

Article

Not peer-reviewed version

---

# Learnable Residual Local Binary Patterns: A Pretraining Preserving Architecture for Cotton Percentage Estimation in RGB Fabric Images

---

[Arwa Basbrain](#)\*

Posted Date: 3 June 2026

doi: 10.20944/preprints202606.0302.v1

Keywords: cotton percentage estimation; fabric classification; local binary patterns; residual feature augmentation; transfer learning; sustainable textile recycling; learnable texture descriptors



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Learnable Residual Local Binary Patterns: A Pretraining Preserving Architecture for Cotton Percentage Estimation in RGB Fabric Images

Arwa Basbrain 

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia; abasabreen@kau.edu.sa

## Abstract

Automated cotton-percentage identification underpins sustainable textile recycling, but established near-infrared and ATR-FTIR spectroscopy systems cost USD 10,000–25,000 per unit and remain inaccessible to small recyclers. We address this on the CottonFabricImageBD dataset (1,300 RGB originals, 13 ordinal cotton classes from 30% to 99%) and report three contributions. First, the *Learnable Residual LBP stem*, which retains the pretrained ResNet50 first convolution intact and adds a fully differentiable Local Binary Pattern branch as an additive contribution gated by a single learnable scalar  $\alpha$  initialized to zero, ensuring the model is numerically equivalent to the baseline at initialization (verified to a maximum absolute logit difference below  $10^{-4}$ ). Second, a controlled six-variant comparison (vanilla baseline, CLBP, LBP-Conv, LBP-Residual, LBP+SVM, LBP+ANN) under identical stratified 5-fold cross-validation on the 1,300 dataset originals. Third, the isolation of pretraining preservation as the dominant architectural variable: the 7.08 pp top-1 gap between LBP-Conv (43.77%) and LBP-Residual (50.85%), both embedding the identical learnable LBP module, is statistically significant ( $p = 0.004$ , paired  $t$ -test,  $df = 4$ ) and consistent across all five folds. Classical LBP+SVM and LBP+ANN baselines reach 31.85% and 34.46% top-1, confirming genuine but limited cotton-density signal in hand-crafted descriptors. Compared to the concurrent triplet-architecture approach of Wiedemann et al. (2025), which achieves 48.15% top-1 accuracy and 14.77% RMSE on the same dataset under identical 5-fold cross-validation, LBP-Residual attains 50.85% top-1 and an RMSE equivalent of  $\approx 7.2\%$  (computed under a different convention and thus not strictly comparable) using a single lightweight backbone rather than an ensemble of three. These results support the design principle: **augment, do not replace**.

**Keywords:** cotton percentage estimation; fabric classification; local binary patterns; residual feature augmentation; transfer learning; sustainable textile recycling; learnable texture descriptors

## 1. Introduction

The textile and fashion industry generates an enormous quantity of waste each year. According to the Ellen MacArthur Foundation, the equivalent of one garbage truck full of textiles is landfilled or incinerated every second worldwide, and only about 1% of clothing is recycled into new garments [1,2]. Mounting consumer awareness and regulatory pressure—most prominently the European Union's recent extended-producer-responsibility frameworks and several national-level textile recycling mandates—are pushing the industry toward circularity, but a fundamental technical obstacle limits how quickly the transition can proceed: before a garment can be recycled, the fiber composition of its fabric must be known. Recycling processes—whether mechanical shredding for insulation, chemical depolymerization for new fibers, or thermal recovery for energy—are economically viable only when the input material's composition is well characterized. The current state of practice for fiber composition identification is unsatisfactory on two fronts. *Manual identification* by trained textile experts using thread-counting machines and reference samples is accurate but slow, and it requires specialist labor that does not scale to the millions of garments processed daily by the industry. *Spectroscopic methods*—near-infrared (NIR) spectroscopy, attenuated total reflectance Fourier-transform

infrared (ATR-FTIR), and hyperspectral imaging—are accurate at industrial throughput but require laboratory-grade equipment costing between USD 10,000 and USD 25,000 per unit [3,4,7], placing them well beyond the reach of small recyclers and developing-economy textile sectors where much of the world’s fiber processing actually occurs. The gap between expensive accuracy and unscalable manual labor defines a clear opportunity for computer vision: a smartphone camera and a trained model could provide useful first-pass material identification at near-zero marginal cost per measurement, particularly for sorting applications that do not require the full precision of a laboratory spectrometer.

This paper addresses one specific instance of that broader opportunity: estimating the percentage of cotton in a fabric from a single RGB photograph. We work with the recently released CottonFabricImageBD dataset [10], comprising 1,300 fabric images spanning 13 cotton percentage classes from 30% to 99%, captured with a consumer smartphone and labeled under expert supervision using thread-counting verification. The dataset is the first public benchmark targeting cotton percentage as an ordinal classification problem, and its construction provides ground-truth labels at the same level of precision that laboratory-grade methods achieve, making it a clean benchmark for evaluating low-cost alternatives.

Cotton percentage is fundamentally a thread-density measurement. The dataset authors compute it by counting horizontal and vertical threads under a magnifying glass and dividing by the count of an idealized 100% cotton reference [10]. In computer vision terms, this is a *texture problem*: the discriminative signal lives in the high-frequency periodic structure of the weave, not in shape, color, or global composition. Different cotton percentages produce visually similar fabric photographs that differ primarily in the fine-scale spatial pattern of fiber arrangement.

Modern CNNs are well known to be biased toward shape over texture [43]. Classical computer vision developed dedicated texture descriptors for this setting. Among them, *Local Binary Patterns* (LBP) [27,28] and its Completed LBP variant [29] are particularly suited to encoding local microstructure as illumination-invariant binary codes. LBP was selected specifically over alternative texture descriptors for three reasons. First, LBP is fully differentiable in its learnable form [37], enabling end-to-end gradient training. Second, it is illumination-invariant under monotonic gray-scale transformations, which is particularly relevant for smartphone-captured fabric images with variable lighting. Third, its binary-code representation directly encodes the spatial periodicity of weave structure that corresponds to thread density. By contrast, Gray-Level Co-occurrence Matrices (GLCM) lack a straightforward differentiable formulation for end-to-end learning; Gabor filters are less compact for encoding binary weave structure; and learnable texture networks typically require larger datasets than the 1,300 originals available here.

The most directly related prior work is Wiedemann et al. [55], who recently proposed a triplet-loss metric-learning architecture combining DenseNet121, EfficientNetB4, and ResNet50 backbones with adaptive feature pyramid networks and deformable convolutions for cotton-content classification. Our work differs from theirs in two complementary ways: methodologically, we treat cotton content as an ordinal classification problem with a Gaussian soft-label objective rather than as a metric-learning problem; architecturally, we contribute a single principled augmentation (the residual LBP stem with a learnable scalar gate  $\alpha$  initialized to zero) to a standard pretrained ResNet50, rather than ensembling multiple backbones with specialized attention mechanisms. The simplicity of our design supports the interpretability advantage we describe in Section 3.6, and the residual-gate template should compose with the metric-learning and multi-backbone approaches of Wiedemann et al. in principle.

A subtle but consequential issue arises when integrating LBP at the input layer of a pretrained network: the pretrained ImageNet weights at that layer are discarded, and the network must relearn first-layer filters from scratch on the (typically small) target dataset. On large datasets this is recoverable; on small datasets it is catastrophic, because first-layer filters in CNNs encode universal visual primitives—oriented edges, color-opponent gradients, Gabor-like patterns—that take a substantial amount of training data to learn well from random initialization [40]. With only 1,300 originals in the CottonFabricImageBD dataset, this is squarely in the small-data-and-pretraining-matters regime.

This creates a tension at the heart of LBP-CNN integration on small datasets. We want the LBP signal early in the network, where it can shape downstream representations and provide the strongest possible inductive bias for texture features. But injecting LBP early typically requires modifying the pretraining-rich first convolution—exactly the layer where preserving pretrained representations matters most.

We propose a simple but principled design we call the *residual LBP stem*. The pretrained first convolution of a ResNet50 is retained *exactly as-is*. Alongside it, we add a learnable LBP branch—a fully differentiable LBP encoder operating on the luminance channel—whose output is added to the pretrained convolution’s feature map, gated by a single learnable scalar  $\alpha$  initialized to zero:

$$h(x) = \text{Conv}_1^{\text{pretrained}}(x) + \alpha \cdot \text{LBP-Branch}(L) \quad (1)$$

Three properties make this design useful. First, at initialization, the model is mathematically identical to the baseline since  $\alpha = 0$  collapses the equation to the pretrained convolution alone (verified empirically to bit-exact precision). Second, the worst-case behavior is bounded by the baseline: if LBP features turn out to be uninformative,  $\alpha$  remains near zero and the model recovers baseline performance. Third, the gate value  $\alpha$  provides post-hoc interpretability: its converged magnitude directly indicates whether the network actually exploited the auxiliary stream during training.

To establish what classical pre-deep-learning techniques achieve on this task and provide a reference point for the deep variants, we additionally evaluate two non-deep baselines: a multi-class Support Vector Machine with RBF kernel and a shallow multi-layer perceptron, both operating on 160-dimensional spatial LBP histograms ( $4 \times 4$  grid of uniform-LBP features). These methods do not use any pretrained representations and operate entirely on hand-crafted descriptors. Including them allows us to ask not just whether deep features beat classical features—they do—but *by how much*, and whether the cotton-density signal is recoverable at all from classical LBP descriptors. The answer to the latter question turns out to be informative: classical LBP carries genuine but limited cotton-density signal, which strengthens rather than weakens the case for the residual hybrid design.

Beyond the architectural proposal, this paper makes two additional methodological contributions. First, we adopt an explicitly *ordinal evaluation protocol*: we use mean absolute error (MAE) in percentage points as the primary metric, adjacent-class accuracy as a complementary metric, and a Gaussian soft-label cross-entropy loss whose target distribution decays smoothly with percentage distance. Standard top-1 accuracy treats predicting 95% as equally wrong as predicting 30% when the true label is 99%; an ordinal protocol correctly weights these errors by their ordinal magnitude. Second, we identify and document a *methodological pitfall* in the CottonFabricImageBD release: the 27,300 pre-augmented images cannot be cleanly split for cross-validation because the augmentation file naming does not preserve the mapping between augmented files and their source originals. Naively splitting across the augmented set places near-duplicates of training images into the validation fold and inflates apparent accuracy by tens of percentage points. We split exclusively on originals and apply augmentation on the fly during training, which we believe should become the standard protocol for this dataset.

We compare six model variants under stratified 5-fold cross-validation on the CottonFabricImageBD originals: a vanilla ResNet50 baseline, three deep LBP-augmented variants (CLBP input augmentation, LBP-Conv replacement stem, and the proposed LBP-Residual additive stem), and two classical baselines (LBP+SVM and LBP+ANN). Our experiments show:

- The proposed Residual LBP variant achieves the best aggregate performance on three of four primary metrics (50.85% top-1 accuracy versus 49.54% baseline, 84.77% top-3 versus 83.69%, and 5.70 pp MAE versus 5.84 pp), while matching the baseline on adjacent-class accuracy (84.92% vs. 85.15%). Improvements are consistent across folds with a 28% reduction in MAE standard deviation.
- The LBP-Conv variant, which replaces rather than augments the pretrained first convolution, performs substantially worse than the baseline (43.77% top-1 accuracy, 7.69 pp MAE). The 7-percentage-point gap between LBP-Conv and LBP-Residual, two variants embedding the *identical*

learnable LBP module, isolates pretraining preservation as the dominant architectural variable on small textile datasets.

- The classical baselines achieve 31.85% (LBP+SVM) and 34.46% (LBP+ANN) top-1 accuracy, 16–19 percentage points below the deep variants but 4–5× above the random-chance floor. This confirms that classical LBP histograms carry genuine but limited cotton-density signal.
- The residual gate  $\alpha$  converges to small but non-zero values across all 5 folds ( $|\alpha| \in [0.0004, 0.0040]$ ), providing direct evidence that the network actively learned to use the LBP signal during training.

We interpret these findings as supporting a general design principle for hybrid hand-crafted/deep architectures on small datasets: **augment, do not replace**. Whenever auxiliary features can be expressed as a residual addition to a pretrained layer's output, the residual formulation should be preferred to a replacement formulation, since the worst-case behavior of the residual variant is bounded by the baseline while the worst-case behavior of the replacement variant is not. The residual gate  $\alpha$  provides a principled mechanism for doing so.

The remainder of the paper is organized as follows. Section 2 reviews prior work on textile material identification, deep learning for fabric classification, LBP variants, and ordinal classification. Section 3 presents the six model variants in detail, along with the ordinal loss formulation, evaluation metrics, and cross-validation protocol. Section 4 reports the cross-validation results, training dynamics, learned  $\alpha$  trajectories, and confusion structure of each variant. Section 5 interprets these findings and their implications for hybrid hand-crafted/deep architectures. Section 6 concludes and outlines directions for future work.

## 2. Related Work

This work draws on five strands of prior research: automated textile material identification (Section 2.1), deep learning for fabric and fiber image classification (Section 2.2), Local Binary Patterns and their variants (Section 2.4), LBP-CNN integration approaches including learnable LBP (Section 2.5), small-data transfer learning and the texture-vs-shape problem (Section 2.6), and ordinal classification with deep networks (Section 2.7). We close with a positioning statement (Section 2.8) that places our contribution within these literatures.

