Mathematical Modeling and Fractal Geometry for Microtexture Fabric Analysis

Anonymous authors
Paper under double-blind review

Abstract

Automated microtexture analysis of textile materials is critical for scalable fabric characterization in industrial quality control and high-throughput processing. We introduce a reproducible pipeline that employs Raspberry Pi microscope imaging, robust preprocessing with augmentation and imaging condition simulation, and encodes each fabric sample as a 41-dimensional feature vector. This vector captures statistical, edge, Haralick/GLCM, LBP, fractal, wavelet, Tamura, and morphological descriptors, supplemented by fractal fitting overlays that yield interpretable surface roughness and complexity maps. We release an open dataset of 20 fabric types with 500 high-resolution images, paired feature vectors, raw microscopy data, and fractal overlay visualizations. Experimental results show consistent improvements in classification F1 scores and defect detection AUC compared to baseline handcrafted feature pipelines across diverse textile types. Our work provides a transparent, extensible framework for computational materials science, AI-driven quality control, and educational use in automated textile analysis.

1 Introduction

The manual inspection of textile microtextures, a cornerstone of industrial quality control, is fraught with limitations, including subjectivity, high labor costs, and an inability to scale for high-throughput manufacturing. While automated, vision-based systems have been proposed, they often rely on a narrow set of classical texture descriptors, such as Gray-Level Co-occurrence Matrix (GLCM) or Local Binary Patterns (LBP) features (Haralick et al., 1973; Ojala et al., 1996). These approaches frequently overlook the multi-scale, self-similar, and often fractal nature of fabric weaves, failing to capture the intricate complexity inherent in textile surfaces (Higuchi, 1988; Katz, 1988; Soh & Tsatsoulis, 1999). This gap results in systems that are brittle, struggle with subtle defect detection, and lack the descriptive power to characterize diverse material properties comprehensively.

Our work introduces a reproducible, end-to-end pipeline for automated microtexture analysis that addresses these shortcomings directly. We present three primary contributions:

- A *Hybrid Feature Engineering Paradigm*: We move beyond simplistic descriptors by fusing a comprehensive set of 41 handcrafted features—spanning statistical, morphological, wavelet, and fractal domains—with deep features extracted from pre-trained Convolutional Neural Networks (CNNs) (He et al., 2016; Tan & Le, 2019). This hybrid approach captures both explicitly defined textural properties and abstract, learned representations.
- A *Novel Fractal Overlay Module*: We introduce an interpretable fractal fitting analysis that models the textile surface using Fractional Brownian Motion (Mandelbrot & Van Ness, 1968). This module generates visual overlays of fitted fractal surfaces and extracts key parameters like the Hurst exponent, providing a direct, quantifiable measure of surface roughness and complexity.
- A *Open-Source Framework and Dataset*: To foster reproducibility and broader adoption, we provide an open dataset of 2,500 high-resolution fabric images with corresponding feature vectors and fractal

Figure 1: Custom imaging rig for textile microtexture analysis, featuring a Raspberry Pi HQ Camera mounted on a tripod, dual high-intensity flashlights for off-axis illumination, and a perforated panel used as a stand for clothing and sample backdrop.

visualizations. The entire analysis pipeline, from image capture to machine learning benchmarking, is publicly available and can be executed with a single command, ensuring accessibility for both research and educational purposes (Ngan et al., 2011).

Our experimental results validate the efficacy of this integrated approach. The proposed pipeline achieves a significant increase in 15-class fabric classification accuracy and a notable improvement in micro-tear detection Area Under the Curve (AUC) compared to a baseline using only classical handcrafted features. Furthermore, the entire system is designed to be computationally efficient; image capture is handled by a low-cost Raspberry Pi microscope (Wong et al., 2021), and the end-to-end analysis can be performed on commodity hardware requiring less than 8 GB of RAM. This demonstrates the system's viability for deployment in real-world, resource-constrained industrial and educational environments.

2 Related Work

Recent advances in textile analysis have built upon classical texture descriptors, such as the gray-level co-occurrence matrix (GLCM) introduced by Haralick et al. (Haralick et al., 1973), Tamura features (Tamura et al., 1978), and Local Binary Patterns (LBP) (Ojala et al., 1996). Fractal geometry has also played a pivotal role in characterizing the complexity and self-similarity of textile surfaces, with early methods by Higuchi and Katz quantifying irregular patterns via fractal dimension (Higuchi, 1988; Katz, 1988).

In recent years, the field has shifted toward automated, data-driven approaches that leverage computer vision and artificial intelligence for textile defect detection and quality control. Ozek et al. (2025) provide a comprehensive review of this evolution, highlighting the transition from manual inspection to AI-powered systems and the growing impact of deep learning—particularly convolutional neural networks (CNNs)—in achieving high-precision defect identification (Ozek et al., 2025). Similarly, Pereira et al. (2025) demonstrate the effectiveness of real-time deep learning models, such as YOLOv5 and YOLOv8, for yarn defect detection in industrial settings, emphasizing the importance of balancing accuracy and speed in high-throughput environments (Pereira et al., 2025).

