

Detection Of Faults In Fabric Using Image Processing In MATLAB

Dr. K. R. Desai¹, Mr. Shree S. Kesarkar²

¹Professor, Department of Electronics & Telecommunication, Bharati Vidyapeeth's College of Engineering, Kolhapur, Maharashtra, India.

²Student, Department of Electronics & Telecommunication, Bharati Vidyapeeth's College of Engineering, Kolhapur, Maharashtra, India.

Abstract - In the textile industry, quality control is essential to competing in the fiercely competitive global market. In these textile industries, faults in the fabric are discovered through tedious, inaccurate manual testing. The profits of the textile business have decreased as a result of fabric faults, which cause unfavorable losses. To solve these issues and speed up and simplify defect diagnosis, an image processing system has been implemented. When the patterns and textures of a fabric are changed, the fabric's appearance and physical properties are affected, which is known as a defect. Either "minor" or "major" problems could exist. The faulty area must be located in order to cut the fabric correctly. In order to easily locate Fabric Texture defects, this project aims to create a tool. The approach outlined in this proposal for labor lessens the physical effort needed from the employees performing the manual examination. To process images in this study, "MATLAB" is employed. Along with centimeters, pixels are used to measure the fault's size. The fault's data representation and its size are displayed using a histogram and the GUI, respectively.

Key Words: Fabric Faults, Flaw Measurement, Image Processing, MATLAB, Textile Industry

1. INTRODUCTION

In every industry, the quality factor is essential. In a similar vein, a textile manufacturer must maintain production quality. Hand testing is still performed today to assess the fabric's quality. The manual inspection result, however, falls short of expectations due to complacency and disinterest. With the help of highly qualified inspectors, only 70% of fabric faults can be identified. A basic cloth's defects can be manually detected in about 90% of cases, but complicated materials are more difficult. Consequently, automated fabric fault identification can lead to quick and high-quality product manufacture. Recently, automated fabric inspection has attracted a lot of attention.

The textile business, like other sectors, strives for high quality and mass manufacturing in order to meet consumer demands and reduce losses resulting from poor fabric quality and flaws. Rapid product flaw detection is necessary due to the rapid pace of manufacturing. The larger roll sizes used to weave the cloth may result in lower production quality before any inspection. In this industry, there are

some companies who prioritize just-in-time delivery over creating high-quality products.

The main goal of today's textile business is to produce First Quality, a high-quality fabric. This premium fabric is devoid of any serious faults as well as any surface or minor structural imperfections. After the first quality, the remaining fabric is recognized as second-class fabric. A few serious defects as well as countless structural or surface flaws in this kind of fabric could cost the manufacturer money. When selling second-quality fabric, just 45% to 65% of the price of first-quality cloth is charged. Manufacturing moves quickly, therefore the manufacturer needs to be able to spot problems, figure out what caused them, and fix them very away. The load on the manufacturer's inspection teams may increase as a result, and the output of second-quality fabric may decline while first-quality fabric production rises. Compared to hiring numerous employees, investing in automated fabric fault identification is less expensive. Real-time fabric inspection is a challenging undertaking because there are many various types of problems based on the weaving process employed and post-production issues like holes and oil stains. While some defects can be detected by sophisticated loom machines on their own, a large number of flaws still require scrutiny after the weaving stage. Therefore, enhancing the quality of materials requires automated fabric inspection. The automated method for spotting fabric flaws will take care of imperfections in the fabric including holes, scratches, stretch, fly yarn, unclean patches, cracked points, misprints, and color bleeding, among others. If these defects are not discovered, the fabric businesses run the danger of losing money.

Automated fabric fault identification is feasible by utilizing MATLAB's image processing method. With minimal work and expenditure, you might get superb precision using this method. The faults in a captured image are discovered by using this MATLAB-based image processing technique. After applying the noise filtering, histogram, and thresholding processes to the image, the output is created. The fabric image can be compared to other common textured images. It is vital to identify the precise position and severity of the faults when deciding that the cloth is faulty. Patterns can be detected anywhere in the visual area, regardless of their size, brightness, or direction. This automation strategy can be used to address these problems

2. PROPOSED WORK

The proposed work can be used by both small and large textile businesses. A single person with modest computer abilities could operate and spot production issues. Human error can be removed by employing a conventional technique, such as manual testing. This suggested effort will make it possible to find and fix incredibly small problems that are invisible to the human eye. The task's defined outcomes and histogram will be shown. The thresholding function is the result. The size of the defect will also be shown by the software. Accurate waste reduction techniques support higher manufacturing quality. The textile industry needs this proposed work as a preventative measure because there are numerous different types of fabric flaws, such as holes, scratches, stretch, fly yarn, dirty spots, slub, cracked points, and colour bleeding that, if improperly detected, can have a significant negative impact on the manufacturing process.

