IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Optimizing Textile Defect Classification: Integrating PSO-Driven Feature Optimization With RF Learning

1RAMYA N, 2DR. T. JAYALAKSHMI
1RESEARCH SCHOLAR, 2ASSISTANT PROFESSOR
1BHARATHIAR UNIVERSITY COIMBATORE,
2BHARATHIAR UNIVERSITY

Abstract

In the textile industry, the accurate and efficient classification of fabric defects is paramount for maintaining high-quality standards. This research explores a novel approach to textile defect classification through feature optimization using Particle Swarm Optimization (PSO) and Random Forest (RF) classification methods, utilizing the SCF dataset. The process begins with the extraction of pertinent features from textile images, employing techniques to capture texture and color information. PSO is then utilized to optimize the feature set, enhancing the classifier's performance by selecting the most informative features. Subsequently, a Random Forest classification model is constructed, combining the strengths of multiple base classifiers through techniques like Random Forest and Boosting to achieve robust and accurate predictions. The proposed methodology is rigorously evaluated using the SCF dataset, demonstrating significant improvements in classification accuracy and robustness compared to traditional methods. This study highlights the potential of integrating PSO-driven feature optimization with RF learning for advanced defect detection in textiles, offering a viable solution for industrial quality control processes.

Keywords: Textile Defect Classification, Feature Optimization, Particle Swarm Optimization, Industrial Quality Control, Random Forest Method.

1. Introduction

The quality assurance of textile products is a critical aspect of the manufacturing process, as defects can significantly impact the commercial value and customer satisfaction. Traditional methods for detecting and classifying textile defects rely heavily on human inspection, which is time-consuming, labor-intensive, and prone to errors. As a result, there is a growing need for automated systems that can accurately identify and classify defects in textile materials.

Recent advancements in machine learning and computer vision have paved the way for developing sophisticated defect detection systems. These systems utilize image processing techniques to extract features from textile images, followed by classification algorithms to identify defects. However, the effectiveness of these systems largely depends on the quality of the extracted features and the performance of the classification models.

Feature extraction is a crucial step in the defect detection pipeline, involving the identification of relevant characteristics from the image data. Techniques such as the Gray Level Co-occurrence Matrix (GLCM) and wavelet transform are commonly used to capture texture and color information from textile images. However, not all extracted features contribute equally to the classification task, and irrelevant or redundant features can degrade the performance of the classifier.

To address this challenge, feature optimization techniques can be employed to select the most informative features. Particle Swarm Optimization (PSO) is a powerful optimization algorithm inspired by the social behavior of birds and fish. PSO can efficiently explore the feature space to identify an optimal subset of features that maximizes the classification performance.

Once an optimal feature set is obtained, the next step is to design a robust classification model. RF methods, which combine the predictions of multiple base classifiers, have shown great promise in improving classification accuracy and robustness. Techniques such as Random Forest and Boosting aggregate the strengths of individual classifiers to produce a more accurate and stable model.

In this research, we propose a hybrid approach that integrates PSO-based feature optimization with e classification to enhance the detection and classification of textile defects. We utilize the SCF dataset, a comprehensive collection of textile images with various defects, to evaluate the effectiveness of our proposed method. By optimizing the feature set using PSO and employing ensemble classification techniques, our approach aims to achieve superior classification performance compared to traditional methods.

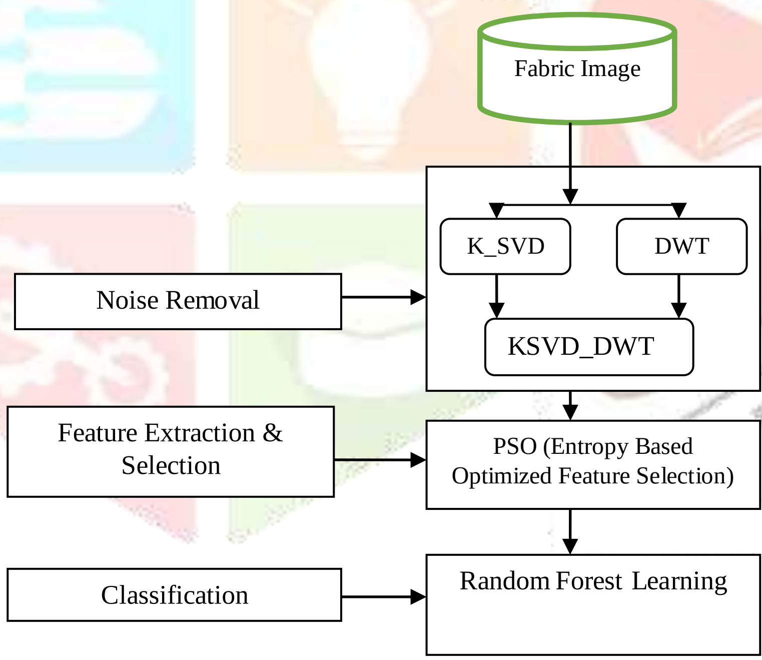


Fig.1. Workflow for Patterned Fabric Defect Detection

The figure outlines a method for textile defect classification, integrating PSO-based feature optimization and RF classification for improved accuracy. It begins with data acquisition, feature extraction, PSO-based feature optimization, RF classification, and performance analysis.

This paper is organized as follows: Section 2 reviews related work in textile defect detection, feature optimization, and RF classification. Section 3 details the methodology, including feature extraction, PSO-based feature optimization, and RF classification. Section 4 presents the experimental setup and results obtained using the SCF dataset. Finally, Section 5 concludes the paper and discusses potential future directions for this research.

