Two-Stage Preprocessing Approach for Background Normalization and Defect Segmentation on Denim Fabric Image Analysis

Thamiris Gire Zine Neves and Diego Minatel and Alexandre da Silva Saito and Matheus de Freitas Oliveira Baffa
National Service for Industrial Training (SENAI-SP)
São Caetano do Sul, SP - 09510-200
Email: thamiris.neves@sp.senai.br, diego.minatel@sp.senai.br, saito@sp.senai.br, matheus.baffa.3@sp.senai.br

***Abstract*—The textile industry is one of the oldest and largest industries in the world. Detecting defects early in manufacturing enables cost reduction through fast intervention and fabric correction. Vision-based software can automatically identify defects in fabrics during pre- or post-production. However, this detection is challenging due to denim fabric color and texture variations, making some defects less visible than others, such as double thread and cut weft. A robust preprocessing approach can reveal intrinsic features of the data. Therefore, this paper investigates background normalization and defect segmentation to highlight features and allow better differentiation during image analysis. We propose a two-stage pipeline using a combination of filters to reduce background information and merge it with the output of a defect segmentation process based on thresholding and morphological operations. To test this hypothesis, we extracted 100 features from the processed and unprocessed images and benchmarked them using several machine learning algorithms, including a deep learning model. With the processed data, we achieved up to a 7% increase in accuracy for the evaluated metrics compared to the unprocessed experiments. This study demonstrates the feasibility of detecting defective regions in denim fabric using computer vision techniques.**

I. Introduction

Brazil plays an essential part in the world exporting scenario of denim fabric — the primary fabric used in jeans production [1]. Also, it is classified as the fifth largest textile industry globally and the fourth country in the clothing segment [2]. In 2021, the cost of investments in the sector was BRL 4,9 billion, and USD 1,14 billion were exported [3], illustrating the textile industry's significant role in the country's economy and highlighting the potential for waste reduction. In general, the textile industry has a variety of segments that require optimization, and it has ample capacity for deploying systems based on artificial intelligence [4]. However, some of its processes, such as quality inspection, still require manual and visual inspection. [5].

This process, which is often inaccurate and time-consuming, leads to waste and reduces the fabric quality. For example, we followed, *in loco*, the production process in a Brazilian factory. We noted that defects caused by environmental conditions (*e.g.*, humidity) or the quality of the raw materials are identified only ten to fifteen days after manufacturing. This late inspection can represent a significant financial loss in the order of millions of dollars.

Based on this problem, researchers have been studying methods to minimize the waste of industry resources to increase the competitiveness of enterprises [6]. Therefore, researchers have been developed to identify and classify fabric defects [6]–[9]. A vision-based system is made up of an image acquisition, a feature extraction, and a classification module [6].

A vision-based system needs to carefully normalize images, reduce unwanted information, and perform a suitable preprocessing pipeline to identify and classify these defects. These steps involve restructuring and highlighting important image details for analysis, segmenting the region of interest, and extracting significant descriptive features. Additionally, a robust preprocessing pipeline can reveal intrinsic features of the data, highlighting patterns in images of both defective and non-defective fabric [5].

This analysis can be challenging in the case of denim fabric. This type of fabric may be dyed in shades of blue and other colors, such as green and black. While the fabric's color does not impact the detection of defects, some datasets might mistakenly present defective fabric in specific colors. Therefore, background normalization is important for defect detection as it helps eliminate such errors.

Another aspect of this problem is the different types of denim defects. Some defects are easy to classify, such as soft weft, which is noticeable visually. On the other hand, further defects, such as double thread and cut weft, require more effort to identify. To minimize these differences, a segmentation method that highlights these defects can significantly enhance the performance of computer vision systems, making it easier to detect and classify the various kinds of defects accurately.

This paper addresses the challenges of background normalization and defect segmentation in denim fabric images for use in the textile industry. We divided our proposed preprocessing pipeline into two parts: one part reduces background impact by normalizing colors through a combination of filters, and the other part segments and highlights fabric defects and features. To evaluate the impact of the preprocessing step on denim fabric images, we conduct a comparative study using image classification based on feature engineering. The findings show an improvement of 7% accuracy when applying our preprocessing step.

The main contribution of this paper is the presentation of a robust preprocessing method based on a two-stage approach. This method can be deployed as part of a vision system to detect defects in factories producing denim fabric, enabling the minimization of waste and increased profits.

