Independent University **Bangladesh (IUB)**

IUB Academic Repository

Computer Science and Engineering | Undergraduate Thesis

2026-08

A novel mobile-captured fabric dataset for benchmarking cnn architectures in multiclass classification

Hasan, Maher

Independent University, Bangladesh (IUB)

https://ar.iub.edu.bd/handle/11348/1619

Downloaded from IUB Academic Repository



A Novel Mobile-Captured Fabric Dataset for Benchmarking CNN Architectures in Multiclass Classification

August 2026

Prepared by:

Maher Hasan

ID: 2030670

Department of Computer Science and Engineering

Independent University, Bangladesh

Supervised by:

Dr. Saadia Binte Alam

Associate Professor

Department of Computer Science and Engineering

Independent University, Bangladesh

A Final Year Design Project (FYDP) Report submitted for the

Degree of Bachelor's of Computer Science & Engineering

Attestation

I am aware of the fact that plagiarism is strictly prohibited and it conflicts with our university's rules and regulations. I guarantee the authenticity of my work. I have used some copyrighted material and models of others in my project which has been properly cited following the international standards and proper guidelines.

Author Name:

Maher Hasan

Signature:

. . . . . .

Evaluation Committee

Supervisor

Name: _____ Signature: _____

Internal Examiner 1

Name: _____ Signature: _____

Internal Examiner 2

Name: _____ Signature: _____

External Examiner

Name: _____ Signature: _____

Acknowledgement

I express my sincere gratitude to my supervisor, Dr. Saadia Binte Alam, Associate Professor, Department of Computer Science and Engineering, Independent University, Bangladesh, for her invaluable guidance, continuous encouragement and expert insight throughout the course of this research. Her mentorship and thoughtful feedback shaped both the direction and the rigour of this work.

I would also like to express my sincere appreciation to Fatin Israq for his valuable guidance, constructive suggestions and continued support throughout the project. His insights and assistance were particularly helpful in shaping the development and progress of this work.

I am grateful to the management and the floor staff of M-YEW FASHION LTD., the textile facility that hosted the data collection. Building this dataset required access to an operating production floor during working hours, and that access was granted by Mr. Nafiul Islam Rizon, General Manager, whose cooperation made the entire study possible. The supervisors who identified the fabric rolls, confirmed the class names against production terminology and measured the areal density of each material made the labelling process considerably more reliable than it would otherwise have been.

I would also like to thank Mahfuzur Rahman Mahmood, who was part of this project and contributed to its early direction and to the planning of the data collection.

I extend my thanks to the faculty members of the Department of Computer Science and Engineering, CCDS, for their comments during the proposal and progress presentations, and to my fellow students for their support and suggestions during the data collection and experimentation phases of this project.

During this work, AI tools including Claude and Gemini were used for technical learning and implementation support, LaTeX preparation, and language refinement, while Grammarly was used for grammar and writing correction. All AI-assisted outputs were reviewed and adapted by the author, who remains responsible for the research, implementation, analysis, and conclusions presented in this thesis.

Finally, I am grateful to all individuals who supported and encouraged me, both directly and indirectly, during this endeavour. Their contributions played a meaningful role in the realisation of this work.

List of Publications

1. Maher Hasan, Mahmood Tauhidul, Fatin Israq, Md. Nadimul Islam, Rashedur Rahman and Saadia Binte Alam, “A Novel Mobile-Captured Fabric Dataset for Benchmarking CNN Architectures in Multiclass Classification,” *2026 3rd International Conference on Advanced Computing Technologies (ICACT)*, Hilton, Sept. 1–2, 2026. (Under Review)

Abstract

Reliable identification of fabric type is a routine but consequential decision on a garment production floor, because weave structure, surface finish and areal density determine how a roll is dyed, cut and routed. The decision is still made visually, it depends on operator experience, and it varies with shift and with available light. Automating it from a camera image is attractive, yet the published evidence rests almost entirely on imagery captured under controlled conditions, which does not describe the conditions under which such a system would actually run.

This report addresses that gap in three parts. First, it introduces a multiclass fabric image dataset of 3,354 photographs spanning six production-relevant classes, namely canvas, twill, denim, ready-for-dyeing (RFD) knits, fleece and corduroy. Every image was captured with a single Samsung Galaxy A34 smartphone inside an operating textile facility under a deliberately uncontrolled protocol, with no tripod, light booth or imposed background. Variation in viewpoint, digital zoom, illumination and material handling was introduced on purpose rather than suppressed, and independent areal-density measurements spanning 186 to 371 g/m^2 were recorded for each class. Second, it benchmarks six ImageNet-pretrained convolutional backbones, namely MobileNetV2, EfficientNetV2-B0, DenseNet-121, ResNet-50, ConvNeXt-Tiny and ResNet-101, under one identical two-stage transfer-learning protocol, evaluated with and without online augmentation. Third, it pairs those results with a Grad-CAM interpretability analysis and with deployment-oriented cost measurements.

EfficientNetV2-B0 produced the strongest recognition performance, reaching a test accuracy of 0.995 without augmentation and 0.998 with augmentation, with precision, recall and F1-score all reaching 0.999 in the augmented setting. The result is notable because EfficientNetV2-B0 is the second smallest model in the benchmark and outperforms ResNet-101, which carries more than six times as many parameters, indicating that balanced compound scaling matters more than raw depth for this task. MobileNetV2 recorded the lowest inference latency at 6.2 ms per image while retaining a precision of 0.993. Augmentation had a real but small and non-uniform effect: four backbones improved by two to six accuracy points per thousand while two declined by amounts within run-to-run variation.

Error analysis showed a single misclassification on the held-out partition, a fleece sample predicted as corduroy, and no confusion between denim and twill despite their shared weave family. Grad-CAM maps concentrated inside the fabric surface rather than on the background or on the handwritten measurement tags, addressing the shortcut-learning risk that an uncontrolled protocol introduces. These findings establish that a compound-scaled, moderate-capacity convolutional network can achieve near-perfect fabric recognition on ordinary smartphone imagery while retaining a practical computational footprint.

Contents

1 Introduction **12**

1.1 Background 13

1.2 Motivation 15

1.3 Research Questions 15

1.4 Objective 16

1.5 Key Contributions 17

1.6 Report Organization 18

2 Literature Review **19**

2.1 Prior Studies 19

2.1.1 Handcrafted Texture Description 19

2.1.1.1 Statistical Descriptors 19

2.1.1.2 Frequency and Filter-Bank Methods 20

2.1.1.3 Local Binary Patterns 21

2.1.1.4 Application to Automated Fabric Inspection 21

2.1.2 Learned Representations for Visual Recognition 21

2.1.2.1 Deep Convolutional Networks 22

2.1.2.2 Deep Representations for Texture 22

2.1.2.3 Transfer Learning under Limited Data 23

2.1.3 Efficiency- and Capacity-Oriented Backbone Design 23

2.1.3.1 Efficiency-Oriented Architectures 23

2.1.3.2 Capacity-Oriented Architectures 24

2.1.3.3 Modernised Convolutional Design 25

2.1.4 Textile-Specific Recognition 25

2.1.4.1 Weave and Fabric Type Classification 25
2.1.4.2 Fabric Image Datasets 26
2.1.4.3 Summary of the Textile Literature 26
2.1.5 Interpretability and Shortcut Learning 26
2.1.5.1 Shortcut Learning and Dataset Bias 26
2.1.5.2 Visual Explanation Methods 27
2.2 Challenges Faced 27
2.3 Scope of Study 28

3 Dataset **29**
3.1 Dataset Description 29
3.1.1 Acquisition Environment and Device 29
3.1.2 Class Definitions 30
3.1.3 Areal Density Measurements 31
3.2 Dataset Preparation 32
3.2.1 Overview and Rationale 32
3.2.2 Capture Variation Protocol 32
3.2.3 Labelling and Verification 33
3.2.4 Stratified Partitioning 34

4 Methodology **36**
4.1 Proposed Method 36
4.1.1 Dataset Pre-processing 37
4.1.2 Data Augmentation 37
4.1.3 Backbone Architectures 38
4.1.3.1 MobileNetV2 39
4.1.3.2 EfficientNetV2-B0 40
4.1.3.3 DenseNet-121 40
4.1.3.4 ResNet-50 41
4.1.3.5 ConvNeXt-Tiny 41
4.1.3.6 ResNet-101 42

4.1.4 Shared Classification Head 42
4.2 Training Protocol 44
4.3 Evaluation Framework 45
4.3.1 Precision 46
4.3.2 Recall 46
4.3.3 F1-Score 46
4.3.4 Accuracy 46
4.3.5 Area Under the ROC Curve 47
4.3.6 Computational Metrics 47
4.4 Interpretability Analysis 47

5 Results and Analysis **48**
5.1 Experimental Overview 48
5.2 Classification Performance 48
5.2.1 Performance Without Augmentation 48
5.2.2 Performance With Augmentation 49
5.2.3 Comparative Analysis and Model Selection 49
5.3 Effect of Data Augmentation 50
5.4 Loss Curve Analysis 51
5.5 Confusion Matrix Analysis 52
5.6 Threshold-Independent Behaviour 53
5.7 Computational Cost 55
5.8 Interpretability Results 57

6 Discussion **59**
6.1 Benchmarking Under a Unified Protocol 59
6.2 Capacity, Scaling and Texture Recognition 59
6.3 Augmentation and Model Capacity 60
6.4 Deployment Trade-offs on Real Hardware 61
6.5 Evidence Against Shortcut Learning 62

7 Conclusion **63**

7.1 Research Questions Answered 64

7.2 Analysis of the Social Effects 65

7.3 Environmental Impact Analysis and Sustainability 66

7.4 Ethical Issues 67

7.5 Limitations 68

7.6 Future Work 69

Bibliography **70**

List of Figures

3.1 Representative samples from the six fabric classes. 31

3.2 Measured areal density per class. 32

3.3 Sources of variation introduced during acquisition. 33

3.4 Distribution of images under the stratified 60/20/20 split. 35

4.1 Experimental pipeline from capture to backbone comparison. 36

4.2 MobileNetV2 inverted residual blocks 39

4.3 EfficientNetV2-B0 stage layout 40

4.4 Dense connectivity within a DenseNet block 41

4.5 ResNet-50 stage architecture 41

4.6 ConvNeXt four-stage layout 42

4.7 ResNet-101 layer configuration 42

4.8 Model architecture used across all six experiments. 43

4.9 Two-stage transfer-learning schedule. 45

5.1 Test accuracy and the effect of augmentation. 50

5.2 Training and validation curves for the fine-tuning stage. 51

5.3 Confusion matrix on the held-out test partition. 52

5.4 Per-class AUROC before and after fine-tuning. 54

5.5 ROC curves for all six classes after fine-tuning. 54

5.6 Parameter count, training time and inference latency. 56

5.7 Test accuracy against inference latency. 57

5.8 Grad-CAM analysis for a representative denim sample. 57

List of Tables

3.1 Summary of the mobile-captured fabric dataset. 29

3.2 Capture variations and measured areal density per class. 33

3.3 Stratified dataset split by class. 34

4.1 Online augmentation policy applied to the training stream in the augmented configuration. 38

4.2 Convolutional backbones evaluated in this study. 39

4.3 Training hyperparameters, held identical across all six backbones and both augmentation settings. 44

5.1 Fine-tuned test performance without augmentation. The best value in each column is shown in bold. 49

5.2 Fine-tuned test performance with online augmentation. The best value in each column is shown in bold. 49

5.3 Computational metrics measured on an NVIDIA GeForce RTX 5070 Ti at batch size one after warm-up. The best value in each column is shown in bold. 55

Chapter-1: Introduction

Textiles are among the oldest manufactured goods and remain among the most economically significant. For Bangladesh in particular, the ready-made garment sector is the dominant export industry, and the mills that constitute it operate at a scale where routine decisions taken hundreds of times a day accumulate into material cost. One such decision is the identification of fabric type. A roll arriving on the floor must be recognised before it can be routed, because weave structure, surface finish and areal density jointly determine which dyeing recipe applies, which cutting table the material should reach, how it is logged in inventory, and what tolerances quality control should enforce. A roll sent to the wrong line does not merely delay a batch; it consumes dye, machine time and labour that cannot be recovered.

This identification is still performed visually and by hand. An operator looks at the material, feels it between two fingers and assigns it to a category drawn from the vocabulary the facility itself uses. Experienced staff perform this task well, and no automated system currently deployed in a typical mill matches the tacit knowledge that such an operator accumulates over years. The difficulty is that the skill is expensive to acquire, unevenly distributed across a shift, and sensitive to conditions that vary within a single working day. Illumination differs between one part of a floor and another, fatigue accumulates toward the end of a shift, and materials that share a weave family can present very similar surfaces under ordinary light. A supporting instrument that returns a consistent second opinion at the moment of the decision would not replace the operator, but it would reduce the variance of the decision.

Image-based recognition is the natural candidate for such an instrument, and the computational machinery required for it now exists. Automatic texture and material analysis progressed for three decades on handcrafted descriptors, in which a human designer specified in advance which local statistics of an image should be measured. Gray-level co-occurrence matrices summarised the spatial relationship between pixel intensities [1], Gabor filter banks captured frequency- and orientation-selective structure [2], and local binary patterns encoded local intensity relationships in a manner invariant to rotation and to monotonic gray-scale change [3]. These descriptors were paired with shallow classifiers and performed acceptably under controlled illumination, but they shared a structural weakness: because the measurement was fixed before any data was observed, a change in imaging conditions that the designer had not anticipated changed the descriptor even though the material had not changed.

The demonstration that a deep convolutional network could be trained end to end on a large la-

belled corpus and substantially outperform such pipelines on large-scale visual recognition [4] redirected the field toward learned representations [5]. The relevant consequence for texture work was established shortly afterwards, when convolutional features combined with orderless pooling were shown to outperform handcrafted descriptors on standardised texture benchmarks [6]. Texture had been regarded as the domain in which engineered descriptors would hold their ground longest, precisely because it lacks the object-level structure that convolutional networks were assumed to exploit; the removal of that expectation is what makes a learned approach the reasonable default for fabric recognition today.

A second development made the approach practical at the scale available to a single-site study. Features learned on a large natural-image corpus transfer to domains far removed from the original labels, with early layers encoding generic primitives that remain useful and later layers encoding task-specific combinations that benefit from adaptation [7, 8]. Transfer learning from an ImageNet-pretrained backbone [9] therefore permits competitive recognition from a few thousand domain images rather than the millions that training from scratch would demand. This is decisive for a report whose dataset is collected by one person, with one device, inside one facility, over a bounded period.

What has not kept pace is the realism of the imagery on which such systems are evaluated. Textile-focused studies report encouraging results, but the benchmarks they rely on are captured under studio illumination, on fixed rigs, or under a microscope [10, 11, 12]. These are appropriate choices for the questions those studies ask. They are not representative of a person holding a phone over a cutting table. Accuracy measured under conditions that cannot be reproduced at deployment is optimistic in a way that only becomes visible after installation, and closing that gap is the concern of this report.

1.1 Background

Fabrics are flexible materials formed by weaving, knitting, or related fibre-bonding processes, and the six categories examined in this study span several of those construction families. Canvas is a heavy plain weave in which warp and weft cross at right angles in a tight, regular grid. Twill is defined by its interlacing rather than by its colour or weight: the filling yarn passes over one or more warp yarns and under two or more, and shifting that pattern by one yarn on each successive row produces the diagonal ridge that gives the class its name. Denim is itself a twill construction, separated from the twill class chiefly by finish, since its warp is dyed while its weft is left undyed, so that the face reads blue and the reverse reads considerably paler. Ready-for-dyeing (RFD) knit is defined by a processing state rather than by a weave: the material has been prepared and scoured but not yet dyed, and structurally it is a knit, in which yarn forms interlocking loops instead of crossing at right angles. Fleece carries a raised pile of fibres that obscures the underlying knit entirely. Corduroy is likewise a pile fabric, but its pile is cut into

parallel ridges, called wales, that run along the length of the material.

Two consequences follow from this taxonomy, and both shape the difficulty of the recognition problem. The first is that the classes are not equidistant. Denim and twill share a construction, so a classifier that attended only to weave geometry would be expected to confuse them, and would have to recover the distinction from colour and finish instead. Fleece and corduroy are both pile fabrics, differing in whether that pile is cut into regular wales, so a sample photographed at low magnification, at an angle that flattens the wales, or under diffuse light that suppresses the shadow between them, could present a soft undifferentiated surface in either case. The second consequence is that the discriminative evidence is carried by recurring local structure, yarn thickness, surface roughness and weak directional cues rather than by object boundaries. A photograph of denim and a photograph of RFD knit may occupy the same region of the frame, cover the same area and share the same overall silhouette; what separates them lies inside that silhouette.

Alongside the visual taxonomy, areal density provides an independent physical descriptor of each material. Expressed in grams per square metre and abbreviated GSM, it is measured with a scale rather than inferred from an image, and it is recorded in this study for every class. Its value is that it is not derived from the imagery at all. A class taxonomy assembled purely from appearance is vulnerable to the objection that its categories are arbitrary partitions of a continuous space, and agreement between the visual classes and independently measured weights is external evidence against that objection.

Convolutional architectures have meanwhile diverged along two axes that matter for a factory deployment. One line of work optimises accuracy per unit of computation so that inference can run on constrained hardware, through depthwise separable convolution and inverted residual blocks [13] and through principled joint scaling of depth, width and input resolution [14]. A second line pursues representational capacity through dense feature reuse [15], residual connectivity at increasing depth [16], and, more recently, a modernised convolutional design space informed by transformer architectures [17]. Which of these positions best suits fabric texture on mobile imagery is an empirical question, and it is one of the questions this report answers.

A final consideration is that high benchmark accuracy is not by itself sufficient evidence that a model has learned the intended concept. Deep networks are known to exploit incidental correlations that happen to be predictive on a particular dataset while bearing no relation to the target concept, a failure mode termed shortcut learning [18]. The risk is elevated precisely where acquisition is least controlled, because backgrounds, lighting artefacts and device signatures are then free to correlate with class labels. Gradient-weighted class activation mapping provides a practical, if coarse, means of checking whether the spatial evidence supporting a prediction lies inside the target object or outside it [19], and it is used in this report for exactly that purpose.

1.2 Motivation

Three specific gaps in the existing literature motivate this work, and each maps onto one of its contributions.

The first concerns dataset realism. Publicly available fabric image collections seldom combine several production-relevant classes with deliberate variation in viewpoint, working distance and illumination. Microscopic corpora such as FabricNET are scientifically valuable and support the prediction of quantitative weaving parameters, but they fix magnification and illumination by construction [12]. Studio collections of woven patterns similarly control the capture geometry [10]. Both choices are correct for their respective research questions and inappropriate as evidence for handheld inspection. No public corpus was found that photographs production-relevant fabric classes under the conditions a deployed system would encounter, and constructing one is the first contribution of this report.

The second concerns comparability. Published architectural comparisons are frequently assembled from results reported in separate papers, under different preprocessing, different optimisation schedules, different data partitions and different amounts of tuning effort. Under those conditions it is not possible to attribute an observed difference to architecture rather than to the care with which each model was trained. A controlled comparison, in which preprocessing, optimiser, loss, schedule and prediction head are all held fixed and only the backbone is exchanged, is required before any claim about relative architectural suitability can be supported. Constructing that protocol and running it across six backbones and two augmentation settings is the second contribution.