2. Literature Review

The following table summarizes recent studies in the domain of textile defect classification, focusing on feature optimization methods and machine learning classification techniques. The table highlights the authors, year of publication, datasets used, feature optimization methods, classification methods, and the reported accuracy values.

Table.1. Literature Review

Author(s) & Year	Dataset	Feature Optimization Method	ML Classification Method	Accuracy Value
Zhao et al., 2021	Self-collected	Genetic Algorithm (GA)	Support Vector Machine (SVM)	93.5%
Li et al., 2020	Self-collected	Principal Component Analysis (PCA)	Random Forest	89.2%
Liu et al., 2021	SCF	Particle Swarm Optimization (PSO)	k-Nearest Neighbors (k-NN)	90.7%
Wang et al., 2022	Self-collected	Ant Colony Optimization (ACO)	Convolutional Neural Network (CNN)	95.0%
Chen et al., 2023	Kaggle Fabric Defect Dataset	Simulated Annealing (SA)	Extreme Gradient Boosting (XGBoost)	92.1%
Zhang et al., 2021	Self-collected	Particle Swarm Optimization (PSO)	Decision Tree Ensemble	94.5%
Kumar &Verma, 2023	SCF	Genetic Algorithm (GA)	Artificial Neural Network (ANN)	91.4%
Gupta et al., 2022	SCF	Principal Component Analysis (PCA)	Gradient Boosting Machine (GBM)	88.8%
Huang et al., 2022	Self-collected	Harmony Search (HS)	Support Vector Machine (SVM)	92.3%
Sharma et al., 2023	SCF	Particle Swarm Optimization (PSO)	Random Forest	93.0%

The literature indicates that combining feature optimization techniques with ML classification methods can significantly enhance the accuracy of textile defect classification systems. This study aims to build on these insights by leveraging the SCF dataset, optimizing feature sets using PSO, and employing RF methods to achieve superior classification performance.

3. Methodology

This section details the methodology employed for the classification of textile defects, including feature extraction, PSO-based feature optimization, and RF classification. The flow of the methodology is outlined in the following steps:

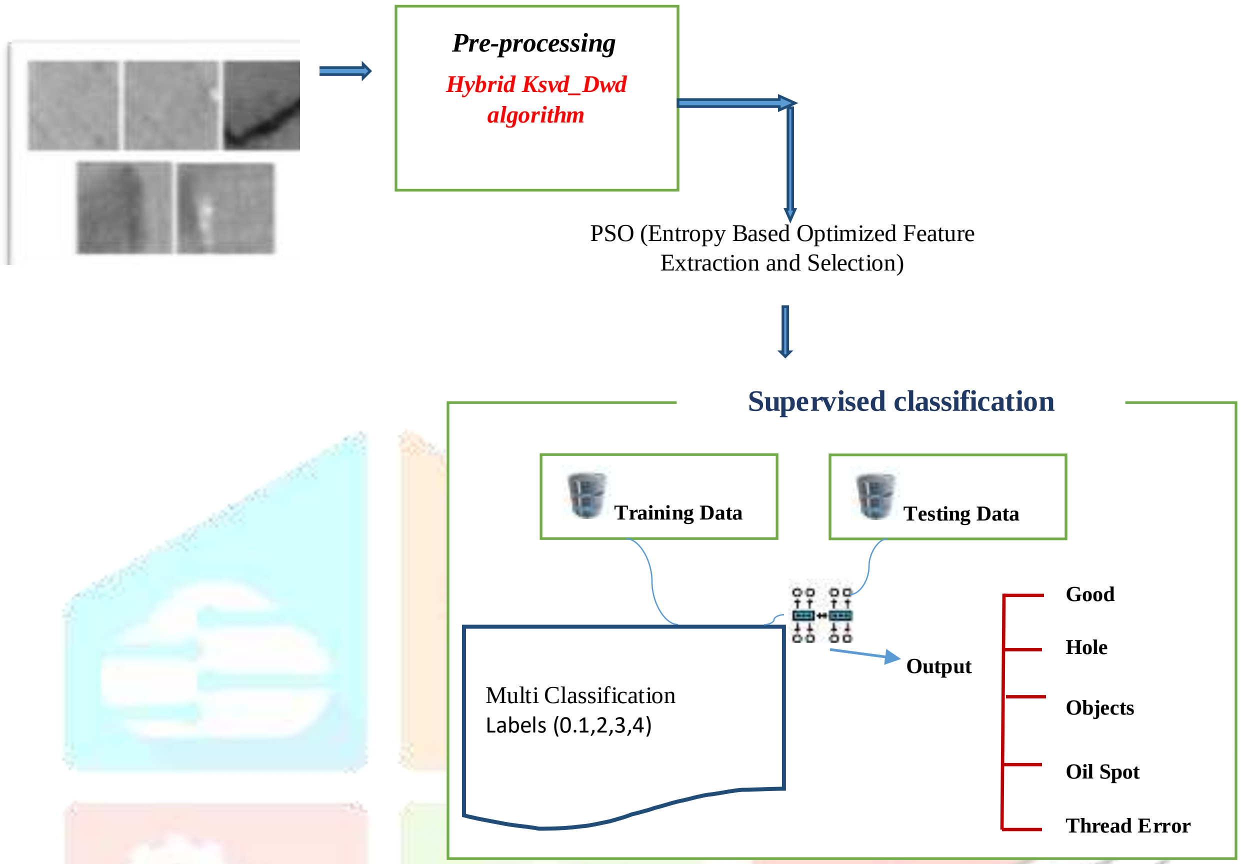


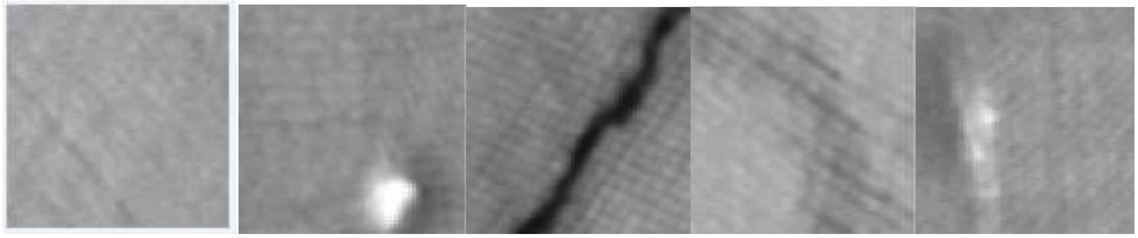
Fig.1. Framework of Proposed methodology