II. Related Papers

In 1994, researchers presented the Fabric Defect Identification and Classification System (FDICS), a vision-based system for identifying and classifying fabric defects, as a solution for the industry [6]. The problem of fabric defects is so recurring that 30 years later, researchers are still proposing solutions to improve detection systems. In their paper, the authors identified the need for preprocessing images through feature extraction, including mean, variance, skewness, kurtosis, and entropy. They used the spatial gray-level dependence method (SGLDM) for the algorithm. For classification, the system employs traditional algorithms such as Bayesian classification.

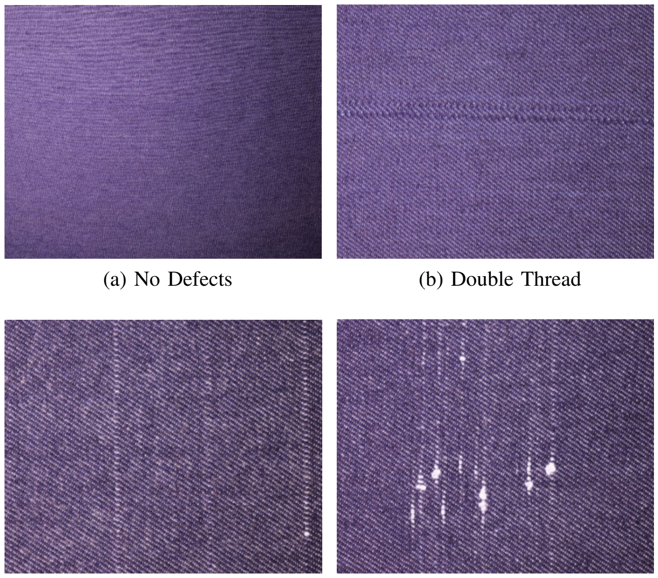
Later, in 2013 and 2014, researchers started exploring artificial neural networks to develop a real-time fabric defect detection [7], [8]. For instance, in Çelik et al. [7], the authors developed an automatic machine vision system containing image acquisition hardware and image processing software. The software incorporates a defect detection algorithm based on wavelet transform, double thresholding binarization, and morphological operations. In the classification stage, they utilize an algorithm based on gray level co-occurrence matrix (GLCM) with a feed-forward neural network. This approach achieved an accuracy of 93.4% in classifying defect and defect-free images of fabrics and 96.3% in identifying five defect classes.

Similarly, Song et al. [8] introduced a non-visual surface texture sensor design that mimics human touch-active texture perception. After data collection, they applied the Fourier Transform and classified the signals using a radial basis function (RBF) neural network based on an unsupervised k-means clustering algorithm.

Recently, in 2021, convolution neural networks (CNN) became more popular in solving the problem of fabric defect identification [9]–[13]. Specifically, Talu, Hanbay & Varjov [9] developed a CNN architecture for this purpose and implemented it in a real-time production on a loom. Furthermore, they applied a Fast Fourier Transform defective patch capture (DPC) algorithm to enhance and suppress defect-free areas. With this methodology, they obtained 96,5% accuracy. Whereas Zheng et al. [10] introduce a novel model for YOLOv5 called SE-YOLOv5, which contains in its structure an attention mechanism and a different activation function (ACON-CSP).

III. Denim Dataset Description

We use a dataset from the "Workshop on Mathematical Solutions for Industrial Problems" (Workshop *de Soluções Matemáticas para Problemas Industriais*) promoted by the Center for Mathematics Science Applied to Industry (CeMEAI) of the University of São Paulo. A Brazilian textile company specializing in denim production provided the dataset, which includes images of three distinct types of defects in denim fabric (cut weft, soft weft, and double thread) as well as images without any defects, as shown in Fig. 1.


(a) No Defects (b) Double Thread

(c) Cut Weft (d) Soft Weft

Fig. 1. Examples from the dataset.

We organized and balanced the dataset for binary classification by categorizing the classes into defect and non-defect. The dataset comprises 1165 patches extracted from 25 distinct fabric rolls, with 583 patches containing defects and 583 patches representing defect-free fabric.

IV. Two-Stage Preprocessing Approach

Denim fabric exhibits distinctive characteristics. Its pattern typically features threads arranged diagonally. While some defects, like soft weft, are easily detectable, others, such as cut weft and double thread, are less visible. Additionally, denim fabrics can be manufactured in various colors but mostly in blue, which can confuse the machine, as it might expect certain colors to indicate defects.