The third concerns evidence. Classification accuracy is reported almost universally; the spatial basis of the predictions and the computational cost of obtaining them are reported far less often. Both omissions matter here. The shortcut-learning risk described above is a direct consequence of the uncontrolled acquisition protocol that makes the dataset realistic, so it must be tested rather than assumed away. Deployment cost matters because a mill choosing between a centralised inspection station and a handheld tool needs latency and memory figures alongside accuracy in order to choose at all. Reporting Grad-CAM evidence and efficiency measurements together with the classification metrics is the third contribution.

1.3 Research Questions

The gaps described above translate into four research questions, which together define what this study set out to answer. They are stated here in the order in which the report addresses them, and each is answered explicitly in the conclusion.

RQ1. Can fabric type be recognised reliably from smartphone photographs captured under or-

dinary factory conditions, where viewpoint, working distance, illumination and material handling are uncontrolled?

RQ2. When preprocessing, optimiser, schedule and classification head are held fixed and only the backbone is exchanged, which convolutional architecture is best suited to fabric texture, and does additional parameter capacity translate into additional accuracy?

RQ3. What effect does online data augmentation have on this task, and is that effect uniform across architectures of different capacity?

RQ4. Do the predictions rest on genuine material evidence inside the fabric surface, or on incidental cues such as the background or the handwritten measurement tags that the uncontrolled protocol introduces?

A further consideration cuts across all four: because the intended setting is a production floor rather than a benchmark suite, every answer is reported together with its computational cost.

1.4 Objective

The overall objective of this study is to determine whether fabric type can be recognised reliably from smartphone photographs captured under ordinary factory conditions, and to establish which convolutional architecture is best suited to that task once accuracy and deployment cost are considered together. Rather than optimising a single model for peak accuracy in isolation, the study builds the infrastructure required to compare architectures fairly, namely a dataset, a fixed protocol and an evaluation framework, and then reports what that infrastructure reveals. This general aim decomposes into four specific objectives, each of which is the instrument through which one of the research questions above is answered.

The first objective, which supplies the evidence for **RQ1**, is to construct and document a labelled, stratified multiclass fabric image dataset collected under realistic, uncontrolled factory capture conditions. This entails designing an acquisition protocol that deliberately introduces viewpoint, zoom, illumination and handling variation rather than eliminating it; establishing a labelling workflow grounded in production terminology and anchored to the physical roll rather than to after-the-fact interpretation of a photograph; recording an independent physical descriptor for each class; and partitioning the collection into stratified training, validation and test subsets suitable for supervised model development.

The second objective addresses **RQ2** and **RQ3** together, and is to benchmark six ImageNet-pretrained convolutional backbones under one consistent two-stage transfer-learning protocol, with and without online augmentation. This requires holding preprocessing, optimiser settings, loss function, fine-tuning schedule and prediction-head design fixed across all six backbones, so that any observed performance difference is attributable to backbone design rather than to

inconsistencies in the training pipeline. Running each backbone in both augmentation settings additionally isolates the effect of augmentation itself and permits its interaction with model capacity to be examined.

The third objective supplies the cost dimension that qualifies the answer to **RQ2**, and is to quantify both classification quality and deployment cost under a single measurement regime. Classification quality is assessed through precision, recall, F1-score, accuracy and the area under the receiver operating characteristic curve on the held-out test partition, supported by a confusion matrix and by per-class analysis. Deployment cost is assessed through parameter count, total training time and per-image inference latency measured on fixed consumer-grade hardware. Reporting both dimensions together supports a recommendation that is conditional on deployment context rather than a context-free claim about a single best model.

The fourth objective answers **RQ4**, and is to assess whether the predictions of the strongest-performing model rest on genuine material evidence. Grad-CAM is applied to representative test samples to determine whether the spatial support for a prediction lies within the fabric surface or on the surrounding background, and in particular whether the handwritten measurement tags introduced by the areal-density workflow are being exploited as a shortcut.

1.5 Key Contributions

This work makes five contributions, listed in the order in which they appear in the chapters that follow.

A public fabric image dataset captured under production conditions. A corpus of 3,354 full-resolution photographs covering six production-relevant fabric classes was collected inside an operating textile facility with a single handheld smartphone, with no tripod, light booth, colour chart or uniform background at any point. It is released publicly, and to the best knowledge of this study it is the first openly available fabric corpus acquired under a deliberately uncontrolled handheld protocol rather than in a laboratory.

A documented acquisition protocol that treats variation as a design variable. Rather than allowing nuisance variation to enter accidentally, the protocol introduces it along five declared axes: front and back faces, four digital zoom levels, flat and rolled presentations, flash and ambient illumination, and whatever light the floor happened to supply. The number of distinct variations is recorded per class, so a reader can see exactly what kind of variation the reported accuracy was measured over.

Independent physical measurements attached to the visual taxonomy. Areal density was measured for every class with a weighing scale by the facility supervisor, giving values from 186 to 371 g/m^2. These numbers are not derived from the imagery, so agreement between the visual classes and the measured weights is external evidence that the six categories separate

materially different products rather than arbitrary regions of a continuous appearance space.

A controlled six-backbone benchmark under one fixed protocol. Six ImageNet-pretrained convolutional backbones spanning a twelvefold range in parameter count were trained under one identical two-stage transfer-learning procedure, with preprocessing, optimiser, loss, schedule and classification head held constant, and each was run twice, once with and once without online augmentation. Because only the backbone changes between runs, the resulting differences can be attributed to architecture rather than to unequal tuning effort, which is not true of comparisons assembled from separately published numbers.

An evaluation that reports cost and evidence alongside accuracy. Parameter count, training time and per-image inference latency were measured for every backbone on fixed hardware, and Grad-CAM was applied to check whether the spatial support for a prediction falls on the fabric or on the background and the handwritten tags. Reporting these together with the classification metrics turns the outcome into a deployment recommendation, and tests rather than assumes the shortcut-learning risk that an uncontrolled protocol creates.

1.6 Report Organization

The remainder of this document is organised into six chapters.

Chapter 2 reviews the literature on texture and material recognition, tracing the field from hand-crafted descriptors through learned convolutional representations and transfer learning, then narrowing to textile-specific recognition and to the interpretability work that motivates the shortcut-learning check used here.

Chapter 3 presents the dataset: the acquisition environment and device, the six class definitions in the vocabulary the facility itself uses, the areal-density measurements, the five-axis capture variation protocol, the labelling workflow, and the stratified partition used throughout.

Chapter 4 describes the methodology: preprocessing, the augmentation policy and the reasoning behind its conservatism, the six backbone architectures and the shared classification head, the two-stage training protocol and its hyperparameters, the evaluation metrics, and the Grad-CAM procedure.

Chapter 5 reports the results, covering classification metrics for both augmentation settings, the effect of augmentation, the loss curves, the confusion matrix, threshold-independent behaviour, computational cost and the accuracy-latency frontier, and the interpretability analysis.

Chapter 6 discusses what those results mean and states plainly which claims the evidence supports and which it does not.

Chapter 7 concludes. It answers the four research questions, examines the social, environmental and ethical dimensions of the work, and sets out the limitations and the further work required.

Chapter-2: Literature Review

2.1 Prior Studies

Image-based fabric analysis draws on two literatures that developed largely independently. The first is general texture and material recognition, which supplies the representations on which any fabric classifier ultimately rests. The second is textile-specific inspection, which supplies the task definitions, the acquisition practices and the benchmarks against which such a classifier is judged. This section surveys both, organised by methodological paradigm rather than by publication date, because the decisive transitions in this field were changes in how the representation was obtained rather than incremental gains within a fixed approach. Four paradigms are treated in turn: handcrafted descriptors paired with shallow classifiers, learned convolutional representations, efficiency- and capacity-oriented backbone design, and textile-specific applications. A fifth subsection reviews the interpretability literature that motivates the verification component of this report.

2.1.1 Handcrafted Texture Description

Before learned representations became practical, texture recognition proceeded by specifying in advance which statistics of an image should be measured, computing those statistics over a region, and passing the resulting feature vector to a classifier. What distinguished one method from another was the choice of statistic. The paradigm produced descriptors that remain in use, and its characteristic limitation, that the measurement is fixed before any data is observed, is what the subsequent paradigm removed.

2.1.1.1 Statistical Descriptors

The earliest systematic treatment of texture as a measurable image property introduced the gray-level co-occurrence matrix, which tabulates how often a pixel of one intensity occurs at a fixed spatial offset from a pixel of another intensity [1]. From this matrix a set of fourteen scalar statistics was derived, including angular second moment, contrast, correlation, entropy and inverse difference moment, each summarising a different aspect of the joint intensity distribution. The descriptor was evaluated on photomicrographs, aerial photography and satellite imagery,

and demonstrated that categories which a human observer distinguishes by surface appearance could be separated by numerical features computed automatically. Its influence is difficult to overstate, and it remains a standard baseline. Its limitations are equally clear in retrospect. The offset and quantisation must be chosen by the designer, the statistics discard phase information entirely, and the representation is sensitive to illumination changes that alter the intensity histogram without altering the material.

Where the co-occurrence formulation summarises pairwise intensity relations, gradient-based descriptors summarise local orientation structure. Scale-invariant keypoint description, which detects extrema in a difference-of-Gaussian scale space and encodes the surrounding gradient orientations into a histogram-based descriptor, established that local image structure could be represented in a form stable under scale and rotation change [20]. Densely computed histograms of oriented gradients, developed for pedestrian detection, showed that the same principle applied on a regular grid with local contrast normalisation produced a representation robust to illumination variation [21]. Neither descriptor was designed for texture specifically, but both entered texture pipelines as region descriptors, and both illustrate the paradigm's central assumption that a well-chosen fixed encoding of local gradients suffices.

2.1.1.2 Frequency and Filter-Bank Methods

A parallel line of work treated texture as a signal and described it in the frequency domain. Gabor filters, which are sinusoids modulated by a Gaussian envelope and are selective in both frequency and orientation, were applied as a multi-scale bank whose response energies form the feature vector [2]. The method was evaluated on browsing and retrieval over the Brodatz texture album, where it substantially outperformed alternative texture features of the period, and it established Gabor energy as a standard texture representation. The formulation is well matched to woven materials in particular, because a weave is close to periodic by construction and the ridge direction of a twill produces a clear orientation signature in the filter responses. Its weaknesses are the computational cost of evaluating a full bank at every location, sensitivity to the choice of scales and orientations, and degradation when the periodicity assumption is violated by folds, drape or non-woven construction.

Filter-bank approaches were subsequently generalised through the texton framework, in which filter responses across a training corpus are clustered into a vocabulary and an image is represented by the histogram of its texton assignments [22]. This introduced two ideas that carried directly into the deep-learning era: that a representation should be learned from data rather than fixed analytically, and that an orderless summary over an image region is an appropriate encoding for texture, because texture is defined by the statistics of local structure rather than by its spatial arrangement.

2.1.1.3 Local Binary Patterns

Local binary patterns take a different route to illumination robustness. Each pixel is compared with its neighbours on a circle of given radius, the comparisons are recorded as a binary string, and the histogram of the resulting codes over a region forms the descriptor. Because every comparison is relative rather than absolute, the representation is invariant to any monotonic transformation of the gray scale. The multiresolution formulation extended this to rotation invariance by identifying uniform patterns, those binary strings containing at most two circular transitions, and mapping all rotations of a pattern to a single canonical bin [3]. Evaluated on rotation- and illumination-controlled texture databases, the method achieved high classification rates at very low computational cost, and its combination of gray-scale and rotation invariance made it one of the most widely adopted texture descriptors of the following decade.

For fabric imagery the appeal is direct, since a sample rotated on a cutting table should retain its class. The limitations, however, bear on precisely the conditions this report studies. The descriptor is defined on a fixed neighbourhood radius, so it encodes structure at a fixed scale and does not by itself accommodate the working-distance variation introduced by digital zoom. It is also sensitive to noise in near-uniform regions, where small intensity fluctuations flip comparison bits, and undyed materials photographed under low contrast present exactly such regions.

2.1.1.4 Application to Automated Fabric Inspection

The industrial application of these descriptors was concentrated on defect detection rather than on type classification. A comprehensive review of automated fabric defect detection catalogued the methods in use across statistical, spectral, model-based, learning-based and structural families [23]. Statistical approaches applied co-occurrence and histogram statistics to local windows and flagged deviations; spectral approaches applied Fourier, wavelet and Gabor analysis and detected disruptions of the periodic weave signature; model-based approaches fitted random-field or autoregressive models to defect-free material and reported likelihood departures. The review's central conclusion is the one most relevant here: the systems surveyed achieved usable performance under controlled and uniform illumination on a moving web, and their performance degraded outside those conditions, because every family depends on a stable relationship between material appearance and measured statistic. This is the recurring constraint of the entire handcrafted paradigm, and it is the constraint that a learned representation relaxes.

2.1.2 Learned Representations for Visual Recognition

The transition away from handcrafted description was driven by evidence from large-scale object recognition rather than from texture work, but it propagated to texture quickly and decisively.

2.1.2.1 Deep Convolutional Networks

The construction of a large, densely labelled image corpus organised on a semantic hierarchy provided the training data and the evaluation protocol without which the transition could not have occurred [9], and the associated annual challenge supplied a common yardstick against which successive architectures were measured [24]. A deep convolutional network trained on this corpus with rectified linear units, dropout regularisation and GPU acceleration reduced top-5 error on the 2012 challenge to 15.3 per cent against 26.2 per cent for the best competing entry, a margin large enough to redirect the field [4].

Subsequent architectures explored the consequences of depth. A systematic study of depth using only small 3×3 convolutions demonstrated that stacking many small kernels achieves a large effective receptive field with fewer parameters than a shallow network with large kernels, and that recognition improves steadily as depth increases to sixteen and nineteen weight layers [25]. An architecture built on parallel multi-scale branches within a repeated module improved accuracy at substantially lower computational cost and won the 2014 challenge [26]. Beyond roughly twenty layers, however, plain networks exhibited a degradation problem in which additional depth increased training error. Residual learning resolved this by reformulating each block to compute a correction to its own input rather than a complete mapping, which permitted networks of 152 layers to be trained successfully and reduced top-5 error to 3.57 per cent on the 2015 challenge [16]. Normalising layer inputs across the mini-batch to reduce internal covariate shift, thereby permitting higher learning rates and faster convergence, was a further enabling contribution of the same period [27]. A retrospective account of the field consolidated these developments and articulated why hierarchical learned features outperform engineered ones [5].

2.1.2.2 Deep Representations for Texture

Texture was expected to be the domain in which engineered descriptors would persist, on the reasoning that texture lacks the object-level structure that convolutional networks exploit. The construction of a texture database annotated with human-describable attributes rather than material identity provided a benchmark on which this expectation could be tested [28]. The decisive result combined convolutional feature extraction with orderless Fisher vector pooling, so that the spatial arrangement of activations is discarded and only their distribution is retained [6]. This construction outperformed handcrafted descriptors on standardised texture benchmarks and, in doing so, removed the principal argument for retaining engineered texture features. The orderless pooling step is itself a direct descendant of the texton histogram, which illustrates continuity between the paradigms rather than a clean break; what changed was that the filters were learned rather than specified.

2.1.2.3 Transfer Learning under Limited Data

Training a deep network from random initialisation requires a corpus far larger than any single-site study can assemble, and transfer learning is what makes the approach tractable at the scale of a few thousand domain images. A systematic layer-by-layer analysis quantified how the transferability of features degrades with depth, establishing that early layers encode generic primitives that transfer across widely separated domains, that later layers become progressively specific to the source labels, and that fine-tuning transferred features generally outperforms both training from scratch and freezing the entire network [7]. A broader survey formalised the setting and its variants [8]. The two-stage schedule adopted in this report, in which a randomly initialised head is trained against a frozen backbone before the deepest backbone layers are released for adaptation, follows directly from this result: the early layers are left alone because they transfer as they are, and the deepest layers are adapted because they encode source-specific combinations.

Data augmentation is the complementary response to limited training data, expanding the effective size of the training set through label-preserving transformation and thereby encouraging invariance to the transformations applied. A survey of the field groups the available operations into geometric transforms, photometric transforms, kernel filters, mixing strategies and generative methods, and emphasises that the choice of policy must respect the semantics of the target labels [29]. That constraint is consequential for fabric: heavy rotation weakens the directional cue separating twill and corduroy from other classes, aggressive colour jitter risks moving an undyed RFD sample or an indigo-warped denim sample toward the appearance of a different class, and strong random cropping at high magnification can remove the weave pattern altogether. The deliberately conservative augmentation policy adopted in this report follows from these considerations.

2.1.3 Efficiency- and Capacity-Oriented Backbone Design

Once learned representations became standard, architectural research diverged along two axes that bear directly on a factory deployment: reducing the computation required for a given accuracy, and increasing capacity where computation is not the binding constraint.

2.1.3.1 Efficiency-Oriented Architectures

Depthwise separable convolution factorises a standard convolution into a depthwise stage that filters each input channel independently and a pointwise 1×1 stage that mixes the filtered channels, reducing multiply-accumulate operations by roughly an order of magnitude at equal width. An architecture built on this factorisation, with width and resolution multipliers exposed as explicit trade-off controls, established the family of models designed for constrained hardware [30].

Its successor introduced the inverted residual block with a linear bottleneck: channels are expanded by a pointwise convolution, spatial filtering is performed in the expanded space, and the representation is projected back down through a layer with no non-linearity [13]. Keeping that final projection linear is the substantive detail, because applying a rectifier at a narrow bottleneck destroys information that cannot subsequently be recovered. The resulting model attains competitive ImageNet accuracy with approximately 3.4 million parameters, and it is included in this report as the lightweight reference point.

A second efficiency-oriented line addressed how a network should be scaled rather than how its blocks should be constructed. Compound scaling establishes that depth, width and input resolution should be increased together according to a fixed ratio rather than independently, on the reasoning that a deeper network without corresponding width has layers too narrow to carry the information the depth could exploit, while higher input resolution without corresponding depth leaves the receptive field too small to cover the enlarged image [31]. The second generation of this family added fused inverted-bottleneck blocks in the early stages, replacing depthwise convolution with regular convolution where depthwise operations underuse modern accelerators, and combined this with a progressive learning schedule that increases image size and regularisation together during training [14]. The reported outcome is faster training and improved parameter efficiency relative to the previous generation. EfficientNetV2-B0, at approximately 7.1 million parameters, occupies the moderate-capacity position in the benchmark reported here.

2.1.3.2 Capacity-Oriented Architectures

Residual connectivity, introduced to resolve the degradation problem, also provides a straightforward axis along which capacity can be increased, and the 50-layer and 101-layer instantiations of the same design differ essentially in the number of blocks in their third stage [16]. Including both in a benchmark isolates the effect of depth alone, since the two share block structure, training treatment and every other design decision.