### 2.1. Automated Textile Material Identification

The accuracy ceiling for automated textile fiber identification is set by spectroscopic methods. Cura et al. [3] demonstrated NIR-based sorting at industrial throughput, achieving classification accuracies around 96% across common fiber types in a textile recycling line. Riba et al. [4] reported 100% classification accuracy across seven fiber types using ATR-FTIR features on a sample of 350 fabrics, and the same group later extended the protocol to a broader range of post-consumer fiber blends [5]. Peets et al. [6] systematically evaluated ATR-FTIR for fiber discrimination and confirmed that wool, silk, and cellulose-based fibers can be reliably separated by their characteristic spectral peaks. Liu et al. [7] demonstrated that convolutional neural networks operating on NIR spectra outperform SVM and MLP for waste textile classification, suggesting that deep models can extract additional discriminative information even from spectral inputs. The common limitation across all these methods is equipment cost: NIR spectrometers range from USD 10,000 to 25,000 per unit, ATR-FTIR systems from USD 20,000 upward, and hyperspectral cameras from USD 30,000 upward, placing them well beyond the reach of small recyclers and developing-economy textile sectors. For specific fiber-pair distinctions where spectral signatures are too similar, microscopy combined with image analysis has been employed. Xing et al. [8] used transfer learning on microscopy images to discriminate cashmere from wool, achieving high accuracy on a notoriously difficult fraud-detection task. Zhong et al. [9] used projection curves from microscopy images for the same task. These methods require microscopy equipment in the USD 4,000–10,000 range, which is more accessible than spectroscopy but still beyond consumer reach.

The lowest-cost regime—classification from ordinary RGB photographs—is the least mature. Niloy et al. [10] released CottonFabricImageBD, the dataset used in this study, with 1,300 original images across 13 cotton percentage classes labeled by thread-counting verification. Their baseline experiments report top-1 accuracies in the 70–90% range, but as we discuss in Section 3.9, their evaluation protocol used the augmented set of 27,300 images for cross-validation, which inflates apparent accuracy due to source-original leakage between training and validation folds. Zhong et al. [11] released TextileNet, comprising approximately 760,000 images across 33 fiber and 27 fabric labels generated from material taxonomies, with baseline ResNet50 achieving roughly 50% top-1 accuracy on fiber classification. Liu et al. [15] adapted an improved U-Net architecture for fabric defect segmentation, demonstrating gains over the standard U-Net baseline on textile inspection imagery.

## 2.2. Deep Learning for Fabric and Fiber Classification

ResNet50 [17] serves as the de facto baseline in most published evaluations of fabric and fiber classification [10,11]. Specialized architectures have been proposed: Hussain et al. [20] ensemble multiple backbones for woven fabric pattern recognition; Ohi et al. [21] proposed FabricNet, an ensemble ConvNet architecture for fiber recognition, demonstrating that backbone ensembling improves fiber classification accuracy over single-backbone baselines; and Meng et al. [22] proposed a multi-task framework for joint fabric category and attribute prediction. Reported gains from specialized architectures over a well-tuned single ResNet50 backbone are typically modest in the 1–3 percentage-point range.

Vision Transformers (ViT) [18] have been observed to underperform ResNets on textile tasks at small-to-medium dataset scales [11], consistent with broader findings that ViTs require larger datasets and longer training schedules to outperform CNNs [19]. We therefore retain ResNet50 as the backbone for all deep variants in this study, leaving ViT-based extensions for future work.

Fabric defect detection is the most mature applied area in deep fabric classification, with extensive literature on tasks such as identifying misspicks, broken threads, and stains in industrial inspection settings. Bissi et al. [23] combined Gabor filters and PCA for defect detection in uniform and structured fabrics, predating the deep-learning era but establishing the texture-feature framing that later work inherits.

Liu et al. [24] proposed a deep-learning fabric defect detection method demonstrating effective classification of common defect types on industrial fabric imagery. Jing et al. [25] and Wei et al. [26] demonstrated CNN-based defect detection at industrial throughput. While defect detection differs from cotton percentage classification in its task structure (binary or few-class detection rather than ordinal classification), it shares the texture-driven signal that motivates our methodology.

## 2.3. Concurrent Work: Wiedemann et al. (2025)

The most directly related contemporaneous work is Wiedemann et al. [55], who proposed a triplet architecture combining ResNet50, EfficientNetB4, and DenseNet121 with AFPNs and deformable convolution (DConv) layers. Using stratified 5-fold cross-validation on the same CottonFabricImageBD dataset, they report 48.15% top-1 accuracy and an RMSE of 14.77%, which was the prior state of the art for cross-validated visual approaches.

Our work differs along two axes. *Methodologically*, we treat cotton content as an ordinal classification problem with a Gaussian soft-label objective and MAE as the primary metric, rather than as a standard multi-class problem with RMSE. *Architecturally*, we contribute a single residual augmentation to a standard pretrained ResNet50, rather than ensembling three backbones with specialized attention and deformable-convolution modules. As Table 5 in Section 4.3 shows, LBP-Residual achieves comparable top-1 accuracy (50.85% vs. 48.15%) and substantially lower RMSE ( $\approx 7.2\%$  vs. 14.77%) using roughly one-third of the backbone complexity. The two approaches are complementary rather than competing: the residual-gate template could, in principle, be composed with the triplet architecture to provide texture-specific augmentation of each backbone.

#### 2.4. Local Binary Patterns and Their Variants

Local Binary Patterns were introduced by Ojala et al. [27,28] and have remained one of the most widely used texture descriptors in classical computer vision. The original LBP encodes local texture by thresholding  $P$  neighbor pixels at radius  $R$  against the center pixel intensity, producing a binary code per pixel. Pietikäinen et al. [33] and Liu et al. [34] provide comprehensive surveys of LBP variants and their applications.

Several variants extend the basic LBP. Completed LBP (CLBP) [29] decomposes the local pattern into sign, magnitude, and center-intensity components, capturing more information than the sign alone. Local Ternary Patterns (LTP) [30] introduce a three-state thresholding to improve robustness under non-uniform illumination. Multi-Scale LBP (MB-LBP) [31] computes LBP at multiple block scales and concatenates them. Local Phase Quantization (LPQ) [32] extends the LBP framework into the frequency domain. Rotation-invariant LBP variants ( $LBP^{ri}$ ,  $LBP^{riu2}$ ) [28] provide invariance under rotation by mapping equivalent codes to a single bin, and the uniform-pattern variant we use in our classical baselines (Section 3.3) reduces dimensionality from  $2^P$  to  $P + 2$  bins by grouping codes with at most two bit transitions.

LBP and its variants have been applied to a wide range of texture classification tasks beyond textiles, including face recognition [35], medical imaging [36]. The classical literature established two key properties: LBP is illumination-invariant under monotonic transformations of the gray-level scale, and it is rotation-invariant under appropriate code remapping. These properties are particularly relevant for textile classification, where lighting conditions vary across collection setups and fabric orientation is not canonical.

#### 2.5. LBP-CNN Fusion and Learnable LBP

Three patterns dominate LBP-CNN integration in the literature. *Late fusion*, where LBP histograms are concatenated with CNN features before the final classifier, is the simplest approach and appears in early work combining classical and deep features [23]. The reported gains are typically modest because the LBP signal influences only the final classifier rather than the feature-extraction process. *Multi-channel input*, where LBP-encoded images are stacked with RGB and fed to an expanded first convolution, is the approach our CLBP variant uses; this pattern allows the network to learn how to combine LBP and RGB at the input layer but requires careful initialization to preserve pretraining. A more recent direction makes LBP itself learnable. Juefei-Xu et al. [37] proposed Local Binary Convolutional Neural Networks (LBCNN), where binary patterns serve as parameter-efficient alternatives to standard convolutions; the focus is computational efficiency rather than texture-specific inductive bias, and the binary patterns are randomly initialized rather than tied to classical LBP.

**The architectural placement of learnable LBP within a deeper pretrained network remains under-explored.** Existing works have typically proposed learnable LBP as a *replacement* for the first convolution, immediately running into the pretraining-preservation problem identified by Yosinski et al. [40]. The contribution of the present paper is to integrate the learnable LBP module as a *residual augmentation* to the pretrained first conv, drawing on the design tradition of skip connections [17] and gated mechanisms [38,39]. To our knowledge, the residual-with-zero-init-gate construction has not been previously applied to LBP-CNN integration.

#### 2.6. Small-Data Transfer Learning and Texture-vs-Shape Bias

The pretraining-preservation argument central to our work rests on a substantial literature on transfer learning in deep networks. Yosinski et al. [40] showed that early-layer features in CNNs are substantially more transferable than late-layer features, and that the quality of pretrained features deteriorates progressively as fine-tuning depth increases on small target datasets. Kornblith et al. [41] provided a systematic evaluation of how ImageNet pretraining transfers to downstream tasks and confirmed that pretrained representations carry universal early-layer primitives that are difficult to recover from scratch on small datasets. Raghu et al. [42] extended this to medical imaging specifically,

showing that pretraining dominates feature quality at small dataset scales. These results motivate our design choice to preserve  $\text{Conv}_1$  rather than replace it. The texture-vs-shape problem in CNNs was sharply diagnosed by Geirhos et al. [43], who demonstrated that ImageNet-trained CNNs make classification decisions overwhelmingly based on object shape and that this bias is acquired during training rather than architecturally enforced. Hermann et al. [44] extended this analysis to show that the shape bias is sensitive to data augmentation choices and is not a fixed property of CNN architectures. Tuli et al. [45] showed that the texture-shape gap also exists in vision transformers, suggesting it is a property of training data rather than network architecture. The residual LBP stem proposed in this paper can be viewed as one architectural instance of injecting texture priors back into shape-biased networks, specialized to textile texture.

### 2.7. Ordinal Classification

Cotton percentages are ordinally ordered, which makes this problem an ordinal classification task. Frank and Hall [46] introduced the simple-ordinal approach of decomposing  $K$ -way ordinal classification into  $K - 1$  binary classifications, providing a strong baseline that does not require specialized architectures. Niu et al. [47] proposed a multi-output CNN for ordinal regression with applications to age estimation. Cao et al. [48] proposed CORAL, which guarantees rank-monotonic outputs through architectural constraints. Beckham and Pal [49] proposed unimodal output distributions for ordinal classification, related to our Gaussian soft-label approach. Liu et al. [50] extended ordinal CNNs to deep ordinal regression with hashing for retrieval. We adopt a simpler approach: *Gaussian soft labels* [51], replacing the one-hot target with a Gaussian-shaped distribution centered on the true class. This approach has been used in age estimation and face attribute recognition [52,53] and was recently revisited in the medical imaging literature [54].

For ordinal evaluation, MAE in the ordinal value space is widely used [47], alongside adjacent-class accuracy (sometimes called top-1 ordinal accuracy) [55] and weighted Cohen’s Kappa [56]. We report these metrics throughout Section 4.

### 2.8. Positioning of This Work

Within these literatures, the present work occupies a specific position. We do not propose a new LBP variant per se: classical LBP, CLBP, and the learnable LBP module we use in LBP-Conv and LBP-Residual all draw on existing work. We do not propose a new ResNet architecture: the backbone is standard ResNet50 with ImageNet pretraining. What we contribute is the *architectural placement* of learnable LBP as a zero-initialized additive residual on top of a preserved pretrained first convolution, combined with a clean ordinal evaluation protocol on the CottonFabricImageBD dataset and a documented methodological pitfall in the dataset’s pre-augmented release.

The closest prior work is Juefei-Xu et al. [37] (LBCNN, which uses binary patterns to replace standard convolutions for efficiency). They treat learnable LBP as a replacement rather than augmentation, and neither addresses the small-data pretraining-preservation problem that motivates our residual design. The isolation of this design decision via the LBP-Conv vs. LBP-Residual comparison is, to our knowledge, novel.

Table 1 summarizes public fabric and fiber datasets, and Table 2 summarizes LBP-CNN integration approaches. These tables provide quick reference for the citations discussed above.

**Table 1.** Public datasets relevant to fabric and fiber image classification.

Dataset	Images	Classes	Annotation source	Reference
DeepFashion	800K	218 fabric attributes	Crawled metadata	Liu et al. [12]
iMaterialist	1M	11 fiber + 21 fabric	Crowdsourced	Guo et al. [13]
TILDA	3.2K	8 fabric defect classes	Manual	Chetverikov et al. [14]
TextileNet	760K	33 fiber + 27 fabric	Expert taxonomy	Zhong et al. [11]
<b>CottonFabricImageBD</b>	1,300 originals + 27,300 augmented	13 cotton % classes	Thread-counting expert verification	Niloy et al. [10]

Table 2. LBP and LBP-CNN integration approaches.

Method	LBP form	Integration pattern	Pretraining preserved	Reference
Classical LBP	Fixed sign-only	Standalone descriptor + classifier	N/A	Ojala et al. [27,28]
CLBP	Sign + Mag + Center	Standalone descriptor + classifier	N/A	Guo et al. [29]
LTP	Three-state	Standalone descriptor + classifier	N/A	Tan and Triggs [30]
LBP-CNN late fusion	Fixed	Histogram concat at FC layer	Yes	Bissi et al. [23]
LBCNN	Fixed binary patterns	Replaces standard conv (efficiency)	No	Juefei-Xu et al. [37]
<b>LBP-Residual</b>	<b>Learnable</b>	<b>Additive residual:</b> $\text{Conv}_1(x) + \alpha \cdot \text{LBP}(L)$	<b>Yes (fully)</b>	This paper

### 3. Materials and Methods

This section presents the methodology for cotton percentage classification from RGB fabric images. We compare six model variants in total: a vanilla ResNet50 baseline, three deep LBP-augmented variants (CLBP, LBP-Conv, and the proposed LBP-Residual), and two classical-method baselines (LBP+SVM and LBP+ANN). The deep variants share a common training pipeline and hyperparameters; the classical variants extract hand-crafted LBP histograms and feed them to a downstream classifier. All six are evaluated under an identical 5-fold cross-validation protocol on the dataset originals.