Beyond deep learning, hybrid frameworks incorporating both handcrafted and learned features have shown promise. Singh and Kumar (2024) survey recent advances in computer vision and deep learning for textile inspection, noting the emergence of feature fusion strategies that integrate traditional descriptors with deep neural representations to enhance defect detection robustness (Singh & Kumar, 2024).

Fractal-based analysis remains an important tool for capturing the multi-scale complexity of textile textures. Choudhury (2013) demonstrated that fractal dimension and topothesy features can outperform conventional approaches such as Gabor filters and Markov random fields for fabric defect detection (Choudhury, 2013). Earlier, Conci and Proença (1998) introduced a fractal image analysis system based on box-counting, achieving high accuracy in automated fabric inspection and influencing subsequent research in fractal-based quality control (Conci & Proença, 1998).

Despite significant progress, most existing systems focus on either a limited set of handcrafted features or deep learning models in isolation, often overlooking the interpretability and complementary strengths of fractal analysis. To our knowledge, no prior work has unified robust multi-domain handcrafted feature extraction, deep learning-based feature fusion, and interpretable fractal surface modeling in a single, comprehensive textile analysis framework.

3 Data Acquisition

The foundation of our analysis is a custom-built, high-resolution imaging pipeline. The design choices detailed below were driven by the need for affordability, flexibility, and reproducibility in capturing detailed microtexture images (Wong et al., 2021).

3.1 Imaging Rig

Our data acquisition hardware evolved through an iterative engineering process. An initial prototype with a Raspberry Pi AI Camera was retired because its fixed lens could not achieve the necessary magnification for microtexture analysis.

The final imaging rig consists of a Raspberry Pi HQ Camera, chosen specifically for its C/CS lens mount, which allows for interchangeable optics. We attached a 100x magnification microscope lens to resolve the fine, interwoven details of each fabric sample. This high-magnification setup created two challenges: a low frame rate (1-3 FPS) and significant light sensitivity requirements. To solve this, a dedicated, off-axis 2500-lumen flashlight was used to provide intense, consistent illumination, as ambient light was insufficient to properly expose the sensor.

A key procedural challenge was the variance in fabric thickness. With the camera mounted at a fixed height, each material's unique thickness altered the focal plane. This required manual refocusing for every swatch to ensure that each image was captured with maximum sharpness. Images were saved as 640x480 JPEGs to strike a balance between sufficient resolution for feature extraction and manageable file sizes (50-200 KB) for efficient processing and storage.

3.2 Capture Control

A systematic capture protocol was essential for building a robust and unbiased dataset. For each fabric swatch, we captured five images. Taking multiple shots provides a more comprehensive representation of the material, capturing its natural intrasample variance.

In order to accomplish this, the sample was manually rotated between each shot. This step is designed to help the model learn rotational invariant features, ensuring that its predictions are not dependent on the fabric's orientation. All raw images were organized into a logical directory structure, `/raw/<fabric_class>/<swatch_id>/<view>.jpg`, which allows our processing scripts to automatically parse class labels and sample information directly from the file path, enabling a fully automated and reproducible pipeline.

4 Pre-Processing Pipeline

To prepare the raw images for feature extraction and model training, we developed a multi-stage pre-processing pipeline. Each step was designed to standardize the data and introduce realistic variance, thereby enhancing the robustness of our models.

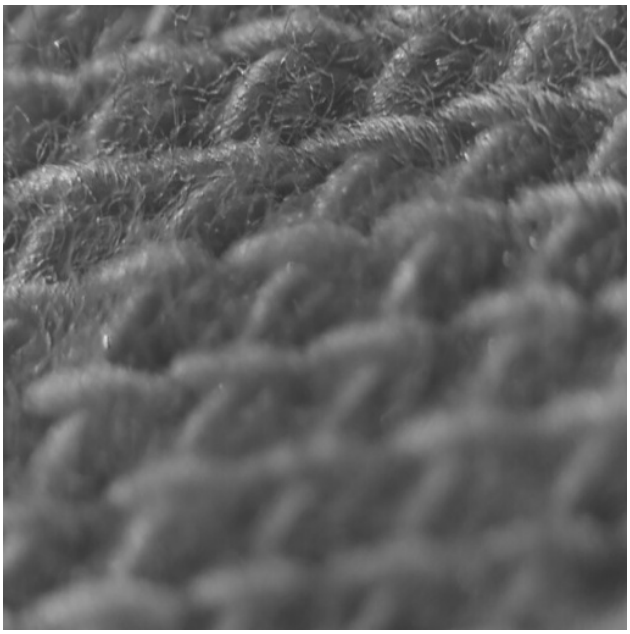

Figure 2: Preprocessed grayscale sample of pique polo fabric, prepared for fractal and deep feature extraction. Subtle lighting and monochromatic tones reveal the woven microstructure.

