Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260503120
Volume 5, Issue3 (September-December 2026), PP: 1008-1013.
www.indjcst.com
ISSN No: 2583-5300

DefectTex AI: Textile Defect Classification, Severity Analysis and Fabric Quality Recommendation System

Prasanna Kumar Naik R[1], Chethan Kumar G S[2], Swarnalatha[3], Preethi K.P[4], Meghana R. Patil[5]

[1,2,3,4],Assistant Professor, Department of Master of Computer Application, U.B.D.T College of Engineering Institute of Engineering, Davanagere, Visvesvaraya Technological University (VTU), Belagavi, Karnataka, India.

[5] Student, Department of Master of Computer Application, U.B.D.T College of Engineering Institute of Engineering, Davanagere, Visvesvaraya Technological University (VTU), Belagavi, Karnataka, India.

To Cite this Article: *Prasanna Kumar Naik R[1], Chethan Kumar G S[2], Swarnalatha[3], Preethi K.P[4], Meghana R. Patil[5], "DefectTex AI: Textile Defect Classification, Severity Analysis and Fabric Quality Recommendation System", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 1008-1013.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Fabric inspection is an essential quality-control activity in the textile industry because defects formed during yarn preparation, weaving, finishing, storage, or handling can reduce the appearance, usability, and commercial value of fabric. Conventional inspection is commonly performed through visual examination by trained personnel. Although this approach remains useful, the inspection process can be time-consuming and its results may vary with the experience, attention, and working conditions of the inspector. This paper presents DefectTex AI, a web-based prototype developed to support fabric inspection through machine-learning-based image classification and quality assessment. The system accepts a fabric image through a web interface and processes it using an ensemble approach combining Random Forest and XGBoost. The project dataset contains five categories: Good, Hole, Objects, Oil Spot, and Thread Error. After classification, the application displays the predicted defect category and provides severity information, a fabric-quality score, a recommendation, and an Accept or Reject outcome. The application is implemented using Python and the Flask framework, providing a simple interface for submitting images and viewing analysis results. The project evaluation table reports 88.48% accuracy for Random Forest, 92.66% for XGBoost, and 92.77% for the ensemble. The ensemble also records 90.95% precision, 92.77% recall, and a 91.18% F1-score in the reported results. These values indicate that the ensemble achieved the highest accuracy among the three evaluated models in the project results. The main contribution of DefectTex AI is the integration of defect classification and quality-related outputs in one web-based workflow, intended to assist users in making more consistent fabric-inspection decisions. The prototype is designed as a decision-support aid rather than a replacement for professional inspection. Further validation using a clearly documented independent test set, class-wise evaluation, and expert-verified severity rules is required before deployment in an industrial environment.*

Keywords: *Textile inspection, fabric defect classification, Random Forest, XG Boost, ensemble learning, severity assessment, quality recommendation.*

I. INTRODUCTION

Fabric defects may arise during yarn preparation, weaving, finishing, storage, or handling. Examples include holes, oil spots, foreign objects, and irregular threads. Identifying these conditions is necessary before fabric is accepted for further processing or use. Manual inspection remains important, but the process depends on the inspector's attention and experience. Earlier research has therefore investigated image processing, texture analysis, and automated visual inspection as ways to support fabric quality control [1], [2].

DefectTex AI is a software-based inspection aid. A user submits a fabric image through a web interface. The application predicts one of five project-defined categories and presents severity information, a quality score, recommendations, and an Accept or Reject result. The system is implemented using Python and Flask and applies Random Forest and XGBoost as its classification components.

The contribution of this work is not a claim that the proposed classifier is superior to every earlier method. Instead, the project integrates a machine-learning classifier with a web-based workflow and additional quality-related outputs. This application-level combination distinguishes the project from earlier studies that concentrate mainly on texture analysis, defect segmentation, or object detection.

II. RELATED WORK

Research on fabric inspection has progressed from frequency-domain and texture-based analysis to machine-learning and deep-learning approaches. Chan and Pang used Fourier analysis to study changes in the spatial-frequency structure of fabric and extracted parameters for identifying structural defects [1]. Bodnarova et al. designed optimized two-dimensional Gabor filters to distinguish defective from non-defective textile texture [2]. Kumar presented a feed-forward

neural-network approach for local defect segmentation and used neighbouring pixel information with dimensionality reduction [3].