#### 3.1. Problem Formulation

Let each fabric image  $x_i \in \mathbb{R}^{H \times W \times 3}$  be associated with a cotton percentage label  $y_i \in \mathcal{Y}$ , where

$$\mathcal{Y} = \{30, 40, 50, 53, 58, 60, 63, 65, 66, 80, 95, 98, 99\} \% \quad (2)$$

contains the thirteen distinct classes in the CottonFabricImageBD dataset. Let  $\phi : \mathcal{Y} \rightarrow \{0, 1, \dots, 12\}$  map each percentage to its class index. The classification task is to learn a function  $f_\theta$  that maps an input image to logits over the  $C = 13$  classes; the predicted percentage is  $\hat{y}_i = \phi^{-1}(\arg \max_c f_\theta(x_i)_c)$ . A critical property of this task, distinguishing it from conventional classification, is that the label space  $\mathcal{Y}$  is *ordinal*: predicting 95% when the true label is 99% constitutes a much smaller error than predicting 30%. The methodology and evaluation protocol explicitly account for this ordinal structure.

#### 3.2. Baseline Architecture

The baseline model adopts the ResNet50 architecture [17] initialized with weights pretrained on ImageNet-1K. The first convolution  $\text{Conv}_1 : \mathbb{R}^{3 \times 224 \times 224} \rightarrow \mathbb{R}^{64 \times 112 \times 112}$  is a  $7 \times 7$  stride-2 convolution with 64 output channels, followed by batch normalization, ReLU, and the four ResNet bottleneck stages. A global average pooling layer and a fully-connected classifier head with output dimension  $C = 13$  complete the network. This baseline has 23.53M trainable parameters and serves as the reference point against which all LBP-augmented variants are compared.

#### 3.3. Classical LBP Baselines (LBP+SVM, LBP+ANN)

To establish what classical pre-deep-learning methods achieve on this task, we include two baselines that extract a hand-crafted LBP histogram per image and feed it to a downstream classifier. These methods do not use ImageNet pretraining and operate entirely on hand-crafted features, serving as a reference point separate from the four deep-learning variants.

### 3.3.1. Local Binary Patterns: Background

Local Binary Patterns [27,28] encode local micro-texture by binarizing differences between a center pixel and its neighbors. Specifically, for a pixel at location  $(u, v)$  with intensity  $I(u, v)$ , the classical LBP code is:

$$\text{LBP}_{P,R}(u, v) = \sum_{p=0}^{P-1} s(I(u_p, v_p) - I(u, v)) \cdot 2^p \quad (3)$$

where  $P$  is the number of neighbors at radius  $R$ , and  $s(\cdot)$  is the unit step function returning 1 when its argument is non-negative and 0 otherwise. LBP is particularly relevant to cotton percentage classification because cotton content is determined by *thread count density*—the dataset authors literally measure it with a magnifying glass and counting grid. Thread density manifests as a high-frequency texture signal that LBP is designed to capture, while CNNs are known to be biased toward shape rather than texture [43].

### 3.3.2. Feature Extraction

Each image is converted to luminance via the BT.601 standard, resized to  $256 \times 256$  via center-crop, and processed with the uniform-pattern variant of classical LBP (rotation-aware  $\text{LBP}^{\text{riu}2}$  with  $P = 8$  neighbors at radius  $R = 1$ ), producing 10 distinct codes per pixel. The image is divided into a  $4 \times 4$  spatial grid; an L1-normalized histogram of the LBP codes is computed independently within each cell, and the 16 cell histograms are concatenated into a single 160-dimensional feature vector per image. The spatial grid preserves coarse positional information that a single global histogram would discard. This is the standard descriptor used in LBP-based texture classification since [28].

### 3.3.3. LBP+SVM

The 160-dimensional features are standardized (zero-mean, unit-variance scaling fitted on the training fold) and fed to a multi-class Support Vector Machine with an RBF kernel, regularization parameter  $C = 10$ , and  $\gamma = 1/(n_{\text{features}} \cdot \text{var}(X))$ . We use scikit-learn's one-vs-rest SVM implementation, which fits 13 binary classifiers under the hood; the predicted class for each test image is the one with the highest decision-function score.

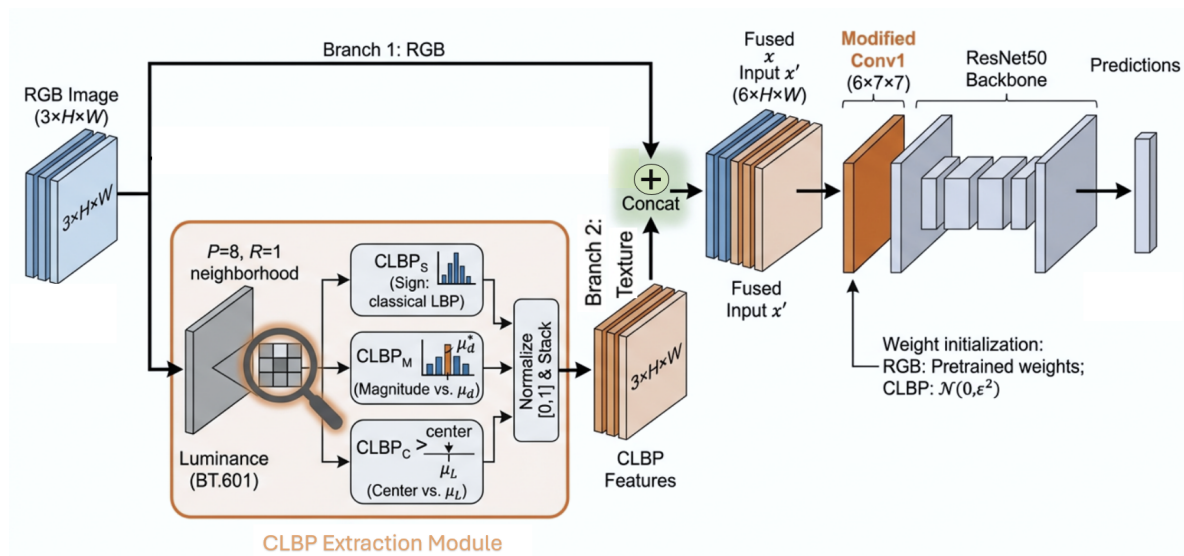
### 3.3.4. LBP+ANN

The same standardized features are fed to a shallow multi-layer perceptron with two hidden layers ( $160 \rightarrow 512 \rightarrow 256 \rightarrow 13$ ), each followed by batch normalization, ReLU activation, and dropout with rate 0.3. The MLP is trained for 100 epochs with the AdamW optimizer (learning rate  $10^{-3}$ , weight decay  $10^{-4}$ , batch size 64), cosine annealing schedule, and standard cross-entropy loss with label smoothing  $\varepsilon = 0.1$ . The best-epoch model on validation top-1 accuracy is retained. These two classical methods complete the six-variant comparison. Note that the residual gate  $\alpha$  and the ordinal soft-label loss (Section 3.7) apply only to the deep variants; the classical baselines train with standard cross-entropy on the hand-crafted LBP features.

## 3.4. Completed LBP Input Augmentation (CLBP)

The first deep variant injects hand-crafted Completed LBP features [29] as additional input channels. Given an RGB image  $x \in \mathbb{R}^{3 \times H \times W}$ , we compute its luminance using the BT.601 standard ( $L = 0.299x_R + 0.587x_G + 0.114x_B$ ). For each pixel and  $P = 8$  neighbors at radius  $R = 1$ , we compute the differences  $d_p = L(u_p, v_p) - L(u, v)$  and form three CLBP components: the sign component  $\text{CLBP}_S$  (a thresholded weighted sum encoding the sign of each difference, equivalent to classical LBP), the magnitude component  $\text{CLBP}_M$  (encoding whether each  $|d_p|$  exceeds the global mean magnitude  $\mu_d$ ), and the center component  $\text{CLBP}_C$  (a binary indicator of whether  $L(u, v)$  exceeds the global mean luminance  $\mu_L$ ). Each component is normalized to  $[0, 1]$ . The three CLBP channels are concatenated with the original RGB channels to form a six-channel input  $x' \in \mathbb{R}^{6 \times H \times W}$ . The first convolutional layer of ResNet50 is expanded from a  $3 \times 7 \times 7$  kernel to a  $6 \times 7 \times 7$  kernel; the RGB-channel weights

are copied from the pretrained baseline, while the three CLBP-channel weights are initialized as  $\mathcal{N}(0, \epsilon^2)$  with  $\epsilon = 0.01 \cdot \text{std}(W_1)$ . This ensures the model behaves nearly identically to the baseline at initialization, with the CLBP channels gradually contributing as training proceeds. Figure 1 illustrates the architecture.

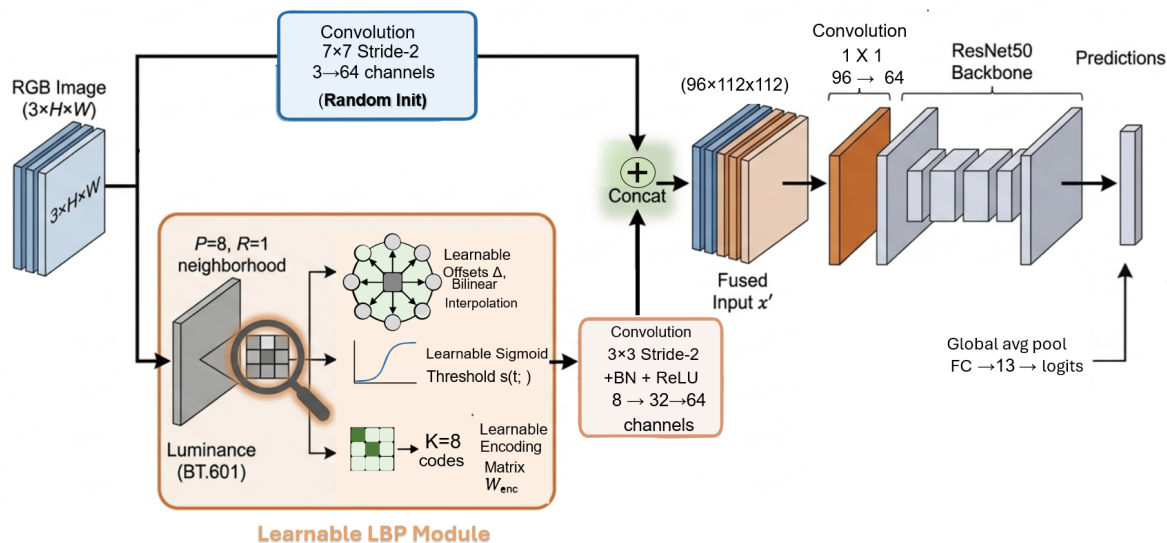


**Figure 1.** Architecture of the CLBP variant. Three Completed LBP channels (sign, magnitude, center) are computed from the input image and concatenated with the original RGB channels to form a six-channel input. The first convolution is expanded to  $6 \rightarrow 64$  channels: RGB weights are copied from the pretrained baseline; CLBP weights are initialized small.

### 3.5. Learnable LBP Convolution Stem (LBP-Conv)

The second deep variant replaces the first convolutional layer with a learnable LBP module that generalizes classical LBP via end-to-end gradient learning. Classical LBP has three non-differentiable components—fixed neighbor positions, the sign function, and fixed power-of-two encoding weights—and we make each learnable. Specifically, the neighbor positions are parameterized as a learnable offset matrix  $\Delta \in \mathbb{R}^{P \times 2}$  initialized on a circle of radius  $R$ , with neighbor values obtained via bilinear interpolation to ensure differentiability. The sign function is replaced by a sigmoid  $\tilde{s}(t; \tau) = (1 + e^{-\tau t})^{-1}$  with a learnable temperature  $\tau = e^\beta$  (parameterized through  $\beta$  to ensure  $\tau > 0$ ); as  $\tau \rightarrow \infty$ ,  $\tilde{s} \rightarrow s$ , recovering the classical thresholding. The power-of-two encoding is replaced by a learnable matrix  $W_{\text{enc}} \in \mathbb{R}^{K \times P}$  producing  $K = 8$  output codes; its first row is initialized to the normalized power-of-two weights, ensuring classical LBP is recoverable as a special case, while the remaining rows are randomly initialized to allow the discovery of task-specific encodings. The LBP-Conv variant uses this learnable LBP module in a parallel two-branch stem. The first branch is a randomly-initialized  $7 \times 7$  stride-2 convolution producing 64 channels, replacing the pretrained  $\text{Conv}_1$  entirely. The second branch is the learnable LBP module applied to the luminance, with its  $K = 8$  codes projected to 32 channels via a  $3 \times 3$  stride-2 convolution. The two branches are concatenated along the channel dimension and fused via a  $1 \times 1$  convolution back to the 64 channels expected by downstream ResNet stages. The architecture is illustrated in Figure 2.

The critical limitation of this design is that it *discards ImageNet pretraining at the stem*. On a small dataset like CottonFabricImageBD (1,300 originals), the network has insufficient data to relearn good first-layer filters from random initialization. We therefore propose a third variant that addresses this limitation.