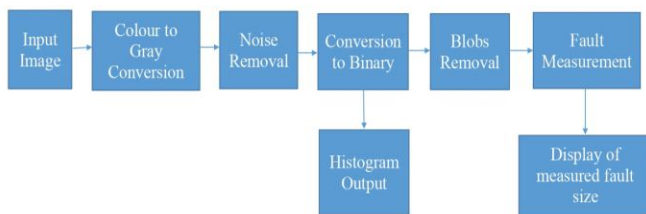


Fig-1: Block Diagram to identify the fault in fabric using image processing in MATLAB

With the use of color photographs that are taken of the cloth being tested, the flaw in the fabric is located and measured. The block diagram shown in Fig. 1 helps to illustrate the entire procedure.

The input for the operation of this technique is a color image. To make algorithms and the extraction of picture descriptors simpler, colour images are transformed to an 8-bit gray-level format. Noise is caused by variations in lighting, the structure of the fabric, and imperfections in the fabric; it is eliminated by filtering the image with a Weiner Low Pass Filter or Median Filter. To clearly identify the problematic area, this filtered image is translated into binary form.

Binary pictures can be used to indicate an image's region of interest. The image is changed into a bi-level image via thresholding. Thresholding is used to isolate the pixels in an image that represent the item. On the thresholded image, the flaws are visible.

The flaws are displayed as data in a histogram output. A histogram is a sort of graph that shows how information is spread visually. A histogram can estimate the probability distribution of a given variable. The histogram is used to categorize the fabric's flaws. The errors are corrected by utilizing the histogram's output.

It is required to eliminate the extremely small noise that is present in the image in order to measure the defect's approximate size. In order to increase the precision of the diameter measurement, we must therefore extensively clean

the image. Filling the image also eliminates small holes. There are little blobs on the edge of the image even after noise has been removed.

Using several MATLAB instructions, blobs smaller than 7 pixels and blobs on the border are deleted. The preferred fault can be chosen for measurement, and the value can be entered in the command box. As a result, the output displayed in the command window will be the centimeter-based measurement of the fabric flaws. It is useful to keep a record of faults since the displayed size of the fabric defect is thought to be the approximate size.

3. WORKING MODULES

Using the imread function in MATLAB, you may read a picture. An image file is read from disc by the imread function, which then returns a matrix containing the image. An image can be read using the simple syntax shown here: `imread('image_file_path')`

The path to the image file you want to read should be substituted for 'image_file_path' in the code above. A relative or an absolute path is possible. Numerous image file formats, including JPEG, PNG, BMP, and TIFF, are supported by the imread function. MATLAB uses the file extension to determine the file format automatically. After being read, the image is stored in a matrix with each element denoting a pixel value. The image's resolution determines the matrix's size. The matrix for a grayscale image will be M by N, where M is the number of rows and N is the number of columns. The matrix for a color image will be M by N by 3, indicating the red, green, and blue color channels. The image can then be subjected to a number of processes utilizing MATLAB's image processing tools, including image enhancement, filtering, and analysis. To obtain the fault size in cm, the fault size noted on the defective image is entered in the command box. The image's flaw can be seen as pixels.

There are various ways to change a colour image to grayscale in MATLAB. Using the image's red, green, and blue channels as a weighted average is one of the often employed techniques. The colour image is changed to grayscale by using the formula.

`grayImage = rgb2gray(colorImage).`

To determine the intensity of each pixel in the final grayscale image, this function computes the weighted average of the RGB channels.

The Wiener filter is implemented via the `wiener2` function, a built-in image filtering tool in MATLAB. The Wiener filter is a well-known method for picture restoration and denoising. By reducing the mean square error between the estimated and original images, it seeks to estimate the original image from a cluttered or degraded version. Utilizing the `wiener2` function requires the following syntax:

`J = wiener2(I, [m n], noiseVar)`

I is the input RGB or grayscale image. The local neighborhood size used for filtering is specified by [m n].