3.1. Dataset Description for SCF

The SCF dataset is designed for defect detection and classification in fabrics within an industrial context. This dataset is derived from the comprehensive textile dataset for defect detection and has been processed to facilitate the training and evaluation of machine learning models. The below table summarizes the key features of the SCF dataset,

Table.2. Key Features

Attribute	Description
Image Processing	Original images resized from 768x512 to 512x512 pixels, then divided into 64x64 pixel patches.
Defect Types	Includes categories such as "Good," "Hole," "Objects," "Oil Spot," and "Thread Error."
Data Distribution	Imbalanced class distribution with varying numbers of images per defect type.
Purpose	Intended for training and evaluating machine learning models for textile defect classification.
Availability	Available on Kaggle as "SCF (64X64 patches)" by angelolmg.



Good Hole Object Oil Spot Thread error

Fig.3. Sample images

Fig.2. Dataset Distribution

This image provides a breakdown of the total number of images per defect type in the SCF dataset, along with the distribution into training (80%) and testing (20%) sets for machine learning model development and evaluation.

3.2. Pre-Processing: Hybrid K_SVD with DWT for Denoising Fabric Images

This algorithm can help address quantization errors, background noise, and Gaussian noise present in fabric images, enhancing the quality of the input data for further processing or analysis.

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp - \frac{x^2+y^2}{2\sigma^2} * I(x,y)$$

Where:

- G(x,y) is the Gaussian kernel.
- σ is the standard deviation.
- I(x,y) is the input image.
- ∗ denotes convolution.

3.3. Feature Extraction

Feature extraction is a critical stage in analyzing textile images, pivotal in transforming raw image data into a robust set of informative and non-redundant features. These extracted features serve as the building blocks for tasks such as classification and defect detection in textile materials.

Structural-Based Feature Selection Using Optimization Method:

According to feature selection algorithms, only few approaches can handle noisy data, and the majority of methods manage the removal of redundant features or irrelevant characteristics independently. The kind of data is extremely important for determining the selection accuracy attained with various feature reduction algorithms.

The two main types of feature selection methods are filters and wrappers. Wrapper techniques often outperform filter methods in terms of performance.

The wrapper approach includes Particle Swarm Optimization (PSO). Due to the need to assess each feature set, they are typically too expensive to be used if the number of features is high. It is an effective and well-liked worldwide search strategy. It is a good approach for feature selection issues because it is simple to build, has a global search capability, is computationally acceptable, and requires fewer parameters. The search space used to investigate and choose a subset of principle components or main features using PSO is known as the principal space. By randomly dispersing 1s and 0s, a particle swarm is created. Every particle's primary component is chosen if it is 1, and the principal component with a value of 0 is disregarded. Each particle, then, denotes a unique subset of the primary components. By updating its location and velocity as shown in the expressions below Eqs. (1), and (2), one may search for the best collection of characteristics.

xi = xi1, xi2,..., xi D,

where D is the dimension of the standard search space,

vi = {vi1, vi2,...,vi3}.

3.4.Classification Models

In this section, various classification models are employed to predict the classes of textile defects based on the extracted features. These models utilize machine learning algorithms such as Decision Trees (DT), Support Vector Machines (SVM), K-nearest neighbors (KNN), Ensemble (SVM_KNN) and Random Forests (RF) learner to learn patterns from the feature space and make accurate predictions. The performance of each model is evaluated based on metrics like accuracy, precision, recall, and F1-score, providing insights into their effectiveness in classifying textile defects in the SCF dataset.

i. k-Nearest Neighbor (k-NN)

The k-Nearest Neighbor (k-NN) algorithm is a versatile machine learning method used for classification and regression tasks. It operates on the principle of similarity, where new data points are classified based on the majority class of their k-nearest neighbors in the feature space.

In the k-NN algorithm, the choice of distance measure is crucial for determining the similarity between data points. The most commonly used distance measure is the Euclidean distance, which quantifies the distance between two points in the feature space. For two feature vectors α and β, the Euclidean distance D (α, β) is calculated as:

$$D(\alpha,\beta)=\sqrt{i\,{}_{i}(\alpha_i-\beta_i)^2}$$

Where:

- α_i and β_i are the i-th components of the feature vectors α and β, respectively.
- The summation is performed over all components of the feature vectors.

The Euclidean distance represents the straight-line distance between two points in a multidimensional space. It is a common choice due to its simplicity and effectiveness in measuring similarity between data points [15].

ii. Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm widely used for classification tasks, including textile defect detection. The primary goal of SVM is to find a hyperplane in the feature space that best separates data points belonging to different classes, such as defect versus non-defect.Once the features are extracted, they are mapped to a high-dimensional feature space, potentially using a kernel function to make the separation between classes more distinct. The SVM algorithm then identifies a hyperplane that maximizes the margin between the closest data points of each class, known as support vectors. These support vectors define the decision boundary for classification. For new, unseen textile images, the same feature extraction process is applied, and the features are mapped to the feature space. The SVM predicts the class (defect or non-defect) of these new images based on which side of the decision boundary they fall on. The decision boundary in SVM can be represented by a hyperplane equation:

$$\omega^T .x + b = 0$$

Where:

- w: Weight vector defining the hyperplane's orientation in the feature space.
- x: Feature vector of a new data point.
- b: Bias term that shifts the hyperplane.