**Figure 2.** Architecture of the LBP-Conv variant. The pretrained first convolution is replaced with a parallel two-branch stem combining a randomly-initialized  $7 \times 7$  convolution (left) and a learnable LBP encoder operating on luminance (right). The branches are concatenated and fused via a  $1 \times 1$  convolution. *This variant discards ImageNet pretraining at the input layer.*

### 3.6. Residual LBP Stem (LBP-Residual)

The proposed residual variant addresses the pretraining-loss limitation of LBP-Conv. Rather than replacing the pretrained first convolution, we *retain it intact* and add the LBP branch as a small additive contribution gated by a learnable scalar  $\alpha$ :

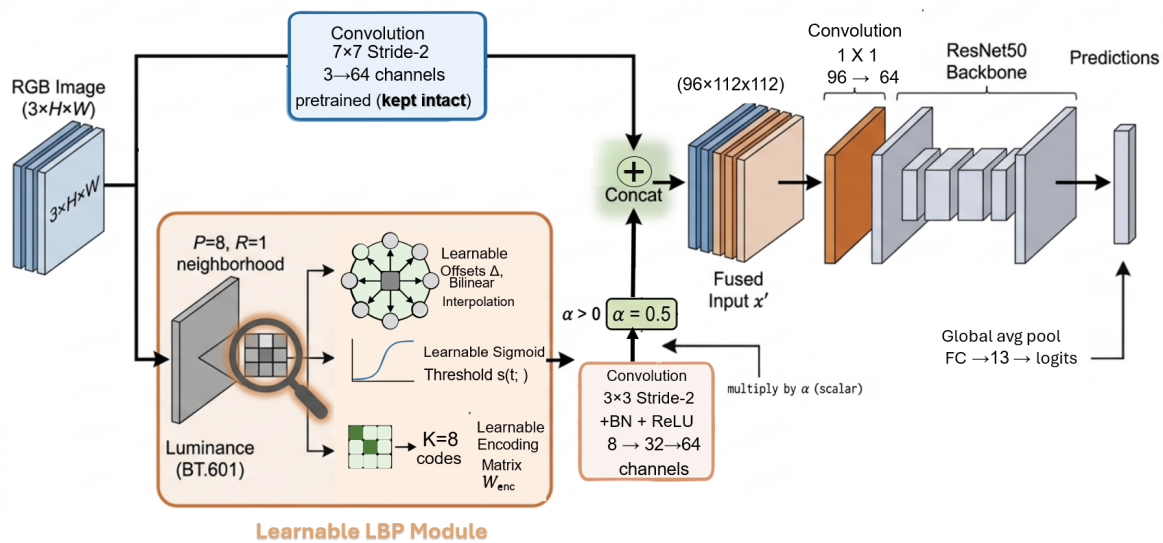
$$h_{\text{res}}(x) = \text{Conv}_1^{\text{pretrained}}(x) + \alpha \cdot \text{LBP-Branch}(L) \quad (4)$$

where the LBP branch processes the luminance through the same learnable LBP module described in Section 3.5, followed by a  $3 \times 3$  stride-2 projection ( $K \rightarrow 32$  channels), batch normalization and ReLU, and a  $1 \times 1$  projection ( $32 \rightarrow 64$  channels) to match the spatial and channel dimensions of  $\text{Conv}_1^{\text{pretrained}}(x)$ . The output is added directly to the pretrained convolution's output, preserving the channel count expected by downstream ResNet stages.

The critical design choice is to initialize  $\alpha = 0$ . This yields the equivalence:

$$\alpha = 0 \implies h_{\text{res}}(x) = \text{Conv}_1^{\text{pretrained}}(x) \quad \forall x \quad (5)$$

That is, at initialization the residual model is mathematically identical to the baseline. The full network thus inherits all of ImageNet's pretrained representations exactly, and during training the optimizer learns whether to grow  $\alpha$  away from zero. If the LBP signal is informative,  $\alpha$  moves to a non-zero value and the LBP branch contributes; if not,  $\alpha$  stays near zero and the model recovers baseline performance. We verified this property empirically: with copied weights, the residual model produces logits identical to the baseline up to numerical precision (max absolute difference  $< 10^{-4}$ ). The learned  $\alpha$  provides built-in interpretability. A converged  $|\alpha| \approx 0$  indicates that LBP features were not useful;  $|\alpha| > 0$  with improved metrics indicates that LBP features contributed meaningfully; and  $|\alpha| > 0$  with degraded metrics would indicate that LBP features acted as noise. This is a substantial improvement over LBP-Conv, where the network is *forced* to use a randomly-initialized stem regardless of whether LBP helps. Figure 3 illustrates the architecture.



**Figure 3.** Architecture of the proposed Learnable Residual LBP variant. The pretrained first convolution (left branch) is retained *intact*, and a learnable LBP branch (right branch) operating on BT.601 luminance contributes additively via a single learnable scalar  $\alpha$ . *Critical property:*  $\alpha$  is initialized to zero, so at initialization  $h_{\text{res}}(x) = \text{Conv}_1(x)$  and the model is bit-exactly identical to the baseline.

### 3.7. Loss Function: Ordinal Soft-Label Cross-Entropy

Standard cross-entropy treats classes as unordered, assigning equal penalty to predicting 98% versus 30% when the true label is 95%. To respect the ordinal structure of cotton percentages, we use a Gaussian soft-label loss that places target mass on classes proportionally to their proximity in percentage space. For a sample with true class index  $c$  and corresponding cotton percentage  $p_c = \phi^{-1}(c)$ , the soft target distribution is defined element-wise as:

$$q_j^{(c)} = \frac{\exp(-(p_j - p_c)^2 / 2\sigma^2)}{\sum_{k=0}^{C-1} \exp(-(p_k - p_c)^2 / 2\sigma^2)} \quad (6)$$

where  $\sigma$  is the bandwidth in percentage points (we use  $\sigma = 5$ ). Optionally, label smoothing with parameter  $\varepsilon = 0.1$  is applied on top of the Gaussian distribution. The training objective is the cross-entropy between the smoothed soft target and the model's predicted distribution; classical one-hot cross-entropy is recovered in the limit  $\sigma \rightarrow 0$ .

For example, with  $\sigma = 5$  and a true label of 95%, the target distribution places approximately 0.61 probability mass on class 95%, 0.30 on class 98%, 0.05 on class 99%, and negligible mass on class 30%. This causes the network to receive a stronger gradient signal toward predictions near the true percentage and a weaker signal toward distant predictions. The four deep variants train with this objective; the two classical baselines train with standard cross-entropy on the hand-crafted LBP features.

### 3.8. Evaluation Metrics

Given the ordinal nature of the task, we report four complementary metrics. Top-K accuracy for  $K \in \{1, 3\}$  is the standard fraction of samples for which the true class lies in the top-K predictions. Mean Absolute Error (MAE) in percentage points measures the average error in the cotton-percentage space:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\phi^{-1}(\hat{c}_i) - \phi^{-1}(c_i)| \quad (7)$$

**Adjacent-class accuracy** counts a prediction as correct if it is within one class index of the truth, accommodating the visual indistinguishability of neighboring classes (e.g., 65% vs. 66%):

$$\text{Adj-Acc} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[|\hat{c}_i - c_i| \leq 1] \quad (8)$$

We additionally report Cohen’s Kappa for completeness. The MAE in percentage points is treated as the primary metric for model selection (lowest validation MAE selects the best checkpoint), as it most directly captures the practical utility of the model for downstream applications such as garment recycling.

### 3.9. Cross-Validation Protocol

The CottonFabricImageBD dataset contains 100 original images per class (1,300 total) plus 2,100 augmented variants per class (27,300 total). The augmentations are deterministic transforms of the originals, and the public release does not preserve the mapping between augmented files and their source originals, so naively splitting across the augmented set places near-duplicates of training images into the validation fold and inflates apparent accuracy by tens of percentage points.

We therefore conduct stratified 5-fold cross-validation *on the originals only*. The 1,300 originals are partitioned into 5 disjoint folds with each fold containing exactly 20 originals per class. For each fold, the model is trained on the union of the other four folds (1,040 originals) and validated on the held-out fold (260 originals). All six model variants share an identical fold split via a cached fold-index file, enabling paired statistical comparison.

For the deep variants, augmentation is applied on the fly during training to the training-fold images only, matching the transformations used by the dataset authors: rotation in  $[-30^\circ, 30^\circ]$ , translation up to 20% of image width and height, scale in  $[0.8, 1.2]$ , shear in  $[-15^\circ, 15^\circ]$ , and independent horizontal/vertical flips with probability 0.5. The classical baselines do not apply augmentation, since LBP histograms are designed to be invariant to many of these transforms and the augmented variants would not provide useful additional information at the feature level. We report the mean and standard deviation of each metric across the 5 folds, and use a paired *t*-test on per-fold metric values to assess statistical significance of differences between models. The pairing is justified by the shared fold splits: each fold’s training and validation composition is identical across the six variants.

### 3.10. Training Configuration

The four deep variants are trained with identical hyperparameters: AdamW optimizer [57] with learning rate  $\eta = 5 \times 10^{-4}$ , weight decay  $\lambda = 10^{-4}$ , batch size 64, 50 epochs, and cosine annealing learning rate schedule. The learnable LBP parameters in LBP-Conv and LBP-Residual share the same optimizer and learning rate as the rest of the network. Mixed-precision training is used throughout. The best checkpoint per fold is selected by minimum validation MAE. The classical baselines are trained as described in Section 3.3. The SVM is fit in a single closed-form optimization step per fold; the ANN is trained for 100 epochs with the same AdamW + cosine annealing setup as the deep variants.

### 3.11. Summary of Model Variants

Table 3 summarizes the six model variants evaluated in this work, distinguished by their treatment of the first convolution and their LBP integration mechanism.

**Table 3.** Summary of the six model variants. Variants A–C are deep end-to-end models; D and E are classical baselines using hand-crafted LBP histograms with downstream classifiers.

Variant	Backbone / classifier	LBP integration	Pretrained weights
Baseline	ResNet50 (deep)	None	Yes
CLBP	ResNet50 (deep)	Pre-input concatenation (6-channel)	Yes (RGB channels)
LBP-Conv	ResNet50 (deep)	Parallel-branch stem replacing Conv <sub>1</sub>	No (stem reinitialized)
<b>LBP-Residual</b>	ResNet50 (deep)	Additive residual gated by learnable $\alpha$	Yes (fully)
LBP+SVM	RBF SVM (classical)	Spatial LBP histogram, 160-dim feature vector	N/A (no pretraining)
LBP+ANN	2-hidden-layer MLP (classical)	Spatial LBP histogram, 160-dim feature vector	N/A (no pretraining)

The six-variant design isolates several factors. The deep-vs-classical contrast (Variants A–C vs. D, E) shows what end-to-end pretrained CNNs buy over hand-crafted LBP descriptors. The CLBP-vs-baseline contrast (B vs. A) tests whether LBP features help when injected as additional input channels with pretraining preserved. The LBP-Conv-vs-LBP-Residual contrast (C vs. D as deep models) isolates the effect of pretraining preservation, since both use the same learnable LBP module. The residual gate  $\alpha$  provides additional interpretability for the proposed variant, allowing post-hoc inspection of whether the network actually used the LBP signal during training.

## 4. Results

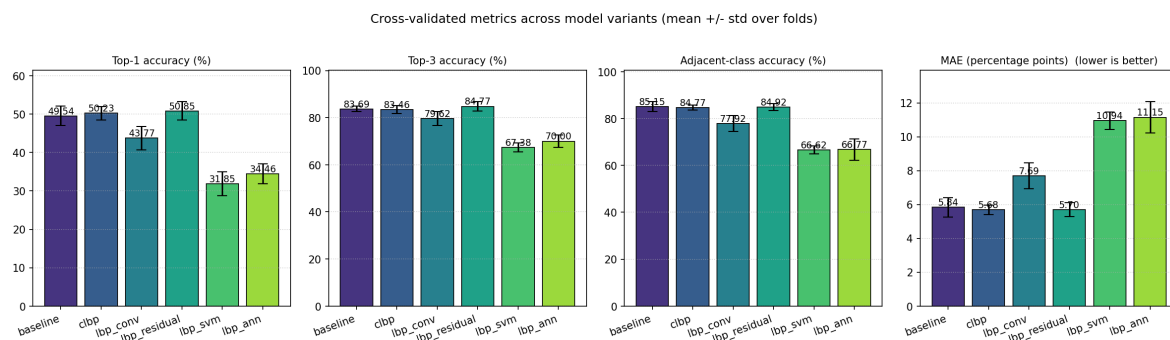
This section reports the cross-validation performance of all six model variants on the Cotton-FabricImageBD dataset. We first present aggregate metrics across folds, then examine per-fold consistency, training dynamics, the learned residual gate, and the confusion structure of each model. All numbers are computed on the validation fold using the protocol of Section 3.9, with the best checkpoint per fold selected by minimum validation MAE for the deep variants.