Therefore, we propose a two-step pipeline to reduce background influence in denim fabric, which consists of standardizing it and highlighting defects and details using a segmentation process. The pipeline, illustrated in Figure 2, starts by receiving the image and extracting the green channel. Then, two paths are followed to achieve the preprocessing goal.

First, to normalize the background, we apply the bilateral filter followed by the top-hat filter. Meanwhile, we segment the region of interest containing defects and other details. This segmentation involves applying thresholding and an opening morphology operation. We improve the segmentation quality

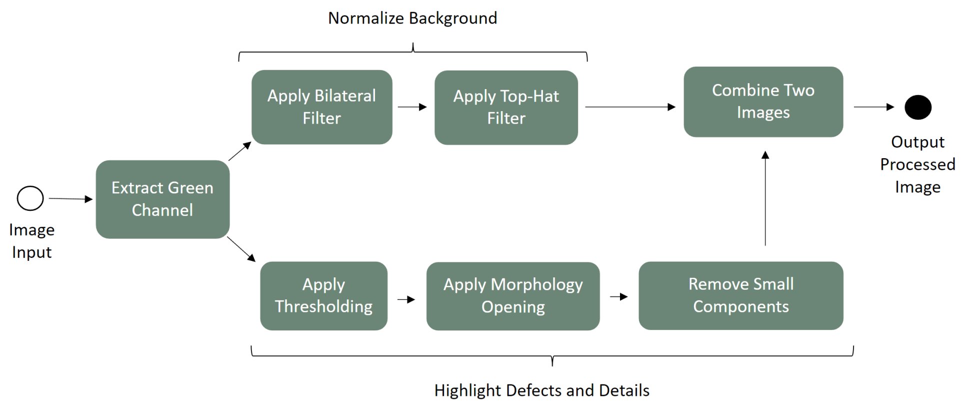


Fig. 2. An overview of the two-stage preprocessing approach for background normalization and defect segmentation.

by removing small connected components. In the end, we combine these operations to produce a unified output.

The resulting image has its background normalized, reducing the impact of denim fabric color. Defects and potential detailed points are highlighted. In the remainder of this section, we describe this pipeline in detail, justifying each technique.

A. Diving Into the Detailed Process

Right at the beginning of the process, we load the denim patch and extract its green channel to use it in grayscale. We choose to extract the green channel instead of simply converting it to grayscale to preserve details and contrast that would be lost with traditional grayscale conversion and other histogram equalization methods.

Besides, the green channel is known to provide a good balance between luminance information and intensity variation, which is particularly relevant in denim fabric analysis, where color differences can indicate defects. Also, denim fabric often contains a significant amount of blue components, which can interfere with defect detection when using grayscale conversion alone. By focusing on the green channel, we can reduce the influence of these potentially misleading blue components. Figure 3 illustrates the difference between the green channel and the traditional grayscale conversion.

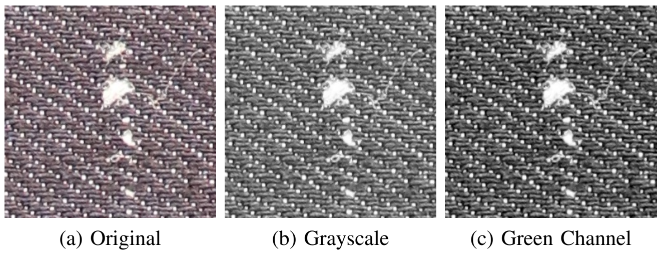


Fig. 3. Differences between the grayscale and the green channel conversion on denim fabric.

After extracting the green channel, we proceed with background normalization to reduce the impact of fabric color on defect detection. To achieve this, we first apply the bilateral filter (Fig. 4b). We chose this filter based on its ability to smooth the image while preserving edges, which is important for maintaining the sharpness of defect boundaries while reducing the background texture variation.

Along with the bilateral filter, we apply the Top-Hat Filter (Fig. 4c). This filter is particularly effective in highlighting small, subtle details in the image, such as defects in the denim fabric. Its operation helps to enhance the contrast between the fabric details and the background, normalizing across various types of denim colors and making them more prominent and easier to detect in the subsequent steps of the analysis.

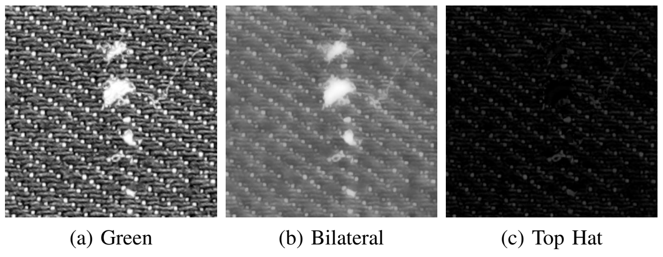


Fig. 4. Stacking the bilateral filter and top hat filter over the green channel.