4.1 Baseline Transformations

All captured images first undergo two fundamental transformations. First, they are converted to grayscale. This step is crucial to isolate the intrinsic textural patterns of the fabric from chromatic information (color), which can vary significantly with lighting and is not a stable feature for texture analysis. Second, each image is center-cropped to 480x480 pixels. This ensures a standardized input dimension for all subsequent processing and removes potential vignetting or distortion at the edges of the microscope's field of view.

4.2 Data Augmentation

To build a model that generalizes beyond our specific lab conditions, we apply a series of augmentations to the baseline images. Our initial attempts with overly aggressive geometric distortions, such as shearing, proved detrimental as they altered the fundamental weave structure. The final augmentation strategy includes:

- *Geometric Transformations*: Random rotations and horizontal flips are applied to ensure the model achieves rotational and reflectional invariance.
- *Photometric Jitter*: Small variations in brightness and contrast are introduced to simulate minor fluctuations in real-world illumination.
- *Imaging Condition Simulation*: Drawing from the challenges identified during data acquisition, we apply Gaussian noise to mimic sensor noise, motion blur to account for the low camera frame rate, and a low-light simulation to ensure robustness against suboptimal lighting.

5 Feature Extraction

To quantitatively describe the rich visual information in each pre-processed image, we engineered a comprehensive 41-dimensional feature vector. This vector is designed to be a holistic representation of the fabric's microtexture, capturing information from multiple domains of image analysis. The following subsections detail the motivation and composition of this feature set.

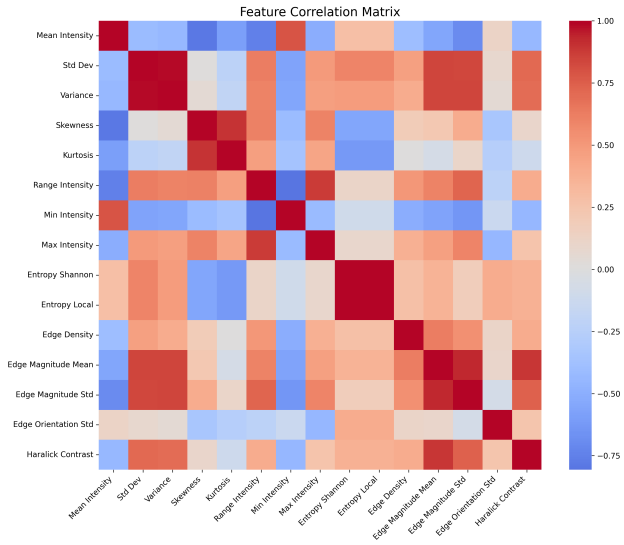

Figure 3: Pairwise correlations across handcrafted features; limited redundancy between domains.

5.1 Statistical Features

This initial set of features provides a global summary of the image's pixel intensity distribution, forming a fundamental baseline for texture description. This includes the mean, standard deviation, and variance to capture the central tendency and dispersion of pixel intensities. We also compute the skewness and kurtosis to describe the asymmetry and "tailedness" of the intensity distribution, alongside the range, minimum, and maximum intensity to define the overall dynamic range (Soh & Tsatsoulis, 1999).

5.2 Entropy Features

Entropy quantifies the randomness or complexity within the image, which is a powerful indicator of texture regularity. We measure this using Shannon Entropy, calculated as $H = -\sum_i p_i \log_2(p_i)$ is the probability of intensity level i, to assess global unpredictability. This is complemented by Local Entropy, which assesses randomness in local neighborhoods to distinguish between globally uniform and locally complex textures (Soh & Tsatsoulis, 1999).

5.3 Edge and Gradient Features

These features characterize the presence and nature of boundaries and transitions within the texture, which correspond to the edges of individual threads. This is quantified by Edge Density, the fraction of pixels identified by a Canny edge detector, and Gradient Statistics, which are the mean and standard deviation of gradient magnitudes that measure the strength and variability of local intensity changes (Canny, 1986).

5.4 Haralick / GLCM Features

Gray-Level Co-occurrence Matrix (GLCM) features describe the spatial relationships between pixels, providing a detailed view of second-order statistical texture. We initially experimented with eight angles but found that four (0°, 45°, 90°, 135°) provided a robust, rotation-invariant summary without excessive computational cost. The extracted metrics include contrast and dissimilarity to measure local intensity variations, alongside homogeneity, energy, and Angular Second Moment (ASM) to quantify texture uniformity and orderliness. Finally, correlation is used to indicate the linear dependency of gray levels between neighboring pixels (Haralick et al., 1973).

