AN INTERNSHIP REPORT
ON
Pattern Sense: Classifying Fabric Patterns using Deep Learning

Team ID : LTVIP2025TMID36589

Team Size : 4

TEAM MEMBERS:

Team Leader : Kudipudi Naga Veera Venkata Satyanarayana

Team member : Lokesh Jonnala

Team member : Palli Chandu

Team member : T Venkata Lakshmi Durga

COLLEGE NAME: SWARNANDHRA COLLEGE OF ENGINEERING AND TECHNOLOGY

Final Report: Pattern Sense - Fabric Pattern Classifier

---

1. INTRODUCTION

1.1 Project Overview

Pattern Sense is an AI-powered web application designed to classify fabric patterns using deep learning. Users can upload fabric images or capture them using a camera to identify the pattern type with confidence scores. The tool supports floral, geometric, polka dot, and striped fabric types.

1.2 Purpose

The goal is to streamline the classification of fabrics for manufacturers, designers, and retailers using a trained Convolutional Neural Network (CNN) model based on ResNet50, enabling faster, more accurate, and consistent fabric pattern identification.

---

2. IDEATION PHASE

2.1 Problem Statement

Manual classification of fabric patterns is time-consuming, subjective, and requires domain expertise. There is a clear need for an automated system that improves efficiency and consistency.

2.2 Empathy Map Canvas

The empathy map outlines the needs and experiences of our users—primarily textile professionals and designers.

- **Thinks:**
  "I need accurate results quickly."
  "Is this pattern floral or geometric?"
- **Feels:**
  Frustrated with manual classification.
  Curious about AI's potential.
- **Says:**
  "Manual checking takes too long."
  "I wish I had a faster tool."
- **Does:**
  Uploads or takes photos of fabric.
  Tries to identify patterns by eye.
- **Pain Points:**
  Inaccuracy, time-consuming, inconsistent results.
- **Gains:**
  Fast, accurate predictions.
  Easy-to-use interface.
  Confidence scores and visual feedback.

2.3 Brainstorming

- Leverage CNN for pattern recognition
- Deploy through a Flask-based web interface
- Use transfer learning (ResNet50)
- Implement drag-and-drop and camera input
- Provide prediction confidence visualization

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

A typical user begins their journey by discovering Pattern Sense through online search, referral, or educational content. Initially curious, they explore the website and review its purpose—automating fabric pattern classification using AI. This sparks interest, especially among textile professionals who are familiar with the challenges of manual pattern checking.

As the user interacts with the application, they upload or capture a fabric image and expect quick, accurate results. The system provides an immediate prediction along with a confidence score, which reassures the user about its reliability. If the prediction is accurate and visually supported with preview images, the user is likely to feel impressed and satisfied.

Over time, repeated successful predictions build trust in the system. Users begin to rely on Pattern Sense for routine tasks and may even recommend it to peers. The journey emphasizes the importance of intuitive design, rapid feedback, and accuracy to drive adoption and retention.

3.2 Solution Requirement

Functional Requirements:

- Upload or capture fabric image
- Preprocess and augment input
- Predict pattern class
- Display prediction results with scores

Non-Functional Requirements:

- Secure image upload handling
- Responsive UI with dark/light mode
- High prediction speed and reliability

3.3 Data Flow Diagram

Data collection ➞ Data Pre-Processing ➞ Model Building ➞ Application Building

3.4 Technology Stack

- **Frontend:** HTML, Bootstrap 5, JavaScript
- **Backend:** Flask (Python)
- **Machine Learning:** TensorFlow, Keras
- **Model Architecture:** ResNet50 (pre-trained, fine-tuned)

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

Pattern Sense automates a traditionally manual task, ensuring faster and more accurate pattern classification while reducing human error.

4.2 Proposed Solution

We trained a ResNet50 model using augmented datasets. The model was integrated into a Flask app supporting live image preview, drag-and-drop, and real-time prediction.