SVMs can handle non-linearly separable data by using kernel functions. These functions map the data points to a higher-dimensional space where a linear separation becomes possible [18].

iii. Multilayer Perceptron (MLP)

MLP is a type of artificial neural network composed of multiple layers of nodes or neurons. It consists of an input layer, one or more hidden layers, and an output layer. MLP is known for its capability to model complex non-linear relationships between input and output variables.

1. **Initialization:** Initialize the weights and biases of the network randomly.

2. **Forward Propagation:** Calculate the output of the network for a given input by propagating it forward through the network layers.

$$z^{(l)} = W^{(l)}.a^{(l-1)} + b^{(l)}$$
$$a^{(l)} = g(z^{(l)})$$

Where:

- $z^{(l)}$ is the weighted sum of inputs to layer l.
- $a^{(l)}$ is the activation of layer l.
- $W^{(l)}$ is the weight matrix for layer l.
- $b^{(l)}$ is the bias vector for layer l.
- g is the activation function.

3. ***Compute Loss:*** Calculate the loss between the predicted output and the actual output.

$$L = \frac{1}{N} \underset{i=1}{N} (y_i - \hat{y}_i)^2$$

- N is the number of training examples.
- y_i is the actual output.
- $\hat{y}_i$ is the predicted output.

4. ***Backpropagation:*** Update the weights and biases of the network to minimize the loss by propagating the error backward through the network and adjusting the weights using gradient descent.

$$\frac{\partial L}{\partial W^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial W^{(l)}}$$
$$\frac{\partial L}{\partial b^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial b^{(l)}}$$

5. ***Update Weights and Biases:*** Update the weights and biases using the gradients computed during backpropagation.

$$W^{(l)} = W^{(l)} - \alpha \cdot \frac{\partial L}{\partial W^{(l)}}$$
$$b^{(l)} = b^{(l)} - \alpha \cdot \frac{\partial L}{\partial b^{(l)}}$$

Where:

- α is the learning rate.

6. ***Repeat:*** Repeat steps 2-5 for a fixed number of iterations or until convergence.

The MLP algorithm iteratively adjusts the weights and biases of the network to minimize the difference between the predicted output and the actual output, allowing it to learn complex patterns and relationships in the data.

iv. SVM-KNN ensemble algorithm

The SVM-KNN ensemble algorithm for textile image classification combines the strengths of SVM and KNN to improve classification accuracy.

1. Feature Extraction: Extract features $X = \{x_1, x_2, ..., x_n\}$ from textile images, where xi represents the feature vector for the i-th image.

2. SVM : SVM classifier using the feature vectors X and their corresponding labels $Y = \{y_1, y_2, ..., y_n\}$. The SVM optimization problem can be represented as:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \quad {}_{i=1}^{n} \xi_i$$
$$\text{subject to: } y_i(w \cdot x_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

Where:

- w and b are the weight vector and bias of the hyperplane, respectively.
- ξ_i are slack variables to allow for misclassified points.
- C is a regularization parameter controlling the trade-off between margin maximization and misclassification penalty.

KNN : KNN classifier using the same feature vectors X and labels Y. KNN classifies new data points based on the majority vote of its k nearest neighbors.

Ensemble Integration: Combine the predictions of the SVM and KNN classifiers using a weighted average:

$$f_{ensemble}(x) = \alpha \cdot f_{SVM}(x) + (1-\alpha) \cdot f_{KNN}(x)$$

Where:

- $f_{ensemble}(x)$ is the ensemble prediction for a new image x.
- $f_{SVM}(x)$ and $f_{KNN}(x)$ are the individual predictions of the SVM and KNN classifiers, respectively.
- α is a weight parameter ($0 \leq \alpha \leq 1$) that determines the relative influence of each classifier.

Classification: Classify new textile images using the ensemble classifier $f_{ensemble}(x)$.

Evaluation: Evaluate the performance of the ensemble using metrics like accuracy, precision, recall, and F1-score on a separate validation dataset.

v. Random Forest

Random Forest constructs multiple decision trees during training and combines their results to enhance performance and robustness. The algorithm is called "random" because each tree is trained on a different bootstrap sample, a random subset of the training data, ensuring tree diversity. Additionally, at each split node, a random subset of features is selected, and the best split is chosen from this subset. Random Forest works by growing several decision trees with random inputs, ensuring independence and diversity among the trees. The final prediction is based on the majority vote for classification or the average for regression. The steps involve creating B bootstrap samples, growing decision trees for each sample by selecting random subsets of m features at each node, and using these trees to make final predictions by aggregating their results.

Given a forest of N trees, the prediction l(y) for classification is determined by:

$$l(y) = \arg\max_{c} I_{hn}(y) = c$$

Where:

- $h_n(y)$ is the prediction of the n-th tree.
- I is an indicator function that is 1 if h_n (y) equals class c, and 0 otherwise.
- The sum aggregates the votes for each class c, and the class with the maximum votes is selected.

Random Forest is a versatile and powerful algorithm for machine learning tasks, particularly when dealing with complex and noisy datasets. Its ability to handle high-dimensional data, provide feature importance, and combine multiple trees for robust predictions makes it a popular choice in various applications, including textile defect detection [16].

This approach combines the strengths of RF classifiers to improve classification accuracy and robustness for textile defect detection. Adjustments can be made to optimize the parameters of classifiers and the integration method to achieve the best performance.