Dense connectivity takes an alternative approach to feature propagation. Within a dense block, each layer receives the concatenated feature maps of every preceding layer in that block and passes its own output to every subsequent one, so that a layer has direct access to the raw early features as well as to the abstractions built from them [15]. Two properties follow. Parameter efficiency improves, because features are reused rather than recomputed, and the 121-layer instantiation requires approximately 8.0 million parameters. More importantly for texture, low-level detail remains available at every depth rather than being progressively replaced. A conventional feed-forward stack gradually substitutes abstraction for detail, which is appropriate for object recognition, where the identity of a chair does not depend on the grain of its wood; for texture classification the low-level detail is the signal.

2.1.3.3 Modernised Convolutional Design

The introduction of transformer architectures to vision, which partition an image into patches and apply self-attention over the resulting sequence [32], and their subsequent adaptation to hierarchical feature maps through shifted local attention windows [33], prompted a re-examination of whether the reported advantage derived from attention itself or from the accompanying macro-design choices. That question was answered empirically by taking a standard residual network and applying, one modification at a time, the design decisions that distinguish a transformer from it: a patch-like stem, fewer and larger stages, depthwise convolution with a 7×7 kernel, an inverted bottleneck, layer normalisation in place of batch normalisation, and GELU in place of ReLU [17]. The resulting purely convolutional architecture matched or exceeded comparable transformers, indicating that much of the reported gain was attributable to design choices transferable to convolutional networks. The large depthwise kernel is the element most relevant to texture, since it allows a single layer to span the repeat period of a coarse weave rather than accumulating that span across several layers. ConvNeXt-Tiny is included in this benchmark as a modern convolutional reference point at approximately 28.6 million parameters.

2.1.4 Textile-Specific Recognition

The general-purpose literature supplies representations; the textile literature supplies the task and the benchmarks. Work in this area falls into three groups.

2.1.4.1 Weave and Fabric Type Classification

Deep convolutional classification has been applied directly to woven-pattern recognition, training a network to discriminate among structurally similar weave types from images captured under studio conditions [10]. The study established that convolutional features can separate interlacing patterns that are difficult to characterise with a fixed descriptor, and reported high classification accuracy on its own corpus. Two properties limit its transferability to the present setting. The capture geometry is fixed by a studio rig, so the reported accuracy is conditional on conditions that a handheld deployment cannot reproduce, and the class definition is a weave taxonomy rather than the processing and product taxonomy that a mill actually uses when routing material.

A subsequent contribution reformulated fabric analysis as joint detection and classification using a single-stage detector, so that the material need not be assumed to fill the frame and multiple samples may be located and categorised within one image [11]. The practical appeal of this formulation is considerable for a production floor, where several materials are frequently visible at once, and it is noted later as a natural extension of the whole-image classification adopted here. Its performance is reported in detection terms, principally mean average precision, which is not

directly comparable to the classification accuracy reported in this report, and its imagery remains controlled.

2.1.4.2 Fabric Image Datasets

The most directly comparable dataset contribution is a microscopic image corpus of woven fabrics assembled specifically to support the prediction of texture and weaving parameters through machine learning [12]. Imaging is performed under a microscope at fixed magnification with controlled illumination, and the accompanying annotations record quantitative weaving parameters rather than product categories alone. For its intended purpose the design is correct: predicting yarn count or weave density requires measurement precision, and measurement precision requires controlled optics.

This corpus is the clearest available point of contrast for the dataset introduced in this report, because the two occupy opposite ends of the acquisition spectrum. One maximises repeatability and accepts that its imagery does not resemble anything a deployed inspection tool would receive; the other accepts substantially reduced control in exchange for imagery that does. Neither choice is superior in general, and the distinction is not a criticism of the earlier work. It is, however, the reason that results obtained on one cannot be assumed to hold on the other, and it is the gap this report addresses.

2.1.4.3 Summary of the Textile Literature

Across these studies a consistent pattern emerges. Textile recognition has followed the general trajectory of computer vision, from handcrafted statistics through learned convolutional representations, and reports strong results at each stage. What has not changed across that trajectory is the acquisition assumption. Whether the imagery is a moving web under uniform illumination, a studio rig, or a microscope stage, the camera is controlled. Consequently the field has extensive evidence about what is achievable when capture conditions are fixed and very little about what survives when they are not.

2.1.5 Interpretability and Shortcut Learning

A separate literature bears on whether reported accuracy should be believed, and it is directly relevant to a dataset collected without background control.

2.1.5.1 Shortcut Learning and Dataset Bias

Deep networks optimise the objective they are given, which is predictive performance on the training distribution, and not the concept the designer intended. Where an incidental feature is

predictive on that distribution, the network is free to use it. This failure mode has been characterised systematically, together with the observation that it is frequently invisible to standard evaluation because the incidental feature is present in the test split as well, and becomes apparent only under distribution shift [18].

The risk applies with particular force to the dataset introduced in this report. The decision not to standardise background and illumination is what makes the corpus realistic, and it is the same decision that leaves room for a shortcut. If a class were photographed predominantly in one part of a facility, the surface beneath the material would become predictive of the label without carrying any information about the material. A related concern arises from the areal-density workflow used here, in which handwritten tags recording roll identifier and measured weight are attached to samples before photography. Such a tag is a high-contrast, class-correlated object and is precisely the kind of feature a network would exploit if it could.

2.1.5.2 Visual Explanation Methods

Class activation mapping established that a convolutional network trained for classification retains localisation information that can be recovered by weighting the final feature maps by the classifier weights for a target class [34], although the formulation requires a specific architecture ending in global average pooling followed by a single linear layer. Gradient-weighted class activation mapping generalised this to arbitrary architectures by using the gradient of the target class score with respect to the feature maps of a chosen convolutional layer as the weighting, pooling those gradients spatially to obtain one importance weight per feature map, forming the weighted sum and applying a rectifier so that only regions with positive influence on the class score are retained [19].

Two properties of the method govern how its output should be read. The map is produced at the spatial resolution of the chosen feature layer, typically 7×7 for a 224×224 input, so it is coarse by construction and cannot localise an individual yarn. And it indicates where the evidence was, not what about that region was informative. For the question posed in this report, namely whether the supporting evidence lies on the fabric or off it, both properties are acceptable, and the interpretability analysis reported here is framed accordingly as a test for a specific failure mode rather than as a complete explanation of the model.

2.2 Challenges Faced

Four challenges shaped the design of this study, and each is a direct consequence of the gaps identified above.

The first is the absence of a suitable public dataset. No available corpus combines production-relevant fabric classes with uncontrolled handheld acquisition, so the dataset had to be built

before any benchmarking could begin. This constrained the study to a single facility and a single device, because that is the access a single investigator could obtain, and it fixes the principal limitation of the work at the outset rather than as an afterthought.

The second is the difficulty of attributing performance differences to architecture. Because published results are not comparable across papers, every backbone had to be trained locally under conditions held identical in every respect other than the backbone itself. This is more expensive than citing published figures and it forgoes the possibility of showing any individual model at its individually tuned best, but it is the only arrangement under which a comparative claim is defensible.

The third is the risk that uncontrolled acquisition introduces exploitable correlations. Realism and shortcut risk arise from the same design decision and cannot be separated by adjusting the protocol; the risk can only be tested after the fact, which is why an interpretability component is treated here as a required element of the evaluation rather than as an illustration.

The fourth is a measurement problem that emerged from the results themselves. Recognition performance proved high enough across all six backbones that the differences between them are of the order of a few images on the test partition. At that level, single-run point estimates cannot reliably order the middle of the field, and conclusions have to be restricted to those comparisons that the evidence actually supports.

2.3 Scope of Study

This study is scoped to the construction of one mobile-captured fabric image dataset and to a controlled benchmarking exercise conducted upon it. The dataset covers six factory-relevant fabric classes photographed with a single smartphone inside one operating textile facility under a deliberately uncontrolled protocol. The benchmark covers six ImageNet-pretrained convolutional backbones spanning lightweight to high-capacity designs, trained under one shared two-stage transfer-learning procedure, evaluated with and without online augmentation, and compared on classification quality, computational cost and Grad-CAM-based interpretability.

Several matters lie deliberately outside this scope. The six classes were selected because they are common in production and because physical rolls of each could be obtained, so they do not exhaust the space of industrial textiles. It does not evaluate transformer-based architectures, although ConvNeXt-Tiny is included partly because its design derives from that literature. It does not test multiple camera models or multiple facilities, so cross-device and cross-site generalisation remain open questions rather than claims. It does not address defect detection, localisation or segmentation, since every image carries a single whole-frame label. And it does not provide a rejection mechanism for materials outside the six classes, which a deployed system would require.

Chapter-3: Dataset

3.1 Dataset Description

The dataset introduced in this report consists of 3,354 full-resolution JPEG photographs, approximately 2.48 GB in total, covering six fabric classes drawn from the vocabulary used on the production floor of an operating textile facility. All images are stored in sRGB colour at 4000×3000 pixels, corresponding to 12 megapixels, with eight bits per sample. Table 3.1 summarises the corpus at a glance. The dataset is publicly available at https://www.kaggle.com/datasets/fabricclassbd/fabric-classification.

Table 3.1: Summary of the mobile-captured fabric dataset.

Property	**Value**
Total images	3,354
Classes	6 (canvas, twill, denim, RFD, fleece, cord)
Capture device	Samsung Galaxy A34 smartphone
Native resolution	4000×3000 pixels (12 MP)
Colour space	sRGB, 8 bits per sample
File format	JPEG
Total size	approximately 2.48 GB
Distinct capture variations	118 across all classes
Acquisition site	One operating textile facility
Availability	Public (Kaggle)

3.1.1 Acquisition Environment and Device

Every photograph was captured with the same handheld smartphone inside an operating facility during working hours. No tripod, light booth, colour chart or uniform background was introduced at any point. Illumination was whatever the overhead lighting of that part of the floor supplied at that moment, backgrounds were the cutting tables and adjacent materials that happened to be present, and working distance was determined by the posture of the person holding the device.

This is an unusual choice and it warrants explanation, because it runs against the standard advice for constructing a clean dataset. A conventional design would fix the working distance, mount

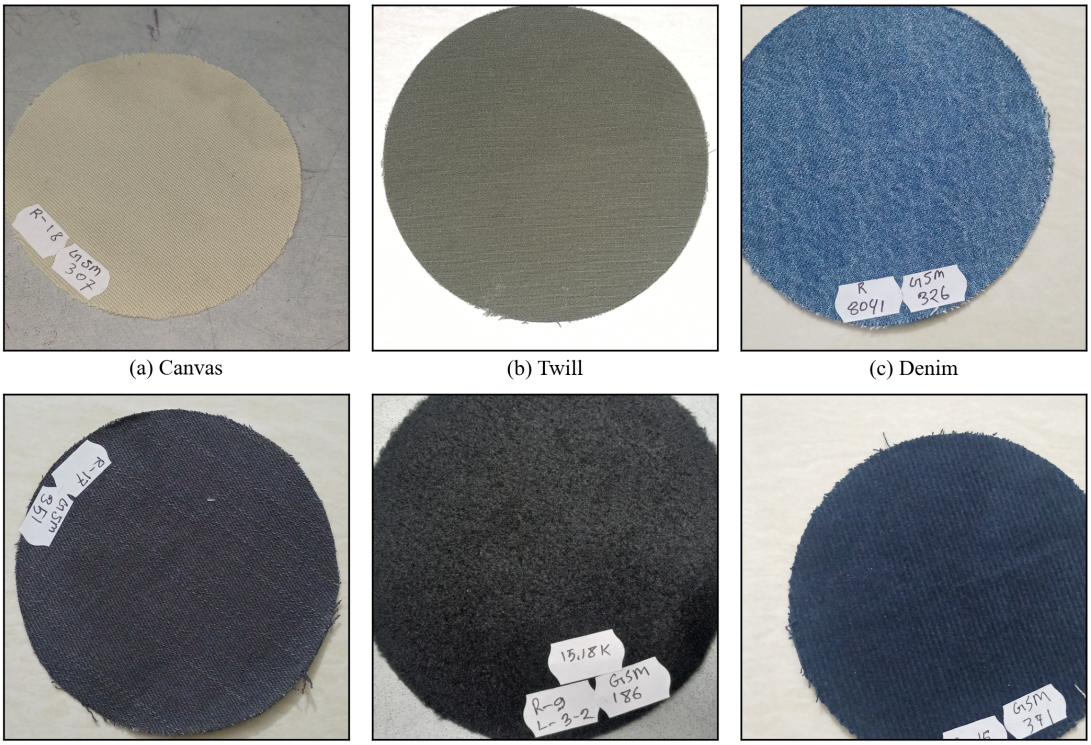
the camera, control the illumination and place every sample on a neutral background, which reduces nuisance variation and makes the learning problem easier. It also produces a model whose reported accuracy is conditional on those conditions being reproduced at deployment. Since the intended deployment scenario is an operator holding a phone over a table, laboratory conditions were never going to be reproduced, and a benchmark built on them would have been optimistic in a way that only becomes apparent after installation. The cost of the decision taken here is that the corpus is noisier and the learning problem is harder; the benefit is that the accuracy figures reported later describe performance on the kind of imagery the system would actually receive.

3.1.2 Class Definitions

The six class names are those used by the staff of the facility when discussing material, and they mix construction categories with processing states because the production vocabulary does so. A taxonomy built on academic weave categories would be more precise in a textile-engineering sense, but it would not correspond to the routing decisions the system is intended to support.

Canvas is a heavy plain weave with a tight, regular grid in which warp and weft cross at right angles with little distortion. Twill is defined by its interlacing: the filling yarn passes over one or more warp yarns and under two or more, and shifting the pattern by one yarn on each successive row produces a diagonal ridge. Denim is itself a twill construction, distinguished chiefly by finish, since its warp is dyed while its weft is left undyed, so the face reads blue and the reverse considerably paler. Ready-for-dyeing knit, abbreviated RFD, is a processing state rather than a weave: the material has been prepared and scoured but not yet dyed, and structurally it is a knit in which yarn forms interlocking loops. Fleece carries a raised pile of fibres that obscures the underlying knit entirely, presenting an irregular surface with no visible periodic structure. Corduroy is also a pile fabric, but its pile is cut into parallel ridges, called wales, running along the length of the material.

Two pairs are therefore expected to be difficult. Denim and twill share a construction and must be separated on colour and finish. Fleece and corduroy are both pile fabrics, differing only in whether the pile is cut into regular wales, so a corduroy sample photographed at low magnification, at an angle that flattens the wales, or under diffuse light that suppresses the shadow between them could present a soft undifferentiated surface resembling fleece. Figure 3.1 shows one representative photograph from each class, ordered (a) canvas, (b) twill, (c) denim, (d) RFD knit, (e) fleece and (f) corduroy. The handwritten tags visible on several of them record the roll identifier and the measured areal density and were attached as part of the measurement workflow described below; their presence in the frame is treated later as a possible confound.



(a) Canvas (b) Twill (c) Denim

(d) RFD (e) Fleece (f) Cord

Figure 3.1: Representative samples from the six fabric classes.

3.1.3 Areal Density Measurements

In addition to the visual class label, areal density was recorded for each fabric type in grams per square metre. The measurements were performed by the factory supervisor using a scale, and the recorded values are canvas 307, twill 242, denim 326, RFD 351, fleece 186 and corduroy 371 g/m^2. These are reported alongside the capture statistics in Table 3.2 and plotted in Figure 3.2.

The value of these measurements is that they are not derived from the imagery. A class taxonomy assembled purely from appearance is open to the objection that its categories are arbitrary partitions of a continuous space, and agreement between visually defined classes and independently measured weights constitutes external evidence against that objection. The measured span is substantial: corduroy at 371 g/m^2 is almost exactly twice the weight of fleece at 186 g/m^2, and the remaining four classes are distributed through that interval rather than clustered at one end.

It is worth noting that the ordering by weight does not follow the ordering by visual similarity. RFD at 351 g/m^2 sits close to corduroy in mass while resembling it in no other respect, and fleece is the lightest of the six despite being the thickest to the touch, because its bulk derives from a raised pile that traps air rather than from dense yarn. A model operating on imagery alone cannot recover weight, which is one reason the areal-density values are recorded as metadata characterising each class rather than used as a prediction target.

One limitation applies to all of these figures. Areal density was recorded once per class, from

the supervisor's measurement of a representative sample, rather than per image or per roll. Production material varies within a class, and that within-class variation is therefore not modelled. The values should be read as characteristic weights for each category rather than as per-sample measurements.

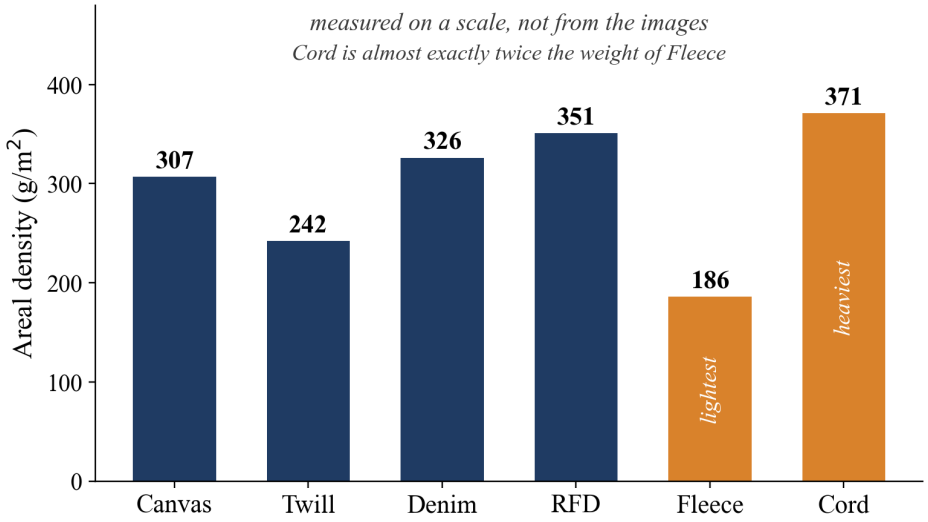


Figure 3.2: Measured areal density per class.

3.2 Dataset Preparation

3.2.1 Overview and Rationale

Preparation of the corpus involved three activities: designing the capture protocol so that variation was introduced systematically rather than incidentally, establishing a labelling workflow whose reliability does not depend on after-the-fact interpretation of the photographs, and partitioning the result into subsets suitable for supervised development and honest evaluation. Each is described below.

3.2.2 Capture Variation Protocol

Rather than suppressing nuisance variation, the acquisition protocol introduced it deliberately along five axes, illustrated in Figure 3.3. Every class was photographed from the front and from the back, at four digital zoom levels (1×, 2×, 4× and 10×), flat and in rolled and backward-rolled layouts, with the camera flash both enabled and disabled, and under whatever ambient illumination was present. At least five images were collected for each such variation, and the number of distinct variations differs by class, ranging from 11 for denim to 35 for canvas and totalling 118 across the dataset. Table 3.2 reports these counts.