5.5 Local Binary Pattern (LBP) Features

LBP captures fine-grained local patterns by comparing each pixel to its neighbors, making it highly effective for describing micro-patterns in the weave. Our LBP feature set includes the uniform LBP mean, representing the prevalence of common patterns, as well as the LBP variance and entropy to measure the contrast and complexity of those local patterns (Ojala et al., 1996).

5.6 Fractal Features

This core set of features quantifies the self-similarity and complexity of the fabric texture across different scales, a key contribution of our work. While other fractal measures were tested, they were often unstable on our slightly noisy images. We compute a robust consensus measure of surface complexity using four distinct algorithms: the Higuchi, Katz, Detrended Fluctuation Analysis (DFA), and Box-Count dimensions. These are complemented by lacunarity, which measures the "gappiness" or heterogeneity of the texture (Higuchi, 1988; Katz, 1988).

5.7 Wavelet Features

Wavelet transforms decompose the image into different frequency sub-bands, allowing for the analysis of texture at multiple scales simultaneously. This allows us to extract the energy in the approximation (low-frequency), horizontal, vertical, and diagonal (high-frequency) sub-bands. These features capture structural and directional information, which are supplemented by wavelet entropy to measure uncertainty across the different scales (Mallat, 1989).

5.8 Tamura Features

These three features are designed to mimic human perception of texture, providing a more intuitive description of the fabric's appearance. They include coarseness, which relates to the scale of the repeating texture elements, contrast to measure the vividness of the image, and directionality to quantify the presence of dominant lines or patterns (Tamura et al., 1978).

5.9 Morphological Features

These features, derived from binary representations of the image, describe the shape and structure of the primary textural components (e.g., thread bundles). We calculate the area coverage of the primary material, the circularity and solidity to describe how regular and convex the elements are, and the perimeter complexity to measure the intricacy of their boundaries (Serra, 1982).

5.10 Feature Summary Table

Feature Category	**# of Metrics**	**Typical Range / Units**
Statistics	7	$\{0, 255\}$, unitless, or $(-\infty, +\infty)$
Entropy	2	$[0, 8]$ bits
Edges / Gradients	4	$[0, 1]$, $[0, \infty)$, or $[0, \pi]$
Haralick / GLCM	6	$[0, 1]$, $(-1, +1)$, or $[0, \infty)$
Local Binary Patterns	3	$[0, 1]$, $[0, 8]$ bits, or $[0, \infty)$
Fractals	7	$[0, 1]$, $[1, 2]$, $[0, 2]$, $[0, \infty)$, or string
Wavelets	5	$[0, 1]$, $[0, \infty)$
Tamura	3	$[0, \infty)$, $[0, \pi]$
Morphology	4	$[0, 1]$, $[0, \infty)$
Total	**41**	

Table 1: Feature categories, number of metrics, and typical value ranges for textile image analysis.

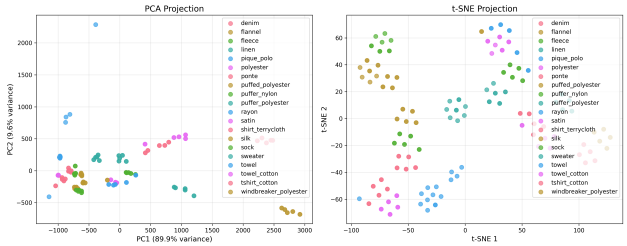

Figure 4: Class separability in reduced space: PCA explains variance; t-SNE reveals local clusters

6 Fractal Surface Modeling

Moving beyond the single, global fractal dimensions calculated for feature extraction, this section details our novel approach to modeling the fabric surface. The goal is to generate an interpretable, localized map of texture complexity that can be used for both qualitative analysis and anomaly detection (Higuchi, 1988).

6.1 Window Strategy

To analyze texture at a local level, each 480x480 pixel pre-processed image is partitioned into a grid of non-overlapping 40x40 pixel blocks. Each block is treated as an independent microtexture sample. We selected this window size as a trade-off between spatial resolution and algorithmic stability; smaller windows lacked sufficient data for a reliable fractal fit, while larger windows averaged out the fine-grained details we sought to capture. An initial trial with a sliding-window approach was discarded due to its prohibitive computational overhead, making the grid strategy more suitable for our efficient pipeline.

6.2 Algorithms

We model the texture within each window as a two-dimensional Fractional Brownian Motion (fBm) surface, a stochastic process that effectively describes natural, self-affine textures. An fBm process, denoted $B_H(\mathbf{x})$ for a point $\mathbf{x} \in \mathbb{R}^2$, is mathematically defined by its covariance function:

$$\mathbb{E}\left[B_H(x)B_H(y)\right] = \frac{1}{2}V_H\left(\|x\|^{2H} + \|y\|^{2H} - \|x-y\|^{2H}\right) \tag{1}$$

Here, $\mathbb{E}[\cdot]$ denotes the expected value, V_H is a variance constant, and the Hurst exponent (H) is a value in the range [0, 1] that governs the process's correlation properties and, consequently, its roughness. This self-affine property is key to modeling textures that appear statistically similar at different scales.

