ResNet50: Automated Fabric Defect Detection and Classification based on a Deep Learning Approach

[1] K. Sharath Kumar, [2] Dr. M. Rama Bai

[1] Research Scholar, JNTU Hyderabad, Telangana, Hyderabad-500075,
[2] Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology Hyderabad, Telangana, India - 500075

E-mail: [1] sharathjohn@yahoo.com, [2] mramabai_cse@mgit.ac.in

Abstract: "The demand for an intelligent computer-based system for visual inspection is on the rise in the textile business, especially among those who place a premium on textile quality. In this research, we present a fully automated AI-driven algorithm for defect detection in fabric, one that makes use of deep neural network models that have already been pre-trained. In order to train the networks, the fabric images go through a series of pre-processing steps where typical image processing techniques are applied. In order to train and categorize various fabric flaws, we use Deep Convolutional Neural Networks (DCNNs) and the pre-trained ResNeT network. The system obtains a best classification accuracy of 96.40% in simulations utilizing preexisting textile datasets. This model's detection and classification system can help human operators spot flaws in the fabric manufacturing process with this degree of precision."
Keywords: Fabric defects, Artificial Intelligence, Defect Classifier, ResNet, Deep Neural Network

1. Introduction

Fabric flaws and deformities are a major issue that can have a negative effect on the quality of textile products. Misuse of the material or a mistake during production are common causes of this kind of problem. Due to the sheer volume and variety of fault types, as well as the varying degrees of certainty associated with each, manual fabric defect inspection presents a formidable challenge. Fatigue and other variables reduce the efficacy of human inspectors, so they can only catch a certain fraction of these flaws. A real-time automated fault detection system, on the other hand, might significantly improve productivity.

The textile sector places a premium on quality and works hard to meet strict deadlines without sacrificing product standards. About 85 percent of textile manufacturing flaws are caused by defective fabric, although manufacturers often only make 45 percent to 65 percent on secondhand or discounted merchandise [3]. As of January 2020, India ranked second globally in textile and apparel production and exports [4] according to data compiled by NITI Aayog. Contamination and low-quality fiber are major issues. India has the world's highest production capacity, but it uses antiquated methods. Modern textile manufacturing cannot survive without using cutting-edge technologies.

Manual examinations by skilled inspectors are currently the primary method for identifying fabric flaws, a process that is not without its own set of difficulties. Long hours of inspecting fabric can cause eye strain and reduce inspectors' ability to detect flaws. Therefore, there is a rising need in the market for an automated fabric fault detection system that can detect, identify, and hopefully eliminate reoccurring flaws. Fabrics frequently have flaws such as those listed in [5], as well as others such as holes, scratches, stretching, loose yarn, unclean patches, cracked points, misprints, color bleeding, and more. There is hope in using computer vision (CV) to solve this problem; we use an AI-based model to categorize several types of flaws such color, cut, holes, metal contamination, and thread. The dataset used to train this machine learning/deep learning model contains at least 500,000 photos.

1.1 AI in Textile Industry

There is an immediate need for high-quality textiles, and AI is at the forefront of the movement to satisfy that need. Over the past decade, artificial intelligence has seen a meteoric rise in use across all industries.

The push to cut production costs without compromising speed or precision is fueling this development. The value of textile products is diminished when defects in the fabric make them unsellable. Neural networks (NN) powered by deep learning techniques have become essential tools in the inspection and identification of problems due to their significantly faster detection and improved accuracy rates. Artificial intelligence is ushering in a new era for the textile industry, and it has immense potential.

2. RELATED WORKS

Since the 1980s, numerous attempts have been undertaken to identify fabric flaws. Spectral, structural, model-based, statistical, learning, hybrid, and comparative investigations are only some of the common classifications for these methods [6].

2.1 Spectral-Based Approaches

Fabric faults can be identified using spectral methods since it has been observed that defect-free fabrics have periodic qualities whereas those with imperfections do not. Images of fabrics are converted into frequency representations [16], with high-frequency components representing flaws. Despite benefits like noise tolerance, spectral-based techniques have fallen out of favor in recent years due to difficulties in producing reliable repeatable outcomes.