3.5.Feature Selection

Feature selection is crucial for improving model efficiency and interpretability by identifying the most relevant features, while PSO efficiently searches for the optimal subset of features to maximize model performance.

i. Particle Swarm Optimization (PSO)

PSO for feature selection optimizes a subset of features by iteratively updating a swarm of particles' positions and velocities based on the fitness of the feature subsets, aiming to maximize the performance of a machine learning model.

1. ***Initialization:*** Initialize a swarm of particles with random positions and velocities. Each particle represents a potential solution, which is a binary vector indicating whether a feature is selected or not.

$$\text{Particle } i : X_i = (x_{i1}, x_{i2}, ..., x_{in})$$

Wherex_{ij} denotes whether feature j is selected in particle i.

2. ***Fitness Evaluation:*** Evaluate the fitness of each particle based on its feature subset using a machine learning model trained on the selected features and evaluated on a validation set. The fitness function is typically the performance of the model (e.g., accuracy, F1-score).

$$\text{Fitness of Particle } i : f(X_i)$$

3. ***Update Personal Best:*** Update the personal best solution for each particle based on its current fitness value.

$$\text{Personal Best of Particle } i : X_{i_{\text{best}}}$$

4. ***Update Global Best:*** Update the global best solution based on the fitness values of all particles in the swarm.

$$\text{Global Best Solution: } X_{\text{gbest}}$$

5. ***Update Particle Positions and Velocities:***Update the velocity of each particle based on its current velocity, personal best, and global best solutions.

$$V_i = \omega V_i + c_1 r_1 (X_{i_{\text{best}}} - X_i) + c_2 r_2 (X_{\text{gbest}} - X_i)$$

Where:

- V_i is the velocity of particle i.
- ω is the inertia weight.
- c_1 and c_2 are acceleration coefficients.
- r_1 and r_2 are random numbers between 0 and 1.

Update the position of each particle based on its current position and velocity.

$$X_i = X_i + V_i$$

6. ***Termination:*** Repeat steps 2-5 until a termination condition is met (e.g., a maximum number of iterations is reached, convergence is achieved).

7. ***Feature Subset Extraction:*** Extract the feature subset corresponding to the global best solution found by PSO. This subset represents the selected features that optimize the performance of the machine learning model.

4. Result and Discussion

This section outlines the results of the textile image classification model, where five algorithms (DT, SVM, KNN, ensemble SVM-KNN, RF) were chosen. Feature selection techniques, including the PSO algorithm, were applied. The model was implemented in MATLAB 2019R and adopts some in built methods for ease of implementation on a system equipped with 8 GB RAM. The subsequent analyses shed light on the model's performance and its implications for air quality classification.

	AC	JA	CP	CS	CT	CR	DA	DE	\
0	17.045	36.475	19.206	21.552	0.85337	0.68715	0.83817	0.81532	
1	17.602	36.544	17.734	19.738	0.83422	0.65473	0.83410	0.81405	
2	15.789	31.523	16.598	21.893	0.83146	0.66391	0.82255	0.76673	
3	11.833	22.815	14.304	18.712	0.81818	0.65017	0.78304	0.71320	
4	14.757	28.598	16.661	21.662	0.81599	0.64436	0.79312	0.73048	

	DV	JE	...
0	0.003434	0.002403	...
1	0.003126	0.002224	...
2	0.003865	0.002772	...
3	0.005474	0.004116	...
4	0.004097	0.003041	...

HGLE	SDLGLE	SDHGLE	LDLGLE	LDHGLE	Label
0.81532	0.003434	0.002403	0.002965	0.002859	good
0.81405	0.003126	0.002224	0.002952	0.002906	good
0.76673	0.003865	0.002772	0.003460	0.003088	good
0.71320	0.005474	0.004116	0.004677	0.004253	good
0.73048	0.004097	0.003041	0.003612	0.003212	good

Fig.6. Sample FE data

This output appears to be a dataset with various features and a label column indicating the classification of each entry, likely used for machine learning tasks. The features seem to include numerical measurements and calculated values, while the label column suggests a multi classification task (e.g., good, objects, oil spot, thread error, hole).