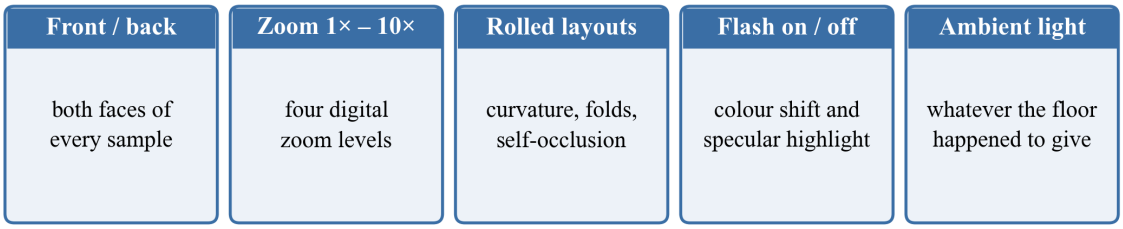


Figure 3.3: Sources of variation introduced during acquisition.

Each axis targets a specific failure mode. Front and back views matter because several of these materials are not symmetric through their thickness: the face of a twill shows a clear diagonal while its reverse can read almost plain, and a fleece carries its raised pile on one side only. A model trained only on faces would fail whenever an operator happened to photograph a reverse. The four zoom levels address working distance: at 1× the frame includes the material together with a substantial amount of its surroundings and the weave is small in the image, while at 10× the weave fills the frame and individual yarns are resolved. Because the operator's distance from the material is not controlled in practice, the model must work across that range. Rolled layouts introduce curvature, self-occlusion and shading gradients, since material on a production floor is rarely lying perfectly flat. Flash captures address illumination directly, altering colour temperature, producing specular highlights and flattening shadow, so that a flash image and an ambient image of the same material can differ considerably while carrying the same label.

The number of variations differs across classes because access differed. This was not planned and it reflects how many distinct rolls of each material were present on the floor during the collection period.

Table 3.2: Capture variations and measured areal density per class.

Class	Images	Variations	GSM (g/m^2)
Canvas	575	35	307
Twill	575	21	242
Denim	483	11	326
RFD	575	13	351
Fleece	575	19	186
Cord	571	19	371
Total	**3,354**	**118**	—

3.2.3 Labelling and Verification

Labels were assigned using the known fabric type at the moment of capture. This detail is worth stating precisely, because it removes a category of error that affects datasets assembled from existing image collections. The label did not come from someone examining a photograph

afterwards and deciding what the material appeared to be; it came from knowing which physical roll was being photographed at the time the photograph was taken.

Where product information or manufacturer labelling was available, it was used to confirm the assigned class. The full collection was reviewed a second time after acquisition, before any sample was admitted to the dataset. That review checked, for each image, that the intended material was actually in frame, that the image was not badly out of focus, and that the filename and folder placement matched the class recorded during capture.

One limitation of this workflow should be stated plainly. Verification relied on review by people who knew the material, but no second independent annotator re-labelled the collection, so no inter-annotator agreement statistic can be reported. For a dataset whose labels derive from knowledge of the source roll rather than from perceptual judgement of an image, the risk this introduces is lower than it would be for a perception-based annotation task, but it remains a gap.

3.2.4 Stratified Partitioning

Each class was divided using a stratified 60/20/20 allocation across training, validation and test subsets, with the proportions applied within each class rather than to the collection as a whole. The resulting counts are given in Table 3.3 and plotted in Figure 3.4. Stratification ensures that every subset contains all six classes in approximately the same ratio as the full dataset; without it, a random split of an unbalanced collection can leave a subset short of one class, which destabilises the resulting metrics.

Table 3.3: Stratified dataset split by class.

Class	Total	Train (60%)	Validation (20%)	Test (20%)
Canvas	575	345	115	115
Twill	575	345	115	115
Denim	483	289	97	97
RFD	575	345	115	115
Fleece	575	345	115	115
Cord	571	342	114	114
Total	**3,354**	**2,011**	**671**	**671**

The training and validation subsets supported parameter updates, learning-rate scheduling, early stopping and checkpoint selection. The test subset was reserved exclusively for the final evaluation and was not consulted at any point during development. Maintaining this separation is what permits the reported test figures to be read as an estimate of performance on unseen data. The validation subset, by contrast, is consulted once per epoch and is used to decide when to reduce the learning rate, when to stop and which checkpoint to retain; because those decisions are made using validation performance, validation accuracy is optimistically biased and cannot be quoted as a generalisation estimate.

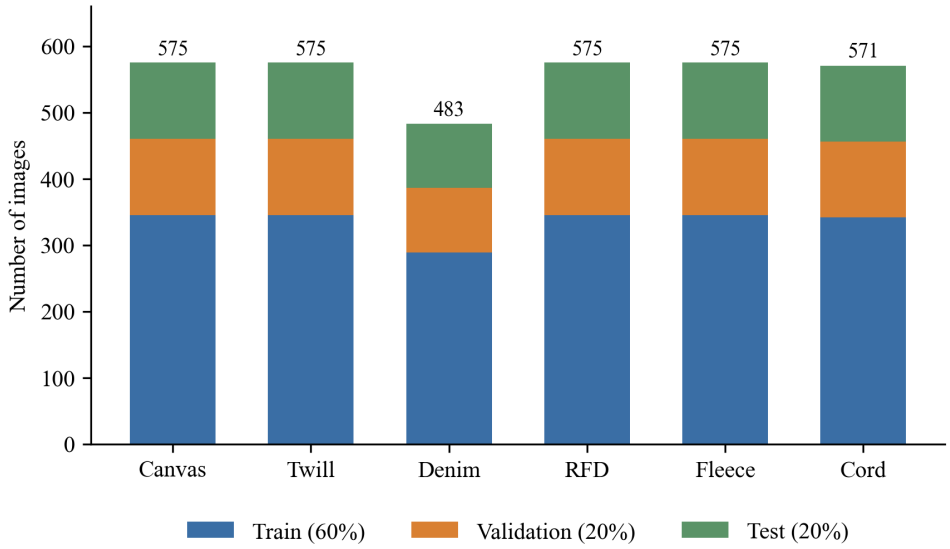


Figure 3.4: Distribution of images under the stratified 60/20/20 split.

Denim contains 483 images against 575 for most other classes. The resulting imbalance is mild, with the largest class exceeding the smallest by 19 per cent, which is far removed from the severe imbalance common in medical imaging corpora. No class weighting or resampling was applied, and whether the minority class suffered as a result is examined directly in the error analysis.

The shortfall has a practical rather than a methodological cause. Denim was the class for which the fewest distinct rolls were available on the floor during the collection period, eleven variations against thirty-five for canvas, and the protocol fixed a minimum of five images per variation rather than a target per class. Two options were available once the shortfall became apparent: leave the counts as collected, or discard images from the larger classes until every class matched the smallest. The second would have produced a balanced corpus of 2,898 images by throwing away 456 usable photographs, which is a poor trade when the imbalance is this mild. The counts were therefore left as collected, and the consequence is carried forward honestly into the evaluation rather than hidden by resampling.

Chapter-4: Methodology

4.1 Proposed Method

The methodology follows a single workflow from acquisition through preparation, training, fine-tuning and evaluation, illustrated in Figure 4.1. Every backbone was trained with the same two-stage transfer-learning procedure implemented in TensorFlow [35]: an initial feature-extraction stage in which the pretrained backbone is frozen, followed by partial fine-tuning in which the deepest backbone layers are released for adaptation. Pretrained weights were taken from the ImageNet benchmark [9].

Holding the protocol fixed across all candidate models is the central methodological commitment of this study, and it is what permits the comparison that follows to support architectural conclusions. Had each backbone been tuned individually, with its own learning rate, schedule and head design, the resulting table would report which model received the most tuning attention at least as much as which model is better suited to the task. Fixing the protocol removes that confound at the cost of not presenting any individual model at its individually optimal configuration, and that trade is accepted deliberately.

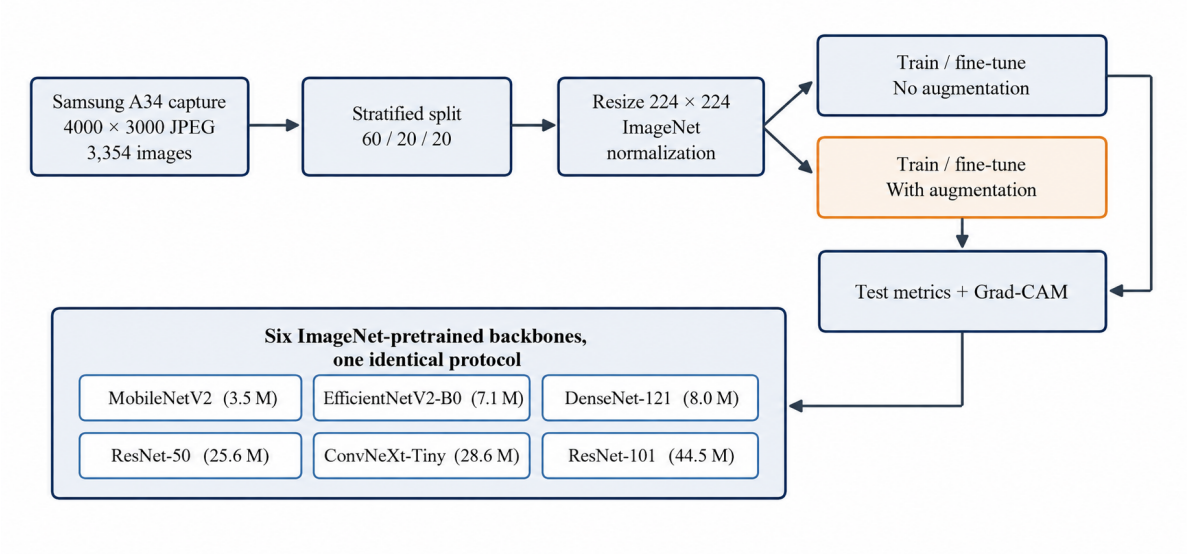


Figure 4.1: Experimental pipeline from capture to backbone comparison.

Acquisition, stratified partitioning, resizing and normalisation are common to every run. Each backbone is then trained twice, once without augmentation and once with the augmentation

policy described below, and both branches are evaluated on the same held-out test partition.

4.1.1 Dataset Pre-processing

Pre-processing was kept deliberately minimal, on the principle that the study is concerned with what a network can extract from ordinary smartphone imagery rather than with how far that imagery can be cleaned before the network sees it. The same steps were applied to every image in every run and in both augmentation settings, and no step is conditioned on the backbone, on the augmentation setting or on the class label.

The native 4000×3000 photographs were rescaled to 224×224 using bilinear interpolation, matching the input resolution expected by the ImageNet-pretrained backbones. This is a substantial reduction, from twelve megapixels to approximately 0.05 megapixels, and it discards most of the captured detail. The resolution is nevertheless fixed by the pretrained weights, whose learned filters are matched to the spatial frequencies present at that scale; supplying a considerably larger input would break that correspondence and would also increase computation by orders of magnitude.

The practical consequence is that fine weave detail survives only for images captured at higher digital zoom. At 1×, a photograph of a fabric roll reduced to 224×224 retains colour and coarse surface character but loses individual yarns. At 10×, the weave occupies enough of the original frame that its structure remains visible after downscaling. Including all four zoom levels in the acquisition protocol therefore does more than add viewpoint variety: it ensures that part of the training data preserves the fine structure that separates the woven classes.

Resized images were then standardised channel-wise using the ImageNet channel statistics, namely a mean of $[0.485, 0.456, 0.406]$ and a standard deviation of $[0.229, 0.224, 0.225]$, according to

$$x_{\text{norm}} = \frac{x - \mu}{\sigma} \tag{4.1}$$

where x is the raw pixel value in a given channel. Using the source-corpus statistics rather than statistics computed from the fabric data keeps the input distribution close to what the pretrained weights expect, which matters most during the first training stage, when the backbone is frozen and cannot adapt. JPEG images were decoded and converted to floating-point tensors of shape $224 \times 224 \times 3$ before normalisation. No colour correction, white balancing, denoising, sharpening or background removal was applied at any stage.

4.1.2 Data Augmentation

Online augmentation was applied only within the training stream of the augmented configuration. The term online indicates that transformations are applied as each batch is drawn, so a given image is presented in a slightly different form on each epoch rather than being expanded into a

fixed set of stored copies. The non-augmented baseline used identical resizing and normalisation without any stochastic transformation.

Two transformations were used, summarised in Table 4.1. Horizontal flipping was applied with probability $p = 0.5$, targeting viewpoint symmetry, since a sample photographed from the left and from the right carries the same label. Brightness jitter of maximum magnitude 0.05, corresponding to ± 5 per cent, was applied with probability $p = 0.3$, targeting the illumination drift that is the single most variable factor across capture sessions.

Table 4.1: Online augmentation policy applied to the training stream in the augmented configuration.

Transform	**Setting**	**Purpose**
Horizontal flip	$p = 0.5$	Viewpoint symmetry
Brightness jitter	max $\pm 5\%$, $p = 0.3$	Illumination drift

Augmentation was never applied to the validation or test streams, which were resized and normalised only. Applying stochastic transformations at evaluation time would make the reported metrics depend on which random draws occurred and would not correspond to any deployment condition.

This policy is deliberately conservative, and the reasoning follows the label-preservation constraint discussed in the literature review. Rotation was excluded because it would interfere with the directional cues that separate twill and corduroy from the remaining classes. Colour jitter beyond brightness was excluded because the RFD class is defined partly by being undyed and the denim class partly by its indigo warp, so shifting hue risks moving a sample toward the appearance of a different class while retaining its original label. Random cropping was excluded because at high magnification a crop can remove the weave pattern entirely. Whether this caution was justified or excessive is examined empirically once the augmented and non-augmented runs are compared.

4.1.3 Backbone Architectures

Six ImageNet-pretrained convolutional backbones were evaluated, listed in Table 4.2 in order of increasing parameter count. The selection was made to span the design space rather than to enumerate available models, and covers efficiency-first design, compound scaling, dense feature reuse, residual connectivity at two depths, and the modernised convolutional design space informed by transformer architectures. Between them the six span more than a twelvefold range in parameter count.

Each of the six is described below, together with the structural feature that makes it a plausible candidate for fabric texture.

Table 4.2: Convolutional backbones evaluated in this study.

Backbone	Design principle	Parameters (M)
MobileNetV2	Depthwise separable, inverted residuals	3.5
EfficientNetV2-B0	Compound scaling, fused MBConv	7.1
DenseNet-121	Dense feature reuse	8.0
ResNet-50	Residual connectivity, medium depth	25.6
ConvNeXt-Tiny	Modernised convolutional design	28.6
ResNet-101	Residual connectivity, high depth	44.5

4.1.3.1 MobileNetV2

MobileNetV2 is built from inverted residual blocks with linear bottlenecks [13], shown in Figure 4.2. A block expands the channel count with a 1×1 convolution, filters each channel independently with a depthwise 3×3 convolution, and projects back to a narrow representation with a second 1×1 convolution that is left linear. Omitting the rectifier at the projection is the design decision that gives the family its name, because applying a non-linearity in a narrow space destroys information that cannot be recovered. That property is directly relevant here, since the evidence separating one weave from another is fine and low in contrast, and it is also the cheapest of the six models to run.

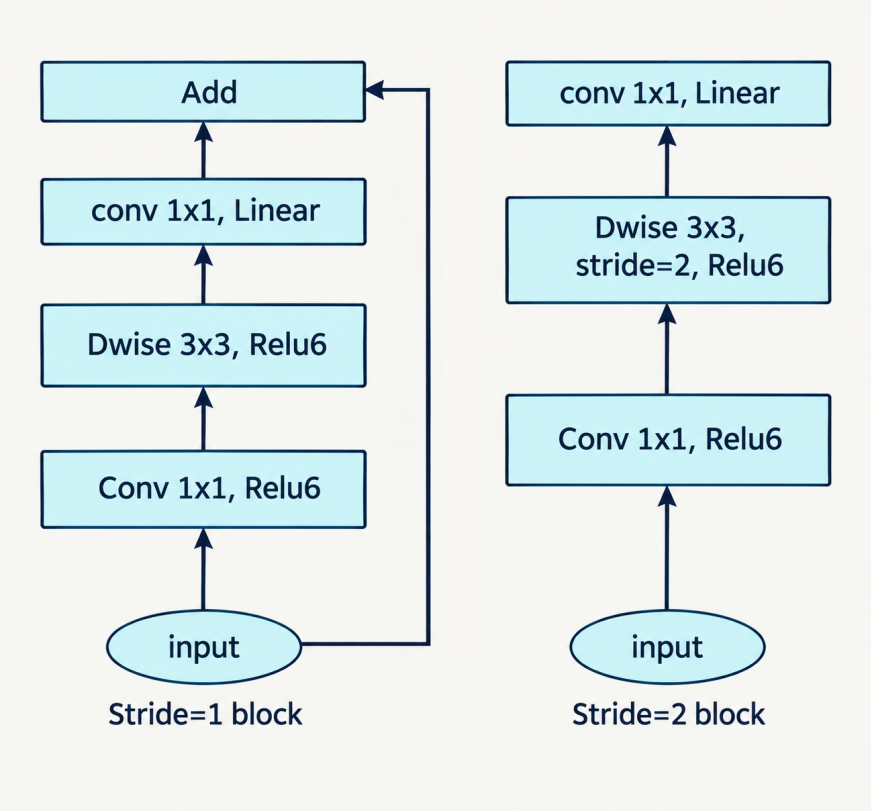


Figure 4.2: MobileNetV2 inverted residual blocks [13].

4.1.3.2 EfficientNetV2-B0

EfficientNetV2 scales depth, width and input resolution together under a single compound coefficient rather than enlarging any one of them alone [14]. Its stage layout is shown in Figure 4.3. The early stages use fused MBConv blocks, in which the expansion and depthwise convolutions of a standard MBConv block are replaced by one ordinary 3×3 convolution, because depthwise operations are poorly served by accelerator hardware at high spatial resolution. Later stages revert to MBConv, where the channel count is high and the depthwise saving is real. The result is a network that reaches a useful capacity level at 7.1 million parameters without the depth of the larger residual models.

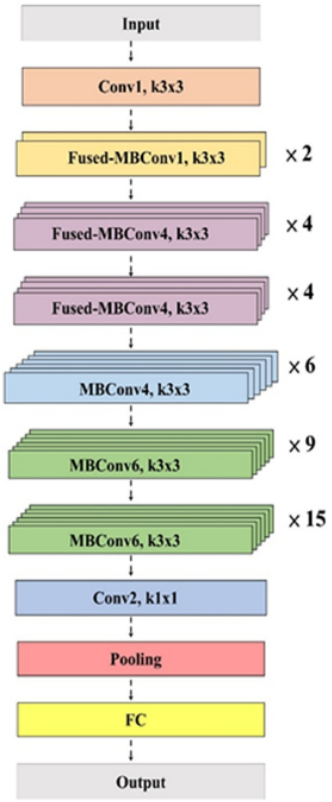


Figure 4.3: EfficientNetV2-B0 stage layout [14].

4.1.3.3 DenseNet-121

DenseNet connects every layer within a block to every later layer in that block, so each layer receives the concatenated feature maps of all its predecessors [15]. Figure 4.4 illustrates one such block. Because features are concatenated rather than summed, low-level detail captured near the input remains directly available at the deepest layers instead of being progressively overwritten by abstraction. This suits a task in which the low-level detail is the signal: yarn thickness and surface roughness are exactly the kind of early-layer evidence that a summation-based network tends to discard. The cost is repeated concatenation of feature maps, which is memory-bandwidth intensive at inference time.

