (resize, normalize)
3. Split into train/validation
4. Train CNN
5. Evaluate & predict

3.4 Technology Stack

- **Language**: Python
- **Libraries**: PyTorch, Torchvision, NumPy, PIL
- **Hardware**: Apple Silicon (MPS) or GPU

---

4. PROJECT DESIGN

4.1 Problem-Solution Fit

The model addresses the core problem by providing an automated, accurate method for imagebased pattern classification.

4.2 Proposed Solution

A custom CNN model trained on preprocessed image data, capable of classifying multiple fabric patterns.

4.3 Solution Architecture

- Input -> CNN Layers -> Fully Connected -> Output
- Three convolution blocks followed by flattening and dense layers
- Model saved after training

---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week	**Task**
1	Dataset collection and preprocessing

Week	Task
2	Model development
3	Training and tuning
4	Evaluation and result analysis
5	Documentation and testing

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Accuracy**: Evaluated using classification report
- **Loss**: Monitored during training
- **Confusion Matrix**: Used to validate model predictions

---

7. RESULTS

7.1 Output Screenshots

(Include the following when available):

- Training loss graph
- Classification report
- Confusion matrix
- Sample prediction result

---

8. ADVANTAGES & DISADVANTAGES Advantages

- High accuracy with sufficient data
- Automation saves time
- Adaptable to different pattern types

Disadvantages

- Requires GPU for efficient training
- Model performance depends on dataset quality

---

9. CONCLUSION

This project successfully demonstrates the potential of deep learning in automating pattern recognition in fabrics, enhancing quality control and operational efficiency.

10. FUTURE SCOPE

- Integration with a GUI for user-friendly prediction
- Expansion to real-time camera-based classification
- Support for multi-label patterns and defects detection

11. APPENDIX Source Code

Included in file: 2e4d4701-ca91-4a4b-b0d5-673fa7e709bc.py

Dataset Link

Local path: C:\Users\Hemanth\Downloads\archive

GitHub & Project Demo Link

https://github.com/ahemanth1/pattern-recognition