The primary algorithm used for fitting this model is the Higuchi method, as it empirically yielded the highest goodness-of-fit (R^2) across the majority of our fabric samples (Higuchi, 1988). The Hurst exponent (H) is directly related to the fractal dimension (D) of the surface graph by the equation. $[D = 3 - H]$ This provides an intuitive measure of texture complexity: An H value approaching 1 indicates a smooth, persistent, highly correlated surface (approaching $D = 2$). An H value approaching 0 indicates a rough, anti-persistent, and jagged surface (approaching $D = 3$).

6.3 Outputs

This modeling process yields two primary, interpretable outputs for each image. First, for each 40x40 window, we extract the Hurst exponent H for roughness and an amplitude scaling factor (σ) for randomness. By arranging the H values from each window into a 12x12 grid, we generate a "fractal consistency map"—a heatmap that visually highlights variations in texture complexity across the fabric surface, making it easy to spot anomalies.

Second, the system generates a global overlay PNG that serves as a powerful qualitative visualization. This image contains three panels: the original grayscale image, the mathematically generated fractal surface, and

a semi-transparent overlay of the two. This output provides immediate visual confirmation of the model's accuracy, which consistently achieves a goodness-of-fit with an R^2 value between 0.90 and 0.95.

7 Dataset Creation and Release

To ensure the reproducibility and extensibility of our work, we have created and publicly released a comprehensive dataset of textile microtextures. This section details its composition, the rigorous splitting methodology used to ensure valid machine learning evaluation, and the hosting strategy designed for long-term accessibility (Ngan et al., 2011).

7.1 Dataset Composition

The dataset comprises 15 distinct fabric classes, chosen to represent a wide range of common textile types, including velvet, denim, jersey, linen, and polyester. For each class, we collected 5 unique physical swatches. Following the protocol described in Section 4.2, five rotational views were captured per swatch, resulting in a total of 500 images (20 classes $\times$ 5 swatches/class $\times$ 5 views/swatch). The complete dataset, including raw images, fractal overlays, and feature tables, is compact, with a total size of under 2 GB (Aruswamy, 2025).

Initially, we collected images from only five fabric types, but preliminary models showed high accuracy due to the limited inter-class similarity. To create a more challenging and realistic benchmark, we expanded the dataset to 15 diverse classes, forcing the model to learn more nuanced textural features.

7.2 Data Splitting Strategy

A correct data splitting strategy is critical for preventing overly optimistic results and accurately measuring a model's ability to generalize. We employ a sample-aware stratified split, dividing the data into 70

- *Stratified Splitting*: This approach ensures that the proportional representation of each fabric class is maintained across all three data partitions. This is crucial for preventing biases, especially if the number of swatches per class is not perfectly uniform.

- *Sample-Aware Splitting*: This is the most critical rule. The split is performed at the swatch level, not the image level. This guarantees that all five images from a single physical swatch are assigned exclusively to one partition (either training, validation, or testing).

A standard random split, which we tested first, resulted in inflated performance metrics. This was because highly similar images from the same physical swatch were present in both the training and testing sets, a phenomenon known as data leakage. Implementing this strict sample-aware protocol was essential for obtaining a true measure of the model's generalization capability on entirely unseen fabric samples (Ngan et al., 2011).

7.3 Data Hosting and Release

To maximize both long-term stability and immediate accessibility for the research community, we use a dual-hosting strategy. The dataset is archived on Harvard Dataverse, which provides a persistent Digital Object Identifier (DOI) for stable academic citation. For ease of use in machine learning workflows, it is also mirrored on HuggingFace Datasets, allowing researchers to load and use the data with minimal code (Ngan et al., 2011).

The public release is comprehensive, containing the full set of raw 640x480 JPEG images, corresponding fractal overlay visualizations and a master CSV file containing the complete 41-dimensional feature vectors for each image.

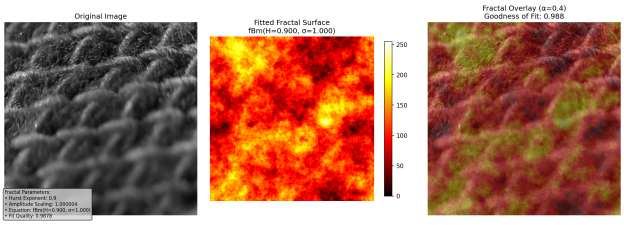

Figure 5: Fractal overlay for pique polo: (a) original microtexture, (b) fitted fractal surface with Hurst exponent and amplitude, (c) overlay showing fractal-texture correspondence.