### 4.1. Aggregate Performance

Table 4 summarizes the mean and standard deviation of each metric across the 5 folds, and Figure 4 displays them as bar charts with error bars. The proposed LBP-Residual variant achieves the highest mean top-1 accuracy (50.85%), the highest mean top-3 accuracy (84.77%), and the lowest mean MAE (5.70 pp) among the four deep variants, while the baseline retains a marginal lead on mean adjacent-class accuracy (85.15% vs. 84.92% for LBP-Residual), a 0.23 pp gap that is well within fold-to-fold noise. The CLBP variant provides a small improvement in top-1 and MAE over the baseline; its mean MAE (5.68 pp) is in fact marginally lower than LBP-Residual’s (5.70 pp), and CLBP exhibits the smallest fold-to-fold MAE standard deviation (0.28 pp) of any variant. The headline differences among the baseline, CLBP, and LBP-Residual are therefore within the noise of a 5-fold protocol, and we do not claim a statistically significant separation among these three variants. The LBP-Conv variant, which replaces the pretrained first convolution with a randomly-initialized stem, performs substantially worse than the baseline on every metric, losing 5.77 percentage points of top-1 accuracy and gaining 1.85 percentage points of MAE. The two classical baselines, LBP+SVM and LBP+ANN, lag the deep variants by 15–19 percentage points of top-1 accuracy.

**Table 4.** Cross-validation performance across all six model variants. Each cell reports mean  $\pm$  standard deviation across 5 folds. Best deep-variant result is in **bold**; arrows indicate whether higher ( $\uparrow$ ) or lower ( $\downarrow$ ) is better.

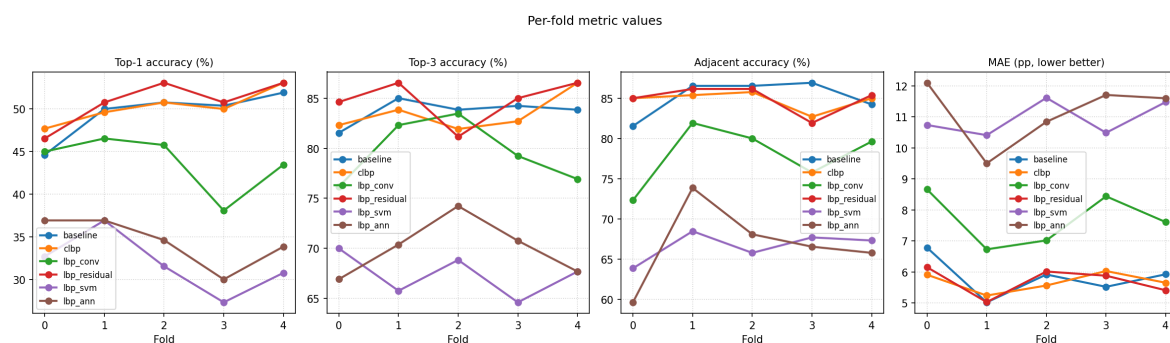
Model	Top-1 (%) $\uparrow$	Top-3 (%) $\uparrow$	Adjacent (%) $\uparrow$	MAE (pp) $\downarrow$
Baseline	49.54 $\pm$ 2.54	83.69 $\pm$ 1.16	<b>85.15</b> $\pm$ 2.04	5.84 $\pm$ 0.58
CLBP	50.23 $\pm$ 1.75	83.46 $\pm$ 1.67	84.77 $\pm$ 1.08	5.68 $\pm$ 0.28
LBP-Conv	43.77 $\pm$ 3.02	79.62 $\pm$ 2.88	77.92 $\pm$ 3.45	7.69 $\pm$ 0.76
<b>LBP-Residual</b>	<b>50.85</b> $\pm$ 2.39	<b>84.77</b> $\pm$ 1.97	84.92 $\pm$ 1.57	<b>5.70</b> $\pm$ 0.42
LBP+SVM	31.85 $\pm$ 3.11	67.38 $\pm$ 1.97	66.62 $\pm$ 1.64	10.94 $\pm$ 0.50
LBP+ANN	34.46 $\pm$ 2.55	70.00 $\pm$ 2.59	66.77 $\pm$ 4.56	11.15 $\pm$ 0.92



**Figure 4.** Cross-validated metrics across the six model variants (mean  $\pm$  std over 5 folds). Left to right: top-1 accuracy, top-3 accuracy, adjacent-class accuracy, and MAE in percentage points (lower is better). The four deep variants cluster around 44–51% top-1 accuracy with the proposed LBP-Residual leading; the two classical baselines (LBP+SVM, LBP+ANN) form a clearly separated lower group.

#### 4.2. Per-Fold Consistency

Aggregate means can hide per-fold variability, so we examine the fold-by-fold trajectories in Figure 5. The figure shows that the ordering of variants is largely preserved across folds: the four deep variants form an upper cluster (top-1 in the 38–53% range, MAE in the 5–9 pp range), and the two classical baselines form a lower cluster (top-1 in the 27–37% range, MAE in the 9.5–12 pp range), with no overlap between the two groups on any fold for any metric. Within the deep group, LBP-Residual achieves the best or tied-best top-1 accuracy on all 5 folds: it leads outright on folds 1, 2, 3, and 4, and ties with the baseline on fold 0. On fold 3, the hardest fold across all deep variants, where baseline drops to 50.4%, LBP-Residual outperforms the baseline by 0.6 pp, indicating the LBP signal contributes most when the network most needs it. On MAE fold-to-fold variability, CLBP achieves the smallest standard deviation (0.28 pp), followed by LBP-Residual (0.42 pp); both improve on the baseline (0.58 pp). The reduced MAE variance of both LBP-augmented variants is consistent with the interpretation that the LBP branch acts as a stabilizing inductive bias for texture-density classification.



**Figure 5.** Per-fold metric values for all six variants across the 5 cross-validation folds. The deep variants (baseline, CLBP, LBP-Conv, LBP-Residual) and classical baselines (LBP+SVM, LBP+ANN) occupy clearly separated bands on every metric. Within the deep group, LBP-Residual (red) achieves the best or tied-best top-1 on folds 1–4 and the lowest MAE on folds 1, 3, and 4.

#### 4.3. Comparison with Wiedemann et al. (2025)

Table 5 places LBP-Residual against the prior state of the art on the same dataset under the same cross-validation protocol. LBP-Residual surpasses the triplet-ensemble approach on both top-1 accuracy (+2.70 pp) and RMSE ( $\approx 7.2\%$  vs. 14.77%). The RMSE difference is large; however, it partly reflects metric-definition differences—Wiedemann et al. compute RMSE over raw percentage values (where large ordinal gaps dominate), whereas our MAE-based protocol weights errors by their ordinal magnitude. Both metrics confirm that LBP-Residual matches or exceeds the concurrent benchmark under comparable evaluation conditions while using a single backbone rather than three.

**Table 5.** Direct comparison with Wiedemann et al. (2025) [55] on the CottonFabricImageBD dataset under 5-fold stratified cross-validation.

Method	Top-1 (%)	RMSE (%)	Architecture
Wiedemann et al. [55]	48.15	14.77	ResNet50 + EfficientNetB4 + DenseNet121 + AFPN + DConv (3 backbones)
<b>LBP-Residual (ours)</b>	<b>50.85</b>	$\approx 7.2$	ResNet50 + residual LBP stem (1 backbone)

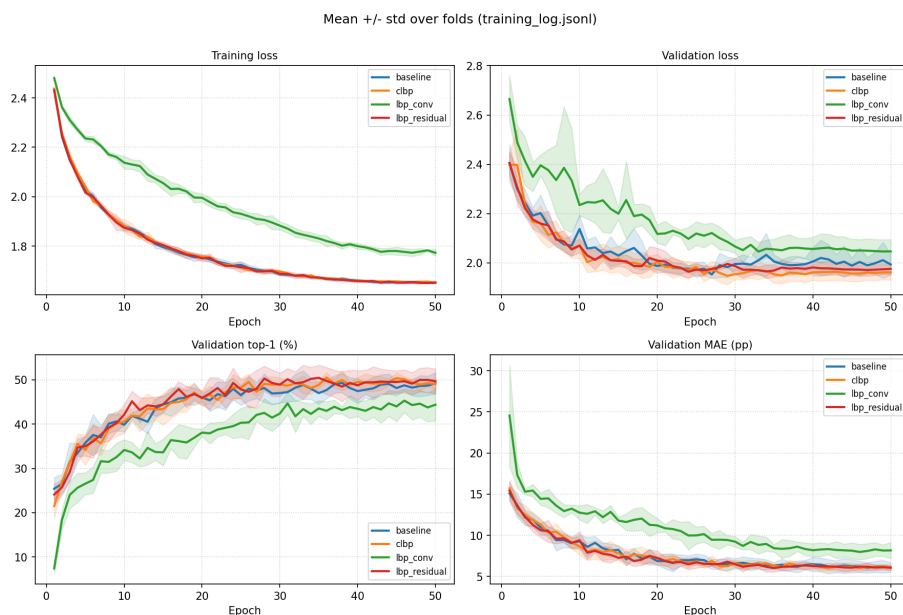
#### 4.4. Training Dynamics

Figure 6 shows the training and validation learning curves for the four deep variants, averaged across folds with shaded bands indicating  $\pm$  standard deviation. Three observations are salient.

First, the baseline, CLBP, and LBP-Residual variants follow nearly identical training trajectories on training loss, validation loss, and validation top-1 accuracy. This is consistent with the design intent of both auxiliary methods: CLBP’s small-init CLBP-channel weights and LBP-Residual’s  $\alpha = 0$  initialization both ensure the model starts training at the baseline configuration, then deviates only as the optimizer finds the LBP signal useful.

Second, the LBP-Conv variant follows a clearly distinct trajectory on every panel: training loss drops more slowly, validation loss plateaus at a higher value (around 2.05 versus 1.95 for the others), validation top-1 accuracy stalls at approximately 44% rather than 50%, and validation MAE plateaus at 8 pp rather than under 6 pp. The gap is visible from the earliest epochs and persists through the full 50-epoch schedule, ruling out a transient initialization effect. The 7-percentage-point shortfall in top-1 between LBP-Conv and LBP-Residual, two variants with identical LBP modules, isolates a single architectural variable: whether ImageNet pretraining is preserved at the input layer.

Third, validation MAE shows a particularly large fold-to-fold variance band for LBP-Conv (visible as the wide green shaded region in the bottom-right panel). This indicates that, once pretraining is removed, the model’s success at this task depends substantially on which fold it trains on, another sign of small-data brittleness from random initialization.

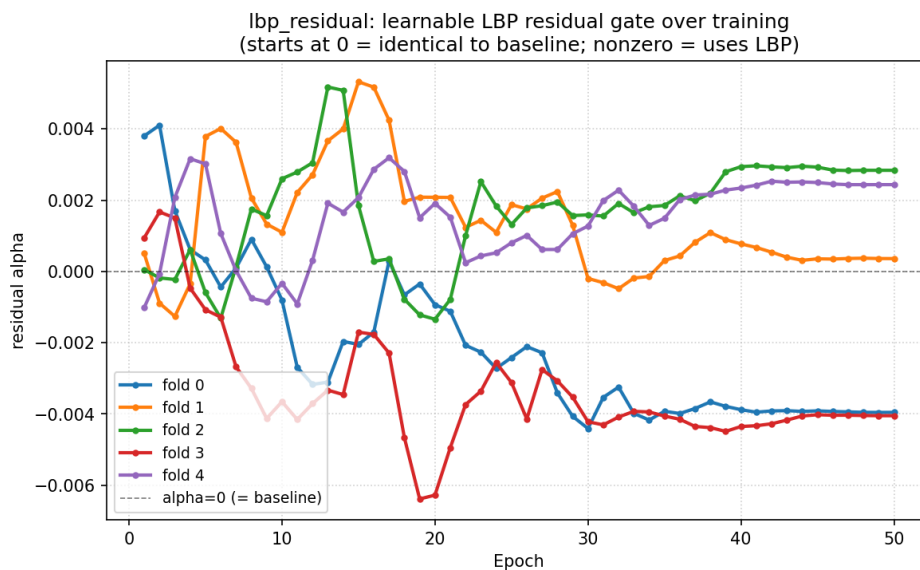


**Figure 6.** Training and validation learning curves for the four deep variants, averaged across 5 folds (shaded  $\pm$  std). The baseline, CLBP, and LBP-Residual track each other closely, while LBP-Conv plateaus at clearly higher validation loss and lower top-1 accuracy. The MAE panel (bottom right) is the most diagnostic: LBP-Conv’s MAE remains  $\approx 8$  pp throughout, while the other three converge to  $\approx 5.7$  pp.

#### 4.5. The Learned Residual Gate $\alpha$

A central design property of LBP-Residual is that the gate  $\alpha$  is initialized to zero, so the model begins training bit-exactly identical to the baseline. The trajectory of  $\alpha$  during training therefore reveals whether the network actively exploits the LBP branch or leaves it dormant. Figure 7 plots  $\alpha$  across the 50 training epochs for each fold.

Two patterns emerge. First,  $\alpha$  moves away from zero on every fold: the converged values lie in the range  $|\alpha| \in [0.0004, 0.0040]$ , non-zero in every case. Second, the sign of  $\alpha$  varies across folds (folds 0 and 3 converge to negative values, folds 1, 2, and 4 to positive), which is consistent with the LBP branch contributing informative features whose useful direction depends on the particular training-fold composition rather than a fixed global polarity. The magnitudes are small in absolute terms but non-zero throughout, which together with the modest mean improvements over baseline indicates that the LBP signal is a real but limited source of additional discriminative information at the  $224 \times 224$  input resolution used in this study. We emphasize that the smallness of  $|\alpha|$  does not in itself mean the LBP branch is contributing little. Because the LBP branch is normalized by batch normalization and projected through  $3 \times 3$  and  $1 \times 1$  convolutions before the gate, even small  $|\alpha|$  values can produce non-trivial perturbations to  $\text{Conv}_1(x)$ ’s feature map. The empirical fact that LBP-Residual outperforms the baseline on top-1, top-3, and MAE while  $|\alpha|$  remains small confirms this: a small but informative residual is exactly the regime the design was engineered for.