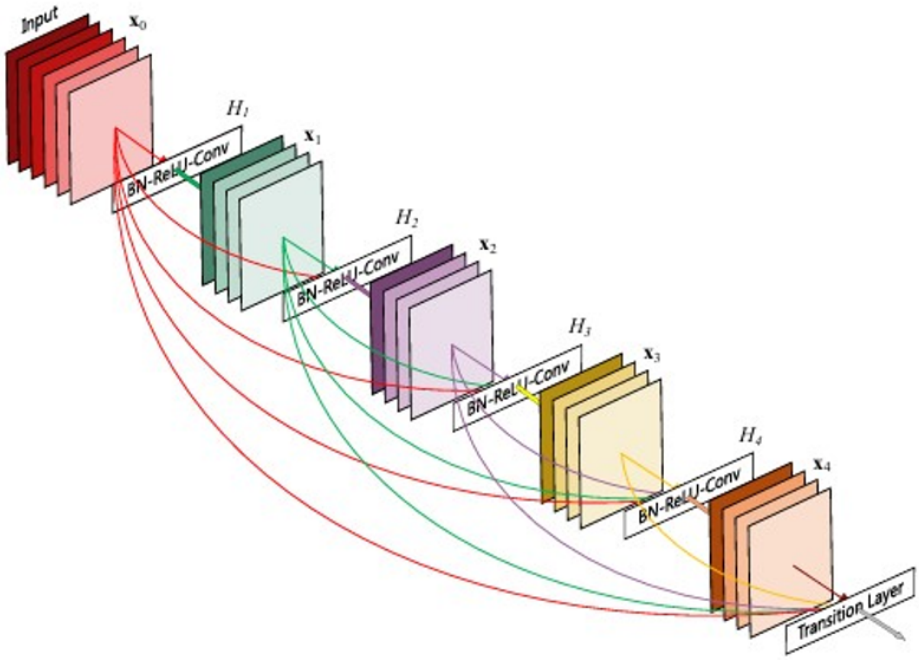


Figure 4.4: Dense connectivity within a DenseNet block [15].

4.1.3.4 ResNet-50

ResNet introduced the residual block, in which a shortcut connection carries the input of a block forward and is added to the block's output, so that each block learns a residual correction rather than a complete mapping [16]. Figure 4.5 shows the stage structure of the 50-layer variant [36]: a stem of convolution, batch normalisation, rectification and max pooling, followed by four stages of bottleneck blocks with progressive downsampling, then global average pooling and a fully connected layer. It is included here as the mid-capacity residual reference point, and it is the most widely reported classification baseline in the texture and material recognition literature.

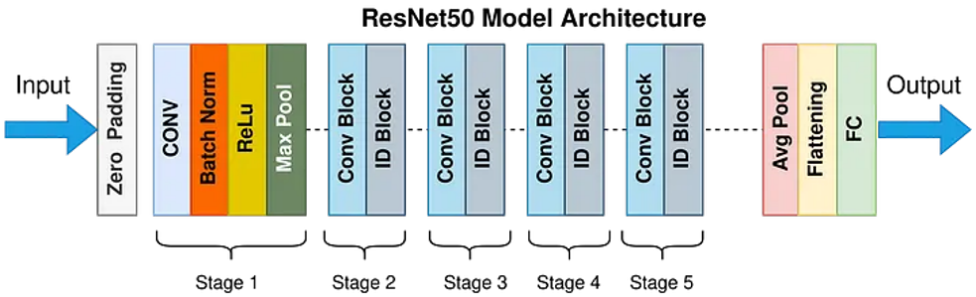


Figure 4.5: ResNet-50 stage architecture [36].

4.1.3.5 ConvNeXt-Tiny

ConvNeXt takes a standard residual network and applies, one at a time, the design choices that distinguish vision transformers, retaining those that help: a patch stem, an inverted bottleneck,

fewer activation and normalisation layers, layer normalisation in place of batch normalisation, and a large depthwise kernel [17]. Its four-stage layout, with block counts of three, three, nine and three, is shown in Figure 4.6. The feature that matters most for fabric is the 7×7 depthwise kernel, since a single layer with that receptive field can span the repeat period of a coarse weave rather than having to assemble it across several layers.

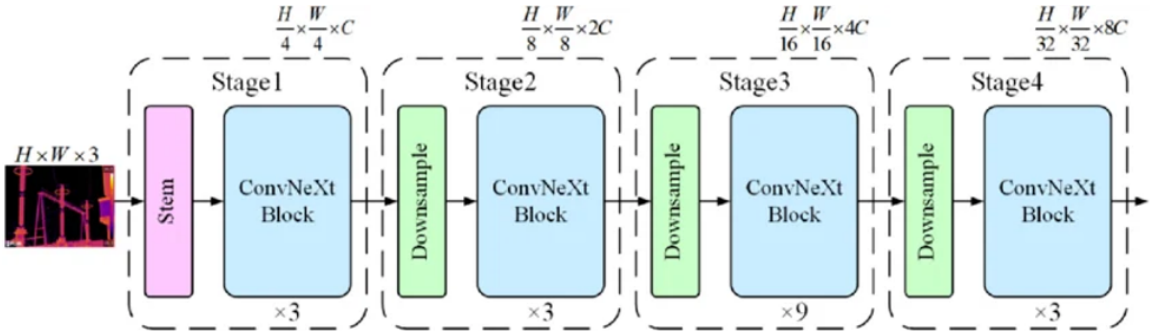


Figure 4.6: ConvNeXt four-stage layout [37].

4.1.3.6 ResNet-101

ResNet-101 shares its design, block structure and training treatment with ResNet-50 and differs only in depth, carrying 23 bottleneck blocks in its fourth stage against six in the shallower model [16], as Figure 4.7 shows [38]. Including both variants is deliberate: because everything except depth is held constant between them, the difference in their results isolates the contribution of depth alone, which no comparison between unrelated architectures can do. At 44.5 million parameters it is also the largest model in the benchmark and therefore sets the upper bound on what additional capacity is worth on this task.

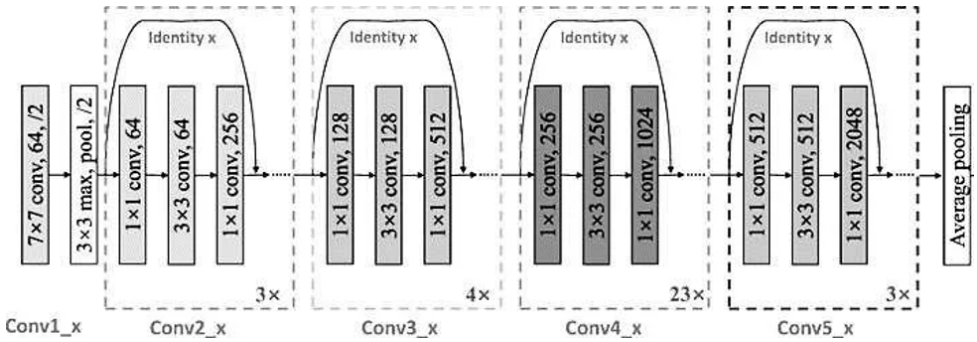


Figure 4.7: ResNet-101 layer configuration [38].

4.1.4 Shared Classification Head

Every backbone feeds into an identical classification head, shown in Figure 4.8. The head consists of global average pooling, layer normalisation with $\epsilon = 10^{-6}$ [39], dropout with $p = 0.3$ [40], and a dense layer of six units with softmax activation.

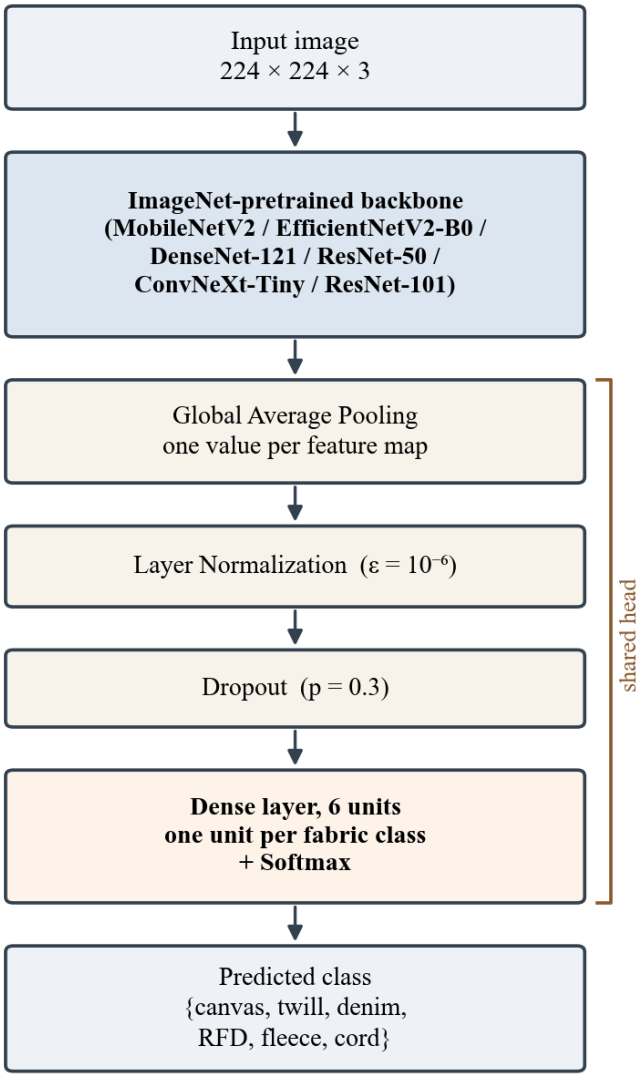


Figure 4.8: Model architecture used across all six experiments.

Only the backbone is exchanged between runs. The input resolution, the classification head and the output layer are identical throughout, the dense layer carries one unit per fabric class, and softmax converts the six activations into a probability distribution.

Each component serves a specific purpose. Global average pooling collapses the spatial dimensions of the final feature map to one value per channel, which makes the head independent of the spatial extent produced by a given backbone and eliminates the large flattened dense layer used by older designs. Layer normalisation standardises the pooled vector across its features, stabilising the scale of the classifier input when different backbones emit features of very different magnitude. Dropout zeroes 30 per cent of pooled features during training, discouraging the classifier from depending on any individual feature. The final dense layer maps to six units, one per class, and softmax converts them to a probability distribution.

Keeping this head fixed across all six backbones is what makes the comparison a backbone

comparison. Were the head tuned per model, differences in the results would reflect head design as well as feature quality.

4.2 Training Protocol

All runs were carried out on a workstation equipped with an NVIDIA GeForce RTX 5070 Ti graphics processor and an AMD Ryzen 9 7900X processor. Training used the Adam optimiser [41] with a learning rate of 1×10^{-4}, a batch size of 16, and sparse categorical cross-entropy loss, which is the appropriate objective for single-label multiclass problems whose targets are integer class indices rather than one-hot vectors. The complete configuration is given in Table 4.3.

Table 4.3: Training hyperparameters, held identical across all six backbones and both augmentation settings.

Parameter	**Value**
Input resolution	$224 \times 224 \times 3$
Normalisation	ImageNet channel mean and standard deviation
Batch size	16
Optimiser	Adam
Learning rate	1×10^{-4}
Loss function	Sparse categorical cross-entropy
Stage 1 epochs (maximum)	60
Stage 2 epochs (maximum)	40
Fine-tuned layers	Deepest 40 of the backbone
Dropout	$p = 0.3$
Layer normalisation ϵ	1×10^{-6}
Early stopping	Validation loss, patience 5, restore best weights
Learning-rate schedule	Reduce on plateau, factor 0.5, patience 3, floor 10^{-6}
Checkpointing	Best validation loss
Output layer	Dense, 6 units, softmax
Hardware	NVIDIA RTX 5070 Ti, AMD Ryzen 9 7900X

The two-stage schedule is illustrated in Figure 4.9. In the first stage, optimisation is restricted to the newly attached classification head while all pretrained backbone weights remain fixed. The reason for this ordering is that the head begins with random weights, so its early gradients are large and noisy; allowing those gradients to propagate into a well-trained backbone from the outset would damage the pretrained features before the head has learned anything useful.

In the second stage, training resumes from the best Stage 1 checkpoint and the deepest 40 backbone layers are released so that higher-level features can adapt to textile-specific structure such as weave direction, ridge continuity and local contrast. Restricting fine-tuning to the deepest layers follows directly from the feature-transferability result of Yosinski et al. [7]: early layers encode generic primitives that transfer without modification and are best left untouched, whereas later layers encode class-specific combinations tied to the source labels and benefit most from

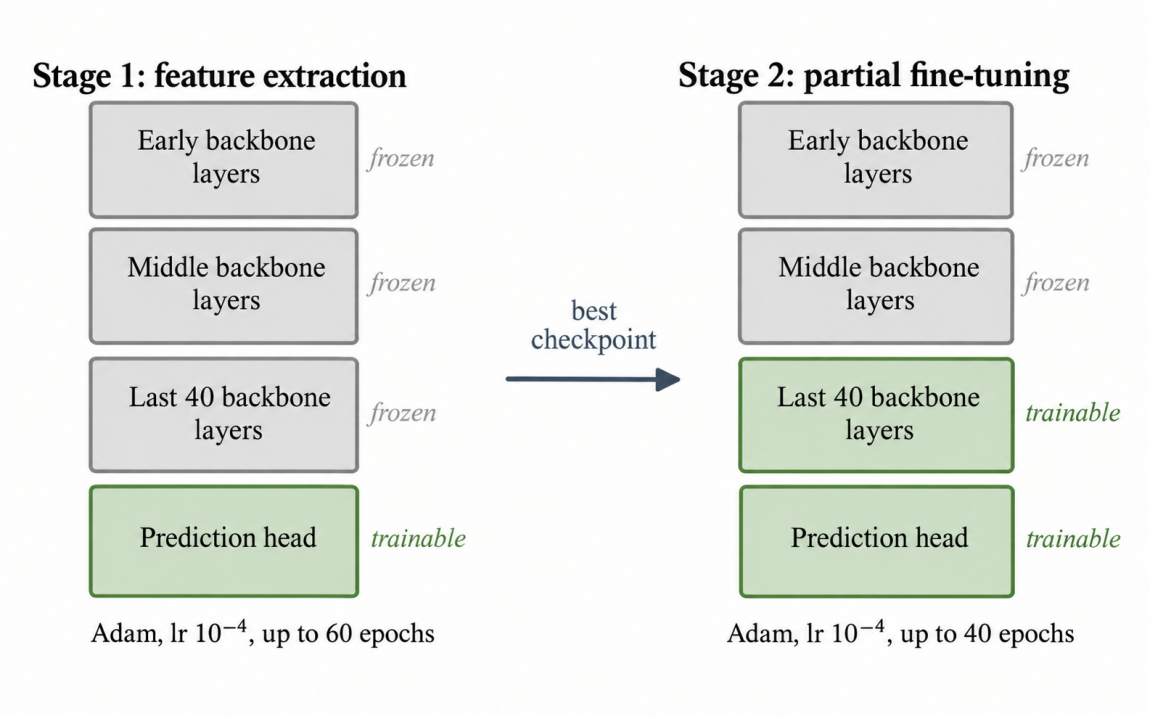


Figure 4.9: Two-stage transfer-learning schedule.

adaptation. Releasing the entire network would additionally risk overfitting, since the training partition contains 2,011 images against the millions on which the source weights were learned.

Four regularisation and training-control measures were applied. Early stopping monitored validation loss with a patience of five epochs and restored the best weights when triggered, so the epoch budgets of 60 and 40 act as upper bounds rather than as fixed run lengths. Reduction of the learning rate on plateau halved the rate after three epochs without validation improvement, down to a floor of 10^{-6}, allowing finer steps near a minimum. Best-model checkpointing retained the weights whenever validation loss reached a new minimum, so that the model carried into evaluation is the best observed rather than the last trained. Dropout in the classifier head provided the fourth measure.

4.3 Evaluation Framework

All metrics were computed on the held-out test partition after fine-tuning. For each class, counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) were obtained under a one-versus-rest formulation and then averaged across the six classes, which is the standard extension of these binary definitions to the multiclass setting [42, 43].

4.3.1 Precision

Precision measures the proportion of predicted positives that are correct:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

It therefore reflects label reliability. In a sorting application, low precision for a class means material is routed into that class when it does not belong there, which appears downstream as contaminated batches.

4.3.2 Recall

Recall measures the proportion of actual positives that are recovered:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

It reflects detection completeness. Low recall means material belonging to a class is being missed and routed elsewhere.

4.3.3 F1-Score

The F1-score is the harmonic mean of precision and recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

The harmonic mean is used rather than the arithmetic mean because it is dominated by the smaller of the two quantities, so a model cannot score well on F1 by performing excellently on one and poorly on the other.

4.3.4 Accuracy

Accuracy gives the overall proportion of correct predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

It is the most intuitive of the four but the least informative under class imbalance, since a model can score well by favouring larger classes. The imbalance here is mild, so accuracy remains meaningful, but it is reported alongside the other metrics rather than alone.

4.3.5 Area Under the ROC Curve

The area under the receiver operating characteristic curve summarises the relationship between true positive rate and false positive rate across all decision thresholds [44]. It differs from the preceding four metrics in one important respect: those four describe behaviour at a single operating point, the point at which the highest softmax score is selected, whereas AUROC describes how well the model ranks samples across every threshold. A model with high AUROC but mediocre F1-score is ranking correctly while placing its decision boundary poorly, which is a correctable condition, and the distinction matters for a deployment that might later add a confidence threshold.

4.3.6 Computational Metrics

Because classification quality alone does not determine whether a model can be deployed, three cost measures were recorded for every backbone. Parameter count comprises the backbone together with the shared head and determines memory footprint. Training time is the wall-clock duration of the complete two-stage run on the single graphics processor described above, including per-epoch validation. Inference latency is the per-image processing time at batch size one after warm-up on the same hardware.

Three qualifications apply, and are stated here so that the reported cost figures are not overread. The latency is graphics-processor latency at batch size one, not smartphone latency, and a mid-range mobile processor would be substantially slower. The training time excludes data collection and labelling, which in practice was the largest time cost of the project. The parameter count is neither the on-disk model size nor the runtime memory requirement.

4.4 Interpretability Analysis

Gradient-weighted class activation mapping was used to examine which image regions drive the predictions of the strongest-performing model [19]. The method takes the gradient of the score for a chosen class with respect to the feature maps of the final convolutional layer, averages it spatially to obtain one importance weight per feature map, forms the weighted sum, and applies a rectifier so that only regions exerting positive influence on the class score are retained. The resulting map is upsampled to the input resolution and overlaid on the original image.

Two properties constrain how the output is read. The map is produced at the resolution of the final convolutional feature map, typically 7×7, so it is coarse and cannot localise a single yarn; and it shows where the evidence lay, not what property of that region was informative. Both are sufficient for the question posed here, whether the evidence lies on the fabric or outside it. The analysis tests for one failure mode, shortcut learning [18], not the model as a whole.