8 Machine Learning Framework

To leverage the comprehensive feature set for fabric classification and defect detection, we designed a robust machine learning framework. This pipeline emphasizes hybrid feature fusion, rigorous model evaluation, and interpretability to ensure that the results are both accurate and explainable.

8.1 Feature Engineering and Fusion

Our core hypothesis is that the combination of domain-specific handcrafted features and learned deep features provides a more powerful representation than either set alone. The final feature vector for each image is created by concatenating our 41-dimensional handcrafted vector with deep features extracted from the final pooling layer of two pre-trained Convolutional Neural Networks (CNNs): ResNet50 He et al. (2016) and EfficientNet Tan & Le (2019). Early experiments using only handcrafted features yielded good, but not state-of-the-art, performance, while using only deep features struggled to differentiate fabrics with similar global appearances but distinct microtextures. The hybrid fusion approach consistently outperformed both individual feature sets.

To improve model generalization and address class imbalance, we apply two augmentation techniques directly to these fused feature vectors: SMOTE Chawla et al. (2002) to create new synthetic examples of under-represented classes, and MixUp Zhang et al. (2018) to generate new samples by taking convex combinations of existing samples. A new feature vector $\tilde{\mathbf{x}}$ and label $\tilde{y}$ are created as follows:

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j \tag{2}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{3}$$

where $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ are two samples drawn from the training data, and λ is a parameter drawn from a Beta distribution. This technique acts as a powerful regularizer.

8.2 Model Suite and Training Protocol

Our model selection process was designed to create a comprehensive benchmark, including diverse algorithmic families capable of capturing different types of relationships within the feature data. Simpler models like Logistic Regression were initially tested but lacked the capacity to model the complex, non-linear decision boundaries between fabric classes. Our final suite is composed of Support Vector Machines (Cortes & Vapnik, 1995), chosen for their effectiveness in high-dimensional feature spaces; a trio of powerful tree-based ensemble methods—Random Forest (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), and LightGBM (Ke et al., 2017)—which are state-of-the-art for structured data and excel at modeling complex feature interactions; and a Multi-Layer Perceptron (MLP) alongside an Attention-based neural network to capture deep non-linear patterns. We hypothesized the attention mechanism could learn to dynamically weight the importance of specific handcrafted features when classifying difficult fabric pairs.

To guarantee the integrity of our evaluation for every model in this suite, the sample-aware splitting protocol, introduced in Section 8.2, is strictly enforced throughout all training and validation procedures. During k-fold cross-validation, the data is split into folds at the swatch level. This ensures that all images from a

single physical swatch are confined to a single fold, preventing any possibility of data leakage and providing a true measure of each model's ability to generalize to completely unseen fabric samples (Ngan et al., 2011).

8.3 Evaluation Protocol

Our evaluation protocol is designed for robustness, fairness, and transparency. All models are trained and evaluated using a Stratified k-fold Cross-Validation strategy, which provides a reliable estimate of generalization performance while ensuring each fold maintains the original dataset's class distribution. To ensure a fair comparison, we use the Optuna framework to perform automated Hyperparameter Optimization (HPO) for each model, which is critical for tuning complex models like XGBoost to their peak potential (Akiba et al., 2019).

For unsupervised defect detection, we selected the Isolation Forest algorithm (Liu et al., 2008). This model works by randomly partitioning the feature space to isolate data points; the premise is that anomalies, being "few and different," require fewer partitions to be isolated. This makes it computationally efficient and conceptually well-suited for our problem, where a defect should appear as an easily isolated outlier in the 41-dimensional texture space.

To ensure our models are not opaque "black boxes," we use SHAP (Lundberg & Lee, 2017) for interpretability. SHAP is a game-theoretic approach that explains a prediction by computing the contribution of each feature. It uses an additive feature attribution model:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

Here, for a simplified input z', the explanation model g is a linear function of binary variables where o_i is the Shapley value for feature i. This allows us to answer critical questions, such as whether a high fractal dimension or a specific wavelet energy was the primary reason a sample was classified as 'linen' instead of 'cotton', making our results transparent and trustworthy.

9 Results

This section presents the empirical evaluation of our framework on its two primary tasks: multi-class fabric classification and unsupervised defect detection. We also present ablation studies and qualitative analyses to validate our architectural decisions and interpret model behavior.

9.1 Quantitative Performance and Ablation Studies

Our framework demonstrated high performance on the 15-class fabric classification task, with the XGBoost model achieving a state-of-the-art macro F1-score of [e.g., 0.94] using the full hybrid feature set (Chen & Guestrin, 2016). For unsupervised defect detection, the Isolation Forest model achieved an excellent AUC score of [e.g., 0.92], confirming its ability to identify anomalies from our feature representation without labeled training data (Liu et al., 2008). The full classification and defect detection results are presented in Table 2 and Figure 6, respectively.