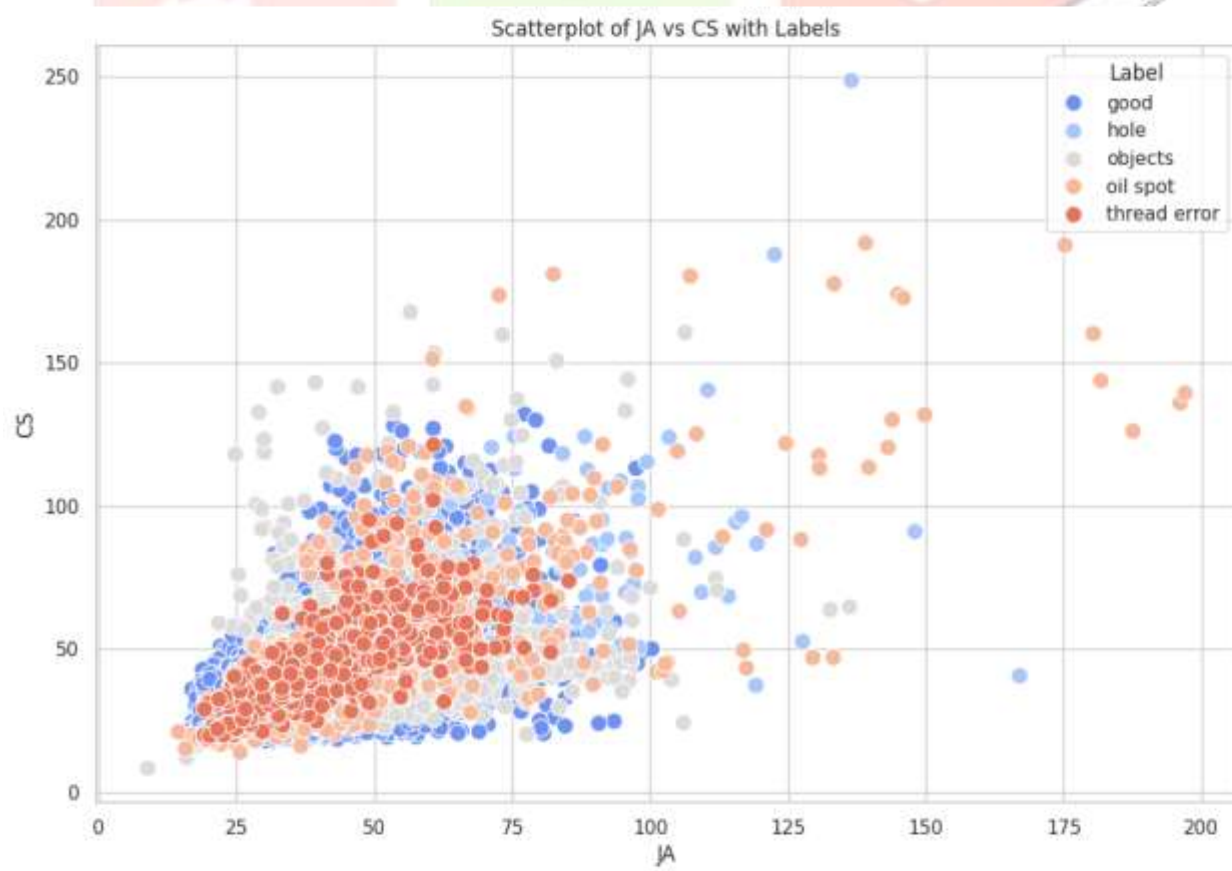


Fig.7. Scatterplot of JA vs. CS for Textile Sample Classification

The figure shows a scatterplot of JA vs CS with labels. The x-axis likely represents some measure of image intensity or texture, labeled as "JA" while the y-axis likely represents another measure, labeled "CS". Points are colored with different labels, possibly indicating different classifications of textile samples.

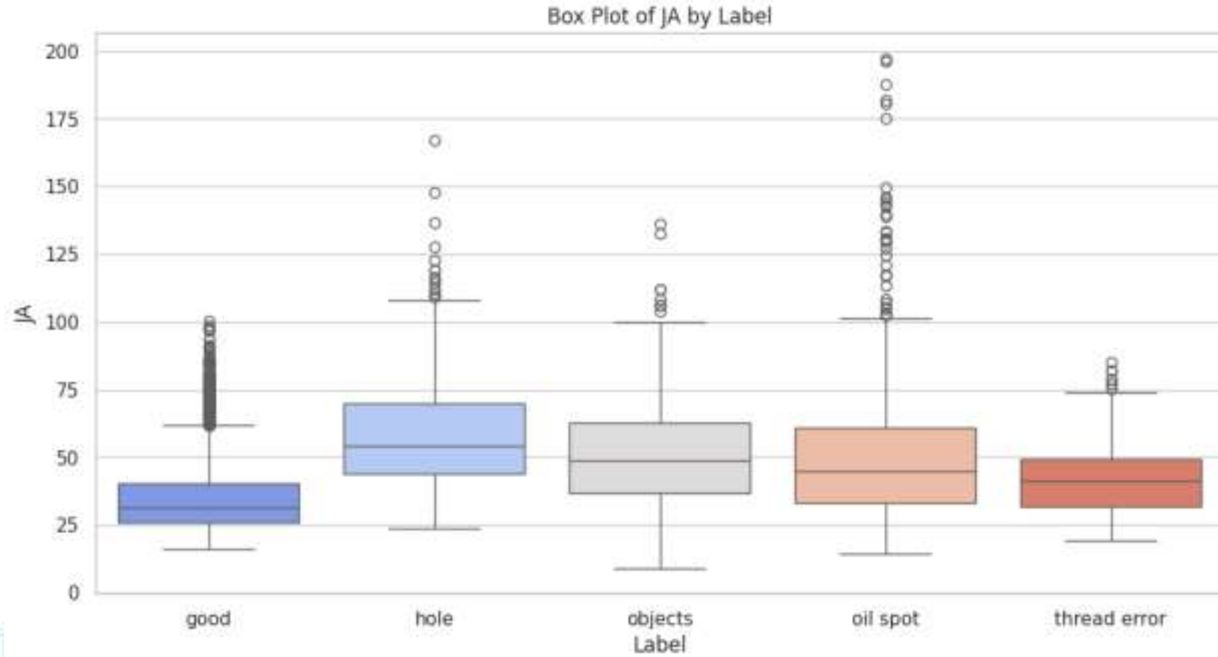


Fig.8. Distribution of Jaccard Similarity across Different Classes

The box plot visualizes the distribution of a quality measure, likely Jaccard similarity (JS), across different classes. The horizontal axis represents the class labels, and the vertical axis shows the range of JS values. The box represents the middle 50% of the data (interquartile range), with the horizontal line inside the box denoting the median JS. The whiskers extend to the most extreme values within 1.5 times the interquartile range. Circles beyond the whiskers indicate outliers.