2.2 Deep Learning-Based Approaches

With the rise of AI has come a proliferation of machine learning and deep learning techniques for spotting outliers, and finding flaws in fabric is no different[2].These machine learning and deep learning-based methods have demonstrated tremendous potential in enhancing the precision and speed with which fabric flaws may be identified, allowing the textile sector to maximize product quality while decreasing waste. With the use of these optimization methods and model advancements, the textile sector can detect fabric faults more quickly and precisely, leading to better finished goods with fewer flaws. Many researchers have taken alternative hybrid approaches to the problem of flaw detection in textiles. Guangzhong Cao, Yundong Li, Cheng Zhang, and Hermanus Vermaak are all well-known for the following approaches and strategies: First, in a study titled "Detection of Large and Complex Surface Defects," conducted by Guangzhong Cao et al.[7]."Hybrid Approach for Warp-Knitted Fabric Defect Detection by Yundong Li and Cheng Zhang " uses Gabor Filters for picture enhancement and segmentation.[8] The numerous properties of faults are categorized by the third hybrid method, the Dual-Tree Complex Wavelet Transform (DTCWT) method developed by Hermanus Vermaak et al.[9]. Different types of fabric defects require different ways of identification, and researchers have utilized a wide range of methodologies, including image processing, machine learning, and feature extraction.

2.3 Dataset

1. In this research, we discuss how we used the publicly accessible MVTec Anomaly Detection dataset [10] to train a machine learning model. Five different types of defects (color, cut, hole, metal contamination, and thread) and a non-defective class (called "good") are represented in the 540,000-image dataset used for training and evaluation. There are two additional categories in this dataset:
2. **Test Set**: This collection of 180,000 photos was most likely utilized to gauge how well the model performed.
3. **Train Set**: The machine learning model was trained with data from a training batch of 360,000 photos.
4. Within the training set, in this subset, there are a total of 72,000 images. These images are equally distributed into six distinct classes, with each class containing 12,000 images.

The sheer number of photos and variety of classes in this comprehensive dataset undoubtedly played a significant impact in training and testing the machine learning model's ability to properly detect a wide range of fabric faults.

The training set consists of 72,000 photos and is evenly divided into 6 classes, with each class containing 12,000 images, as shown in Figure 1.

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

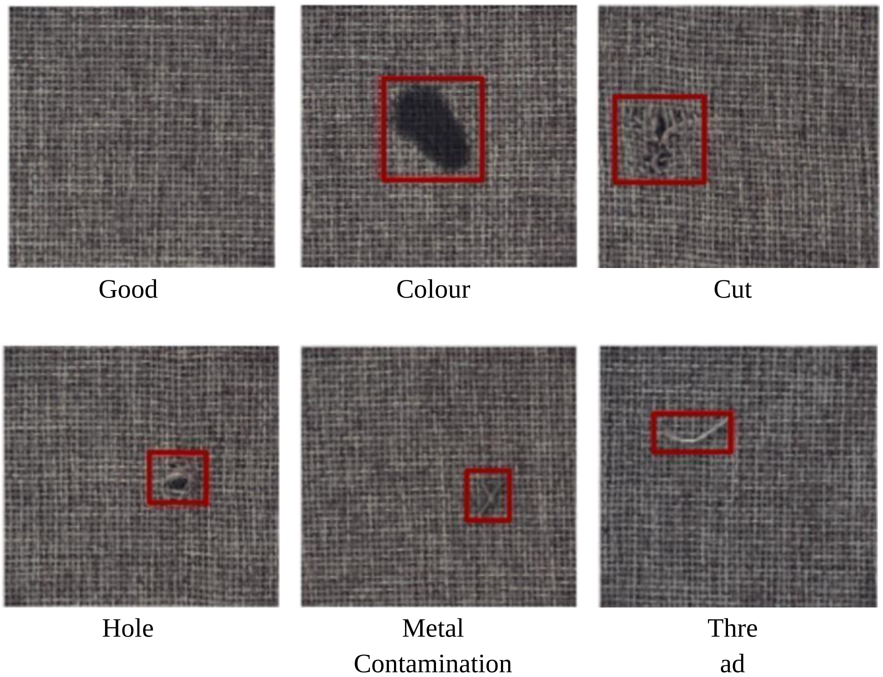

Fig 1: Real time image samples of defects

3. Methodology

The collected fabric dataset undergoes an initial data integrity check to identify and address any corrupted pixels, ensuring data quality [11,12]. Subsequently, a series of preprocessing steps are applied to prepare the dataset for use in training and testing a fabric defect detection system. The key steps of this process are as follows:

1. **Preprocessing Techniques**: The training dataset's fabric images go through a number of preparation steps. Examples of such methods include:
   - Scaling images to a uniform size before training can increase computing efficiency and maintain a high level of reliability [13].
   - To help neural networks train, pixel normalization ensures that image pixels all fall within the same range of values.
   - Dimension Reduction: A technique for decreasing the number of dimensions in a dataset in order to facilitate more straightforward processing. [14-15]
2. **Feature Extraction with CNN Model**: For this reason, several researchers use Convolution Neural Network (CNN) models for feature extraction. These models work exceptionally well for describing and analyzing the visual characteristics of fabrics with and without defects. The CNN's feature extraction is then used to create a more informative and discriminatory representation of the fabric images.
3. **Training the System:** The system is trained for defect detection and classification using datasets of both faulty and normal fabric photos. The model is taught to recognize clean (defect-free) and damaged cloth samples during training.
4. **Defect Classification**: After being taught, the system can determine if a given fabric image is "good" (defect-free) or "defective." When an image flaw is found, the system further specifies the type of flaw it is. This may be anything from a color flaw to a cut flaw to a hole flaw to a thread flaw to metal contamination.

Deep Neural Networks (DNN) image processing techniques are used in this method for accurate flaw detection and classification in textiles. Quality control procedures in the textile industry can benefit substantially from the system's capacity to automatically detect and classify fabric flaws. You can get a good idea of the layout and operation of this fabric flaw detection system from Figure 2.

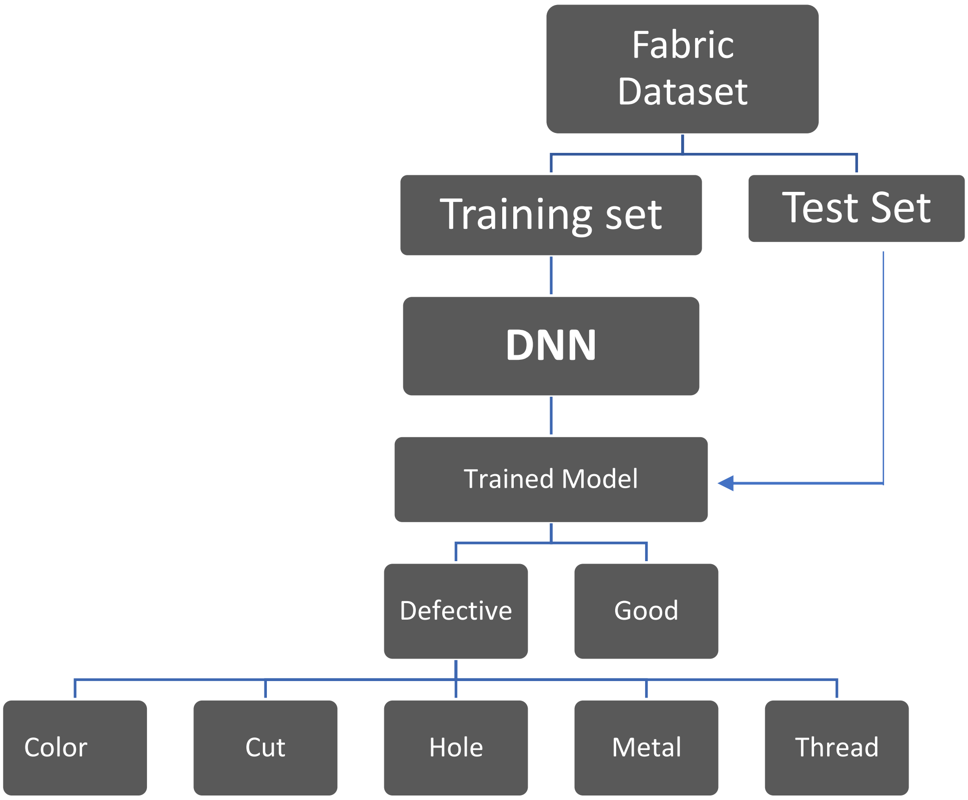

Fig 2: Fabric defect detection system

3.1 Image Pre-Processing Techniques

1. When preparing images for machine learning models, image preprocessing is essential because it increases the reliability of the data and the efficiency of the subsequent image processing methods. To ensure that the machine learning model is trained on high-quality data, it is necessary to remove distracting artifacts and improve important image attributes. Several standard procedures for preparing images for further analysis are used here:
2. Resizing the Image
3. Denoising
4. Segmentation
5. Morphing

Cleaning and standardizing the dataset in this way makes it more suitable for machine learning methods. In addition, they can help improve model training efficiency, which in turn helps boost performance. In order to increase the model's generalization to new situations and data changes, data augmentation techniques can be used to further enhance the dataset by adding variances in the photos.