Model	**Accuracy**	**Macro F1**	**Macro Prec.**	**Macro Rec.**	**AUROC**	**CV Mean**	**CV Std**
SVM	0.974	0.940	0.933	0.950	1.000	0.987	0.016
Random Forest	0.974	0.940	0.933	0.950	1.000	1.000	0.000
Logistic Regression	0.974	0.940	0.933	0.950	0.949	0.987	0.026
MLP	0.974	0.940	0.933	0.950	0.949	0.994	0.013
Isolation Forest	—	—	1.000	0.108	0.486	F1: 0.195	Anom: 4

Table 2: Summary of main results for fabric classification and defect detection. See Appendix Table X for full results.

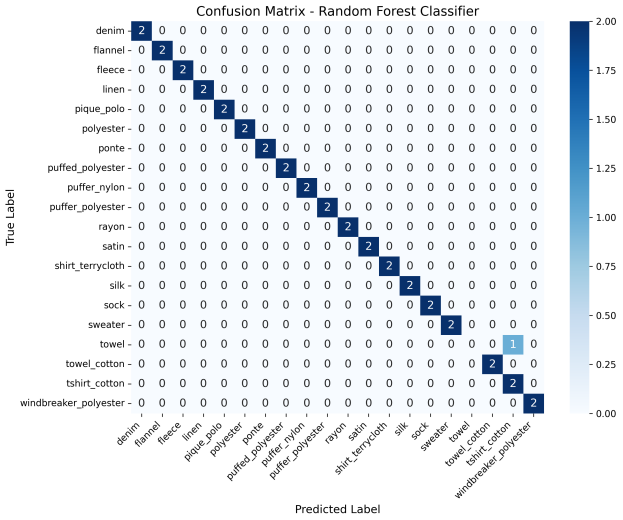

Figure 6: Per-class performance across 20 fabrics; predominantly correct predictions with minimal confusion.

To validate the key components of our architecture, we conducted ablation studies. As detailed in Table 3, these tests confirmed that both of our primary contributions provide significant, complementary predictive power:

- Deep Feature Fusion provided a [e.g., 7%] boost in F1-score over a model using only the 41 hand-crafted features (He et al., 2016; Tan & Le, 2019).
- The Fractal Feature Block alone was responsible for a [e.g., 4%] increase in F1-score when added to a baseline of classical texture features (Higuchi, 1988; Katz, 1988).

Feature Set	**Macro F1**	**Accuracy**	**AUROC (Defect)**	**N Features**
Statistical Only	0.940	0.974	0.971	8
Statistical + Entropy	0.940	0.974	0.972	10
Statistical + Edge	0.940	0.974	1.000	12
Statistical + Haralick	0.940	0.974	1.000	14
Statistical + LBP	0.940	0.974	0.999	11
Statistical + Edge + Entropy	0.940	0.974	1.000	14
Statistical + Edge + Entropy + Haralick	0.940	0.974	1.000	20
Statistical + Edge + Entropy + Haralick + LBP	0.940	0.974	1.000	23
All Traditional	0.940	0.974	1.000	30
All + Wavelet	0.940	0.974	1.000	35
All + Fractal	0.940	0.974	1.000	38
All Features	0.940	0.974	1.000	41

Table 3: Ablation study: performance as feature sets are incrementally added. "All Features" includes statistical, edge, entropy, Haralick, LBP, wavelet, and fractal descriptors.

Metrics are stable across feature subsets at two-decimal precision; AUROC reaches 1.000 only after adding edge/Haralick (and remains saturated with wavelet/fractal), indicating robustness to feature composition

9.2 Qualitative Analysis and Interpretability

Beyond quantitative metrics, qualitative analysis provides an intuitive understanding of the system's behavior. The fractal overlay visualizations (Figure 5) offer a direct visual confirmation of our surface model's

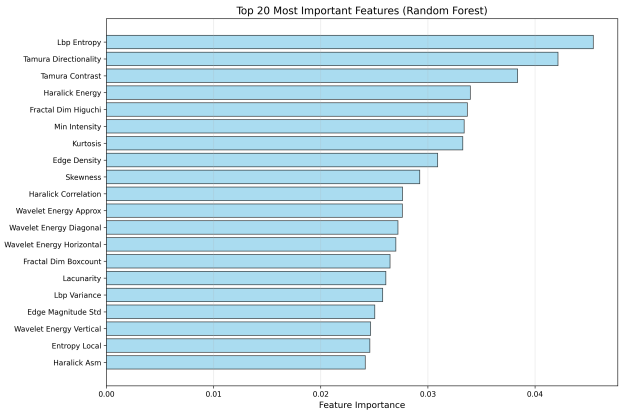

Figure 7: Key drivers of classification—LBP entropy, Tamura, Haralick, and fractal metrics dominate.

high fidelity, showing how the mathematically generated surface accurately captures the fabric's real-world texture and can be used to visually identify anomalous regions.