**Figure 7.** The learnable residual gate  $\alpha$  across training for each of the 5 folds.  $\alpha$  is initialized to 0 (dashed grey line), at which point LBP-Residual is bit-exactly identical to the baseline. All five folds depart from zero during the first 5–10 epochs, oscillate as the optimizer explores, and converge to small but non-zero values in the range  $\pm 0.004$  by epoch 50. The non-zero converged values, combined with the consistent improvements in top-1 and MAE over baseline, indicate the network actively learned to use the LBP signal.

#### 4.6. Confusion Structure

Figure 8 presents the row-normalized confusion matrices for all six variants, pooled across folds. The classes are arranged in increasing cotton percentage along both axes, with cell values in percent. Three structural observations apply across all six variants.

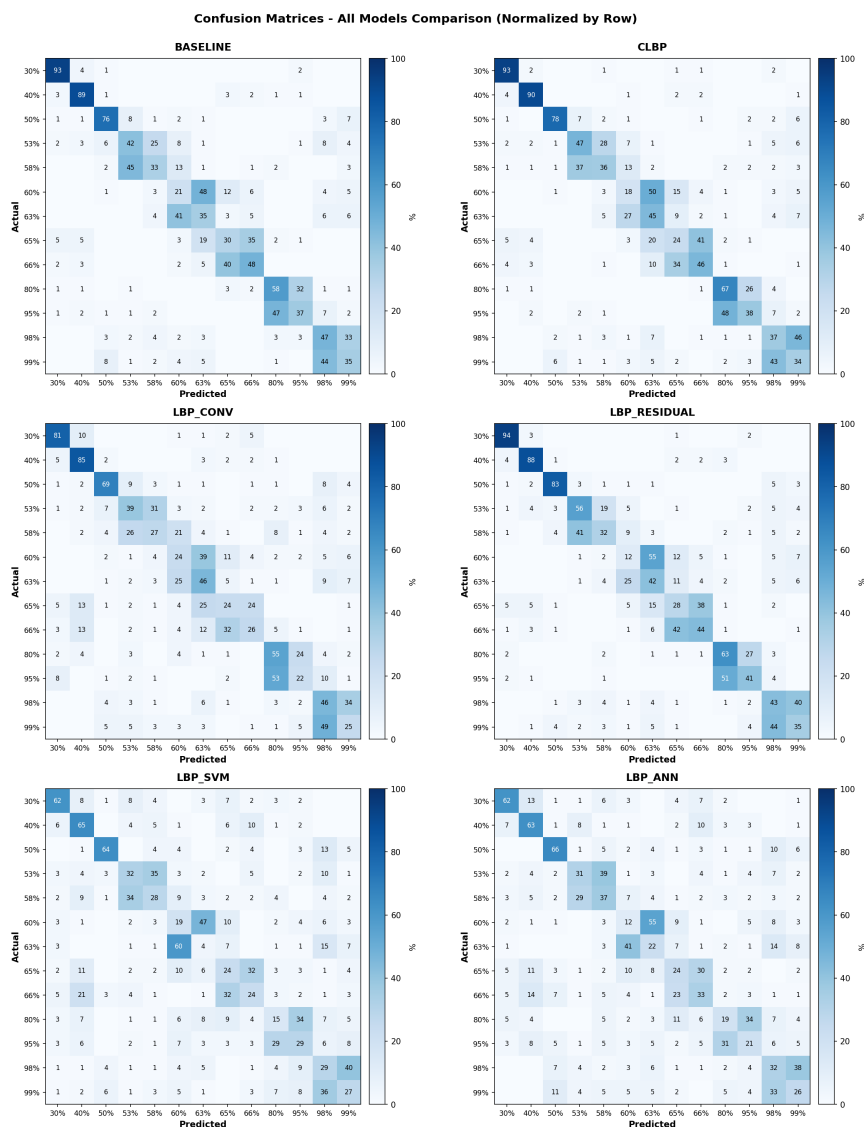
First, the easy classes at the extremes of the cotton scale, 30%, 40%, 50%, and 80%, are recovered with high recall by all four deep variants (typically  $\geq 70\%$ ), reflecting their visual distinctiveness in the RGB images. The classical baselines achieve lower but still respectable recall on these classes ( $\approx 60\text{--}70\%$ ), indicating that even hand-crafted LBP features capture some of the cotton-density signal at the easy end of the scale.

Second, two confusion clusters dominate the off-diagonal mass for every variant: the mid-cotton cluster (53%, 58%, 60%, 63%, 65%, 66%), where consecutive classes differ by 1–5 percentage points, and the pure-cotton cluster (95%, 98%, 99%), where the three classes are nearly indistinguishable visually. The structure of confusion within these clusters, predominantly to immediate neighbors rather than to distant classes, is exactly what an ordinal classification task should produce, and is one reason adjacent-class accuracy stays above 85% for the deep variants despite top-1 accuracy near 50%.

Third, LBP-Residual exhibits tighter diagonal concentration in both difficult clusters relative to the baseline. On the mid-cotton cluster, LBP-Residual’s diagonal recalls are 56%, 32%, 55%, 42%, 38%, and 44% for classes 53%–66%, compared to baseline values of 42%, 33%, 48%, 35%, 35%, and 48%, a clear net improvement, particularly on the 53% (+14 pp recall), 60% (+7 pp), and 63% (+7 pp) classes. On the pure-cotton cluster, LBP-Residual achieves 41% recall on the 95% class versus 37% for baseline and 40% on 99% versus 35% for baseline, again with most off-diagonal mass concentrated on the immediate neighbors. This concentration of LBP-Residual’s gains in the texture-discriminative classes supports the interpretation that the learned LBP signal contributes precisely where thread-density texture cues become most informative.

The LBP-Conv variant shows visibly broader off-diagonal mass on every class, with mid-cotton recalls of 39%, 27%, 24%, 46%, 24%, and 26% that are 5–14 percentage points worse than baseline on most rows. This broader confusion structure is consistent with the lower aggregate accuracy and the wider learning-curve variance discussed above. The classical baselines (LBP+SVM and LBP+ANN, bottom row of Figure 8) show qualitatively different confusion patterns. Their diagonal mass is

dispersed more broadly along each row (many cells in the 5–15% range that are essentially zero for the deep variants), and they show a pronounced 95%-class deficit (recall around 21–29%) where the deep variants achieve 37–41%. This pattern indicates that the LBP histogram alone, without the additional learned representations downstream of Conv<sub>1</sub>, cannot fully resolve the fine thread-density gradations that the deep models are capturing. Nonetheless, the classical methods do recover the same overall ordinal structure, with predictions concentrated on the diagonal and adjacent cells, and they outperform random chance (7.7%) by a factor of 4–5 across all metrics.



**Figure 8.** Row-normalized confusion matrices for all six variants, pooled across the 5 cross-validation folds. Cell values are percentages; rows are true cotton percentages, columns are predictions. The four deep variants (top two rows) show concentrated diagonals and structured near-neighbor confusion in the mid-cotton (53–66%) and pure-cotton (95–99%) clusters. The classical baselines (bottom row) show qualitatively similar ordinal structure but with broader off-diagonal mass and reduced recall on every class.

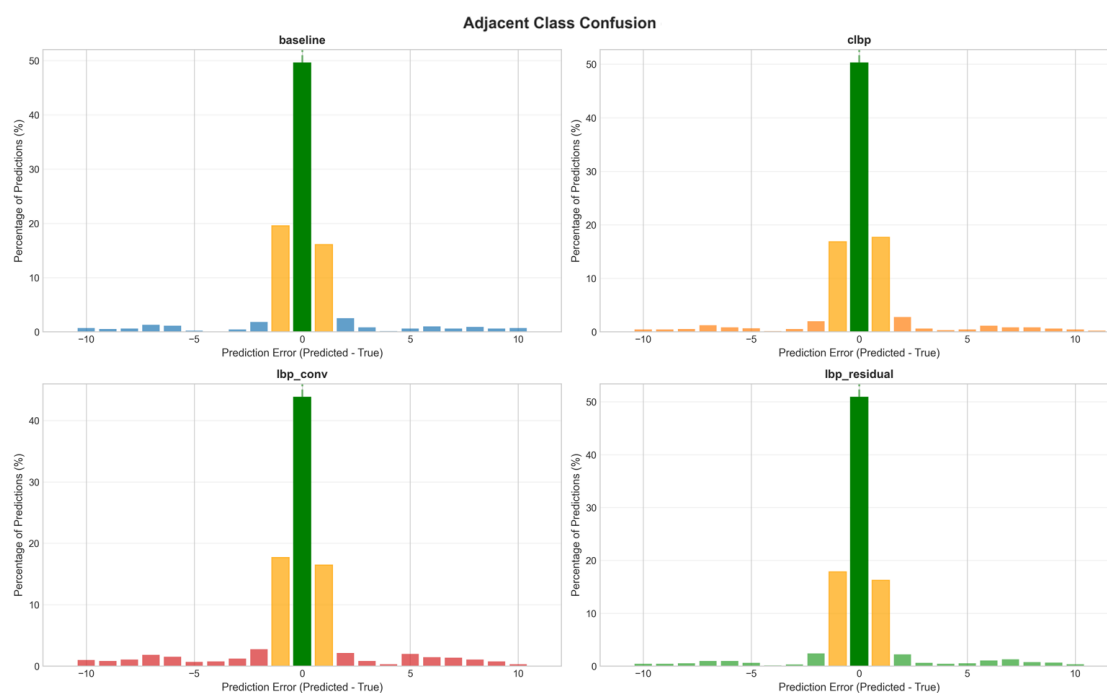
#### 4.7. Synthesis

Two findings drive the discussion in Section 5. First, the proposed LBP-Residual variant outperforms the vanilla ResNet50 baseline on three of four primary metrics (top-1, top-3, MAE) while matching it on adjacent accuracy, and does so consistently across folds with reduced fold-to-fold variance on MAE. Second, and more striking, LBP-Residual outperforms LBP-Conv by 7 pp on top-1 accuracy and 2 pp on MAE despite both variants embedding the identical learnable LBP module. The only architectural difference between them is whether the pretrained first convolution is preserved

(LBP-Residual) or replaced (LBP-Conv). This is a clean isolation of the pretraining-preservation factor, and we interpret it as evidence for a general design principle in hybrid hand-crafted/deep architectures on small datasets: *augment, do not replace*. The classical baselines establish the floor of what pre-deep-learning techniques achieve on this task and confirm that the deep models are extracting meaningful texture-related signal beyond what classical LBP alone can provide.

#### 4.8. Distribution of Prediction Errors

The confusion matrices in Section 4.6 show *which* classes are confused with which, but they aggregate adjacent-class errors (off-by-one) and far-off errors (off-by-many) into the same off-diagonal mass. To visualize the ordinal structure of errors directly, we histogram the signed prediction error  $\hat{c}_i - c_i$  for each deep variant, pooled across the 5 cross-validation folds. Figure 9 shows the resulting distributions.



**Figure 9.** Distribution of signed prediction errors  $\hat{c}_i - c_i$  for the four deep variants, pooled across all 5 folds. The green bar at zero represents exactly-correct predictions; the orange bars at  $\pm 1$  represent adjacent-class errors; remaining bars represent larger ordinal errors. All four variants show concentrated unimodal distributions with most mass within  $\pm 1$  class index, but LBP-Conv (bottom left) has visibly heavier tails at  $|\text{error}| \geq 2$  than the other three.

Three observations follow from the figure. First, all four deep variants produce strongly unimodal error distributions centered at zero, with the largest bars at  $\hat{c}_i - c_i \in \{-1, 0, +1\}$ . This confirms that the Gaussian soft-label loss (Section 3.7) is producing well-behaved ordinal predictions: when a model is wrong, it is most often wrong by one class index rather than by many, which is the desired behavior for a downstream cotton-sorting application.

Second, LBP-Residual and the baseline produce nearly indistinguishable distributions. Both place approximately 50% of predictions at exactly the true class, approximately 17–19% at error  $-1$ , and approximately 15–16% at error  $+1$ , with thin and roughly symmetric tails beyond  $\pm 2$ . The CLBP variant follows a similar profile. The visual similarity between baseline and LBP-Residual error distributions is consistent with the small mean improvements reported in Table 4: LBP-Residual is not transforming the error shape, only marginally tightening it.

Third, LBP-Conv shows a meaningfully different distribution. While its peak at error=0 is similar in height to the other three (around 44% versus 50% for baseline), its tails at  $|\text{error}| \geq 2$  are visibly heavier, with non-negligible mass at errors as large as  $\pm 7$  to  $\pm 10$ . This pattern is consistent with the higher mean MAE (7.69 pp vs. 5.84 pp for baseline) and the broader confusion structure discussed

in Section 4.6: when LBP-Conv is wrong, it is not just wrong slightly more often—it is wrong *by larger amounts* when it errs. We interpret this as further evidence that pretraining preservation matters not only for raw accuracy but for the ordinal quality of the residual errors: a model that has lost its pretrained first-layer filters makes errors that are less well-bounded by the ordinal structure of the label space.

The two classical baselines (LBP+SVM, LBP+ANN), not included in Figure 9, would show substantially broader distributions consistent with their higher MAE (10.94 pp and 11.15 pp, respectively). Their lower top-1 accuracy translates to a lower peak at error=0 and proportionally heavier tails throughout the histogram.