Cho et al. described a real-time vision-based fabric inspection system intended for industrial web materials [4]. Serdaroglu et al. combined wavelet transformation with independent component analysis to identify defective textile image regions [5]. Kumar's survey grouped computer-vision fabric-defect methods into statistical, spectral, and model-based categories and discussed their respective limitations [6]. Ngan, Pang, and Yung later reviewed a wider range of automated fabric-inspection approaches, including statistical, spectral, learning, structural, hybrid, and motif-based methods [7].

More recent work has applied deep-learning detectors. Jia et al. improved Faster R-CNN through transfer learning and architectural changes for fabric-defect detection [8]. Dhamal et al. proposed a Mask R-CNN-based framework with a ResNeXt-101 backbone and feature-pyramid components [9]. These detection-oriented approaches differ from the present project, which uses a Random Forest and XGBoost ensemble for five-category image classification and focuses on delivering a compact web-based quality-assessment workflow. The Random Forest and XGBoost components are grounded in established tree-ensemble methods [10], [11].

III. COMPARISON WITH EARLIER APPROACHES

Table I compares representative earlier studies with the scope of DefectTex AI. The comparison is based on the methods and objectives described in the cited publications; it is not a reimplementation or a head-to-head benchmark.

Earlier work	Main approach reported	Difference from DefectTex AI
Chan and Pang [1]	Fourier-domain analysis of fabric structure and frequency characteristics.	DefectTex AI uses supervised tree-based classification and adds severity, quality score and recommendations.
Bodnarova et al. [2]	Optimized Gabor filters for detecting flawed textile texture.	DefectTex AI uses an RF–XGBoost ensemble rather than a Gabor-filter detection pipeline.
Kumar [3]	Neural-network segmentation of local defects using neighbouring-pixel features and PCA.	DefectTex AI classifies complete uploaded samples into five application classes and provides a quality summary.
Jia et al. [8]	Transfer learning with an improved Faster R-CNN detector.	DefectTex AI uses tree-based ML rather than a CNN-based object detector; its goal includes a Flask-based inspection result.
Dhamal et al. [9]	Mask R-CNN / Detectron2 framework for defect detection.	DefectTex AI is a lighter classification-oriented prototype with integrated severity, score and accept/reject output.

The key project-level difference is the integration of five-class classification and post-classification quality information in one web application. The comparison does not establish that DefectTex AI is more accurate or faster than the cited systems, because the datasets, labels, evaluation protocols, and tasks are different.

IV. PROBLEM STATEMENT AND OBJECTIVES

The problem this project addresses is the need for a straightforward software tool that can classify visible fabric defects from uploaded images and present quality-assessment information in an understandable way. Manual inspection can be more time consuming and may also produce differences between inspectors. So there must be a system that should reduce the inspection of manual work and also easily detect the defect categories of fabric

So that the user can easily take the decision to accept or reject the fabric easily it recommend of the fabric . So this reduces the more time and more efforts of the fabric analyst.

The objectives are: (1) train and evaluate a machine-learning model across 5 fabric categories; (2) integrate the trained classifier into a Flask application; (3) provide an image-upload workflow that displays the predicted category; (4) present a fabric quality score and an accept/reject decision using the application's built-in rules; and (5) provide recommendations based on the analysis results.

V. PROPOSED SYSTEM AND METHODOLOGY

The processing sequence begins with image submission through the web interface. Flask receives the image and passes it through the image-preparation stage used by the application. The prepared input is supplied to the Random Forest and XGBoost classification components. The ensemble output determines the predicted category. The application then uses its implemented severity and quality-assessment logic to prepare the result page. The final page presents the predicted class, severity information, quality score, recommendation, and Accept or Reject status.

A. Dataset and Classes

The current project dataset is described as a Kaggle textile-image dataset organized into five class folders. The labels and their meanings are shown in Table II. Exact image counts per class, dataset version, and train/validation/test partition details were not present in the project document available for this revision; these values are therefore not invented here.