3.2 Data Augmentation

In machine learning, data augmentation is a helpful strategy, especially for smaller datasets. It entails applying multiple modifications to the photos to produce diverse copies of the original data. The fundamental objective of data augmentation is to expand the dataset's useful size, which in turn can enhance the model's generalizability to new settings and data variances.

When applied to the original dataset, these augmentation methods generate much more data, with 108,000 photos added for each of the five augmentation methods. This expanded dataset gives the model more

data from which to draw insights, making it better equipped to deal with complex situations that arise in the real-world Indian textile sector. Rotation, horizontal and vertical flipping, blurring, and noise addition are the five augmentation techniques used in our dataset.
Sample photos enhanced with the aforementioned "Hole" class methods are shown in Figure 3.

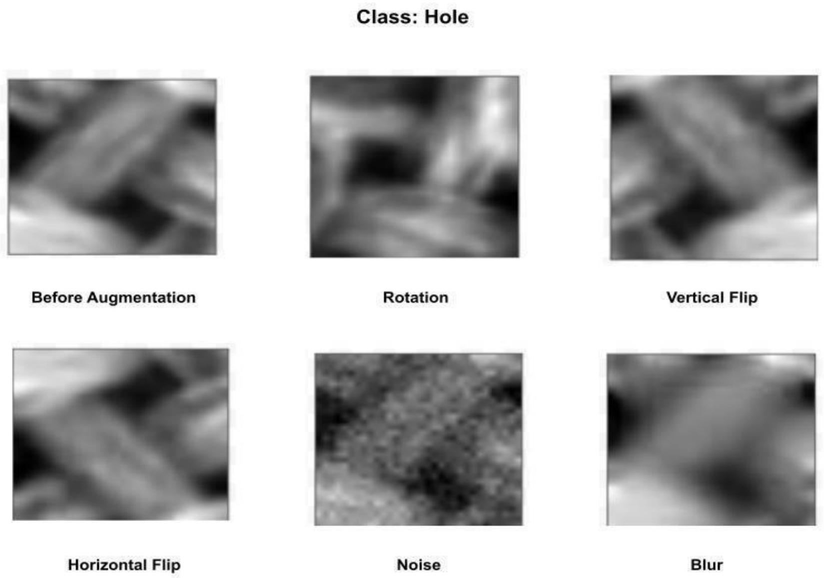

Fig 3: Image patches after data augmentation

3.3 Network Architecture

The ResNet-50 architecture adds an extra 12 weighted layers to the original ResNet design (ResNet-34). The revolutionary notion of ResNet (short for Residual Network) allowed for the training of very deep convolutional neural networks (CNNs) by overcoming the vanishing gradient problem. One of ResNet's most distinctive features is its reliance on "shortcut connections," also known as "skip connections" or "identity mappings."

When it comes to training deep networks efficiently, ResNet-50 is an architecture that uses residual blocks with shortcut connections and a bottleneck design. The success of deep learning in computer vision applications can be directly attributed to its superior performance on tasks like picture classification.