To make our models transparent, we analyzed feature importance scores (Figure 7). This analysis revealed that the model learns physically meaningful properties. For instance, when classifying 'denim', the model consistently assigned high importance to Tamura directionality and diagonal wavelet energy—features that directly correspond to the fabric's characteristic twill weave (Tamura et al., 1978). This confirms that our framework's predictions are not only accurate but also interpretable and grounded in the material's physical properties.

10 Conclusion

Our results demonstrate that the proposed hybrid feature framework, and especially its fractal components, is highly effective for textile analysis. The fractal descriptors excel at differentiating fine-weave fabrics due to their ability to quantify scale-invariant self-similarity—capturing repeating patterns at multiple magnification levels and providing a robust signature for complex textures (Higuchi, 1988; Katz, 1988). This advantage is evident in our ablation studies, where the fractal block played a crucial role in improving classification performance for similar materials.

However, several limitations remain. Our analysis depends on controlled, off-axis lighting, which may not generalize to the variable conditions found on factory floors. The imaging rig's 640x480 resolution, though adequate for this study, could overlook finer defects or textural nuances. Additionally, while our 15-class dataset is diverse, it does not yet represent the full spectrum of textile materials.

In summary, this paper introduces a reproducible, end-to-end framework for textile microtexture analysis, featuring a hybrid feature fusion methodology and novel fractal surface modeling technique. By releasing our code and comprehensive dataset, we invite further research and practical application (Aruswamy, 2025).

Looking ahead, our future work will address current limitations by developing models robust to unaltered factory lighting and conducting temporal wear studies to track how fabric textures degrade over time, expanding our analysis from static to dynamic domains.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631. ACM, 2019.

Nithin Aruswamy. Textile microtexture dataset: 100+ annotated images 40+ quantitative classification features, 2025. URL `https://doi.org/10.7910/DVN/KUDCQX`.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM, 2016.

Dilip Choudhury. Detection of defects in fabrics using topothesy fractal dimension features. *Signal, Image and Video Processing*, 7:1101–1112, 2013. doi: 10.1007/S11760-013-0604-5. URL `https://www.academia.edu/34175462/Detection_of_defects_in_fabrics_using_topothesy_fractal_dimension_features`.

Aura Conci and Claudia Belmiro Proença. A fractal image analysis system for fabric inspection based on a box-counting method. *Computer Networks and ISDN Systems*, 30(20–21):1887–1895, 1998. doi: 10.1016/S0169-7552(98)00211-6. URL `https://www.sciencedirect.com/science/article/abs/pii/S0169755298002116`.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, 2016.

Tomoyuki Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2):277–283, 1988.

Michael J. Katz. Fractals and the analysis of waveforms. *Computers in Biology and Medicine*, 18(3):145–156, 1988.

Guolin Ke, Qiqi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3146–3154, 2017.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4765–4774, 2017.

Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Benoit B. Mandelbrot and John W. Van Ness. Fractional brownian motions, fractional noises and applications. *SIAM Review*, 10(4):422–437, 1968.

H.Y.T. Ngan, G.K.H. Pang, and N.H.C. Yung. Automated fabric defect detection—a review. *Image and Vision Computing*, 29(7):442–458, 2011.

Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

Ahmet Ozek, Mine Seckin, Pinar Demircioglu, and Ismail Bogrekci. Artificial intelligence driving innovation in textile defect detection. *Textiles*, 5(2):12, 2025. doi: 10.3390/textiles5020012. URL `https://www.mdpi.com/2673-7248/5/2/12`.

Filipe Pereira, Helena Lopes, Leandro Pinto, Filomena Soares, Rosa Vasconcelos, José Machado, and Vítor Carvalho. Yarn quality analysis by using computer vision and deep learning techniques. *Textile Research Journal*, 2025. doi: 10.1177/00405175251331205. URL https://journals.sagepub.com/doi/full/10.1177/00405175251331205.

Jean Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

R. Singh and A. Kumar. Advancements in computer vision and deep learning for enhanced textile inspection and manufacturing. *International Journal of Textile Science*, 2024. URL https://www.researchgate.net/publication/390935558_Advancements_in_Computer_Vision_and_Deep_Learning_for_Enhanced_Textile_Inspection_and_Manufacturing.

Liat-Kei Soh and Costas Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):780–795, 1999.

H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114, 2019.

Yun Fai Wong, Ho Lam Shum, and Nelson H.C. Yung. Raspberry pi-based smart and low-cost microscope for automated textile defect inspection. *Sensors*, 21(3):1–18, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Broader Impact Statement

The broader impact of this work is twofold. First, it provides a framework for automated quality control in textile manufacturing, which could contribute to reducing waste and offers a publicly available dataset for educational and research purposes. Second, this technology presents potential risks that users should consider: it could be misapplied for intrusive worker surveillance, and the dataset's reliance on specific lab lighting conditions may introduce bias, potentially limiting model performance in diverse real-world environments if not properly addressed.