Class label	Description
Good	Fabric sample without a labelled defect.
Hole	Visible hole or opening in the fabric.
Objects	Foreign object or material visible on the fabric.
Oil Spot	Oil stain or spot on the fabric surface.
Thread Error	Thread-related irregularity.

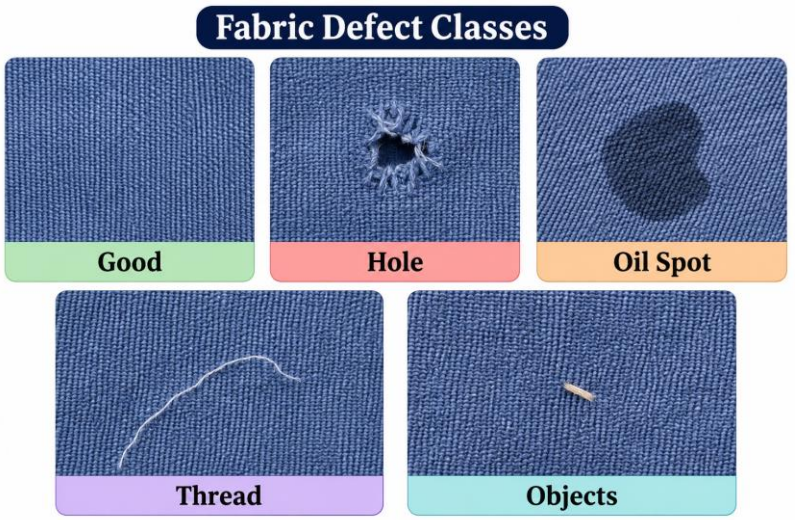


Fig. 1. Representative images for the five fabric categories used in the project.

B. Classification Algorithms

Random Forest constructs a collection of decision trees and combines their outputs to produce a class prediction [10]. XGBoost is a gradient-boosting method in which trees are added sequentially to improve the model's objective [11]. DefectTex AI uses both algorithms as an ensemble. The precise probability-combination rule or weighting parameter was not specified in the available implementation description; accordingly, this paper does not assume a weighted-average formula. The final prediction is the class returned by the implemented ensemble.

C. Severity, Quality Score and Recommendation

After classification, the application presents severity information and a fabric-quality score. It then displays a recommendation and an Accept or Reject outcome. The project document confirms that these modules are implemented, but it does not provide the exact rule thresholds, weights, or equations used internally. To avoid describing a formula that may not match the source code, this paper reports these components functionally rather than introducing an unverified severity or quality equation.

D. System Architecture

Defect Tex AI System Architecture

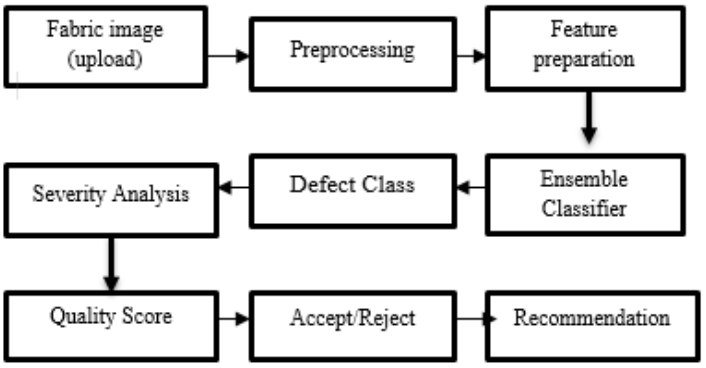


Fig. 2. DefectTex AI system architecture supplied with the project materials.

The architecture shows the user-facing image submission, Flask application, image-processing stage, ensemble classifier, five-class output, severity and quality assessment, and final recommendation. The diagram is retained as a project illustration; implementation-specific details should be checked against the final source code before publication.