The top hat filter washes away defective regions in the image, leaving an empty space there. Therefore, another pipeline running in parallel is necessary to highlight the defects themselves. To achieve this, we developed a second pipeline running exclusively for this purpose.

The second pipeline part receives the green channel and applies thresholding, using a threshold of 127 (Fig. 5a). This step produces an image where pixel values above the threshold are set to white, indicating potential defect regions, while pixel values below the threshold are set to black.

In the binary image, we observed not only the regions with defects but also many other white spots corresponding to denim fabric threads. To remove these white spots, we utilized the opening morphology operation (Fig. 5b). This operation involved applying erosion twice, which removed most of the uninteresting information, followed by two applications of dilation, restoring the defective areas to their original size.

Some small noise was still present in the image. We avoided using a stronger erosion to prevent the loss of important information and the correct shape of the defect. Therefore, we included an improvement step that involved removing small connected components (Fig. 5c). In this step, we measured the mass of each small component in the image, and any component with a density of less than 44 pixels was removed. This threshold was determined empirically to eliminate noisy components while retaining important information in the image.

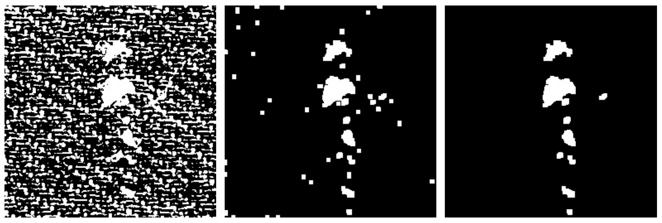

(a) Threshold (b) Morphology (c) Small Connected Components

Fig. 5. Stacking the bilateral filter and top hat filter over the green channel.

Finally, after the two processes, we merge the segmentation mask with the normalized background. This merging step combines the enhanced defect regions from the segmentation process with the background-normalized image, resulting in a final image where defects are highlighted against a uniform background. Figure 6 illustrates the two-stage preprocessing applied to both defective (Fig. 6a - 6b) and non-defective fabrics (Fig. 6c - 6d).

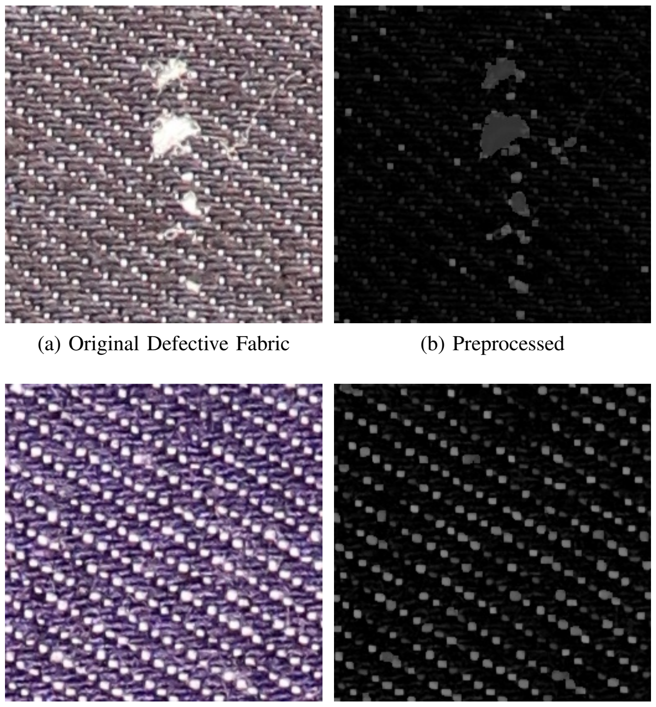
V. Feature Extraction and Benchmark Methodology

Following the preprocessing of the images, we evaluated their performance against the unmodified images. Our goal was to determine which features worked best for each dataset and assess the impact of the preprocessing pipeline on training with both traditional and deep learning methods.