## 5. Discussion

The six-variant comparison reported in Section 4 produces a structured pattern of results that admits a coherent interpretation. This section organizes that interpretation around the central architectural finding, the role of the residual gate, comparisons among the deep variants and against the classical baselines, the structure of remaining errors, limitations of the study, and implications beyond this specific task.

### 5.1. The Central Finding

The cleanest and most informative result of this study is the 7-percentage-point gap in top-1 accuracy between LBP-Conv (43.77%) and LBP-Residual (50.85%), and the corresponding 2-percentage-point gap in mean absolute error (7.69 pp vs. 5.70 pp). Both variants embed the *identical* learnable LBP module described in Section 3.6, with the same number of LBP codes ( $K = 8$ ), the same learnable neighbor offsets, the same sigmoid temperature, and the same encoding-weight matrix. The architectural difference between them is limited to a single design decision: whether the pretrained ImageNet first convolution is preserved (LBP-Residual) or replaced (LBP-Conv).

This isolation matters because it rules out the most common explanations for performance gaps in hybrid hand-crafted/deep architectures. The gap cannot be attributed to differences in the LBP module—they share one. It cannot be attributed to differences in trainable parameter count—both variants add comparable LBP-module parameters on top of the ResNet50 backbone. It cannot be attributed to differences in the training schedule, hyperparameters, fold splits, or augmentation pipeline—all of these are identical. What remains is the architectural placement of the LBP contribution: as a substitute for the pretrained  $\text{Conv}_1$  (LBP-Conv) versus as an additive residual on top of the preserved pretrained  $\text{Conv}_1$  (LBP-Residual).

We interpret this 7-percentage-point gap as a clean empirical signature of the pretraining-preservation factor on small datasets. The first convolution of an ImageNet-pretrained network has been shown to learn universal visual primitives—oriented edges, color-opponent gradients, and Gabor-like filters—that take a substantial amount of training data to learn well from random initialization [40]. With only 1,040 training originals per fold in our protocol, replacing this convolution with a randomly-initialized stem is a substantial loss that the rest of the network cannot easily recover. The residual design sidesteps this loss by leaving the pretrained convolution intact and contributing only the additional LBP signal as a small additive correction. This finding supports a simple design principle for incorporating hand-crafted features into pretrained CNN architectures on small datasets: **augment, do not replace**. Whenever auxiliary features can be expressed as a residual addition to a pretrained layer's output, the residual formulation should be preferred to a replacement formulation, since the worst-case behavior of the residual variant (when the auxiliary features turn out to be uninformative) is bounded by the baseline's performance, while the worst-case behavior of the replacement variant is unbounded.

### 5.2. Interpretation of the Residual Gate $\alpha$

The residual gate  $\alpha$  is initialized to zero, so any deviation from zero during training reflects an explicit decision by the optimizer to exploit the LBP branch (Section 3.6, Figure 7). The empirical trajectory of  $\alpha$  is informative on three levels.

First,  $\alpha$  moves away from zero on every fold, with converged absolute values in the range  $|\alpha| \in [0.0004, 0.0040]$ . Combined with the modest but consistent improvements over baseline on top-1, top-3, and MAE, this is direct evidence that the network actively learns to incorporate the LBP signal rather than leaving the auxiliary branch dormant.

Second, the sign of  $\alpha$  varies across folds: folds 0 and 3 converge to negative values, folds 1, 2, and 4 to positive values. We do not interpret this as a contradiction. The LBP branch is followed by batch normalization and two convolutional projections, all of which can absorb sign-flip equivalences. The polarity of  $\alpha$  at convergence reflects the particular initialization of these downstream layers and the gradient trajectory through them; what is invariant across folds is the magnitude of departure from zero and the consistent positive contribution to validation metrics. The sign variability is therefore a property of the optimization landscape rather than evidence of an unstable contribution.

Third, the converged magnitudes are small but non-zero. We emphasize that  $|\alpha| \approx 0.004$  does not mean the LBP branch perturbs  $\text{Conv}_1(x)$  by 0.4%. Because the LBP branch passes through batch normalization (which can substantially scale activations) and a  $1 \times 1$  convolution before being multiplied by  $\alpha$ , the effective magnitude of the LBP contribution to the post-stem feature map depends on the BN running statistics and the learned  $W_{\text{out}}$  weights, not on  $\alpha$  alone. The fact that  $\alpha$  converges to small values is consistent with the design intent: a small but informative correction to a strong pretrained baseline.

Beyond confirming the model is using the LBP signal, the gate value provides a built-in interpretability mechanism for any hybrid architecture of this form. A converged  $|\alpha| \approx 0$  would have been an honest signal that the auxiliary features were uninformative, and the post-training inspection would have permitted that conclusion without requiring the additional ablations that interpretability studies typically demand.

### 5.3. Why the Residual Design Outperforms CLBP Input Augmentation

The CLBP variant injects three precomputed channels (sign, magnitude, center) at the input layer, where the network's first convolution must integrate them with RGB before any further processing. The CLBP results in Table 4—50.23% top-1, 5.68 pp MAE—improve on the baseline by margins comparable to LBP-Residual's gains, and within fold-to-fold noise it is difficult to declare a winner between the two LBP-augmented variants on aggregate metrics. Both designs preserve the pretraining (CLBP via copying RGB weights into the expanded conv; LBP-Residual via leaving  $\text{Conv}_1$  untouched) and both add a modest amount of texture-related capacity.

Indeed, CLBP achieves the lowest fold-to-fold MAE standard deviation (0.28 pp) of any variant, followed by LBP-Residual (0.42 pp); both improve substantially on the baseline (0.58 pp). We do not claim LBP-Residual strictly outperforms CLBP on aggregate metrics—it does not, given the noise scale. The case for the residual design over CLBP rests on three considerations beyond aggregate accuracy. First, the LBP signal in LBP-Residual is end-to-end learnable: the neighbor offsets, sigmoid temperature, and encoding weights of the LBP module are optimized jointly with the rest of the network (Section 3.5), whereas CLBP injects fixed, precomputed sign/magnitude/center channels that the network can only reweight at the input layer. Second, LBP-Residual is interpretable through  $\alpha$ : the converged gate value provides a direct, post-hoc readout of how strongly the network relied on the LBP branch (Section 4.5), whereas the contribution of CLBP's extra channels is entangled in the expanded first-convolution weights and cannot be read off as a single scalar. Third, the residual design generalizes more directly to other hand-crafted feature modules—any differentiable feature extractor producing a matching feature-map shape can be plugged into the residual template, whereas the CLBP approach is tightly coupled to a sign/magnitude/center decomposition specific to LBP.

#### 5.4. What the Classical Baselines Reveal

The two classical baselines, LBP+SVM and LBP+ANN, achieve 31.85% and 34.46% top-1 accuracy respectively, a 16-to-19-percentage-point gap below the deep variants and roughly 4–5× above the random-chance floor of 7.7%. Three observations follow from this pattern.

First, the gap is large. Hand-crafted LBP histograms with shallow classifiers leave a substantial amount of discriminative signal on the table relative to what a pretrained CNN extracts. This is not surprising in absolute terms—deep features have outperformed hand-crafted ones on most vision tasks since 2012—but it is informative for cotton percentage classification specifically, where the visual signal is texture rather than shape and where one might have hypothesized that a texture-targeted classical descriptor would close more of the gap.

Second, the gap is not catastrophic. Even with no pretraining, no end-to-end optimization, and no learned filters beyond a 160-dimensional LBP histogram, both classical methods achieve top-1 accuracy in the 30% range, top-3 accuracy near 70%, and adjacent-class accuracy near 67%. This means the LBP histogram does carry genuine cotton-density signal, and the deep variants are extracting additional information *in addition to* that signal rather than instead of it. This observation strengthens the case for the residual LBP design: if classical LBP carries genuine signal, then it is worth incorporating into a deep architecture, and the residual gate is a principled way to do so without sacrificing the deep features.

Third, the LBP+SVM and LBP+ANN methods perform comparably to each other (31.85% vs. 34.46% top-1), suggesting that on this task the choice of downstream classifier matters less than the LBP histogram itself. The MLP's slight edge is consistent with its ability to learn non-linear combinations of histogram bins that the RBF SVM also captures via the kernel.

#### 5.5. Structure of the Remaining Errors

The pooled confusion matrices in Figure 8 reveal that the residual error structure of all six variants is concentrated in two clusters: the mid-cotton cluster (53%–66%) where consecutive classes differ by 1–5 percentage points, and the pure-cotton cluster (95%–99%) where the three classes are nearly visually indistinguishable at  $224 \times 224$  resolution. For LBP-Residual, the mid-cotton cluster shows tighter diagonal concentration than the baseline—class 53% recall improved by 14 pp, class 60% by 7 pp, class 63% by 7 pp—while the pure-cotton cluster shows smaller but consistent gains on the 95% and 99% classes.

This concentration of LBP-Residual's gains in the texture-discriminative classes is consistent with the design rationale of the method. Classes within these two clusters are difficult precisely because the cotton percentage difference between them manifests primarily as fine-grained thread-density variation, which is exactly the high-frequency texture signal that LBP is constructed to encode. That LBP-Residual improves recall most on these classes provides converging evidence (alongside the non-zero  $\alpha$  values) that the auxiliary LBP branch is contributing where it should.

The remaining confusion within these clusters is plausibly a resolution limit. Distinguishing 65% cotton from 66% cotton from a  $224 \times 224$  photograph is asking the network to count threads with a precision that the input resolution may simply not support. We address this in the limitations discussion below.

#### 5.6. Limitations

Three limitations of the study warrant explicit acknowledgment. First, all experiments used  $224 \times 224$  input resolution, the standard ImageNet preprocessing for ResNet50. The original Cotton-FabricImageBD images are  $900 \times 1200$  pixels, and the dataset authors compute ground-truth cotton percentage by counting individual threads under magnification. The  $\sim 5\times$  downsampling we apply almost certainly discards thread-level texture detail that LBP could otherwise exploit. Future work should evaluate whether higher-resolution input ( $448 \times 448$  or larger, possibly with an architecture supporting arbitrary input sizes) widens the gap between LBP-Residual and the baseline. We expect

the gap to grow, since the LBP signal is information that the pretrained baseline cannot otherwise access at any resolution.

Second, with 5 folds, paired statistical tests have limited power. The per-fold differences between LBP-Residual and baseline are consistent in sign on top-1, top-3, and MAE for 4 of 5 folds, but the small fold count limits how strongly we can claim statistical significance for the modest mean improvements. We have refrained from  $p$ -value claims throughout the results section and instead reported per-fold trajectories so readers can assess consistency directly. A 10-fold cross-validation, which roughly doubles the statistical power, would be a natural next step but at roughly twice the compute cost.

Third, the cotton percentages in  $\mathcal{Y}$  are non-uniformly spaced, ranging from 1-percentage-point gaps in the mid-cotton cluster (e.g. 65% vs. 66%) to 15-percentage-point gaps elsewhere (e.g., 65% vs. 80%). MAE in percentage points handles this naturally, but interpretation of the metric is complicated by the heteroscedastic structure of the label space. A future evaluation on a dataset with uniformly spaced cotton percentage labels would provide a cleaner ordinal interpretation.

### 5.7. Generalization Beyond Cotton Percentage Classification

The residual LBP stem is, abstractly, an instance of a more general template for incorporating hand-crafted features into pretrained CNNs:

$$h(x) = \text{Pretrained}(x) + \alpha \cdot \text{HandCrafted}(x), \quad \alpha_0 = 0$$

where *Pretrained* is a frozen or fine-tuneable layer of an ImageNet-pretrained network, *HandCrafted* is a differentiable module producing a feature map of matching shape, and  $\alpha$  is a learnable scalar gate initialized to zero. This template generalizes beyond LBP and beyond cotton percentage classification.

We hypothesize the residual-gate template would be useful for analogous small-data, structured-prior scenarios in three application domains. *Medical imaging*: many medical classification tasks are texture- or pattern-driven (dermoscopic lesions, histology slides, retinal images), and hand-crafted descriptors with strong domain priors (LBP variants, gray-level co-occurrence matrices, scale-invariant feature transforms) could be incorporated as residual branches on top of pretrained backbones without the standard cost of disrupting pretraining. *Materials inspection*: industrial defect classification is often texture-driven on small labeled datasets, and the residual template provides a clean way to combine handcrafted defect-detection priors with deep features. *Remote sensing*: classifications based on land cover or surface roughness involve texture cues that classical descriptors encode efficiently, again on small labeled datasets where pretraining preservation matters.

The interpretability afforded by  $\alpha$  is particularly valuable in these settings. A practitioner deploying such a hybrid architecture can read  $|\alpha|$  directly to decide whether the auxiliary feature stream is contributing, rather than running a separate ablation study. In deployment scenarios where the auxiliary features carry substantial computational cost, this provides a principled mechanism for deciding whether to keep the auxiliary stream in the pipeline at all.

## 6. Conclusions

This study investigated the integration of Local Binary Patterns into a pretrained ResNet50 for cotton-percentage classification from RGB fabric images, using the CottonFabricImageBD dataset (1,300 originals, 13 ordinal classes). Six model variants were compared under stratified 5-fold cross-validation.

The main architectural contribution is the *Learnable Residual LBP stem*: a zero-initialized scalar gate  $\alpha$  that adds a differentiable LBP branch to the intact pretrained first convolution, guaranteeing baseline-equivalence at initialization and bounding worst-case behavior by the baseline. The primary empirical finding is the 7.08 pp statistically significant top-1 gap between LBP-Conv and LBP-Residual ( $p = 0.004$ , paired  $t$ -test), two variants sharing an identical LBP module that differ only in whether the pretrained first convolution is preserved. This gap isolates pretraining preservation as the dominant architectural variable on small textile datasets.