The function uses a default neighbourhood size of [3 3] if this optional parameter is not given. noiseVar is the additive noise's variance. Furthermore, it has an optional value. In the event that it is not supplied, the function guesses it from the input image. It is also an optional parameter. The wiener2 function uses the input image to estimate a local image variance, which it then uses to conduct adaptive noise reduction. Then, using the estimated variance, a Wiener filter is applied to each pixel in the image. In comparison to fixed filters like Gaussian smoothing, the Wiener filter offers better noise reduction because it changes its filter coefficients to the local noise characteristics.

Utilize the imbinarize function to threshold the grayscale image and create a binary image. Based on the image and the intended outcome, you can set an acceptable threshold value. As an alternative, you can utilize adaptive thresholding techniques like 'imbinarize' with the 'adaptive' option.

A binary image's histogram gives a visual depiction of how the image's pixel values are distributed. Since there are only two potential pixel values in a binary image (usually 0 and 1), the histogram will display the frequency at which each value occurs. The vertical axis of the histogram represents the frequency or quantity of pixels that have a specific value, while the horizontal axis displays the values of the pixels. A binary image will have two bars that represent the frequency of 0s and 1s. The function used is `histogram = imhist(binaryImage);`

Based on their area (number of pixels), related components of a binary picture can be filtered using MATLAB's `bwareafilter` function. It's frequently used to eliminate little or big items from binary images, keeping only the components that are wanted according on their area.

`filteredImage = bwareafilter(binaryImage, areaRange);`
 where: The input binary image with connected components is called `binaryImage`. The required range of component areas is specified by the two-element vector `areaRange`. The lower threshold is the first element, while the upper barrier is the second. Areas outside of this range in components will be filtered away.

The following actions are carried out by the `bwareafilter` function: Label the connected components first: Using the `bwlabel` function, it first labels the connected components in the input binary image. Each component has its own specific label. Component area computation: Using the `regionprops` function and the 'Area' attribute, it then determines the area of each connected component. The locations are kept in a vector. Sort the elements: The function decides which components to maintain and which to eliminate based on the supplied `areaRange`. Areas outside the defined range for a component are discarded. The function creates a new binary image in which only the desired related components are kept and all other pixels are set to 0. This is the filtered image. The `imdistanline` function is used to estimate distances in a picture interactively. You can use it to draw a line on an image and get the length in pixels or a chosen physical unit. It activates the interactive mode by invoking `imdistanline` without any arguments. A draggable measurement tool that

shows the distance in pixels as you move it will emerge when you click and drag on the image to draw a line. The formula 'H*0.0264583333' is used to convert the measured fault size to centimeters. The command pane displays the measured defect in millimeters.