First, we extract a large number of features, such as texture and shape, from both datasets. For this purpose, we utilized the Python frameworks PyRadiomics [14] and Mahotas [15] to extract 217 features. These features include 18 first-order features, nine shape features, 24 gray-level co-occurrence matrix (GLCM) features, 16 gray-level run length matrix (GLRLM) features, 16 gray-level size zone matrix (GLSZM) features, five neighborhood gray-tone difference matrix (NGTDM) features, 14 gray-level dependence matrix (GLDM) features, 25 Zernike moments features, 36 local binary pattern (LBP) features, and 54 thresholding adjacency statistics (TAS) features.

After extracting the features, we applied standard scaler normalization to scale each feature to a similar range. This step is important for ensuring that each feature contributes equally to the classification process and prevents features with larger scales from negatively impacting the training process.

Additionally, we used the SelectKBest algorithm to detect and select the best 100 features ($k = 100$). This algorithm


(a) Original Defective Fabric (b) Preprocessed

(c) Original Non-Defective Fabric (d) Preprocessed

Fig. 6. Example of defective and non-defective denim fabrics after preprocessing.

utilizes the chi-squared (χ^2) statistical test to evaluate the independence between each feature and the class labels.

We evaluated two approaches to detect patterns in the feature dataset: traditional machine learning algorithms and a deep fully-connected neural network (FCNN). Our goal was to identify the best classifier for the problem. The traditional machine learning benchmark used nine algorithms: support vector machine (SVM) with linear, radial, and polynomial kernels; decision tree (DT); k-nearest neighbors (KNN); stochastic gradient descent (SGD); random forest (RF); extreme gradient boost (XGBoost); and multilayer perceptron (MLP). These algorithms were developed using the Scikit-Learn framework with their default parameter values, except for the KNN, where the k value was set to 86, and the max iteration parameter of the MLP was set to 1000.

The FCNN was developed using 12 layers: one input layer, ten hidden layers, and one output layer. The input layer had 100 neurons corresponding to the number of features selected previously. These neurons were fully connected to the subsequent hidden layers, each containing 86 neurons and using the rectifier linear unit (ReLU) activation function. After each hidden layer, a dropout layer was attached with a probability of 20% to turn off neurons, preventing overfitting. Finally, the output layer had two neurons representing the two classes, utilizing the softmax activation function.

The other hyperparameters used in the FCNN included the Adam optimizer for efficient gradient-based optimization and the categorical cross-entropy loss function for measuring the performance of the classification model. We ran the model for 500 epochs, as this was sufficient for convergence, and used a batch size of 128 to balance the training stability and computational efficiency. Figure 7 illustrates the customized FCNN architecture developed for this work.

VI. Experimental Design and Results

The development and experiments for this work were conducted using a dedicated workstation running the Ubuntu Server 22.04 operating system. The workstation was equipped with two Intel Xeon Silver processors, 128 GB of DDR4 memory, and one NVIDIA Tesla T4 graphics card. We utilized the Python programming language, specifically version 3.10.12, along with the TensorFlow 2.10, Scikit-Learn 1.3.2, OpenCV 4.9, Mahotas 1.4.15, and PyRadiomics 3.1.0 frameworks.

The experiments followed the 10-fold cross-validation protocol, which involved separating the dataset into ten parts. Each iteration used one part for validation, while the remaining $k - 1$ parts were used for training. To ensure consistent separability across all folds, we defined a random seed of 1337.

We used five evaluation metrics to measure the performance of each algorithm over the two classification strategies: accuracy (Acc), sensitivity (Sens.), specificity (Spec.), area under the curve (AUC), and precision (Prec.). The accuracy and the AUC provide us with the overall performance of the method, while the sensitivity measures the ability of the model to identify defective instances correctly. On the other hand, the specificity indicates the model's ability to identify the non-defective instances correctly. Finally, precision measures the proportion of correctly identified defective instances among all instances classified as defective by the model.

A. Results

The FCNN for the processed dataset obtained the best results in our experiments. Table I contains a detailed report of the per-fold performance of the network. Among the folds, the FCNN achieved the highest accuracy of 98.28% on fold nine, 97.41% on fold ten, and 96% on folds two, four, and six. The lowest results were 92.24% on fold seven and 93.16% on folds one and three. The FCNN achieves an overall accuracy of 95.53%, with an overall sensitivity of 95.53% and an AUC of 98.31%.

TABLE I
Results obtained from the FCNN for the processed dataset.