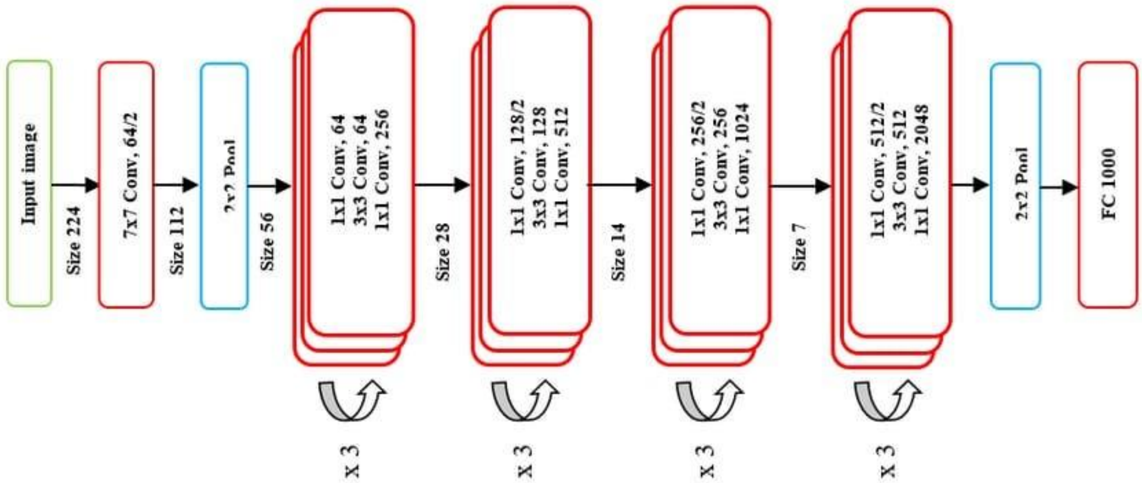

Fig 4: Architecture of ResNet50

Computer vision tasks, especially image identification, have benefited greatly from ResNet-50, a deep convolutional neural network (CNN) architecture. Key features and developments of ResNet-50 are outlined below.

1. **Overlapping Max Pooling**: ResNet-50 makes use of max pooling layers that overlap with one another. The performance of the network is enhanced by the overlapping windows present in these levels. When comparing overlapping and non-overlapping pooling windows of size 2x2 with a stride of 2, the top-1 error rate is reduced by 0.4% and the top-5 error rate is reduced by 0.3%.
2. **Fully Connected Layers**: After the convolutional layers of ResNet-50, you'll find the fully linked layers. In order to accomplish classification tasks, these fully connected layers multiply their inputs by trainable weight vectors and biases. A softmax classification layer with 1000 class labels is used in ResNet-50.
3. **Rectified Linear Unit (ReLU) Activation**: All of the nodes in the network use the ReLU activation function. When compared to saturated activation functions like tanh or sigmoid, ReLU allows for quicker training and aids the network in overcoming the vanishing gradient problem. Learning is facilitated and performance is enhanced.
4. **Bottleneck Design**: The convolutional layers of ResNet-50 use a "bottleneck" design. Each of the blocks in this design is made up of 1x1, 3x3, and 1x1 convolutional layers. By minimizing the number of parameters and the computational complexity, these 1x1 convolutions speed up and improve the effectiveness of the training process.
5. **Depth and Layer Count**: ResNet-50 is a 50-layer deep neural network. It's important to remember that skip connections were first introduced in the original ResNet architecture, making it possible to train networks with hundreds of hidden nodes. In the world of deep learning, this was a major development.
6. **Residual Blocks**: ResNet is composed primarily of residual blocks, which are comprised of the skip connections. By providing the network with these residual blocks, it is able to learn residual features, which are the deviations between the input and the desired output. As a result, the vanishing gradient problem can be mitigated and deeper neural networks can be trained.
7. **Innovative Breakthrough**: ResNet-50 and the ResNet architecture were first introduced in a 2015 work titled "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. This research was a watershed moment for the fields of deep learning and computer vision since it made it possible to build extraordinarily deep neural networks.
8. Overall, ResNet-50 is a powerful and widely used CNN architecture that has been successful in a variety of computer vision tasks, including image recognition, object detection, and more. Its innovations, such as skip connections and the bottleneck design, have had a lasting impact on the field of deep learning.

3.4 Optimization Algorithm

The main goal of training a neural network is to maximize accuracy while minimizing error through a series of iterative adjustments to the network's weights. This can be achieved by carefully picking an optimization method that will govern how the model adjusts its weights during the training process. Adaptive Moment Estimation (Adam) is an approach for optimization that is frequently substituted for standard implementations of the Stochastic Gradient Descent (SGD) method. Adam is a well-known optimizer because it overcomes some of the drawbacks of both the AdaGrad and RMSProp algorithms. Specifically, it keeps track of a moving average of previous squared gradients (denoted "vt") and a moving average of past gradients (denoted "mt"), both of which decay exponentially.

Here's a brief overview of how Adam works:

1. **Exponentially Decaying Averages**:

This moving average remembers prior gradients, and its name stands for "First Moment Estimate," or "o mt." It aids in redistributing the load in a manner that minimizes waste.

As a moving average, o vt (Second Moment Estimate) keeps tabs on the past squared gradients. It aids in adjusting the rates of learning for each variable.

Initialization: Adam initializes the moving averages mt and vt to zero.

Parameter Updates:

Adam calculates the weight gradients with respect to the loss function at each iteration. To do this, it takes the current gradients and adds them to the mt and vt moving averages. After taking their starting points into account, Adam calculates mt and vt estimations that are free of bias. The revised estimations are then used to revise the model's weights.

Adaptive Learning Rates: Each parameter's learning rate is adjusted by Adam dividing mt by the square root of vt. This implies that greater gradient parameters will have lower effective learning rates, protecting against training overshoot.