Chapter-5: Results and Analysis

5.1 Experimental Overview

Twelve training runs were carried out in total, one for each of the six backbones in each of the two augmentation settings, under the protocol set out in the previous chapter. Every run used the same stratified partition, the same preprocessing, the same classification head and the same optimiser configuration, so differences between rows in the tables that follow reflect architecture and augmentation rather than tuning effort.

The chapter proceeds from the aggregate to the specific. It first reports the classification metrics for both settings and compares the backbones, then isolates the effect of augmentation, examines the loss curves, analyses the structure of the residual errors, reports threshold-independent behaviour and computational cost, and closes with the interpretability analysis.

5.2 Classification Performance

5.2.1 Performance Without Augmentation

Table 5.1 reports the fine-tuned test metrics obtained without augmentation. EfficientNetV2-B0 achieves the highest value on every threshold-based metric, reaching 0.995 accuracy with precision, recall and F1-score all at 0.996. ResNet-101 follows at 0.992 accuracy, then ConvNeXt-Tiny at 0.991, ResNet-50 at 0.989 and DenseNet-121 at 0.988. MobileNetV2 records the lowest accuracy at 0.978.

One row deserves separate comment. MobileNetV2 records a precision of 0.993, which exceeds its own recall of 0.985 and considerably exceeds its accuracy of 0.978. A model whose precision exceeds its recall makes relatively few false positive assignments while missing more true members of each class. In sorting terms this is the safer of the two failure modes: material routed to a class is usually correct, while material belonging to a class is more often missed and diverted for review.

Table 5.1: Fine-tuned test performance without augmentation. The best value in each column is shown in bold.

Backbone	Precision	Recall	F1-score	Accuracy	AUROC
MobileNetV2	0.993	0.985	0.989	0.978	**1.000**
EfficientNetV2-B0	**0.996**	**0.996**	**0.996**	**0.995**	**1.000**
DenseNet-121	0.988	0.988	0.988	0.988	**1.000**
ResNet-50	0.991	0.990	0.990	0.989	**1.000**
ConvNeXt-Tiny	0.991	0.991	0.991	0.991	**1.000**
ResNet-101	0.991	0.991	0.991	0.992	**1.000**

5.2.2 Performance With Augmentation

Table 5.2 reports the corresponding metrics with augmentation enabled. EfficientNetV2-B0 again leads on every threshold-based metric, improving to 0.998 accuracy with precision, recall and F1-score all at 0.999. It is the only model in the benchmark to reach 0.999 on any of the three. ResNet-101 improves to 0.995, DenseNet-121 to 0.994 and ResNet-50 to 0.991, while ConvNeXt-Tiny declines to 0.989 and MobileNetV2 to 0.976.

Table 5.2: Fine-tuned test performance with online augmentation. The best value in each column is shown in bold.

Backbone	Precision	Recall	F1-score	Accuracy	AUROC
MobileNetV2	0.992	0.988	0.990	0.976	**1.000**
EfficientNetV2-B0	**0.999**	**0.999**	**0.999**	**0.998**	**1.000**
DenseNet-121	0.994	0.994	0.994	0.994	0.999
ResNet-50	0.996	0.993	0.994	0.991	**1.000**
ConvNeXt-Tiny	0.993	0.992	0.992	0.989	**1.000**
ResNet-101	0.996	0.996	0.996	0.995	**1.000**

5.2.3 Comparative Analysis and Model Selection

Two observations govern the interpretation of both tables.

The first concerns the spread. Across twelve runs, test accuracy ranges from 0.976 to 0.998, a span of just over two percentage points, and every configuration exceeds 0.97. The task is therefore well within reach of transfer learning at this data scale, which answers the feasibility question directly. It also means the differences between backbones are small in absolute terms and should not be overstated: a gap of two accuracy points on a partition of this size corresponds to roughly thirteen images.

The second concerns the ordering, which is nevertheless consistent. EfficientNetV2-B0 is best in both settings and on every threshold-based metric. MobileNetV2 is last in both settings on accuracy. The four models between them change relative position depending on setting and met-

ric, which is what one would expect when the differences separating them amount to a handful of images.

The result of principal interest is that EfficientNetV2-B0, at 7.1 million parameters, is the second smallest model in the benchmark and outperforms ResNet-101 at 44.5 million, which carries more than six times as many parameters. The two residual models also allow the effect of depth to be read directly, since they share design, block structure and training treatment: ResNet-101 exceeds ResNet-50 by 0.003 accuracy without augmentation and 0.004 with it. Depth therefore helps, but the return is small relative to its cost, and the comparison with EfficientNetV2-B0 indicates that balanced compound scaling is a more efficient use of a parameter budget than depth alone on this task.

5.3 Effect of Data Augmentation

Figure 5.1 presents the accuracy of each backbone in both settings together with the change produced by augmentation. The effect is real but small, and it is not uniform across architectures. Four of the six models improved and two declined. DenseNet-121 gained most at six accuracy points per thousand, followed by EfficientNetV2-B0 and ResNet-101 at three each and ResNet-50 at two. MobileNetV2 and ConvNeXt-Tiny each lost two.

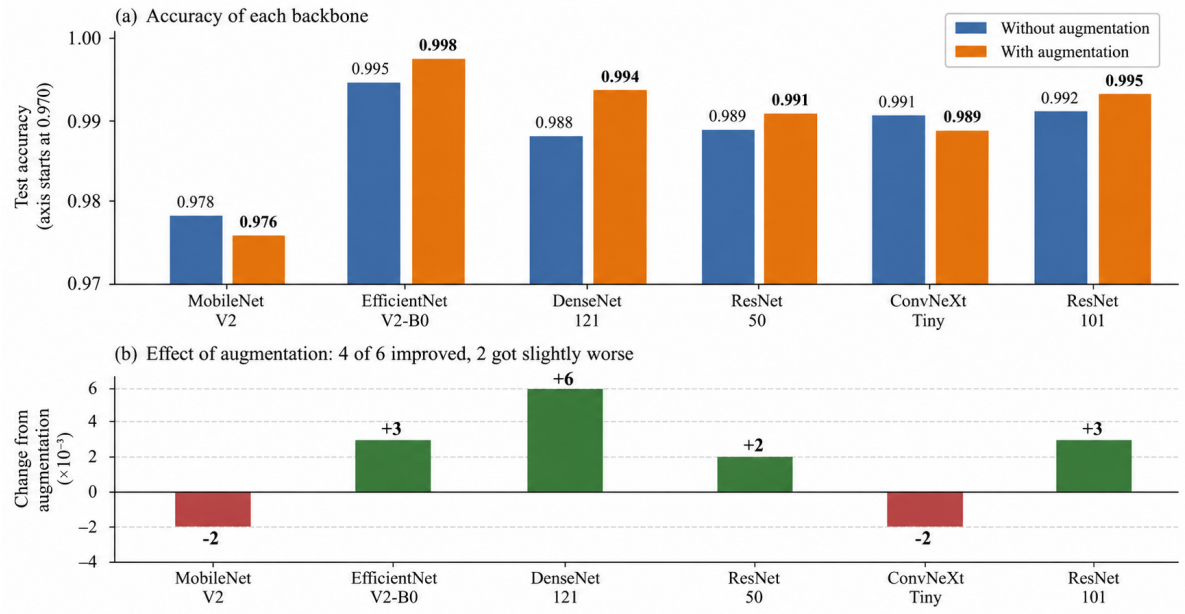


Figure 5.1: Test accuracy and the effect of augmentation.

Panel (a) gives the test accuracy of each backbone in both settings. Its vertical axis begins at 0.970 so that the differences remain visible; on a full axis from zero to one the twelve bars would be indistinguishable. Panel (b) gives the change produced by augmentation, expressed in accuracy points multiplied by one thousand.

That two models declined is stated explicitly because a summary reporting only the average

effect would conceal it. The declines amount to two accuracy points per thousand in each case, which on a partition of this size is a difference of one or two images and falls within the variation expected from stochastic training. The defensible reading is that augmentation clearly assisted the moderate-capacity and deeper models and made no reliable difference to MobileNetV2 or ConvNeXt-Tiny.

The proposed explanation is a capacity argument. Augmentation operates by adding variation to the training distribution, and a model can exploit that variation only if it possesses spare representational capacity in which to encode the additional invariance. MobileNetV2, designed to be as small as possible, has the least spare capacity of the six and correspondingly the least to gain. DenseNet-121, which retains access to low-level features at every depth, has the most to gain from training data that varies precisely those features, and it recorded the largest improvement.

The magnitude of these effects should also be read against the mildness of the policy, which combines one geometric and one photometric transformation, both at low intensity. A gain of six accuracy points per thousand from a horizontal flip and a five per cent brightness jitter is a reasonable return on that intervention, and it licenses no conclusion about what a stronger policy would achieve.

5.4 Loss Curve Analysis

Figure 5.2 shows loss and accuracy across the fine-tuning stage of a representative run. The curves are informative less for what they show than for what they do not show, namely any sign of the failure modes that would undermine the reported test figures.

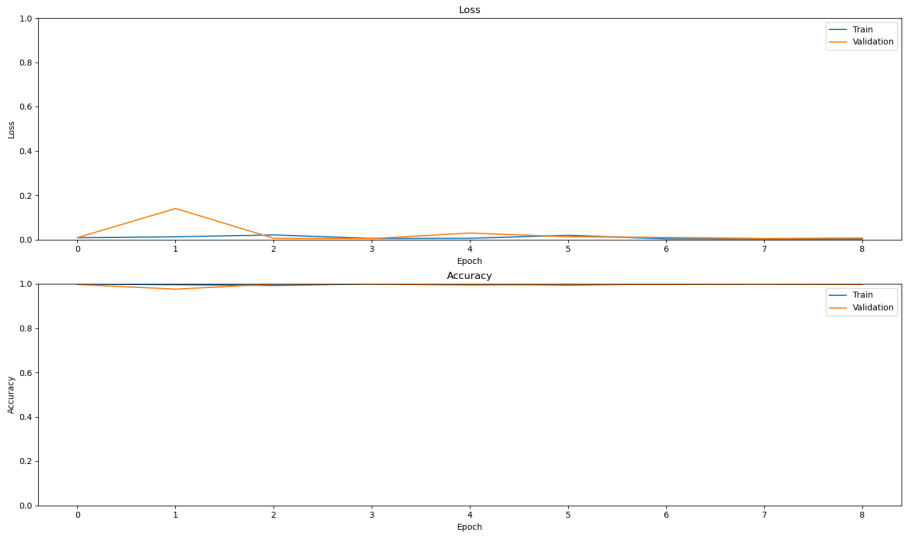


Figure 5.2: Training and validation curves for the fine-tuning stage.

Three features are relevant. Convergence is rapid, with both training and validation loss falling

close to zero within the first three epochs. This is characteristic of transfer learning on a task the pretrained features already suit: the network is not learning to represent texture from scratch, it is reweighting representations it already possesses. Validation loss exhibits a pronounced disturbance at the first epoch before settling, which coincides with the point at which the deepest 40 backbone layers are released and begin to move; the important detail is that it recovers within a single epoch rather than destabilising the run, indicating that the fine-tuning learning rate was low enough. Most importantly, the training and validation traces remain close together throughout. The classic signature of overfitting is a training loss that continues to fall while validation loss turns upward, opening a widening gap, and no such pattern appears. Combined with the early stopping and best-model checkpointing used throughout, this supports reading the test figures as genuine generalisation rather than memorisation of the training partition.

One caveat applies. Because accuracy is already close to unity from the first epoch, the accuracy panel is compressed into the top of its range and carries little visual information; the loss panel is the more informative of the two at this level of performance.

5.5 Confusion Matrix Analysis

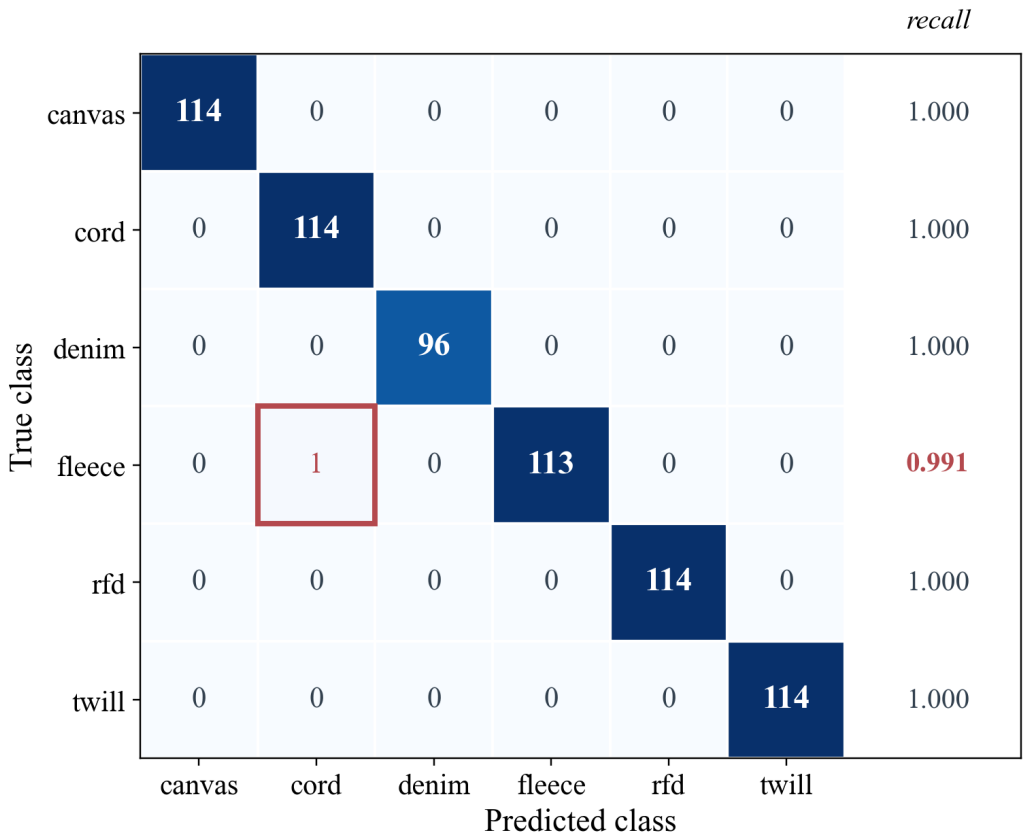


Figure 5.3: Confusion matrix on the held-out test partition.

Scalar metrics report how often a model is wrong but not what it is wrong about. Figure 5.3

presents the confusion matrix for the best-performing configuration on the held-out partition, together with per-class recall.

Rows are true classes and columns are predicted classes, so the diagonal contains correct predictions and every off-diagonal entry is an error, with per-class recall shown at the right. The matrix is almost entirely diagonal. Canvas, corduroy, denim, RFD and twill are classified without error in either direction. The only off-diagonal entry in the whole matrix is a single fleece sample predicted as corduroy. Reading the per-class figures directly from the matrix, recall is 1.000 for canvas, corduroy, denim, RFD and twill, and 0.991 for fleece, which lost one of its 114 test samples. Precision is 1.000 for every class except corduroy, which received one image that did not belong to it and therefore scores 114 out of 115, or 0.991.

Two observations follow, and the second is the more interesting.

The expected confusions did not occur. Denim and twill were identified earlier as the pair most likely to be confused, since denim is itself a twill construction and the two share their underlying interlacing. Not a single denim sample was labelled twill, nor the reverse. The model is evidently able to exploit the colour and finish differences that separate them, and the dyed indigo warp of denim appears to remain a sufficient cue even after downscaling to 224×224.

The one confusion that did occur is the one that the class definitions would have judged least likely on the basis of appearance alone. Fleece was described as the most visually distinctive class in the dataset, an irregular pile with no periodic structure, while corduroy was described as strongly directional with parallel raised wales; on paper the two are opposites. The most plausible account is that both are pile fabrics, in which fibre is raised off the surface, and that the distinction rests on whether that pile is cut into regular ridges. A corduroy sample photographed at low magnification, at an angle that flattens the wales, or under diffuse illumination that suppresses the shadow between them, could plausibly present a soft undifferentiated pile surface. With a single error there is no basis for a stronger claim, and this should be read as a hypothesis rather than a finding.

The absence of any denim error also addresses the concern raised by the class imbalance. Denim is the smallest class, with 483 images against 575 for most others, and the natural worry with an unbalanced corpus is that the smallest class absorbs a disproportionate share of the errors. That did not occur. Denim was classified perfectly, indicating that the mild imbalance present here did not materially disadvantage the minority class.

5.6 Threshold-Independent Behaviour

The metrics discussed so far describe behaviour at a single operating point. The area under receiver operating characteristic curve describes behaviour across them. Figure 5.4 reports per-class AUROC before and after fine-tuning, and Figure 5.5 shows corresponding Stage 2 curves.

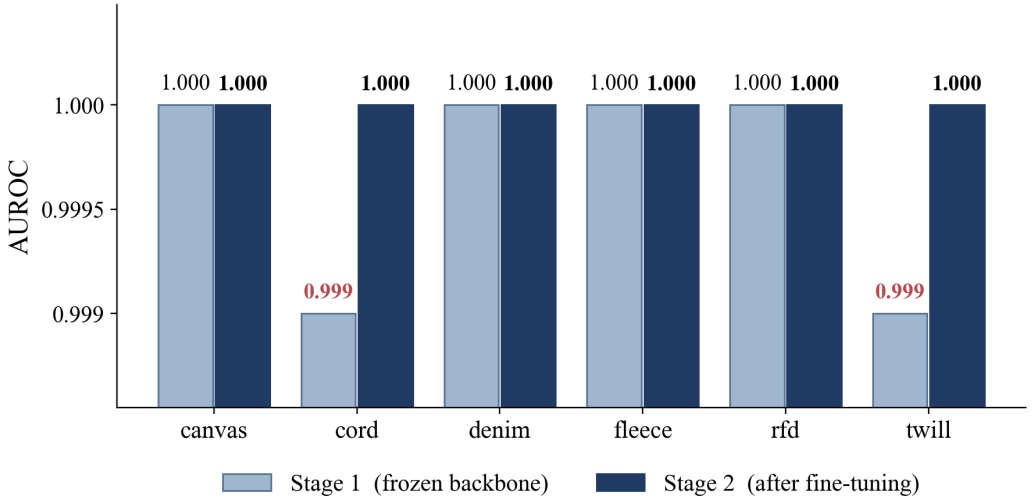


Figure 5.4: Per-class AUROC before and after fine-tuning.

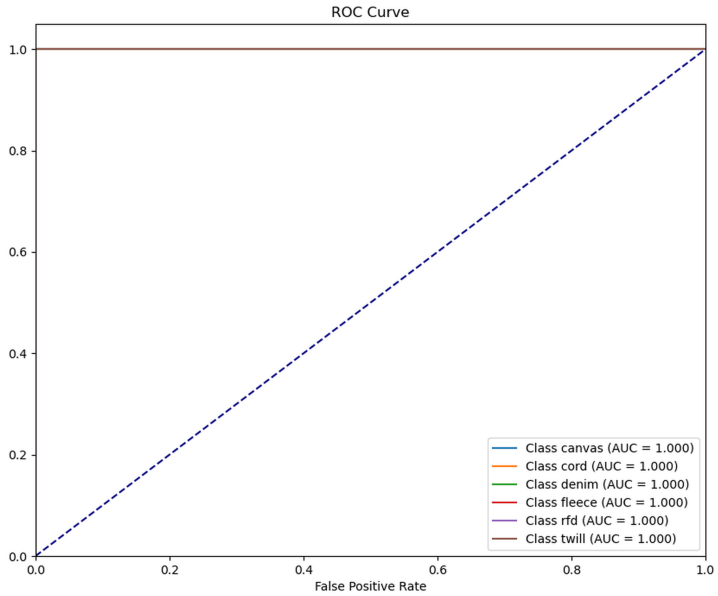


Figure 5.5: ROC curves for all six classes after fine-tuning.