Fold	Acc	Sens	Spec	AUC	Prec
1	93.16%	93.13%	93.13%	96.04%	93.39%
2	96.58%	96.57%	96.57%	98.89%	96.65%
3	93.16%	93.15%	93.15%	98.36%	93.22%
4	96.55%	96.55%	96.55%	98.84%	96.77%
5	95.69%	95.69%	95.69%	97.90%	95.81%
6	96.55%	96.55%	96.55%	99.70%	96.55%
7	92.24%	92.24%	92.24%	95.82%	92.56%
8	95.69%	95.69%	95.69%	99.11%	95.70%
9	98.28%	98.28%	98.28%	99.97%	98.28%
10	97.41%	97.41%	97.41%	98.44%	97.43%
Mean	**95.53%**	**95.53%**	**95.53%**	**98.31%**	**95.64%**

In our benchmark of traditional machine learning algorithms (Table II), four methods achieved results exceeding 90% aside from the FCNN: SVM with radial basis, random forest, XGBoost, and MLP. The MLP scored only 1% lower than the FCNN, suggesting that with more data, deeper networks may discern superior patterns, potentially allowing the FCNN approach to attain even higher results without the constraints of the MLP.

DT and KNN reached the lowest results of only 82.45% and 74.46%, respectively, indicating that these algorithms may face difficulties in effectively capturing the complex patterns present in the dataset. This lower performance suggests that DT and KNN may struggle with the inherent complexity of the data, potentially resulting in suboptimal predictive capabilities.

TABLE II
Benchmark with traditional machine learning algorithms for processed dataset.

Classifier	Acc	Sens	Spec	AUC	Prec
KNN	74.46%	74.48%	74.48%	74.48%	76.68%
DT	82.45%	82.45%	82.45%	82.50%	82.45%
SVM P.	85.81%	85.81%	85.81%	86.07%	85.81%
SVM L.	88.21%	88.21%	88.21%	88.53%	88.21%
SGB	89.94%	89.94%	89.94%	90.05%	89.94%
SVM R.	90.11%	90.11%	90.11%	90.52%	90.11%
RF	91.06%	91.06%	91.06%	91.12%	91.06%
XGBoost	92.34%	92.34%	92.34%	92.42%	92.34%
MLP	94.50%	94.49%	94.49%	94.49%	94.49%
FCNN	**95.53%**	**95.53%**	**95.53%**	**98.31%**	**95.64%**

While testing on the unprocessed dataset, FCNN obtained considerably lower results, as illustrated in Table III. The highest performance was observed in folds one and two, which achieved 92.31% accuracy and AUC values of 98.19% and 97.37%, respectively. However, the majority of the folds yielded results below 90%, with fold seven achieving 84.48% and fold three achieving 86.32%. The average performance of the FCNN for this experiment was 88.82%.

TABLE III
Results obtained from the FCNN for the unprocessed dataset.

Fold	Acc	Sens	Spec	AUC	Prec
1	92.31%	92.31%	92.31%	98.19%	92.31%
2	92.31%	92.29%	92.29%	97.37%	92.43%
3	86.32%	86.35%	86.35%	95.06%	86.48%
4	87.93%	87.93%	87.93%	92.15%	87.98%
5	87.93%	87.93%	87.93%	95.72%	88.11%
6	88.79%	88.79%	88.79%	96.22%	88.80%
7	84.48%	84.48%	84.48%	89.88%	84.86%
8	89.66%	89.66%	89.66%	94.31%	89.70%
9	90.52%	90.52%	90.52%	96.65%	90.63%
10	87.93%	87.93%	87.93%	96.05%	87.98%
Mean	**88.82%**	**88.82%**	**88.82%**	**95.16%**	**88.93%**

The benchmark for the unprocessed dataset yielded the best performance from the MLP, achieving an overall accuracy of 90.03%. In contrast to previous experiments where more complex algorithms achieved the best results, SVM with a linear kernel, along with FCNN and XGBoost, achieved results above 85%. Conversely, other algorithms, such as DT and