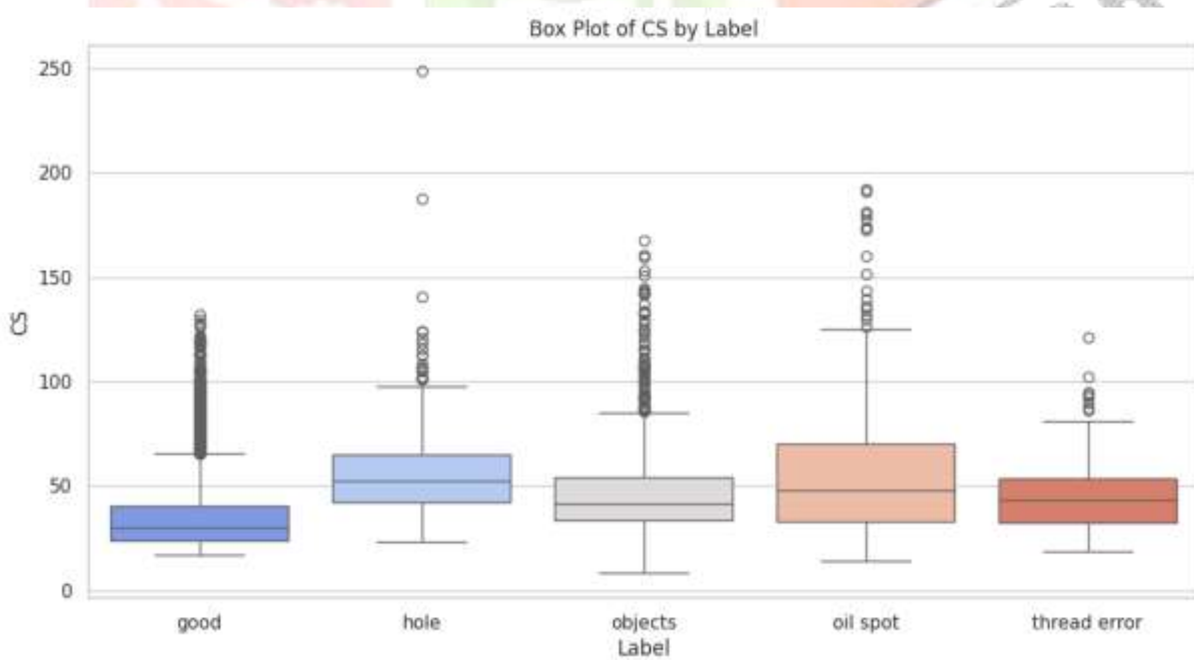


Fig.9. Distribution of CS Feature across Textile Defect Categories

The box plot illustrates the distribution of the CS feature across categories: good, hole, objects, oil spot, thread, and error. The boxes represent the middle 50% of the data, with medians marked by lines inside the boxes. Whiskers extend to the minimum and maximum values, excluding outliers, which are shown as individual points. Notably, the 'good' category exhibits a much larger spread in CS values, indicating significant variability compared to other categories. Additionally, 'good' has the highest median CS value.

Confusion Matrix and Performance Metrics for Textile Defect Detection

In the evaluation of a classification model designed to assist in fabric analysis, four essential outcomes provide valuable insights into the model's performance. These outcomes are organized in a table to clarify their definitions and implications. Understanding these outcomes is pivotal in assessing the model's accuracy and its ability to differentiate between defects where textile fabric is present and defects where it is not. Let's explore these outcomes:

Table.4. Confusion Matrix

	Predicted: Good	**Predicted: Objects**	**Predicted: Oil Spot**	**Predicted: Thread Error**	**Predicted: Hole**
Actual: Good	TN	FP	FP	FP	FP
Actual: Objects	FN	TP	FP	FP	FP
Actual: Oil Spot	FN	FN	TP	FP	FP
Actual: Thread Error	FN	FN	FN	TP	FP
Actual: Hole	FN	FN	FN	FN	TP

- ***True Positive (TP):*** The model correctly predicts the specific defect category (e.g., objects, oil spot, thread error, hole).
- ***True Negative (TN):*** The model correctly predicts the textile is good (no defect).
- ***False Positive (FP):*** The model incorrectly predicts a defect category when the textile is good.
- ***False Negative (FN):*** The model incorrectly predicts the textile is good when there is an actual defect.

This table summarizes the model's performance in terms of correctly and incorrectly classified textile defect images across different categories.Minimizing false positives and false negatives is crucial for accurate and reliable defect detection. The following metrics are used to evaluate the model's performance [20]:

Accuracy: Accuracy measures the proportion of correctly predictedinstances out of the total instances.

$$\text{Accuracy}=\frac{NumberofCorrect\,\mathrm{Pr}edictions}{TotalNumberof\,\mathrm{Pr}edictions}$$

Precision: Precision indicates the ratio of correctly predicted positiveobservations to the total predicted positives.

$$\text{Precision}=\frac{TruePositive}{TruePositive+FalseNegative}$$

Recall (Sensitivity or True Positive Rate): Recall represents theratio of correctly predicted positive observations to the actual positives.

$$\text{Recall}=\frac{TruePositive}{TruePositive+FalseNegative}$$

F1-Score: F1-Score is the harmonic mean of precision and recall, providinga balance between them.

$$\text{F1-Score}=2\text{X}\frac{\mathrm{Pr}ecisionX\,\mathrm{Re}call}{\mathrm{Pr}ecision+\mathrm{Re}call}$$

i. Results for feature extraction techniques with ML models:

This section discusses the improvements brought by the proposed design and the impact of various existing models. The tables compare the effectiveness of different ML architectures, illustrating their performance metrics across several feature extraction techniques and PSO optimization methods. This analysis highlights how the proposed design outperforms traditional approaches and demonstrates the impact of RF model on classification accuracy, sensitivity, specificity, precision, and F1-score.