Combining RMSProp's dynamic learning rates with AdaGrad's gradient history, the Adam optimizer achieves optimal performance. As a result, it may be used for a variety of deep learning problems and, in many cases, achieves faster convergence and better training performance than standard SGD.

While Adam optimizer implementations may differ slightly in the details of how mt and vt are updated, they all adhere to the same general principles laid out in your text: where is the learning rate, 1 and 2 are exponential decay rates for mt and vt, and is a small constant for numerical stability. During training, these equations guarantee that the optimizer can effectively respond to changes in the gradient dynamics and move about the parameter space.

$$m_t=\beta_1 m_{t-1}+(1-\beta)g_t \tag{1}$$

$$v_t=\beta_2 v_{t-1}+(1-\beta)g_t^2 \tag{2}$$

Where β1, β2 are the decay rates and g_t is the current gradient.

4. Experimental Results

4.1 Performance Evaluation

RESNET50 provides better performance than the other networks with an accuracy of 96.42%. The test accuracy of various architectures is given inTable1.

Table 1: Accuracy of Various Architectures

Model	Accuracy, %
ResNet-50	96.42
AlexNet after augmentation	92.60
Differential evolution	93.40
Co-occurrence matrix	90.78
Mathematical morphology	90.41
CNN with AlexNet	81.20
CNN	73.80
MLP	74.13

4.2 Learning Curve

Figure 5 indicates the decrease in training loss and validation loss and Figure 6 indicates the increase in the training accuracy and validation accuracy with subsequent epochs. After training the ResNet model for 27 epochs, the training and validation losses were 0.268 and 0.204, respectively, and the training and validation accuracies were 90.10% and 96.42%, respectively