4. RESULTS

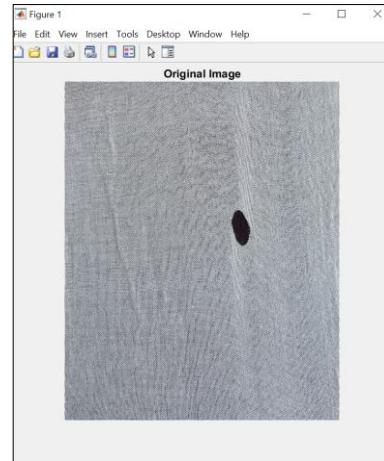


Fig-2: Original Image of fabric under test

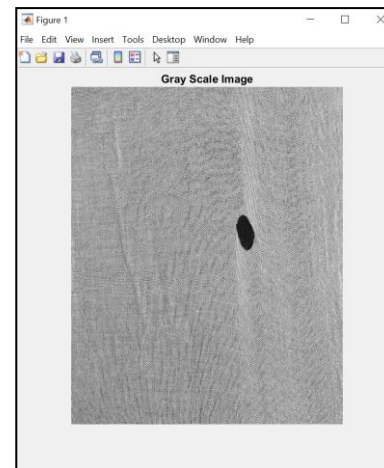


Fig-3: Gray Scale Image of fabric under test

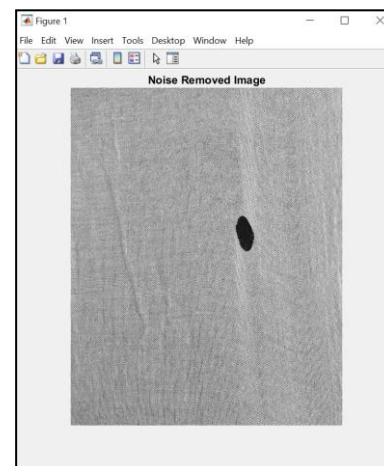


Fig-4: Noise Removed Image of fabric under test

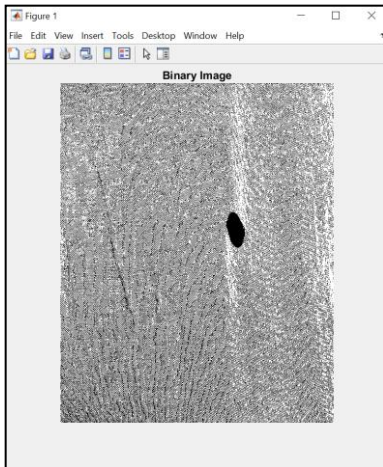


Fig-5: Binary Image of fabric under test

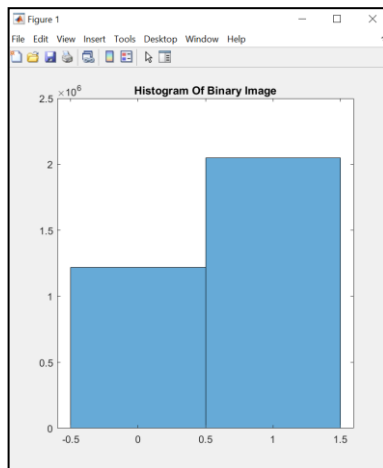


Fig-6: Histogram of Binary Image

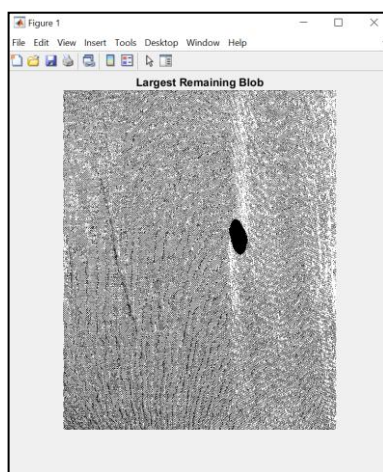


Fig-7: Image showing largest remaining blob/fault

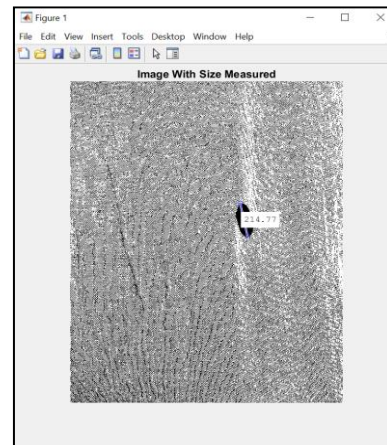


Fig-8: Image with measured fault size

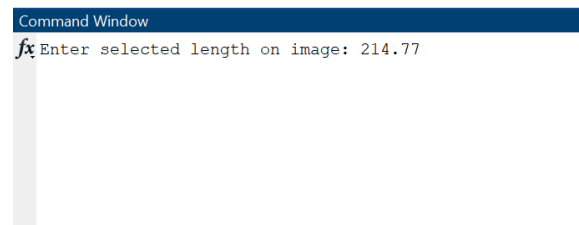


Fig-9: Measured size entered in command window in pixels

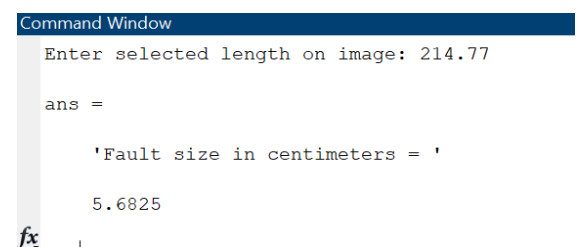


Fig-10: Fault size in centimeters

5. CONCLUSIONS

In this paper, a supervised defect detection approach has been used to identify a class of fabric faults. The experimental results on several flawed images show that it is possible to spot fabric faults. The size of the fault is quantified in pixels as well as in centimeters, and the faults in fabric can be discovered and their data represented using histograms. If there is no flaw on the cloth being tested, there will be only one data representation in the histogram. Using the MATLAB App feature, all of this functionality may be performed in a GUI.