The comparison between the two stages is the informative part. After the first stage, with the backbone frozen and only the prediction head trained, the model already separates the classes very well, but corduroy and twill fall marginally short of a perfect area at 0.999. After fine-tuning the deepest 40 backbone layers, every class reaches 1.000. This is direct evidence that the second stage performs useful work rather than merely consuming computation: adapting the deepest backbone features to textile-specific structure closes the remaining ranking errors.

Two consequences follow. The near-perfect AUROC values reported in Tables 5.1 and 5.2 indicate that the six classes are largely separable under the present acquisition protocol, so that the differences among the stronger backbones arise less from coarse class confusion than from how consistently each preserves correct top-ranked predictions under appearance variation. And an area of 1.000 implies that a threshold exists which separates the classes cleanly, which is the

precondition for a confidence-based rejection scheme that would route uncertain samples to a human operator.

A caution is nevertheless warranted. An AUROC of exactly 1.000 on a partition of this size means only that no negative sample outranked any positive sample for that class. It is a ceiling effect, and it indicates that the metric has stopped discriminating between the stronger models rather than that the problem is solved. The discussion in this chapter accordingly rests on the threshold-based metrics and on the confusion matrix rather than on AUROC.

5.7 Computational Cost

Table 5.3 reports the three cost measures for every backbone, and Figure 5.6 plots them together. Parameter count determines memory footprint, training time determines the expense of retraining when new material is introduced, and inference latency determines how long an operator waits for a result.

Table 5.3: Computational metrics measured on an NVIDIA GeForce RTX 5070 Ti at batch size one after warm-up. The best value in each column is shown in bold.

Backbone	Parameters (M)	Training time (h)	Inference latency (ms)
MobileNetV2	**3.5**	**5.1**	**6.2**
EfficientNetV2-B0	7.1	6.3	8.9
DenseNet-121	8.0	7.8	15.2
ResNet-50	25.6	8.2	12.3
ConvNeXt-Tiny	28.6	10.4	16.8
ResNet-101	44.5	14.5	18.7

Read across the rows, the three columns of Table 5.3 do not rise together. Parameter count spans a twelvefold range, from 3.5 million for MobileNetV2 to 44.5 million for ResNet-101, but training time spans less than threefold and latency only threefold, so the largest model is far from twelve times the most expensive to run. Training time is the column that tracks size most faithfully, because every additional parameter must be updated on every batch. Latency is the column that departs from it, since inference at batch size one is governed as much by how many sequential operations a network performs, and how much data those operations move, as by how many weights it holds. The rows of principal interest are the first two: MobileNetV2 is the cheapest on all three measures simultaneously, and EfficientNetV2-B0 doubles its parameter count for 2.7 ms of additional latency, which is the price of the accuracy difference reported earlier in this chapter.

MobileNetV2 is the fastest option at 6.2 ms per image and is therefore attractive for on-device sorting, whereas EfficientNetV2-B0 offers the best combination of recognition quality and latency for a graphics-processor-backed station. ResNet-101 also delivers strong metrics, but its

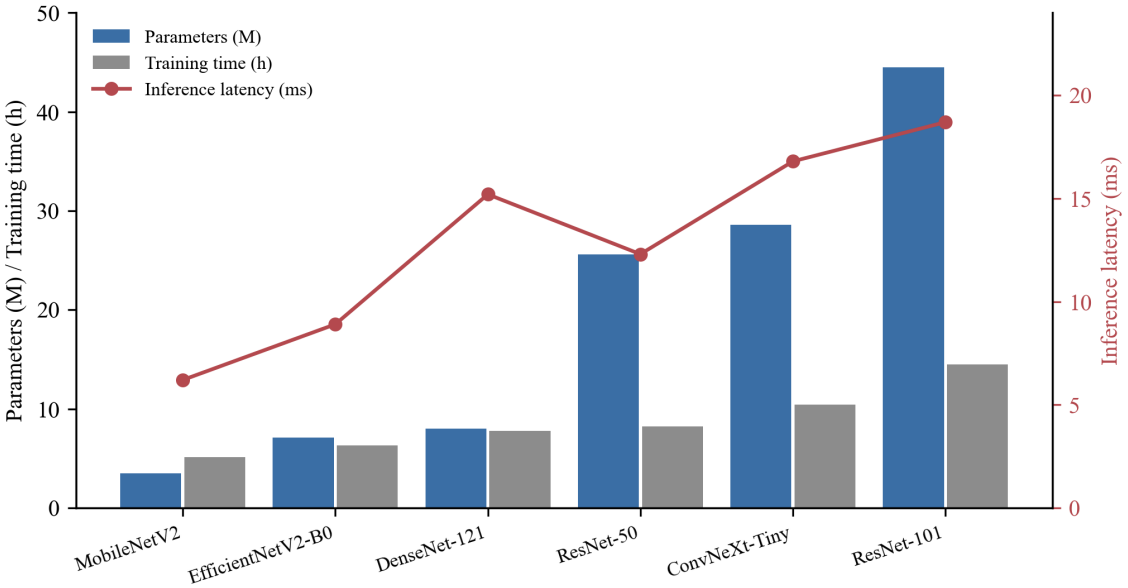


Figure 5.6: Parameter count, training time and inference latency.

44.5 million parameters and 18.7 ms latency make it unattractive for edge deployment unless the modest accuracy gain justifies the additional cost.

The cost measures do not track one another monotonically, and the exception is informative. DenseNet-121 carries only 8.0 million parameters yet requires 15.2 ms per image, which is slower than ResNet-50 at 12.3 ms despite ResNet-50 having more than three times as many parameters. Dense connectivity requires repeated concatenation of feature maps, which is memory-bandwidth intensive and does not parallelise as cleanly on a graphics processor as the element-wise additions of a residual block. Parameter count is therefore a poor proxy for speed, and a deployment decision taken on model size alone would be misled here.

Figure 5.7 combines accuracy and cost into a single view.

Accuracy with augmentation is plotted against per-image inference latency, with marker area proportional to parameter count. Positions toward the upper left are preferable, since they combine high accuracy with low latency. Two models occupy that frontier. MobileNetV2 sits at the extreme lower left: nothing in the benchmark is faster or smaller, and any model with higher accuracy costs more time. EfficientNetV2-B0 sits at the upper left: nothing is more accurate at any latency, and it achieves this at 8.9 ms with 7.1 million parameters. Every other backbone is dominated, being simultaneously less accurate, slower and larger than EfficientNetV2-B0. This simplifies the recommendation considerably: for this task and this dataset the choice is between two models, and it turns on whether the deployment context can afford 8.9 ms and 7.1 million parameters or requires 6.2 ms and 3.5 million.

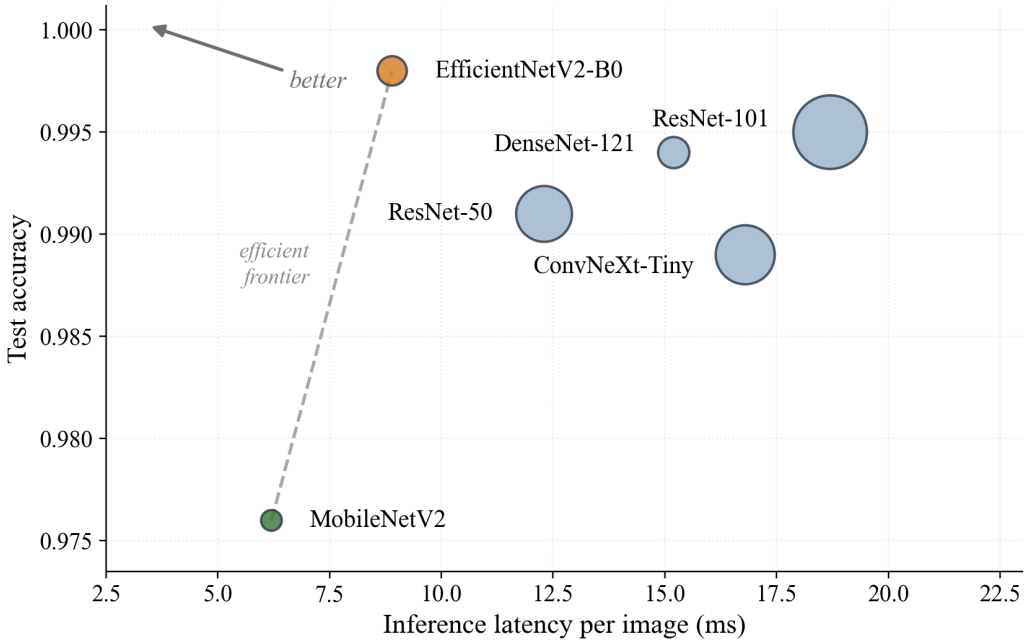


Figure 5.7: Test accuracy against inference latency.

5.8 Interpretability Results

The results above establish that the models are accurate. They do not establish that the models are accurate for the right reasons, which is the shortcut-learning concern raised earlier and is a genuine risk for a corpus collected without background control. Figure 5.8 addresses it directly.

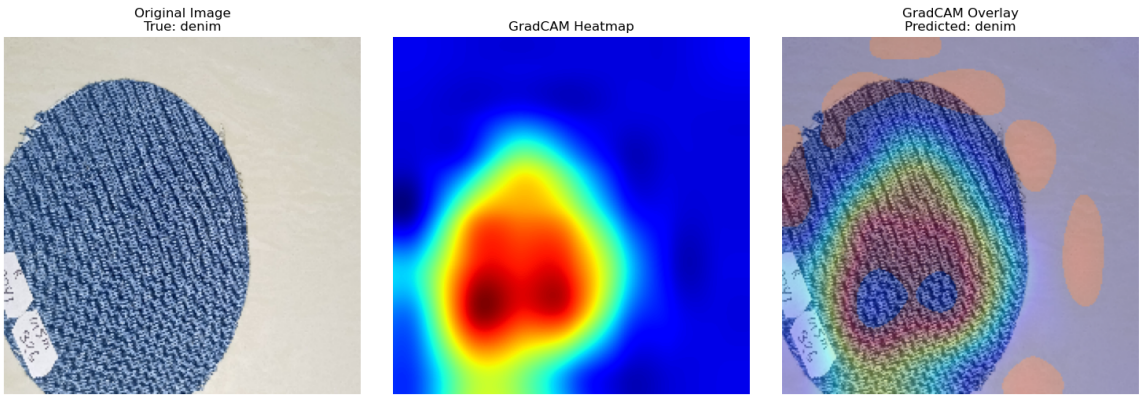


Figure 5.8: Grad-CAM analysis for a representative denim sample.

The three panels show the original image, the Grad-CAM heatmap and the heatmap overlaid on the image together with the predicted class. Activation concentrates within the fabric surface.

Two details of this map bear on the specific concerns raised earlier. The first is that the surrounding table occupies a substantial fraction of the frame and receives almost no attention. This is the shortcut-learning question in its most direct form: had the network learned to identify denim by the surface on which it was photographed, the response would sit outside the swatch rather than within it. It does not.

The second is more specific. Many samples carry a small handwritten tag recording the roll identifier and the measured areal density, and those tags were flagged during dataset construction as a possible confound. A tag is a bright, high-contrast, class-correlated object and is precisely the feature a network would exploit if it could. In Figure 5.8 the tag is clearly visible at the lower left of the swatch and the Grad-CAM response there is low. The evidence supporting the prediction lies in the weave rather than in the label attached to it. Given that the tags were an unintended by-product of the measurement workflow rather than a designed element of the dataset, this is a reassuring result.

Across representative samples the activation concentrates on weave structure and local surface texture rather than on empty background regions, which supports the interpretation that the model responds to material-specific visual cues rather than to incidental capture artefacts. Where the highlighted region shifts away from the textile area, that sample would be a reasonable candidate for manual review in a practical inspection workflow, which suggests a concrete operational use for Grad-CAM beyond analysis.

The limits of this evidence should be stated. Grad-CAM produces a coarse map at the resolution of the final convolutional feature layer, and the analysis rests on representative samples rather than a quantitative measure across the whole test partition. Figure 5.8 supports the negative claim that the model is not relying on background or measurement tags, rather than the stronger claim that it has learned weave structure in depth.

Two further qualifications follow from how the method works. The map records where the evidence supporting a prediction was located, not what property of that region carried the information, so a response centred on the weave is consistent with the model attending to yarn periodicity, to local contrast, or to a sensor-specific artefact that happens to coincide with the fabric. A shortcut lying inside the fabric region would therefore be invisible to this test. And because the maps were inspected for a model that misclassifies one image in 671, there are effectively no failure cases available against which to compare a successful explanation, which is the comparison that would make the analysis diagnostic rather than merely reassuring. Converting the check into a number, by segmenting the fabric region across the test partition and measuring the fraction of activation mass that falls inside it, would remove both qualifications and is the form this analysis should take in future work.

Chapter-6: Discussion

6.1 Benchmarking Under a Unified Protocol

The central methodological commitment of this study was to hold every element of the training pipeline fixed and exchange only the backbone. That decision governs what the reported results can and cannot support, and the consequence is worth stating in both directions.

What it supports is attribution. Because preprocessing, optimiser, loss function, schedule, prediction head and data partition were identical across all twelve runs, the ordering observed in Tables 5.1 and 5.2 is attributable to backbone design rather than to differences in tuning effort. This is the property that comparisons assembled from separately published results cannot offer, and it was the second of the three gaps that motivated this work.

What it forgoes is peak performance. No backbone is shown at its individually optimal configuration, and it is likely that each could be improved by tuning its learning rate, schedule or head capacity separately. The absolute figures reported here should therefore be read as what each architecture achieves under a common and reasonable protocol rather than as an upper bound. For the purpose of choosing among architectures this is the more useful quantity, but the distinction matters when comparing these numbers against published results obtained under individually optimised conditions.

A third consequence emerged from the results themselves rather than from the design. Every configuration exceeded 0.97 accuracy, and the spread across twelve runs is just over two percentage points. At that level the differences separating the middle of the field amount to a handful of images, and single-run point estimates cannot reliably order them. The claim that EfficientNetV2-B0 is the strongest model on this dataset is supported, because it leads in both settings and on every threshold-based metric. The finer ordering among DenseNet-121, ResNet-50, ConvNeXt-Tiny and ResNet-101 is not something this experiment can resolve.

6.2 Capacity, Scaling and Texture Recognition

The result of principal interest is that the second smallest model in the benchmark is also the most accurate. EfficientNetV2-B0, at 7.1 million parameters, outperforms ResNet-101 at 44.5 million

in both augmentation settings. The paired residual models sharpen the point: ResNet-50 and ResNet-101 share design, block structure and training treatment, differing essentially in the number of blocks in their third stage, and the deeper model gains only 0.003 to 0.004 accuracy for an additional 18.9 million parameters, 6.3 hours of training and 6.4 ms of inference latency.

A plausible reading rests on where the discriminative evidence lies. Fabric classification depends on local texture statistics, namely yarn thickness, surface roughness, weave periodicity and weak directional cues, and these are available early in a convolutional hierarchy. Additional depth principally adds capacity for combining features into higher-level abstractions, which is what object recognition requires, since the identity of an object is not determined by its local surface. For a task in which the local surface is itself the target, that additional capacity is largely unused. A network that maintains sufficient width at moderate depth, which is what compound scaling produces [14], therefore has an advantage over one that spends its parameter budget on depth alone.

This account is consistent with the DenseNet-121 result but does not require it. Dense connectivity preserves access to low-level features at every depth [15], which on the same reasoning should suit texture, and DenseNet-121 does reach 0.994 accuracy with augmentation at only 8.0 million parameters. It does not, however, exceed EfficientNetV2-B0, and its inference latency of 15.2 ms is the second worst in the benchmark. The evidence therefore supports the general claim that moderate-capacity designs suit this task, and does not isolate which specific architectural mechanism is responsible.

6.3 Augmentation and Model Capacity

The augmentation results are the clearest instance in this study of a finding that must be reported carefully rather than summarised. Four backbones improved and two declined. Reporting only the mean effect would conceal that split and would misrepresent what the experiment showed.

The direction of the split is consistent with a capacity account. Augmentation adds variation to the training distribution, and exploiting that variation requires spare representational capacity in which to encode the resulting invariance. The models that gained, namely DenseNet-121, ResNet-101 and EfficientNetV2-B0, occupy the moderate-capacity and deeper positions. MobileNetV2, engineered to be as small as possible, has the least spare capacity and gained nothing. ConvNeXt-Tiny sits at the other end of the size range and also gained nothing, which the capacity account does not explain; its decline is therefore left as an observation rather than folded into the explanation.

The magnitudes constrain the conclusion further. The two declines amount to two accuracy points per thousand, or one to two images on this partition, which is within the variation expected from stochastic initialisation and data ordering. Since each configuration was trained once, no

variance estimate is available against which to test that expectation formally. The defensible statement is therefore that augmentation assisted the moderate and deeper models by a small margin and made no reliable difference to the other two, and that a stronger claim would require repeated runs.

The policy itself bounds the conclusion further still. It comprised one geometric and one photometric transformation, both at low intensity, chosen conservatively because rotation would interfere with the directional cues separating twill and corduroy and because colour jitter would threaten the definitions of the RFD and denim classes. A gain of six accuracy points per thousand from a horizontal flip and a five per cent brightness jitter is a reasonable return on that intervention, and it licenses no inference about what a stronger policy would achieve.

6.4 Deployment Trade-offs on Real Hardware

Reporting cost alongside accuracy changes the shape of the recommendation. On accuracy alone the answer is EfficientNetV2-B0 and the discussion ends. Once latency and parameter count enter, the frontier analysis in Figure 5.7 shows that four of the six backbones are dominated outright, being simultaneously less accurate, slower and larger than EfficientNetV2-B0. Only two models remain defensible, and the choice between them is determined by deployment context rather than by any property of the models themselves.

For a centralised inspection station with graphics-processor support, EfficientNetV2-B0 is the appropriate choice, delivering 0.998 accuracy at 8.9 ms per image. For on-device or edge-oriented sorting, MobileNetV2 trades 2.2 accuracy points for a 30 per cent reduction in latency and a halving of parameter count. Whether that trade is acceptable depends on the application. For a fully automated line with no operator in the loop the accuracy difference is material; for a handheld assistant that flags a suggestion to an operator who retains the final decision, the difference is much less consequential and the size and latency advantages may dominate.

The measurement conditions qualify these figures, and were stated explicitly with the methodology for that reason. The reported latency was measured at batch size one on a desktop graphics processor after warm-up. It is not smartphone latency, and a mid-range mobile processor would be substantially slower. MobileNetV2 is recommended for that setting precisely because its design targets it, but the figure of 6.2 ms should not be quoted as a mobile inference time. Similarly, the reported training times exclude data collection and labelling, which constituted the largest real time cost of this project, and the parameter counts are neither on-disk model sizes nor runtime memory requirements.

