Pattern Sense: Classifying Fabric Patterns Using Deep Learning

Introduction

"Pattern Sense" is an AI-powered system that leverages deep learning to identify and classify fabric patterns from images. It utilizes Convolutional Neural Networks (CNNs) to extract visual features and assign labels such as floral, geometric, abstract, striped, or ethnic. This solution is designed for the fashion, apparel, and textile industries to improve operational efficiency and reduce manual labor.

1.1 Purpose

The core objective is to automate the fabric pattern classification process using image recognition, reducing the subjectivity and time involved in manual classification. It enables:

- Faster product cataloging
- Enhanced inventory management
- Improved e-commerce filtering
- Support for automated quality control systems

2. IDEATION PHASE

2.1 Problem Statement

Manual identification of fabric patterns is inefficient, inconsistent, and labor-intensive. As fashion brands and textile businesses scale, there's a growing need for AI-based solutions that can automatically classify fabric patterns with high accuracy, even in complex and noisy images.

2.2 Empathy Map Canvas

Aspect	Insights
Says	“It takes too long to manually sort fabrics.”
Thinks	“We could boost productivity with automation.”
Feels	Frustrated with misclassifications and manual effort.
Does	Reviews images, sorts manually, maintains spreadsheets.
Pains	Inconsistency, fatigue, human error.
Gains	Speed, reliability, and scalability with automation.

2.3 Brainstorming Ideas

- Use pretrained CNN architectures (e.g., VGG16, ResNet50)
- Augment training data to simulate real-world variability
- Develop a Streamlit or Flask-based UI

- Optimize model for mobile or embedded use
- Integrate quality control checks using pattern-type filtering

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Phase	User Action	System Response	Emotional State
Upload	Uploads fabric image	Accepts and preprocesses image	Curious
Analysis	Clicks "Predict"	Model runs prediction	Anticipation
Feedback	Sees prediction and label	Displays confidence level	Informed/Trusting
Export	Downloads result	Offers downloadable report	Empowered

3.2 Solution Requirement

- **Functional:**
  - Upload image through web UI
  - Output predicted pattern class
  - Show confidence score
- **Non-Functional:**

- Response time < 2 seconds
- Accuracy > 85%
- Secure and lightweight web deployment

3.3 Data Flow Diagram

User → Web UI → Flask Backend → CNN Model → Prediction Output → UI

3.4 Technology Stack

Layer	Tools
Frontend	HTML
Backend	Python, Flask
Deep Learning	TensorFlow, Keras
DevOps	GitHub, Streamlit (optional), Docker (optional)
Dataset	Kaggle, Custom Labeled Data
Model	Custom CNN / Transfer Learning

4. PROJECT DESIGN

4.1 Problem Solution Fit

The automation of pattern classification is crucial for modernizing textile industries. A CNN-based solution ensures visual consistency, speed, and accuracy. The problem is well-suited for deep learning, which excels at image recognition tasks.

4.2 Proposed Solution

1. **Train a CNN model** (e.g., MobileNet, ResNet) on a labeled dataset of fabric pattern images, ensuring the dataset includes diverse categories like floral, striped, abstract, geometric, and ethnic.
2. Apply **data augmentation techniques** such as flipping, rotation, zooming, and brightness adjustments to improve the model's ability to generalize to unseen data.
3. **Evaluate model performance** using accuracy, precision, recall, and F1-score metrics, and fine-tune hyperparameters (like learning rate and batch size) to optimize performance.
4. **Save and export** the trained model and label map for use in real-time prediction through a web interface.
5. Develop a **Flask-based web application** where users can upload fabric images via a clean and responsive UI.
6. The Flask backend handles image **preprocessing**, such as resizing and normalization, before passing it to the model for inference.

4.3 Solution Architecture

[User]

↓ uploads image

[Frontend Web App]

↓ sends to backend API

[Flask Server]

↓ preprocess image

[Trained CNN Model]

↓ classify pattern (e.g., Floral)

[Flask Server]

↓ return result

[Frontend Web App] → Display Output to User

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Task	Outcome
Data Collection & Labeling	Clean, labeled dataset
Model Training & Evaluation	Trained CNN model with metrics
Flask Web Interface	Interactive UI for image upload
Integration & Testing	Model integrated with Flask
Documentation & Deployment	Full deployment, GitHub upload

1. Gathered requirements and finalized the problem statement.
2. Collected and labeled fabric pattern images from online sources.
3. Trained a CNN model using TensorFlow with data augmentation.
4. Developed a Flask web app to integrate the trained model.
5. Conducted testing, documentation, and prepared final deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Test Type	Metric	Result
Classification	Accuracy	87.2%
Precision	86.5%	86.5%
Recall	88.0%	88.0%
F1-Score	87.2%	87.2%
Inference Time	Mean time per image	1.89 sec
UI Responsiveness	Upload & Predict	< 2 sec

7. RESULTS

7.1 Output Screenshots

Include visuals such as:

- Home screen (upload section)
- Example fabric images (input)
- Predicted class label with confidence
- Comparison between actual and predicted labels

8. ADVANTAGES & DISADVANTAGES

Advantages

- Reduces manual labor and human error
- Increases speed of fabric sorting
- Easily scalable and modifiable
- Works with a wide variety of images

Disadvantages

- Dependent on image quality
- Needs retraining for new pattern types
- May fail with poor lighting or occlusions

9. CONCLUSION

"Pattern Sense" successfully demonstrates the viability of deep learning for textile pattern classification. With over 85% accuracy and a user-friendly web interface, the system can be extended to industrial applications. It provides a reliable foundation for AI adoption in fashion-tech.

10. FUTURE SCOPE

- Real-time prediction via mobile cameras
- Classify fabric type (e.g., cotton, silk) along with pattern
- Deploy as REST API for third-party use
- Support edge deployment using TensorFlow Lite

11. APPENDIX

Source Code

- GitHub:

https://github.com/kowshik791/Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning.git

- Demo Link:

https://drive.google.com/file/d/1uOmduIZAH9CElyh8Sz1Gg55QcMymQSH2/view?usp=sharing