E. Processing Flowchart

Defect Tex AI: proposed processing pipeline

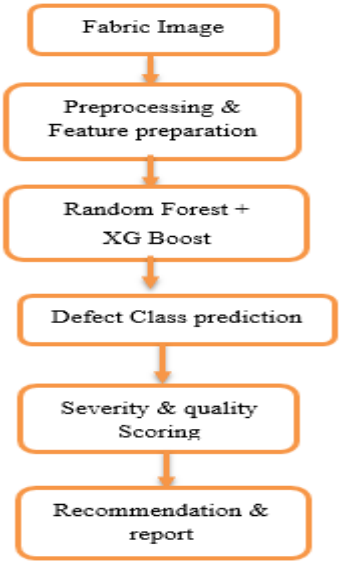


Fig. 3. Processing flow of DefectTex AI, redrawn from the workflow described in the project paper.

VI. EVALUATION METRICS

For a multi-class classification task, accuracy is calculated as the number of correctly classified samples divided by the total number of evaluated samples. Precision, recall, and F1-score are calculated for each class using a one-versus-rest interpretation. TP, FP, and FN refer to true positives, false positives, and false negatives for the class being evaluated.

$$\text{Accuracy (\%)} = (\text{Number of correct predictions} / \text{Total number of evaluated samples}) \times 100$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

For a five-class confusion matrix, the diagonal cells represent correct predictions and the off-diagonal cells represent class confusions. If aggregate precision, recall, and F1-score are reported, the averaging method (macro, micro, or weighted) should also be stated. The averaging method was not identified in the available project results table.

VII. RESULTS AND DISCUSSION

Table III reproduces the numerical results recorded in the project paper. These are project-reported values and should be checked against the final evaluation script and saved model before formal publication.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Random Forest	88.48	87.60	88.48	87.29
XGBoost	92.66	90.70	92.66	91.08
Ensemble	92.77	90.95	92.77	91.18

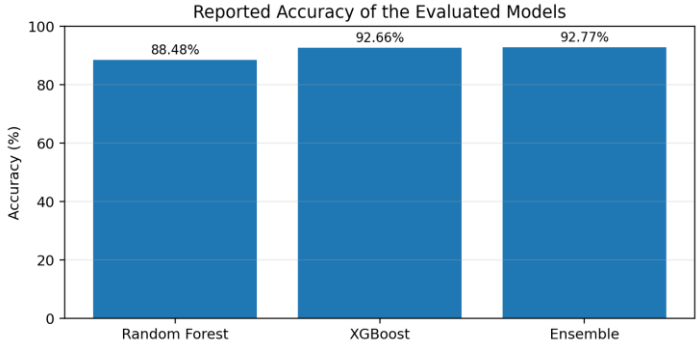
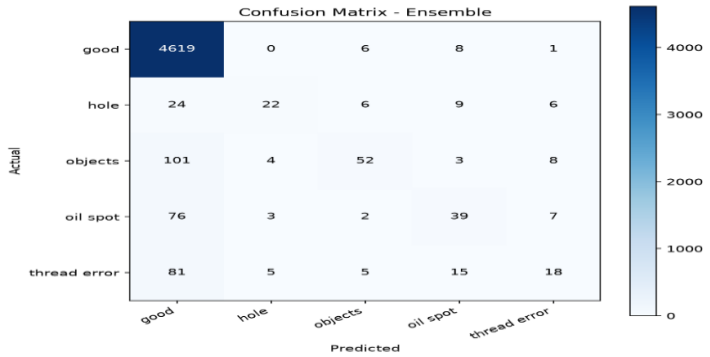


Fig. 4. Accuracy values reported in the project results table.

According to the recorded table, the ensemble has the highest reported accuracy at 92.77%, which is 0.11 percentage points above XGBoost and 4.29 percentage points above Random Forest. The reported ensemble F1-score is 91.18%. Precision is 90.95%, while recall is 92.77%. These values suggest that the ensemble performed best among the three listed models in the recorded evaluation. However, the small difference between XGBoost and the ensemble should not be treated as statistically meaningful without repeated trials or a suitable significance analysis.

The current documentation does not include the total number of test images, per-class support, data split ratio, random seed, class-wise metrics, or the numerical confusion matrix. These details are important for reproducibility and should be added after checking the model evaluation output. No fabricated confusion matrix or class-wise scores are included in this draft.

Confusion matrix:



Rows in confusion matrix represents actual classes and columns represents predicted classes. The diagonal contains correct classifications, while off-diagonal entries show confusions.