One further observation from Table 5.3 deserves emphasis because it affects practical decisions. Latency does not track parameter count monotonically. DenseNet-121 carries 8.0 million parameters yet is slower than ResNet-50 at 25.6 million, because dense connectivity requires repeated

concatenation of feature maps and is memory-bandwidth intensive rather than arithmetic-bound. A deployment decision taken on model size alone would therefore be misled, and latency must be measured rather than inferred.

6.5 Evidence Against Shortcut Learning

The interpretability analysis was included because the acquisition protocol that makes this dataset realistic is the same decision that creates room for a shortcut. Backgrounds and illumination were not standardised, so any feature of the environment that happened to correlate with class membership was available for the network to exploit [18]. A second and more specific risk arose from the areal-density workflow, in which handwritten tags recording roll identifier and measured weight were attached to samples before photography. Such a tag is a bright, high-contrast, class-correlated object of exactly the kind a network would use if it could.

The Grad-CAM evidence addresses both. Activation concentrates within the fabric surface. The surrounding table, which occupies a substantial fraction of the frame, draws almost no response, and the measurement tag likewise draws almost none. On the available evidence the model is responding to material rather than to context.

The confusion structure points the same way, and it is worth noting because it is independent of the attribution method. The single error observed on the test partition is between fleece and corduroy, two pile fabrics differing in whether the pile is cut into regular wales. That is a material-based confusion. Had the model been keying on the acquisition environment, one would expect errors distributed according to where samples were photographed rather than according to surface construction. Equally, the complete absence of denim and twill confusion, despite those classes sharing a weave family, indicates that the model is using colour and finish information that survives downscaling rather than relying on weave geometry alone.

These two lines of evidence are mutually reinforcing, but neither is conclusive. Grad-CAM produces a coarse map at the resolution of the final convolutional feature layer, and the analysis rests on representative samples rather than on a quantitative measure computed across the whole partition. What the evidence supports is the negative claim, that the model is not relying on background or on the measurement tags. It does not support the positive claim that the model has acquired an understanding of weave structure in any deeper sense, and this report does not make that claim.

Chapter-7: Conclusion

This report set out to determine whether fabric type can be recognised reliably from smartphone photographs captured under ordinary factory conditions, and to establish which convolutional architecture is best suited to that task once accuracy and deployment cost are considered together. It approached the question by constructing the infrastructure such a comparison requires, namely a dataset, a fixed training protocol and an evaluation framework covering both recognition quality and computational cost, and then reporting what that infrastructure revealed.

Three contributions follow. The first is a public dataset of 3,354 mobile photographs covering six production-relevant fabric classes, captured with a single smartphone inside an operating textile facility under a deliberately uncontrolled protocol, accompanied by independent areal-density measurements spanning 186 to 371 g/m^2 and released for reuse. The second is a controlled benchmark of six ImageNet-pretrained backbones under one identical two-stage transfer-learning procedure, evaluated with and without online augmentation on the same stratified partition, which permits observed differences to be attributed to architecture rather than to tuning effort. The third is an evaluation that reports deployment cost and interpretability evidence alongside classification metrics rather than accuracy alone.

The principal empirical findings are as follows. Recognition under uncontrolled handheld capture proved entirely feasible: all twelve runs exceeded 0.97 test accuracy despite deliberate variation in viewpoint, digital zoom, illumination and material handling. EfficientNetV2-B0 achieved the strongest overall performance, reaching 0.998 accuracy with augmentation together with precision, recall and F1-score of 0.999, and it did so as the second smallest model in the benchmark, outperforming ResNet-101, which carries more than six times as many parameters. This indicates that balanced compound scaling is a more efficient use of a parameter budget than depth alone for a task whose discriminative evidence is local surface texture. MobileNetV2 recorded the lowest inference latency at 6.2 ms per image while retaining a precision of 0.993, and the frontier analysis showed that only these two backbones are undominated, which reduces the deployment decision to a single trade between 2.2 accuracy points and a 30 per cent latency reduction.

The effect of augmentation was real but small and not uniform. Four backbones improved by two to six accuracy points per thousand and two declined by amounts falling within run-to-run variation, a pattern consistent with the interpretation that spare representational capacity is required to exploit added training variation, although the mildness of the augmentation policy

bounds how far that interpretation can be pushed. Error analysis revealed a single misclassification on the held-out partition, a fleece sample assigned to corduroy, and no confusion whatever between denim and twill despite their shared weave family. Grad-CAM analysis showed activation concentrated within the fabric surface rather than on the surrounding background or on the handwritten measurement tags, which addresses the shortcut-learning risk that an uncontrolled acquisition protocol necessarily introduces.

Taken together, these results establish that a compound-scaled, moderate-capacity convolutional network can achieve near-perfect fabric recognition on ordinary smartphone imagery while retaining a computational footprint compatible with practical deployment, and that lighter architectures remain a defensible choice where latency and memory are the binding constraints. The broader implication for the field is that the acquisition assumption common to the textile recognition literature, namely a controlled camera, can be relaxed considerably without the task becoming intractable.

7.1 Research Questions Answered

The four research questions posed in the introduction are answered here in turn, each against the evidence that bears on it.

RQ1: Can fabric type be recognised reliably from smartphone photographs captured under ordinary factory conditions? Yes. All twelve training runs exceeded 0.97 test accuracy on a corpus collected with one handheld phone inside an operating mill, with no tripod, no light booth and no imposed background, and with viewpoint, digital zoom, illumination and material handling deliberately allowed to vary. The best configuration misclassified one image out of 671. The acquisition control that the textile recognition literature has treated as necessary is therefore not necessary for this task at this number of classes, although the answer is bounded by the single device and single site from which the evidence comes.

RQ2: Which architecture is best suited to fabric texture, and does additional capacity translate into additional accuracy? EfficientNetV2-B0 is best, and additional capacity does not reliably help. It led on every threshold-based metric in both augmentation settings, reaching 0.998 accuracy, and it did so as the second smallest model in the benchmark, outperforming ResNet-101 which carries more than six times as many parameters. The paired residual models isolate the effect of depth directly, since they differ in nothing else: ResNet-101 gained only 0.003 to 0.004 accuracy over ResNet-50 for an additional 18.9 million parameters and 6.4 ms of latency. Balanced compound scaling is therefore a more efficient use of a parameter budget than depth alone for a task whose discriminative evidence is local surface texture. The recommendation is conditional on deployment context rather than absolute: MobileNetV2 and EfficientNetV2-B0 are the only two undominated models, and the choice between them trades 2.2 accuracy points against a 30 per cent reduction in latency.

RQ3: What effect does augmentation have, and is it uniform across architectures? The effect is real, small, and not uniform. Four backbones improved by two to six accuracy points per thousand and two declined by two, which on a partition of this size is a difference of one or two images and lies within the variation expected from single-run training. The models that gained occupy the moderate-capacity and deeper positions, which is consistent with the reading that exploiting added training variation requires spare representational capacity, but ConvNeXt-Tiny declined despite being the second largest model and that reading does not account for it. The answer is also specific to a deliberately mild policy of one horizontal flip and a five per cent brightness jitter, and licenses no claim about what a stronger policy would achieve.

RQ4: Do the predictions rest on genuine material evidence? On the available evidence, yes. Grad-CAM activation concentrated inside the fabric surface rather than on the surrounding table, which occupies a substantial fraction of the frame, and the handwritten measurement tags drew almost no response despite being bright, high-contrast and correlated with class. That is a meaningful negative result, since those tags were an unintended by-product of the areal-density workflow and are exactly the kind of feature a network would exploit if it could. It remains a negative result rather than a positive one: the analysis rests on representative samples rather than a quantitative overlap measure, and it cannot detect a shortcut that happens to lie inside the fabric region, such as a sensor-specific noise signature.

7.2 Analysis of the Social Effects

The question a factory floor will ask of this work is whether it removes jobs. The honest answer, for the system as it stands, is that it cannot. The model assigns one of six labels to a photograph. It does not unroll material, does not feel weight or hand, does not judge whether a roll has been damaged in transit, and cannot say anything at all about a material outside its six classes, which on any real floor is most of them. What it can do is give a second opinion at the moment an operator is already making a decision, which is a different economic proposition from replacing that operator.

The group most affected is therefore not the experienced inspector but the new one. In the facility where this dataset was collected, fabric identification is learned by standing beside someone who already knows, over months, and the quality of that transfer depends on who is available on a given shift and how busy they are. A tool that returns a consistent answer in under ten milliseconds shortens that apprenticeship and makes the third shift as well served as the first. It also narrows the gap between a large mill that can afford senior staff on every shift and a subcontractor that cannot, which matters in a sector where a large share of Bangladesh's output passes through smaller units.

Two social risks deserve to be named rather than assumed away. The first is deskilling: if operators come to accept the tool's answer without examining the material, the tacit knowledge

that made the dataset possible will thin out over a few years, and it is precisely that knowledge that would be needed to notice when the model has started to fail. The second is surveillance drift. A device that photographs material on a table also photographs the hands holding it and records when each roll was checked, and a system introduced as a quality aid can be repurposed as a productivity monitor without any technical change. Whether that happens is a matter of policy rather than of architecture, and it should be settled with the workforce before deployment rather than after. Further work on this system should therefore be evaluated on whether it keeps the operator in the decision, not only on whether it raises accuracy.

7.3 Environmental Impact Analysis and Sustainability

The environmental case for this work rests on avoided waste rather than on the efficiency of the models themselves. A misidentified roll is dyed to the wrong recipe, and the correction consumes water, dye, thermal energy and, in the worst case, the roll. Estimates for the sector place dyeing in the range of fifty to a hundred litres of water per kilogram of fabric, so a single mistaken roll carries a water cost several orders of magnitude larger than the electricity used to classify every image in this dataset. Reducing routing errors is thus an environmental intervention before it is a productivity one, and the appropriate next step is to measure it as such: instrumenting a trial deployment to count misroutes before and after, and converting that count into litres of water and kilowatt-hours rather than into accuracy points, would give the argument the evidence it currently lacks.

The training cost of the study should be stated on the same terms. The six backbones together consumed roughly 52 hours of graphics-processor time per augmentation setting, so approximately 104 hours across the twelve runs, on a single desktop machine, and no architecture search or hyperparameter sweep was performed. Retraining a chosen model when new material enters the catalogue costs between five and fifteen hours depending on the backbone, which is a small recurring cost against continuous operation. Choosing EfficientNetV2-B0 over ResNet-101 halves the training energy and cuts inference energy by roughly half again at equal or better accuracy, so the recommendation this report reaches on accuracy and latency grounds happens to be the lower-carbon recommendation as well. That coincidence should not be overstated, but it does mean nothing is being traded away by taking it.

Sustainability also has a data dimension that is easy to overlook. The dataset is 2.48 GB of full-resolution imagery released publicly, so every group that reuses it avoids repeating the collection and the training that would otherwise be run from scratch. Extending the release with the trained weights, so that reuse begins from a fine-tuned model rather than from ImageNet, would extend that saving further. A useful piece of future work would be to report per-experiment energy consumption alongside accuracy in the results tables, which is now standard practice in parts of the machine learning literature and is not yet standard in textile applications.

7.4 Ethical Issues

Four ethical issues arise from this work, and each of them constrains what should be built on top of it.

The first is consent and provenance. The imagery was captured inside one operating facility with the knowledge and cooperation of its management, and the areal density measurements were taken by a supervisor there. No individual is identifiable in any image, since the frames contain material, tables and occasionally a hand, and no personal data was recorded. The facility is not named. Anyone extending the dataset to further sites should treat that arrangement as the minimum standard rather than as an optional courtesy, since the commercial sensitivity of a mill's material catalogue is a real interest even when no personal data is involved.

The second is the risk of overclaiming. The headline result, 0.998 accuracy with a single misclassified image, is a genuine measurement and also an easy figure to quote out of context. It was obtained on one device, in one building, over a bounded period, on six classes chosen because that facility handles them. Quoting it as evidence that fabric classification is solved would be misleading, and a purchaser making an investment decision on that basis would be misled. This report states the boundary conditions alongside the number wherever it appears, and any publication or product built on it carries the same obligation.

The third is accountability when the system is wrong. A model that is right on 670 of 671 images will still be wrong occasionally, and in a production setting a wrong answer has a cost that falls on somebody. If the operator is held responsible for a misroute the tool suggested, the tool has transferred risk downward without transferring authority; if nobody is responsible, errors will not be investigated and the model will drift unnoticed. The explicit rejection option proposed later as further work is partly an ethical mechanism rather than a purely technical one, because it gives the system a way to decline rather than to guess, and any deployment should log its predictions so that a disputed decision can be reconstructed afterwards.

The fourth concerns the interpretability evidence itself. The Grad-CAM analysis shows that activation falls on the fabric rather than on the background or the handwritten tags, which is a meaningful negative result and is reported as such. It is not a guarantee that the model has learned weave structure, and it cannot detect a shortcut that happens to lie inside the fabric region, such as a sensor-specific noise signature. Presenting a heatmap as proof of correct reasoning would be a misuse of the method, and the distinction between what the evidence rules out and what it establishes has been maintained deliberately throughout this report.

7.5 Limitations

Several limitations constrain the generality of these conclusions, and they are stated here in order of how much they restrict the claims.

The most consequential is that the dataset derives from a single facility and a single camera. Every image shares one sensor, one lens, one image signal processor and one JPEG encoder, and every image was captured in one building. Nothing in these results establishes that a model trained here would transfer to a different device or a different mill, and the shortcut-learning literature [18] gives specific reason to expect that device and site signatures could contribute to the reported accuracy without being visible in a Grad-CAM map. No cross-device or cross-site claim is made, and this is the first item addressed in future work.

A second limitation concerns statistical resolution. The best configuration misclassified one test image, and every configuration exceeded 0.97 accuracy. At that level there is very little room remaining to distinguish architectures, and because each configuration was trained once, no estimate of run-to-run variance is available. Differences of a few images therefore cannot be separated from stochastic noise, and the finer ordering among the four middle-ranked backbones cannot be resolved by this experiment.

A third concerns dataset composition and labelling. Denim contains 483 images against 575 for most other classes, although the confusion matrix indicates the class was not disadvantaged in practice. Label verification relied on review by people who knew the source material, without a second independent annotator, so no inter-annotator agreement statistic can be reported. Because labels derive from knowledge of the physical roll rather than from perceptual judgement of an image, the risk this introduces is lower than it would be for a perception-based annotation task, but it remains a gap.

A fourth concerns the augmentation policy, which was deliberately mild. The conclusions drawn about the interaction between augmentation and model capacity are conclusions about this specific policy rather than about augmentation in general, and a stronger policy could change both the size and the direction of the effect for individual models.

A fifth concerns the areal-density measurements, which were recorded once per class from a representative sample rather than per image or per roll. Within-class variation in density is therefore not modelled, and the values characterise each category rather than each sample.

Finally, the task addressed is whole-image classification into one of six classes. The system does not localise fabric within a scene, does not segment it, does not detect defects, and cannot express uncertainty or reject an input belonging to none of the six classes. A deployed tool would require at least the last of these, since a production floor contains materials outside the training taxonomy.

7.6 Future Work

The limitations above map onto a programme of work, ordered here by how much each would strengthen the conclusions.

The most valuable next step is cross-site and cross-device evaluation. Collecting imagery from additional mills and additional camera models, and evaluating under leave-one-site-out and leave-one-device-out protocols, is the only way to establish whether the representations learned here generalise or whether they encode site and device characteristics. This would also broaden the range of production variation the dataset represents.

The second concerns statistical rigour. Training each configuration several times from different random initialisations and reporting confidence intervals would allow the architectural ranking to be stated with proper uncertainty, and would determine whether the two small declines observed under augmentation are genuine or artefacts of single-run estimation. Testing stronger augmentation policies, including rotation, random resized cropping and mixing-based methods, would establish whether the modest and uneven gains reported here are a property of augmentation on this task or a consequence of a deliberately cautious policy.

The third concerns architecture. Adding transformer baselines would test whether attention-based designs offer any advantage on texture-dominated imagery, a question this study raises but does not answer, since ConvNeXt-Tiny borrows transformer design choices without employing attention. In the opposite direction, knowledge distillation from EfficientNetV2-B0 into MobileNetV2 would test whether the accuracy gap between the two undominated models can be narrowed without incurring the latency cost of the larger one, which is the most practically useful open question arising from the benchmark.

The fourth concerns interpretability. The Grad-CAM analysis presented here is qualitative and rests on representative samples. Segmenting the fabric region in a subset of test images and computing the fraction of activation mass falling inside that region would convert the check into a reportable number that could be compared across architectures and tracked as the dataset changes.

Finally, moving from a benchmark to a usable instrument would require confidence calibration and an explicit rejection option, so that samples about which the model is uncertain are routed to an operator rather than guessed. The near-perfect AUROC values reported here indicate that a suitable threshold exists. Extending the pipeline to detection rather than whole-image classification, following the formulation of Mao et al. [11], would remove the assumption that the material fills the frame. Recording per-image rather than per-class areal density, and extending the taxonomy to finer grades within a single weave family, would each bring the system closer to the decisions actually taken on a production floor.

Bibliography

[1] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[2] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[3] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description, and segmentation," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 65–94, 2016.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3320–3328, 2014.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

[10] M. A. I. Hussain, B. Khan, Z. Wang, and S. Ding, "Woven fabric pattern recognition and classification based on deep convolutional neural networks," *Electronics*, vol. 9, no. 6, p. 1048, 2020.

[11] M. Mao, A. Lee, and M. Hong, "Efficient fabric classification and object detection using yolov10," *Electronics*, vol. 13, no. 19, p. 3840, 2024.

[12] M. Seçkin, A. Ç. Seçkin, P. Demircioglu, and I. Boğrekçi, "Fabricnet: A microscopic image dataset of woven fabrics for predicting texture and weaving parameters through machine learning," *Sustainability*, vol. 15, no. 21, p. 15197, 2023.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

[14] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10096–10106, 2021.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022.

[18] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005.

[22] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1–2, pp. 61–81, 2005.

[23] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated fabric defect detection: A review," *Image and Vision Computing*, vol. 29, no. 7, pp. 442–458, 2011.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

[28] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3606–3613, 2014.

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019.

[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[31] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[33] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283, 2016.

[36] S. Mukherjee, "The annotated ResNet-50: Explaining how ResNet-50 works and why it is so popular," aug 2022. Accessed 20 August 2026.

[37] Y. Zhang, A. Xu, D. Lan, X. Zhang, J. Yin, and H. H. Goh, "ConvNeXt-based anchor-free object detection model for infrared image of power equipment," *Energy Reports*, vol. 9, pp. 1121–1132, 2023.

[38] R. Beg, R. K. Pateriya, and D. S. Tomar, "ACMFNN: Design of an augmented convolutional model for intelligent cross-domain malware localization via forensic neural networks," *Research Square Preprint*, 2022. doi:10.21203/rs.3.rs-2055847/v1.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[42] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining and Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.

[43] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.

[44] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.