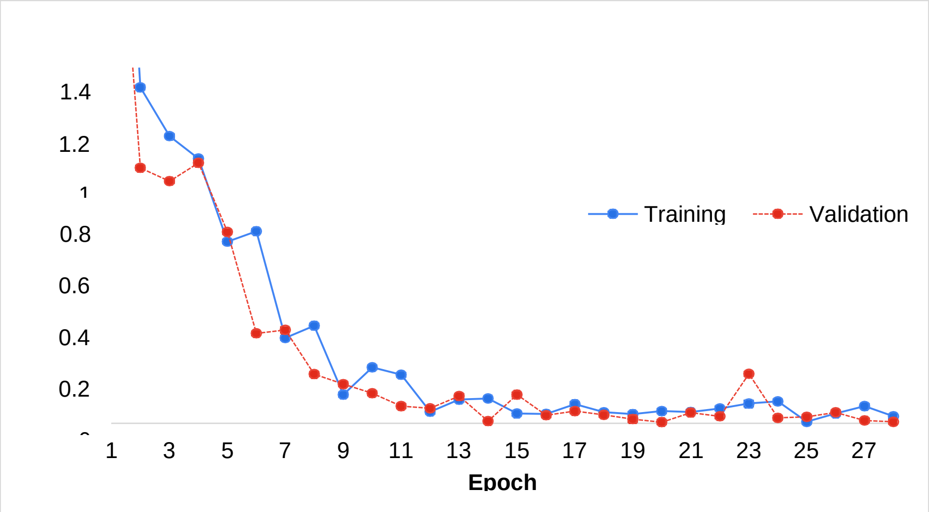

Fig 5: Training and Validation loss

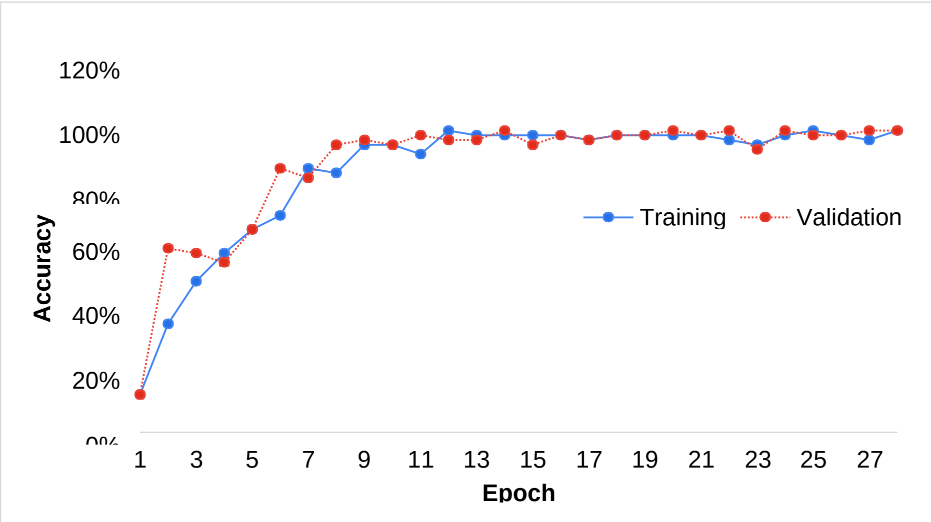

Fig 6: Training and Validation Accuracy

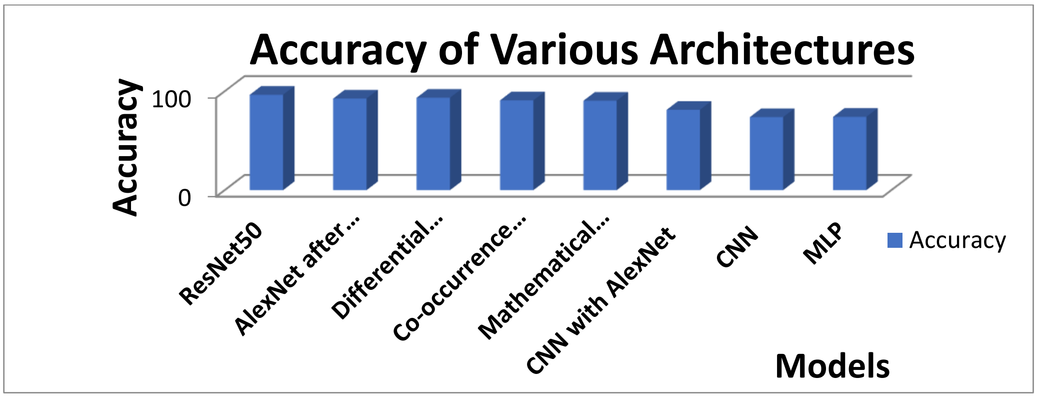

Fig 7: Accuracy of Various Architectures

4.3 Error Matrix

When assessing the efficacy of a machine learning model, the confusion matrix proves very useful for classification tasks. It provides a quick overview of how many times the model was right and how many times it was wrong. The confusion matrix appears to be employed in your setting to evaluate the efficacy of a ResNet-trained model.

When evaluating a model's efficacy, these indicators are essential, especially for binary classification jobs. The confusion matrix can be used to calculate a number of useful measures for assessment, such as:

Sensitivity (True Positive Rate or Recall) quantifies how often the model produces accurate predictions for true positive cases. It is the ratio of true positives to true negatives and measures the accuracy of the model.

Specificity (True Negative Rate): This metric assesses how accurately the model identified "negative" cases. The accuracy with which the model properly recognizes negative situations is represented by this metric, which is computed as TN / (TN + FP).

Accuracy of the model's positive predictions is measured by its Positive Predictive Value (Precision). It is the fraction of times the model's positive predictions are right, and it is computed as TP / (TP + FP).

• NPV, or Negative Predictive Value, measures how well a model does at making negative predictions. It is the fraction of times the model's negative predictions are accurate, and it is computed as TN / (TN + FN).

•The model's strengths and limitations in correctly identifying different scenarios can be evaluated using these measures. Specifically, they help you weigh the costs and benefits of different approaches to positive/negative case identification and the avoidance of Type 1 and Type 2 errors.

Table 2: Confusion matrix of the trained ResNet

ACTUAL CLASS

	Colour	Cut	Good	Hole	Metal	Thread
Colour	94	1.5	2.5	1.1	0.82	0.7
Cut	1.2	94.5	0.9	2.1	1.3	1.2
Good	2.1	1.3	94.1	1.2	0.48	1.1
Hole	1.1	1.5	1.5	94.8	0.8	1.2
Metal	0.84	0.37	1.68	0.7	95.3	1.45
Thread	0.86	1.2	0.41	0.81	1.5	95.3

PREDICTED CLASS

4.4 Testing and Cross-Validation

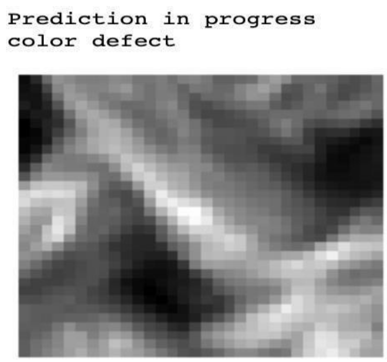

Fig10: Output from testing the trained network

As can be seen in Figure 8, the network performed admirably when presented with a sample image depicting a color flaw.