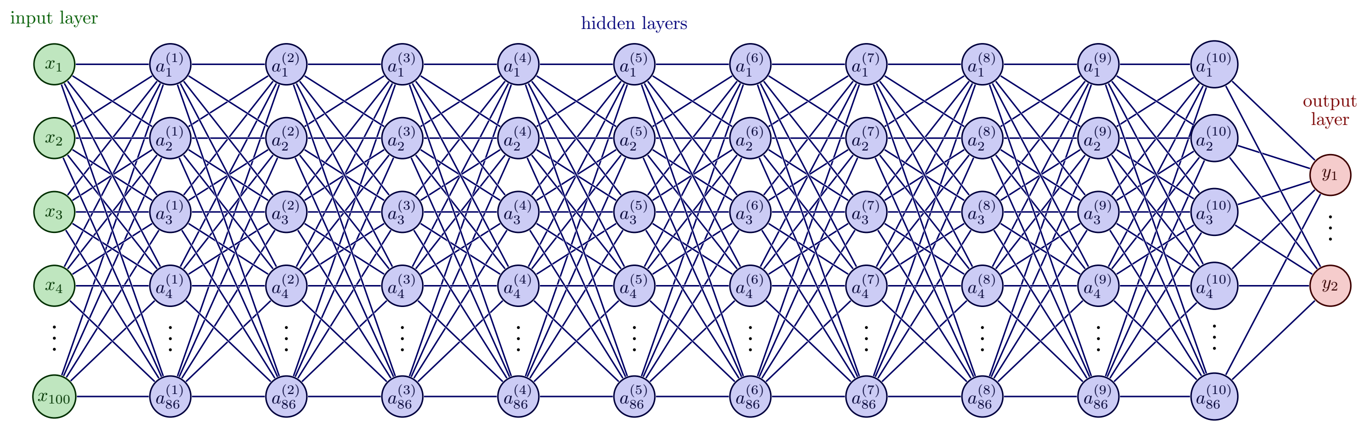


Fig. 7. Proposed FCNN architecture for feature classification.

SVM with a polynomial kernel, reached the lowest results of 75.93% and 79.37%, respectively.

TABLE IV
Benchmark with traditional machine learning algorithms for unprocessed dataset.

Classifier	Acc	Sens	Spec	AUC	Prec
DT	75.93%	75.94%	75.94%	75.94%	76.08%
SVM P.	79.37%	79.37%	79.37%	79.37%	79.90%
KNN	83.66%	83.66%	83.66%	83.66%	83.85%
SVM R.	83.75%	83.75%	83.75%	83.75%	84.07%
SGB	84.27%	84.27%	84.27%	84.27%	84.40%
RF	84.87%	84.87%	84.87%	84.87%	85.05%
XGBoost	86.67%	86.67%	86.67%	86.67%	86.75%
SVM L.	86.85%	86.85%	86.85%	86.85%	87.03%
FCNN	88.82%	88.82%	88.82%	95.16%	88.93%
MLP	**90.03%**	**90.03%**	**90.03%**	**90.03%**	**90.20%**

VII. Discussion

Throughout the results obtained in the experiments, the proposed approach for background normalization and defect segmentation significantly enhanced the differentiation between defective and non-defective fabric. We observed an increase of 7% in accuracy when employing deep neural networks compared to experiments using unprocessed images.

Additionally, deep neural networks achieved superior results compared to traditional machine learning approaches, such as XGBoost and support vector machines. Although the difference was not substantial, both FCNN and MLP demonstrated excellent performance. With the inclusion of more data, we can further expand our analysis using deeper networks, thereby enhancing performance and quality in detecting fabric defects.

Unlike previous works, we propose a preprocessing pipeline based on a two-stage approach and measure its impact on final classification efficiency through a benchmark with traditional and deep machine learning models. Additionally, we approach the classification model using robust feature extraction and a fully connected neural network that achieves competitive results while using fewer computational resources.

A limitation of this study is the need to work with patches of the entire image. The process involves obtaining the entire image, dividing it into patches, extracting features patch-by-patch, and then classifying each patch. This approach can be computationally expensive, particularly when analyzing fabric production in either pre- or post-production settings, where the speed of fabric production may vary.

In addition to this limitation, the patch-based approach offers another perspective on the problem: the detection of error regions within the entire image. By employing the patch approach, we can precisely identify the location of errors and subsequently suggest the type of error, aiding operators in decision-making regarding how to address the issue. This fine-grained analysis enables targeted interventions, potentially streamlining the defect correction process and improving overall efficiency.

The ability to process and detect fabric defects during production, thereby inspecting the fabric's quality, can facilitate real-time corrections while the denim is still being produced. This enables quick intervention, allowing for swift defect rectification and enhancing the overall product quality. Consequently, this approach can contribute to increased company revenue and cost savings by minimizing the production of defective units and optimizing resources.

VIII. Conclusion

This paper introduced a novel two-stage preprocessing pipeline for background normalization, feature enhancement, and defect segmentation for denim fabric images. In industrial settings, quality inspection typically occurs during the later stages of product development. However, by integrating it into early processes, our approach enables cost reduction through early intervention and fabric correction. The proposed preprocessing pipeline resulted in an improvement of over 7% in accuracy compared to the analysis of unprocessed datasets using deep neural networks.