VIII. COMPARISON WITH MANUAL INSPECTION

Aspect	Manual inspection	DefectTex AI
Input	Fabric is visually examined by an inspector.	A fabric image is submitted through the web application.
Classification	Based on the inspector's visual judgement.	Predicted by the RF–XGBoost ensemble across five classes.
Consistency	May vary with experience, attention and viewing conditions.	Same model applies the same learned classification process to each input.
Severity and quality	Assessed by the inspector or a separate grading procedure.	Application displays severity and a quality score.
Final output	Inspector decides whether to accept or reject.	Application presents a recommendation and Accept / Reject outcome.
Human role	Inspector performs the assessment.	Inspector reviews the system output; the application is an aid, not a replacement.

This is a qualitative workflow comparison, not a measured performance comparison. A valid time or accuracy comparison with human inspectors would require the same labelled image set, multiple inspectors, a defined inspection protocol, and recorded time measurements.

IX. LIMITATIONS

The current prototype is limited to the five categories represented in its dataset. Its predictions depend on the quality and coverage of the training images. The available paper records do not specify class-wise image counts, the independent test partition, the ensemble-combination rule, or the exact severity and quality-score equations. In addition, the current description concerns image-based software inspection and does not establish real-time factory-line integration. These limitations should be considered when interpreting the reported performance.

X. CONCLUSION AND FUTURE WORK

This paper presented DefectTex AI, a web-based textile inspection prototype that combines Random Forest and XGBoost for five-category fabric image classification. The application extends the classification result with severity information, a fabric-quality score, recommendations, and an Accept or Reject outcome. The project results table reports 92.77% ensemble accuracy, with 90.95% precision, 92.77% recall, and 91.18% F1-score. The system's main contribution is the integration of classification and quality-related outputs in a single Flask-based workflow. Future work includes documenting the dataset and split in full, validating the reported metrics on a held-out test set, adding the actual confusion matrix and class-wise results, and validating severity and quality rules with textile-domain experts.

REFERENCES

1. C.-H. Chan and G. K. H. Pang, "Fabric defect detection by Fourier analysis," IEEE Transactions on Industry Applications, vol. 36, no. 5, pp. 1267–1276, 2000, doi: 10.1109/28.871274.
2. A. Bodnarova, M. Bennamoun, and S. Latham, "Optimal Gabor filters for textile flaw detection," Pattern Recognition, vol. 35, no. 12, pp. 2973–2991, 2002, doi: 10.1016/S0031-3203(02)00017-1.
3. A. Kumar, "Neural network based detection of local textile defects," Pattern Recognition, vol. 36, no. 7, pp. 1645–1659, 2003, doi: 10.1016/S0031-3203(03)00005-0.
4. C.-S. Cho, B.-M. Chung, and M.-J. Park, "Development of real-time vision-based fabric inspection system," IEEE Transactions on Industrial Electronics, vol. 52, no. 4, pp. 1073–1079, 2005, doi: 10.1109/TIE.2005.851648.
5. A. Serdaroglu, A. Ertuzun, and A. Ercil, "Defect detection in textile fabric images using wavelet transforms and independent component analysis," Pattern Recognition and Image Analysis, vol. 16, no. 1, pp. 61–64, 2006, doi: 10.1134/S1054661806010196.
6. A. Kumar, "Computer-vision-based fabric defect detection: A survey," IEEE Transactions on Industrial Electronics, vol. 55, no. 1, pp. 348–363, 2008, doi: 10.1109/TIE.1930.896476.
7. 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, doi: 10.1016/j.imavis.2011.02.002.
8. Z. Jia, Z. Shi, Z. Quan, and M. Shunqi, "Fabric defect detection based on transfer learning and improved Faster R-CNN," Journal of Engineered Fibers and Fabrics, vol. 17, 2022, doi: 10.1177/15589250221086647.
9. S. K. Dhamal, C. Joshi, P. Chakrabarti, and D. S. Chouhan, "An automated framework for fabric defect detection in textile inspection," Journal of Information Systems Engineering and Management, vol. 10, no. 46s, pp. 353–367, 2025, doi: 10.52783/jisem.v10i46s.8815.
10. L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
11. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.