4.3 Solution Architecture

The architecture of Pattern Sense is designed to support fast, secure, and accurate fabric pattern classification. It consists of the following key components:

1. **Frontend Interface:**
   A web-based interface built using HTML, Bootstrap, and JavaScript. It allows users to upload images or capture them using a webcam. The interface also displays prediction results and confidence scores in a clean, user-friendly layout with dark/light mode support.

2. **Flask Backend Server:**
   Acts as the core processing unit. It handles routing, manages file uploads, invokes the model for predictions, and returns the results. It ensures input sanitization and manages augmented image generation.

3. **Deep Learning Model (ResNet50):**
   A fine-tuned version of ResNet50 trained on a dataset of labeled fabric images. The model processes incoming images, performs preprocessing, and predicts the most likely pattern class.

4. **Static and Upload Folders:**
   Stores user-uploaded images and augmented previews for rendering on the result page.

5. **Prediction Pipeline:**
   Uploaded images are resized, normalized, and passed through the model. Predictions are generated, sorted by confidence, and returned to the frontend for visualization.

This modular architecture ensures scalability, reliability, and smooth integration between user input, processing, and output display.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase 1 – Problem Analysis & Research (Week 1):

- Defined the problem statement
- Identified target users and pain points
- Researched existing solutions and datasets

Phase 2 – Model Development (Weeks 2–3):

- Prepared and augmented the dataset
- Built and trained the ResNet50-based CNN model
- Evaluated performance using test data

Phase 3 – Web App Design (Week 4):

- Designed user interface using HTML, CSS (Bootstrap)
- Integrated dark/light themes and image upload features

Phase 4 – Flask Integration (Week 5):

- Connected model to frontend via Flask
- Added image preprocessing, prediction logic, and result rendering

Phase 5 – Testing & Deployment (Week 6):

- Conducted functional and performance tests
- Collected feedback
- Deployed the app locally and prepared for hosting

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

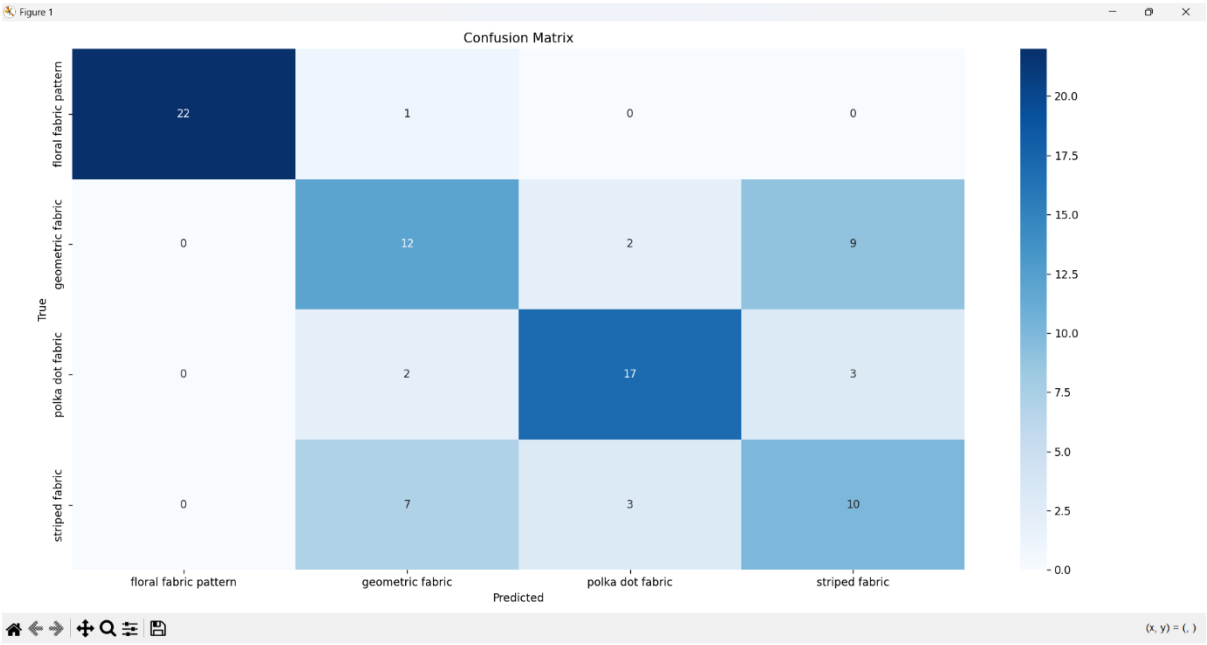
Using test.py, we evaluated the model's performance on unseen test data. It produced accurate predictions with strong class-wise confidence.