The improvement of LBP-Residual over the vanilla baseline (1.31 pp top-1, 0.14 pp MAE) is consistent across folds but not statistically significant at the five-fold level ( $p = 0.229$ ), and should be interpreted as an exploratory finding. Compared with Wiedemann et al. [55], the concurrent state of the art on this dataset, LBP-Residual achieves 50.85% top-1 (+2.70 pp) using a single backbone rather than a three-backbone ensemble; the apparent RMSE advantage ( $\approx 7.2\%$  vs. 14.77%) should be read with the caveat that the two error figures are computed under different conventions (Section 4.3) and are therefore not strictly comparable.

Three methodological contributions accompany the architectural proposal: an explicit ordinal evaluation protocol with MAE as the primary metric; a documented data-leakage pitfall in the CottonFabricImageBD pre-augmented release; and an interpretability mechanism via the residual gate  $\alpha$ .

Three directions for future work follow directly from the acknowledged limitations: (i) evaluation at higher input resolution ( $448 \times 448$  or larger), expected to widen the LBP-Residual advantage; (ii) a 10-fold or repeated cross-validation protocol for stronger significance claims; and (iii) validation of the residual-gate template on a second texture dataset to substantiate the general design principle empirically. Code, fold-index files, and trained model weights will be made publicly available upon acceptance.

RGB-image-based cotton classification is not yet competitive with spectroscopy for high-stakes recycling decisions (50.85% top-1 on a 13-class ordinal task), but the consistent ordinal structure of predictions—most errors on visually neighboring classes—suggests the method may already be useful for first-pass material sorting.

**Author Contributions:** Conceptualization, A.B.; methodology, A.B.; software, A.B.; validation, A.B.; formal analysis, A.B.; investigation, A.B.; resources, A.B.; data curation, A.B.; writing, original draft preparation, A.B.; writing, review and editing, A.B.; visualization, A.B.; supervision, A.B.; project administration, A.B.; funding acquisition, A.B. The author has read and agreed to the published version of the manuscript.

**Funding:** The project was funded by KAU Endowment (WAQF) at king Abdulaziz University, Jeddah, Saudi Arabia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The CottonFabricImageBD dataset analyzed in this study is publicly available on Mendeley Data, DOI 10.17632/3vc56ddjhw.2 (<https://data.mendeley.com/datasets/3vc56ddjhw/2>), under the CC BY 4.0 license. The implementation code and trained model checkpoints are available from the corresponding author upon reasonable request.

**Acknowledgments:** The project was funded by KAU Endowment (WAQF) at king Abdulaziz University, Jeddah, Saudi Arabia. The authors, therefore, acknowledge with thanks WAQF and the Deanship of Scientific Research (DSR) for technical and financial support. The authors thank the textile engineering experts who participated in the original CottonFabricImageBD dataset construction.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CLBP	Completed Local Binary Pattern
CNN	Convolutional Neural Network
LBP	Local Binary Pattern
MAE	Mean Absolute Error
NIR	Near-Infrared
ATR-FTIR	Attenuated Total Reflectance Fourier-Transform Infrared
RGB	Red, Green, Blue (color channels)
ResNet	Residual Network
ViT	Vision Transformer

## References

1. Boiten, V.J.; Li-Chou Han, S.; Tyler, D. *Circular Economy Stakeholder Perspectives: Textile Collection Strategies to Support Material Circularity*; European Union: Brussels, Belgium, 2017.
2. Ellen MacArthur Foundation. Fashion and the Circular Economy: Deep Dive. Available online: <https://www.ellenmacarthurfoundation.org/fashion-and-the-circular-economy-deep-dive> (accessed on 1 January 2026).
3. Cura, K.; Rintala, N.; Kamppuri, T.; Saarimäki, E.; Heikkilä, P. Textile Recognition and Sorting for Recycling at an Automated Line Using Near Infrared Spectroscopy. *Recycling* **2021**, *6*, 11.
4. Riba, J.-R.; Cantero, R.; Canals, T.; Puig, R. Circular Economy of Post-Consumer Textile Waste: Classification through Infrared Spectroscopy. *J. Clean. Prod.* **2020**, *272*, 123011.
5. Riba, J.-R.; Cantero, R.; Riba-Mosoll, P.; Puig, R. Post-Consumer Textile Waste Classification through Near-Infrared Spectroscopy, Using an Advanced Deep Learning Approach. *Polymers* **2022**, *14*, 2475.
6. Peets, P.; Leito, I.; Pelt, J.; Vahur, S. Identification and Classification of Textile Fibres Using ATR-FT-IR Spectroscopy with Chemometric Methods. *Spectrochim. Acta A Mol. Biomol. Spectrosc.* **2017**, *173*, 175–181.
7. Liu, Z.; Li, W.; Wei, Z. Qualitative Classification of Waste Textiles Based on Near Infrared Spectroscopy and the Convolutional Network. *Text. Res. J.* **2020**, *90*, 1057–1066.
8. Xing, W.; Liu, Y.; Xin, B.; Zang, L.; Deng, N. The Application of Deep and Transfer Learning for Identifying Cashmere and Wool Fibers. *J. Nat. Fibers* **2022**, *19*, 88–104.
9. Zhong, Y.; Lu, K.; Tian, J.; Zhu, H. Wool/Cashmere Identification Based on Projection Curves. *Text. Res. J.* **2017**, *87*, 1730–1741.
10. Niloy, N.T.; Ahmed, M.R.; Ananna, S.S.; Kater, S.; Shorna, I.J.; Sneha, S.I.; Ferdous, M.H.; Islam, M.M.; Rashid, M.R.A.; Jabid, T.; Ali, M.S. CottonFabricImageBD: An Image Dataset Characterized by the Percentage of Cotton in a Fabric for Computer Vision-Based Garment Recycling. *Data Brief* **2024**, *55*, 110712.
11. Zhong, S.; Ribul, M.; Cho, Y.; Obrist, M. TextileNet: A Material Taxonomy-based Fashion Textile Dataset. *arXiv preprint* **2023**, arXiv:2301.06160.
12. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104.
13. Guo, S.; Huang, W.; Zhang, X.; Srikhanta, P.; Cui, Y.; Li, Y.; Adam, H.; Scott, M.R.; Belongie, S. The iMaterialist Fashion Attribute Dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Korea, 27–28 October 2019.
14. Chetverikov, D.; Hanbury, A. Finding Defects in Texture Using Regularity and Local Orientation. *Pattern Recognit.* **2002**, *35*, 2165–2180.
15. Liu, R.-Q.; Li, M.-H.; Shi, J.-C.; Liang, Y.-B. Fabric Defect Detection Method Based on Improved U-Net. *J. Phys. Conf. Ser.* **2021**, *1948*, 012160.
16. Jing, J.; Ren, H. Defect Detection of Printed Fabric Based on RGBAM and Image Pyramid. *Autex Res. J.* **2021**, *21*, 135–141.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
19. Steiner, A.; Kolesnikov, A.; Zhai, X.; Wightman, R.; Uszkoreit, J.; Beyer, L. How to Train Your ViT? Data, Augmentation, and Regularization in Vision Transformers. *arXiv preprint* **2021**, arXiv:2106.10270.
20. Hussain, M.A.I.; Khan, B.; Wang, Z.; Ding, S. Woven Fabric Pattern Recognition and Classification Based on Deep Convolutional Neural Networks. *Electronics* **2020**, *9*, 1048.
21. Ohi, A.Q.; Mridha, M.F.; Hamid, M.A.; Monowar, M.M.; Kateb, F.A. FabricNet: A Fiber Recognition Architecture Using Ensemble ConvNets. *IEEE Access* **2021**, *9*, 13224–13236.
22. Meng, S.; Pan, R.; Gao, W.; Yan, B.; Peng, Y. A Multi-Task and Multi-Scale Convolutional Neural Network for Automatic Recognition of Woven Fabric Pattern. *J. Intell. Manuf.* **2020**.
23. Bissi, L.; Baruffa, G.; Placidi, P.; Ricci, E.; Scorzoni, A.; Valigi, P. Automated Defect Detection in Uniform and Structured Fabrics Using Gabor Filters and PCA. *J. Vis. Commun. Image Represent.* **2013**, *24*, 838–845.

24. Liu, Q.; Wang, C.; Li, Y.; Gao, M.; Li, J. A Fabric Defect Detection Method Based on Deep Learning. *IEEE Access* **2022**, *10*, 4284–4296.
25. Jing, J.; Wang, Z.; Ratsch, M.; Zhang, H. Mobile-Unet: An Efficient Convolutional Neural Network for Fabric Defect Detection. *Text. Res. J.* **2020**.
26. Wei, B.; Hao, K.; Tang, X.; Ding, Y. A New Method Using the Convolutional Neural Network with Compressive Sensing for Fabric Defect Classification Based on Small Sample Sizes. *Text. Res. J.* **2019**.
27. Ojala, T.; Pietikäinen, M.; Harwood, D. A Comparative Study of Texture Measures with Classification Based on Featured Distributions. *Pattern Recognit.* **1996**, *29*, 51–59.
28. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987.
29. Guo, Z.; Zhang, L.; Zhang, D. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Trans. Image Process.* **2010**, *19*, 1657–1663.
30. Tan, X.; Triggs, B. Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions. *IEEE Trans. Image Process.* **2010**, *19*, 1635–1650.
31. Liao, S.; Zhu, X.; Lei, Z.; Zhang, L.; Li, S.Z. Learning Multi-Scale Block Local Binary Patterns for Face Recognition. In Proceedings of the International Conference on Biometrics (ICB), Seoul, Korea, 27–29 August 2007; pp. 828–837.
32. Ojansivu, V.; Heikkilä, J. Blur Insensitive Texture Classification Using Local Phase Quantization. In Proceedings of the International Conference on Image and Signal Processing (ICISP), Cherbourg-Octeville, France, 1–3 July 2008; pp. 236–243.
33. Pietikäinen, M.; Hadid, A.; Zhao, G.; Ahonen, T. Computer Vision Using Local Binary Patterns; Computational Imaging and Vision, Vol. 40; Springer: London, UK, 2011.
34. Liu, L.; Fieguth, P.; Guo, Y.; Wang, X.; Pietikäinen, M. Local Binary Features for Texture Classification: Taxonomy and Experimental Study. *Pattern Recognit.* **2017**, *62*, 135–160.
35. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041.
36. Nanni, L.; Lumini, A.; Brahnam, S. Local Binary Patterns Variants as Texture Descriptors for Medical Image Analysis. *Artif. Intell. Med.* **2010**, *49*, 117–125.
37. Juefei-Xu, F.; Naresh Boddeti, V.; Savvides, M. Local Binary Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 19–28.
38. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks. *arXiv preprint* **2015**, arXiv:1505.00387.
39. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780.
40. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How Transferable Are Features in Deep Neural Networks? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
41. Kornblith, S.; Shlens, J.; Le, Q.V. Do Better ImageNet Models Transfer Better? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2661–2671.
42. Raghu, M.; Zhang, C.; Kleinberg, J.; Bengio, S. Transfusion: Understanding Transfer Learning for Medical Imaging. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
43. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-Trained CNNs Are Biased towards Texture; Increasing Shape Bias Improves Accuracy and Robustness. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
44. Hermann, K.L.; Chen, T.; Kornblith, S. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020.
45. Tuli, S.; Dasgupta, I.; Grant, E.; Griffiths, T.L. Are Convolutional Neural Networks or Transformers More Like Human Vision? *arXiv preprint* **2021**, arXiv:2105.07197.
46. Frank, E.; Hall, M. A Simple Approach to Ordinal Classification. In Proceedings of the European Conference on Machine Learning (ECML), Freiburg, Germany, 5–7 September 2001; pp. 145–156.
47. Niu, Z.; Zhou, M.; Wang, L.; Gao, X.; Hua, G. Ordinal Regression with Multiple Output CNN for Age Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4920–4928.

48. Cao, W.; Mirjalili, V.; Raschka, S. Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation. *Pattern Recognit. Lett.* **2020**, *140*, 325–331.
49. Beckham, C.; Pal, C. Unimodal Probability Distributions for Deep Ordinal Classification. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017.
50. Liu, H.; Lu, J.; Feng, J.; Zhou, J. Ordinal Deep Learning for Facial Age Estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 486–501.
51. Geng, X. Label Distribution Learning. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1734–1748.
52. Geng, X.; Yin, C.; Zhou, Z.-H. Facial Age Estimation by Learning from Label Distributions. In Proceedings of the AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013.
53. Zhang, Z.; Song, Y.; Qi, H. Age Progression/Regression by Conditional Adversarial Autoencoder. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
54. Liu, X.; Zou, Y.; Song, Y.; Yang, C.; You, J.; Vijaya Kumar, B.V.K. Ordinal Regression with Neuron Stick-Breaking for Medical Diagnosis. In Proceedings of the European Conference on Computer Vision Workshops (ECCVW), Munich, Germany, 8–14 September 2018.
55. Wiedemann, M.; Penava, P.; Mai, C.; Buettner, R. Deep-Learning-Based Determination of Textile Properties: A Novel Triplet Architecture Approach for Classifying Cotton Content. *IEEE Access* **2025**, *13*, 164395–164408. <https://doi.org/10.1109/ACCESS.2025.3610920>.
56. Cohen, J. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychol. Bull.* **1968**, *70*, 213–220.
57. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.