$$\text{Accuracy (Acc)} = \frac{\text{TN+TP}}{TN+FP+FN+TP}$$

5. Conclusion

Using transfer learning on a ResNet pretrained network, your deep learning model performed exceptionally well, reaching an astonishing 96.40% accuracy on all five defect classifications (cut, color, hole, thread, metal contamination). The consequences of this success for the textile industry are substantial, as they may result in higher fabric quality and lower production costs. Efficiency, cost reduction, and quality improvement are just a few of the potential gains from using a DL-based classifier in the textile production process.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 computer vision research paper titled 'Deep Residual Learning for Image Recognition'

[2] NC Sandhya, Nihal Mathew Sashikumar, M Priyanka, Sebastian Maria Wenisch, Kunaraj Kumarasamy, Automated Fabric Defect Detection and Classification: A Deep Learning Approach Textile & Leather Review. 2021; 4:315-335 https://doi.org/10.31881/TLR.2021.24

[3] Sengottuvelan P, Wahi A, Shanmugam A. Automatic Fault Analysis of Textile Fabric Using Imaging Systems. Research Journal of Applied Sciences.2008;3(1):26-31.

[4] National Portal of India. Available from: http://niti.gov.in/weaving-way-indian-textile-industry

[5] Patel J, Jain M, Dutta P, Detection of Faults Using Digital Image Processing Technique. Asian Journal of Engineering and Applied Technology.2013;2(1):36-40. https://doi.org/10.51983/ajeat-2013.2.1.644

[6] Hanbay K, Talu MF, Özgüven F. Fabric defect detection systems and methods—A systematic literature review. Optik.2016;127(24):11960-11973. https://doi.org/10.1016/j.ijleo.2016.09.110.

[7] Cao G, Ruan S, Peng Y, Huang S, Kwok N. Large-Complex-Surface Defect Detection by Hybrid Gradient Threshold Segmentation and Image Registration. IEEE Access. 2018, 6:36235-36246.https://doi.org/10.1109/ACCESS.2018.2842028

[8] LiY, Zhang C. Automated vision system for fabric defect inspection using Gabor filters and PCNN. SpringerPlus5.2016,765.https://doi.org/10.1186/s40064-016-2452-6

[9] Vermaak H, Nsengiyumva P, Luwes N. Using the Dual-Tree Complex Wavelet Transform for Improved Fabric Defect Detection. Journal of Sensors.2016; 9794723. https://doi.org/10.1155/2016/9794723

[10] Bergmann P, Fauser M, Sattlegger D,Steger C. MVTec AD—A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition.2019June16–20LongBeach, California; Computer Vision Foundation;2019:9584-9592.https://doi.org/10.1109/CVPR.2019.00982

[11] Kunaraj K, Wenisch SM, Balaji S, Don Bosco MFP. Impulse Noise Classification Using Machine Learning Classifier and Robust Statistical Features. In: Smys S, Tavares J, Balas V, Iliyasu A (eds)Computational Vision and Bio-Inspired Computing. ICCVBIC 2019. Advances in Intelligent Systemsand Computing. 2020; vol 1108. Springer, Cham. p.631-644.https://doi.org/10.1007/978-3-030-37218-7_72

[12] Kumarasamy K, Wenisch SM, Balaji S, Suriya LJJ, Jerlin A, Rajkumar R. Improving Impulse Noise Classification Using Ensemble Learning Methods. Series: Advances in Intelligent Systems andComputing.2021; 1299:187-199.https://doi.org/10.1007/978-981-33-4299-6_16

[13] Mathematics. Resizing and scaling image. Available from:https://math.stackexchange.com/questions/3078121/resizing-and-scaling-image

[14] Han YJ, YuHJ. Fabric Defect Detection System Using Stacked Convolutional Denoising Auto-Encoders Trained with Synthetic Defect Data. Applied Sciences. 2020; 10(7)2511. https://doi.org/10.3390/app10072511

[15] Ma J, Yuan Y.Dimension reduction of image deep feature usingPCA. Journal of Visual Communication and Image Representation.2019;63,102578.https://doi.org/10.1016/j.jvcir.2019.102578

[16] T Bhaskar, D Vasumathi, "A reversible data embedding scheme for MPEG-4 video using non-zero AC coefficients of DCT" 2013 IEEE ICCICR-2013,1-4.