Classification Report and Accuracy:

Classification Report:

	precision	recall	f1-score	support
floral fabric pattern	1.00	0.96	0.98	23
geometric fabric	0.55	0.52	0.53	23
polka dot fabric	0.77	0.77	0.77	22
striped fabric	0.45	0.50	0.48	20
accuracy			0.69	88
macro avg	0.69	0.69	0.69	88
weighted avg	0.70	0.69	0.70	88

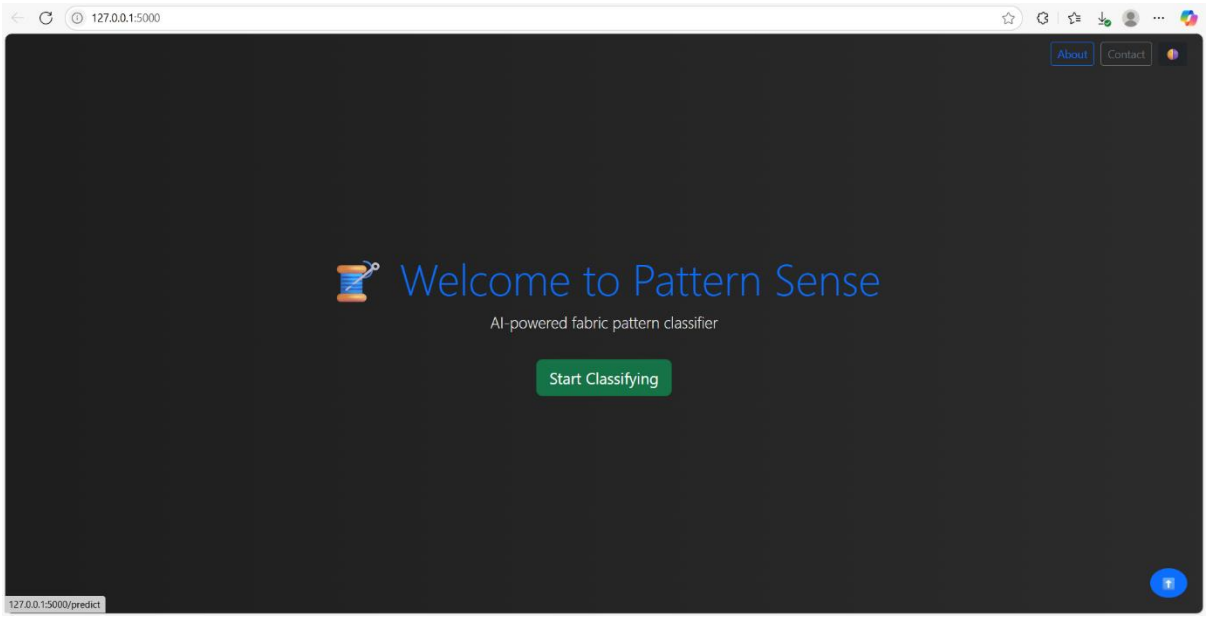
Confusion Matrix Analysis:



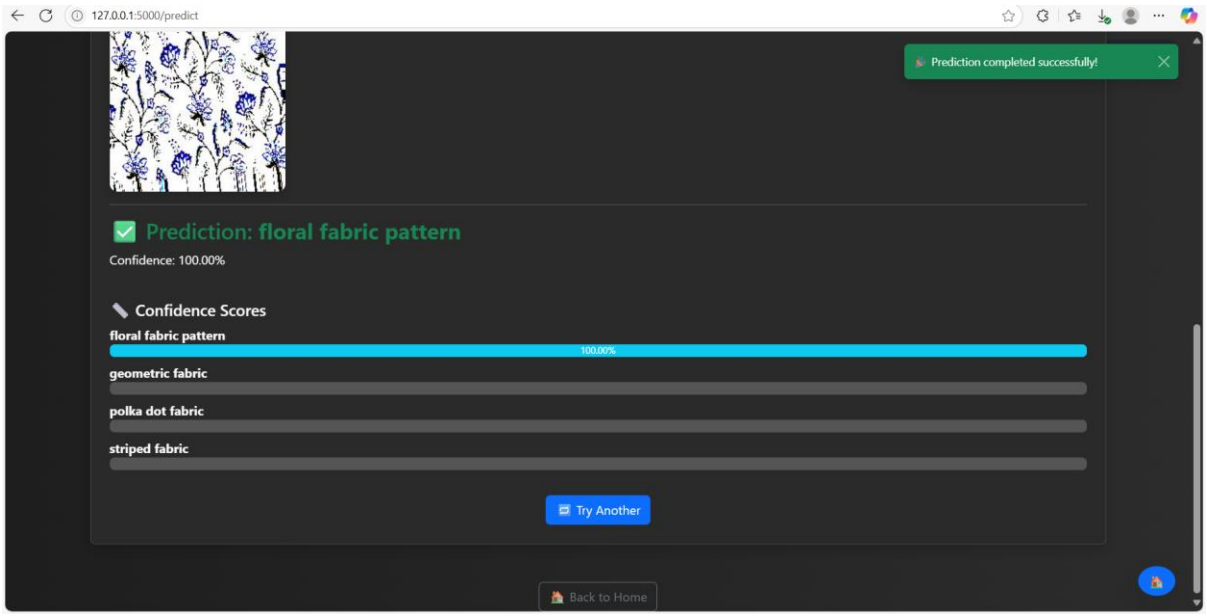
7. RESULTS

7.1 Output Screenshots

- **Home Interface:**



- **Prediction Interface with Confidence Bars:**



- **Augmented Image Output:**

---

8. ADVANTAGES & DISADVANTAGES

Advantages:

- High classification accuracy (powered by ResNet50)
- Drag-and-drop upload and webcam support
- Real-time feedback and prediction explanation
- Responsive light/dark UI

Disadvantages:

- Limited to 4 pattern classes
- Requires GPU for optimal performance
- Dataset must be updated for broader coverage

---

9. CONCLUSION

Pattern Sense provides a practical, intuitive, and efficient solution to classify fabric patterns using AI. It improves upon traditional methods by enhancing reliability, usability, and speed. The project showcases how deep learning can enhance real-world workflows.

---

10. FUTURE SCOPE

- Expand to more fabric patterns and textures
- Integrate feedback loop for model retraining
- Launch Android/iOS version
- Include multilabel prediction and multilingual support
- Real-time prediction via webcam streaming

---

11. APPENDIX

Source Code: Included (app.py, train.py, test.py, home.html, predict.html, etc.)

Source Code Included:

- app.py – Flask backend
- train.py – Model training script
- test.py – Model evaluation
- templates/*.html – Web UI pages
- fabric model_cnn.h5 – Saved model file

🔗 **GitHub**: Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning/Project files at main · satyanarayana5403/Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning
(or provide ZIP/Google Drive link)

Dataset Link: Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning/dataset at main · satyanarayana5403/Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning
GitHub & Project Demo Link: Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning/Video Demo/fabric pattern demo.mp4 at main · satyanarayana5403/Pattern-Sense-Classifying-Fabric-Patterns-using-Deep-Learning

---