Acknowledgment

The authors would like to thank the National Service for Industrial Training (SENAI-SP) for supporting this research through the Center for Skills Development in Artificial Intelligence (CDC-IA) program. CeMEAI is partially financed by FAPESP, through the process Cepid-CeMEAI 2013/07375-0.

References

[1] JCR-VIS Credit Rating Company Limited, "Denim industry: Sector update," https://docs.vis.com.pk/docs/Denim201803.pdf, 2018, accessed: 2024-05-15.

[2] Delta Máquinas Têxteis, "Textile industry in brazil: discover the panorama," https://deltamaquinastexteis.com.br/en/textile-industry-in-brazil-discover-the-panorama/, 2023, accessed: 2024-05-15.

[3] Brazilian Textile and Fashion Industry, "Brazilian textile and apparel sector for 2022," https://texbrasil.com.br/en/press/brazilian-textile-and-apparel-sector/, 2023, accessed: 2024-05-15.

[4] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, "Fabric defect detection in textile manufacturing: A survey of the state of the art," *Security and Communication Networks*, vol. 2021, p. 1–13, May 2021. [Online]. Available: http://dx.doi.org/10.1155/2021/9948808

[5] K. Hanbay, M. F. Talu, and O. F. Özgüven, "Fabric defect detection systems and methods—a systematic literature review," *Optik*, vol. 127, no. 24, p. 11960–11973, Dec. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijleo.2016.09.110

[6] H. Balakrishnan, S. Venkataraman, and S. Jayaraman, "Fdics: A vision-based system for the identification and classification of fabric defects," *Journal of the Textile Institute*, vol. 89, no. 2, p. 365–380, Jan. 1998. [Online]. Available: http://dx.doi.org/10.1080/00405009808658623

[7] H. Çelik, L. Dülger, and M. Topalbekiroğlu, "Development of a machine vision system: real-time fabric defect detection and classification with neural networks," *The Journal of The Textile Institute*, vol. 105, no. 6, p. 575–585, Sep. 2013. [Online]. Available: http://dx.doi.org/10.1080/00405000.2013.827393

[8] A. Song, Y. Han, H. Hu, and J. Li, "A novel texture sensor for fabric texture measurement and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, p. 1739–1747, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1109/TIM.2013.2293812

[9] M. F. TALU, K. HANBAY, and M. HATAMİ VARJOVİ, "Cnn-based fabric defect detection system on loom fabric inspection," *Tekstil ve Konfeksiyon*, vol. 32, no. 3, p. 208–219, Sep. 2022. [Online]. Available: http://dx.doi.org/10.32710/tekstilvekonfeksiyon.1032529

[10] L. Zheng, X. Wang, Q. Wang, S. Wang, and X. Liu, "A fabric defect detection method based on improved yolov5," in *2021 7th International Conference on Computer and Communications (ICCC)*. IEEE, Dec. 2021. [Online]. Available: http://dx.doi.org/10.1109/ICCC54389.2021.9674548

[11] A. Durmusoglu and Y. Kahraman, "Detection of fabric defects using convolutional networks," in *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, Oct. 2021. [Online]. Available: http://dx.doi.org/10.1109/ASYU52992.2021.9599071

[12] J. Barman, H.-C. Wu, and C.-F. J. Kuo, "Development of a real-time home textile fabric defect inspection machine system for the textile industry," *Textile Research Journal*, vol. 92, no. 23–24, p. 4778–4788, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1177/00405175221111477

[13] R. Thakur, D. Panghal, P. Jana, Rajan, and A. Prasad, "Automated fabric inspection through convolutional neural network: an approach," *Neural Computing and Applications*, vol. 35, no. 5, p. 3805–3823, Oct. 2022. [Online]. Available: http://dx.doi.org/10.1007/s00521-022-07891-1

[14] J. J. M. Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. Beets-Tan, J.-C. Fillon-Robin, S. Pieper, and H. J. Aerts, "Computational radiomics system to decode the radiographic phenotype," *Cancer Research*, vol. 77, no. 21, pp. e104–e107, 2017. [Online]. Available: https://doi.org/10.1158/0008-5472.CAN-17-0339

[15] L. P. Coelho, "Mahotas: Open source software for scriptable computer vision," *Journal of Open Research Software*, vol. 1, no. 1, p. e3, 2013. [Online]. Available: https://dx.doi.org/10.5334/jors.ac