ACKNOWLEDGEMENT

We are grateful to our mentor Dr. K. R. Desai (Electronics and Telecommunication Engineering Department, BVCOEK) for giving us the opportunity to work on this project and for his continued intellectual support in the form of his astute recommendations, instructions, and constructive criticism. We are grateful that the members of the Electronics and

Telecommunication Engineering department accepted our project-related recommendations. We want to sincerely thank everyone who helped us finish this project, whether they were directly involved or not.

REFERENCES

- [1]. Padmavathi S., Prem P., and Praveen D., 'Locating Fabric Defects Using Gabor Filters', International Research Engineering Journal of Scientific & Technology (IJSRET), Volume 2 Issue 8 pp 472-478 November 2013 ISSN 2278-0882.
- [2]. Utkarsha Singh, Teesta Moitra, Neha Dubey and Mrs M.V.Patil, 'Automated Fabric Defect Detection Using Matlab', International Journal of Advanced Technology in Engineering and Science, Volume No.03, Issue No. 06, June 2015, ISSN (online): 2348 – 7550.
- [3]. R.Thilepa and M.Thanikachalam, 'A Paper On Automatic Fabrics Fault Processing Using Image Processing Technique In Matlab', Signal & Image Processing: An International Journal (SIPIJ) Vol.1, No.2, December 2010.
- [4]. Miss. Pallavi A. Patil, Dr. M. S. Kumbhar, 'Fabric Defect Detection Using Image Processing Technique' International Conference on Academic Research in Engineering and Management, IETE, 30th April 2017, (ICAREM-17), ISBN: 978-93-86171-43-6.
- [5]. A. Kumar and G. K. H. Pang, "Defect detection in textured materials using Gabor filters," in IEEE Transactions on Industry Applications, vol. 38, no. 2, pp. 425-440, March-April 2002. doi: 10.1109/28.993164 keywords: {Gabor filters; Inspection; Fabrics; Textile industry; Automation; Filtering; Quality assurance; Industry Applications Society; Consumer electronics; Throughput}, URL:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=993164&isnumber=21415>Tan jim Mah mud, Juel Sikder and Rana Joyti Chakma, 'Fabric Defect Detection System', Research gate Conference Paper, February 2021, DOI: 10.1007/978-3-030-68154-8_68.
- [6]. S. L. Bangare, N. B. Dhawas, V. S. Taware, S. K. Dighe, P. S. Bagmare, 'Fabric Fault Detection using Image Processing Method', International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 4, April 2017.
- [7]. S.Priya, T. Ashok Kumar, Dr. Varghese Paul, 'A Novel Approach to Fabric Defect Detection Using Digital Image Processing', Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).
- [8]. Engr. Anum Khowaja, Engr. Dinar Nadir, 'Automatic Fabric Fault Detection Using Image Processing', 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS).
- [9]. Jagruti Mahure, Y.C.Kulkarni, 'Fabrics Fault Processing Using Image Processing Technique in MATLAB', IJCST Vol. 4, Issue 2, April - June 2013 ISSN: 0976-8491 (Online) | ISSN: 2229-4333.
- [10]. Matthew Wesolowski, 'Using MATLAB to Measure the Diameter of an Object within an Image'.
- [11]. R. C. Gonzalez, R. E. Woods, S. L. Eddins, "Digital Image Processing using MATLAB", ISBN 81-297-0515-X, 2005, pp. 76-104,142-166.
- [12]. Kenneth R. Castleman, "Digital image processing", Tsinghua Univ Press, 2003.
- [13]. [Histogram plot - MATLAB.mht](#)
- [14]. <https://in.mathworks.com/help/images/apply-gaussian-smoothing-filters-to-images.html>
- [15]. <http://matlab.izmiran.ru/help/toolbox/images/enhance6.html#:~:text=To%20create%20an%20image%20histogram.histogram%20based%20on%2064%20bins>
- [16]. <https://in.mathworks.com/help/images/ref/medfilt2.html>
- [17]. <https://in.mathworks.com/help/images/ref/regionprops.html>
- [18]. <https://sisu.ut.ee/imageprocessing/book>
- [19]. <https://in.mathworks.com/help/images/linear-filtering.html>
- [20]. <https://in.mathworks.com/help/images/image-type-conversions.html>
- [21]. <https://www.onlineclothingstudy.com/2019/02/classification-of-fabric-defects.html#:~:text=A%20fabric%20defect%20corresponds%20to,faulty%20yarns%20or%20machine%20spoils>.