**Pattern Sense: Classifying Fabric Patterns Using Deep Learning**

**Project Documentation format**

**1. Introduction**

**Project Title: [Pattern Sense: Classifying Fabric Patterns Using Deep Learning]**

* **Team Members:**

1. Golla Narasimha[22HM1A0536]
2. Boosi Reddy Tejaswini [22HM1A0517]
3. Gaddam Bhagya Lakshmi [22HM1A0532]
4. Batchanaboina Bhavya Srisai [23HM5A0501]

**2. Project Overview**

* **Purpose:**
* The purpose of "PATTERN SENSE: CLASSIFYING FABRIC PATTERNS USING DEEP LEARNING" is to develop a system that automatically identifies and categorizes different fabric patterns using deep learning techniques. This aims to automate a task that is currently often done manually, improving efficiency and accuracy in the textile industry.
* To automatically recognize and classify different fabric patterns (e.g., plain, satin, twill, stripes, plaids, floral) using deep learning, replacing manual inspection and handcrafted feature extraction with an end-to-end, scalable image analysis approach
* **Goals:**
* The main goal of "Pattern Sense: Classifying Fabric Patterns Using Deep Learning" is to automate the process of classifying fabric patterns, specifically using deep learning techniques to improve accuracy and efficiency compared to traditional manual methods.
* This involves developing a system that can accurately identify and categorize different fabric patterns from images.
* The primary goal is to move away from manual, labor-intensive methods of classifying fabric patterns, which are prone to errors and time-consuming
* Automated classification can significantly speed up the process of identifying and categorizing fabric patterns, leading to increased efficiency in textile production and management.
* **Features:**

**Dataset & Preprocessing**

* High-quality fabric images captured under controlled illumination, using consistent focal length and ISO settings for clarity
* Data augmentation to create robust variance: flips, rotations (e.g., every 30°), zoom, shear, brightness changes — boosting generalization and avoiding overfitting

**CNN Architectures & Transfer Learning**

* Pre-trained models like ResNet‑50, VGG‑16/19, Google Net/Inception are fine-tuned for fabric textures — combining strong feature abstraction with task adaptation
* Architecture improvements include identity shortcuts (ResNet) to combat vanishing gradients, small-kernel stacks (VGG), and inception modules for multi-scale feature capture

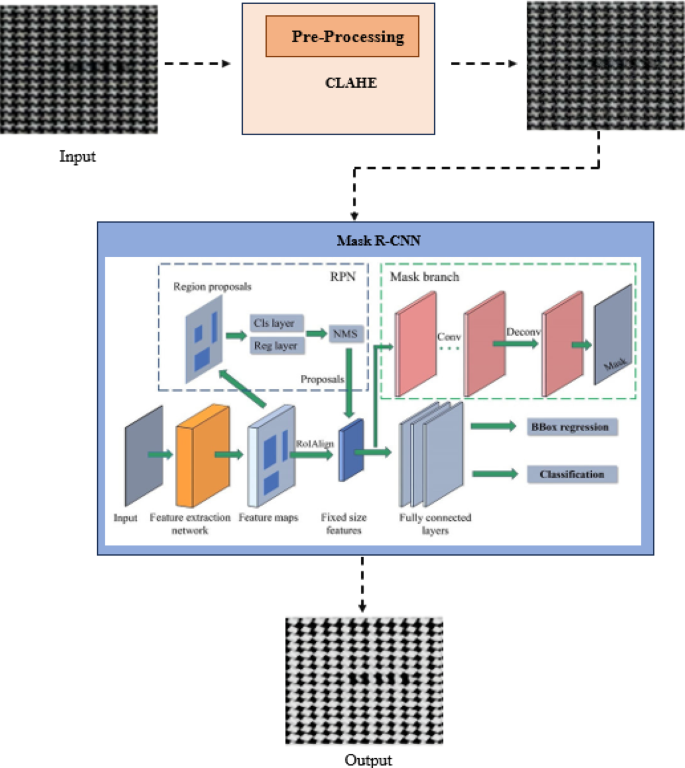
**Texture-Specific Feature Enhancements**

* Feature fusion: Combine CNN features with classical descriptors like HOG, HSV histograms, LBP, and GLCM to enrich shape and color cues
* Attention-enhanced networks such as DenseNet variants emphasize discriminative texture regions, boosting accuracy