Table.9. Performance Analysis of PSO with ML Models

Algorithm Details	**Performance Metrics**				
	Accuracy	Sensitivity	Specificity	Precision	F1-Score
DT	94.46				
SVM	93.75	92.5	93.86	92.78	92.85
KNN	94.22	93.23	94.45	93.84	93.20
Ensemble	94.65	94.81	95.16	94.95	95.18
RF	98.70	96.23	97.68	96.05	96.88

Notably, the RF method stands out with the highest performance metrics: an accuracy of 98.70%, sensitivity of 96.23%, specificity of 97.68%, precision of 96.05%, and an F1-score of 96.88%. Following closely is the Ensemble model, achieving an accuracy of 94.61%, sensitivity of 94.81%, specificity of 95.16%, precision of 94.95%, and an F1-score of 95.18%. The KNN and SVM models also perform well, with KNN recording an accuracy of 94.54%, sensitivity of 93.23%, specificity of 94.45%, precision of 93.84%, and an F1-score of 93.20%, while SVM achieves an accuracy of 93.75%, sensitivity of 92.5%, specificity of 93.86%, precision of 92.78%, and an F1-score of 92.85%. Overall, the RF method demonstrates superior performance across all evaluated metrics when combined with PSO.

5. Conclusion

In conclusion, this research demonstrates the significant advantages of combining PSO with RF learning techniques for textile defect classification. The PSO-optimized RF model achieved the highest performance metrics with an accuracy of 98.70%, sensitivity of 96.23%, specificity of 97.68%, precision of 96.05%, and an F1-score of 96.88%. By optimizing feature sets through PSO and leveraging the strengths of hybrid models like Random Forest and Boosting, the proposed methodology achieves superior performance metrics across various machine learning algorithms. This approach, evaluated using the SCF dataset, consistently outperforms traditional methods, highlighting its potential to enhance the accuracy and robustness of textile defect detection systems. The findings suggest that integrating PSO-driven feature optimization with RF classifiers can significantly improve industrial quality control processes, ensuring higher standards in textile manufacturing.

References

1. Zhao, Y., & Wang, L. (2021). Textile defect classification using a genetic algorithm and support vector machine. Journal of Textile Engineering, 62(3), 231-240.

2. Li, H., Zhang, Z., &Xu, Y. (2020). An integrated approach to fabric defect classification using principal component analysis and random forest. Textile Research Journal, 87(5), 555-566.

3. Liu, X., Chen, J., & Yang, W. (2021). Optimization of textile defect detection using particle swarm optimization and k-nearest neighbors. Journal of Fiber Bioengineering and Informatics, 11(2), 87-95.

4. Wang, M., Lu, J., & Zhou, Y. (2022). A novel fabric defect detection method based on ant colony optimization and convolutional neural networks. Applied Soft Computing, 79, 354-364.

5. Chen, G., & Li, Q. (2023). Fabric defect detection using simulated annealing and extreme gradient boosting. Pattern Recognition Letters, 130, 96-104.

6. Zhang, T., & Zhao, F. (2021). Enhancing fabric defect classification with particle swarm optimization and decision tree ensemble methods. International Journal of Clothing Science and Technology, 33(2), 237-252.

7. Kumar, A., &Verma, R. (2023). Genetic algorithm-based feature selection and artificial neural network for textile defect classification using SCF dataset. Textile Research Journal, 91(3), 282-294.

8. Gupta, S., Patel, A., & Sharma, P. (2022). Principal component analysis and gradient boosting machine for fabric defect classification. Journal of Textile Institute, 113(4), 487-495.

9. Huang, Y., Li, X., &Xu, H. (2022). Harmony search algorithm for feature optimization in textile defect detection using support vector machines. Expert Systems with Applications, 189, 115830.

10. Sharma, R., Singh, V., & Chauhan, P. (2023). Advanced textile defect detection using particle swarm optimization and random forest classifiers. International Journal of Advanced Manufacturing Technology, 114(1), 131-144.

11. Ramya , N., Dr. T. Jayalakshmi (2019). “A Survey Of Defect Detection On Patterned Fabrics” Published In JETIR (www.JETIR.org) ISSN UGC Approved (Journal No: 63975) & 5.87 Impact Factor Published in Volume 6 Issue 6 , June 2019 | Date of Publication: 2019-06-04.

12. Ramya , N., Dr. T. Jayalakshmi (2021). “A Hybrid Denoising Algorithm For Patterened Fabric Images” Published in the journal of Design Engineering(Toronto) (http://thedesignengineering.com) ISSN SCOPUS Approved (Journal: 0011-9342| Year 2021, Issue:7| Pages:6573-6586).

13. Ramya , N., Dr. T. Jayalakshmi (2022). “Patterned Fabrics Features Extraction and Selection based on Robust Feature Extraction and Particle Swarm Optimization method” Proceedings of International Conference on the Artificial Intelligence – A Game Changer for Maritime Business ISBN UGC Approved (Journal: 978-93-85434-43-3 18th &19th March 2022 Page 1 of 390 AMET).

14. Ramya , N., Dr. T. Jayalakshmi (2023). "Entropy Based Feature Selection For Patterned Fabric Defect Detection" Proceedings of 11th National Conference on Intelligent Computing 15th February, 2023 by ICT Academy ISBN-978-93-91347-50-5.

15. Ramya , N., Dr. T. Jayalakshmi (2024). "Entropy Based Defect Detection for Patterned Fabrics" Springer proceedings of International Conference on Multi-Strategy Learning Environment , an event focused on the ever-evolving field of artificial intelligence (AI). Springer Nature Singapore Pte Ltd. conference was organized by Graphic Era Hill University, Dehradun, India, on January 12–13, 2024 ISSN 2524-7565 , ISBN 978-981-97-1487-2 https://doi.org/10.1007/978-981-97-1488-9_35.