**Scalability & Efficiency**

* Integration of depth wise-separable convolutions (e.g., Mobile Net-style) and channel pruning for lightweight and fast inference—vital for deployment on embedded devices
* Optional ensemble or segmentation heads for defect detection, allowing multi-task operation in production

**3. Architecture**



**4. Setup Instructions**

* **Prerequisites**:
* **Installation:**

**5. Folder Structure**

**6. Running the Application**

1. **Backend (Flask Application):**
2. **Frontend:**

**7. API Documentation**

**8. Authentication**

**Future Scope for Authentication (Optional Enhancements):**

**Recommended Future Features:**

**9. User Interface**

**10. Testing**

**Testing Strategy**

**Tools Used**

**11. Screenshots or Demo**

**Screenshots**

The complete execution of the Smart Sorting application is shown in the images step by step as shown below.

**Project Demo Link:**

**12. Known Issues**

**Limited & Biased Datasets**

* Small dataset sizes restrict coverage of pattern diversity. Fabric image datasets are often limited (e.g., 3K–10K images), hurting generalization and risking overfitting
* Sampling bias: Majority class images dominate, underrepresenting rare patterns, so models generalize poorly to unseen types

**CNN Bias Toward Texture Over Shape**

* Pretrained CNNs (e.g., ResNet‑50) tend to overly rely on texture, neglecting shape information. This bias can lead to misclassification under distortion or when fabrics vary substantially
* Mitigation: training with stylized-image augmentation or shape-texture debiasing methods can improve robustness

**Sensitivity to Rotation, Scale, & Lighting**

* Fabric textures change wildly with orientation, zoom, or lighting. Standard CNNs struggle without specific augmentation or encoding mechanisms.
* Wavelet CNNs or Deep‑TEN encoding layers help gain invariance to scale and viewpoint

**Insufficient Texture-Specific Feature Encoding**

* Typical fine-tuning can't fully capture micro-structures in patterns. Advanced modules (e.g., Deep‑TEN, bilinear pooling) improve representation but add complexity and training data requirements

**Computational Bottlenecks**

* High-capacity CNNs (e.g., DenseNet, ResNet) with encoding layers are expensive in memory/compute—problematic for edge devices
* Solutions include compact models, pruning, or knowledge distillation—but may reduce accuracy.

**13. Future Enhancements**

**Topological Deep Learning for Structural Awareness**

* Incorporate topological layers (e.g. persistence homology) to explicitly learn fabric’s multi-scale structure and weave topology—offering robustness to distortions and enhancing texture understanding beyond pixel-level features

**Multi-Modal & Depth-Enhanced Inputs**

* Add RGB-D or multi-view inputs (e.g., depth maps, multi-angle captures) to capture 3D surface features like fabric drape, thickness, and texture shadows—ideal for distinguishing similar weaves

**Advanced Texture Encoding Modules**

* Integrate state-of-the-art modules such as Deep-TEN, wavelet-based CNNs, or mixture-enhancement + attribute clustering to learn richer, more invariant texture representations

**Multi-Task Learning: Defect Detection + Classification**

* Implement unified pipelines combining classification + segmentation/detection heads (e.g., MobileNetV2-SSD-FPN, YOLOv5, U-Net) to detect defects alongside pattern types in industrial contexts

**Lightweight & Efficient Models**

* Apply model compression, pruning, quantization, or distillation to tailor models for edge devices—enabling real-time deployment in resource-constrained manufacturing workflows

**Unsupervised Anomaly Detection**

* Incorporate unsupervised or self-supervised techniques (e.g., motif-based CNNs trained on defect-free fabric) to detect rare or unseen defects with minimal labelling effort

**Domain Adaptation & Robustness Strategies**

* Deploy advanced augmentations (adversarial, style, lighting, geometric), as well as self-training / domain adaptation approaches, to ensure stability across new fabrics, lighting conditions, and production lines

**Explainability & Model Interpretability**

* Use Grad‑CAM, topological insights, or feature-importance mappings to highlight the fabric structures driving decisions—crucial for user trust and model validation in industrial settings.

**Automated Robotic Feedback Integration**

* Connect with robotic knitting/fabrication systems (e.g., reverse-engineering pipelines or CAM integrations) to adapt manufacturing based on detected pattern/defect insights.

**Quantum & Optical Neural Network Prototypes**

* Explore experimental deployment using optical neural networks or quantum ML layers to accelerate on-device inference and support ultra-fast fabric processing systems