

Automated Texture Defect Detection using the Non-Extensive Entropy with Gaussian gain

A Major Project Report submitted in the partial fulfilment of the
requirements for the award of the degree of

MASTER OF TECHNOLOGY

(INFORMATION SYSTEMS)

Submitted by:

MONIKA SHARMA

(Roll No. 2K13/ISY/13)

Under the esteemed guidance of

Dr. SEBA SUSAN

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

SESSION:2013-2015

CERTIFICATE

This is to certify that work entitled “**Automated Texture Defect Detection using the Non-Extensive Entropy with Gaussian gain**” submitted by **Monika Sharma (2K13/ISY/13)** to Delhi Technological University, Delhi for the award of the degree of Master of Technology is a bonafide record of research work carried out by her under my supervision.

The content of this thesis, in full or parts, have not been submitted to any other institute or university for the award of any degree or diploma

Dr. Seba Susan

Project Guide

Assistant Professor

Department of Computer Science and Engineering

Delhi Technological University

Shahbad Daulatpur, Bawana Road, Delhi-110042

Date:-----

ACKNOWLEDGEMENT

I would like to thank my project guide, **Dr. Seba Susan** for her valuable guidance and wisdom in coming up with this project. I humbly extend my words of gratitude to **Dr. O. P. Verma**, Head of Department, and other faculty members of COE department for providing their valuable help and time whenever it was required. I thank all my friends at DTU who were constantly supporting me throughout the execution of this thesis.

Special thanks to the Almighty Lord for giving me life and the strength to persevere through this work. Last but not least, I thank my family for believing in me. May you all be blessed.

Monika Sharma

Roll. No. 2K13/ISY/13

M.Tech (Information Systems)

E-mail: monika.sh1989@gmail.com

Department of Computer Science and Engineering

Delhi Technological University

ABSTRACT

Inefficient processes in industries can cost a lot of time, money and customer satisfaction. Quality assurance is a very important aspect for industries. As a result, to improve their processes and become more efficient at the global level, they have started to automate their certain tasks for which humans were generally considered to perform them. And one such popular task is Industrial Inspection . As a result, intelligent visual inspection systems are developed to ensure high quality of products in production lines. Advancement in image processing and computer vision techniques has led to the development of Automated Visual Inspection Systems. These systems are required in industries to inspect the manufactured products, so as to identify any discrepancies in them. Thus, they play an important role in industries by ensuring that only good quality products enter into the market because selling defective products in market can lead to large losses for goods manufacturing industry. One such industry is Textile Industry where these systems are of great importance. They identify defects occurring in textile cloth. Several algorithms have been developed based on different approaches so as to identify defects in texture patterns. Presence of complex defects is a major hurdle for many such algorithms and that is why new ideas keep on emerging to develop more efficient algorithms which can identify such type of defects in texture patterns. On the same lines, this major project report presents a new algorithm based on a new approach to identify complex defects in texture in a more efficient and accurate manner.

Table of Contents

Certificate.....	i
Acknowledgement.....	ii
Abstract.....	iii
Table of Contents.....	iv
List of Figures.....	vii
List of Tables.....	viii
1. Introduction.....	2
1.1 Fabric Defects.....	2
1.2 Traditional Inspection.....	3
1.3 Automation for Inspection.....	3
2. Different Types of Approaches of Defect Detection.....	6
2.1 Defect Detection using Co-occurrence Matrix Features.....	6
2.2 Defect Detection using Edge Detection.....	7
2.3 Defect Detection using Bi-Level Thresholding.....	8
2.4 Defect Detection using Neural Networks.....	8
2.5 Defect detection using Discrete Fourier Transform.....	9
2.6 Defect Detection using Morphological Operators.....	10
2.7 Defect detection using Gabor Filter.....	10
3. Comparison Method – 1 : Texture Defect Detection using Segmentation proposed by K.N. Sivabalan, Dr. D. Ghanadurai.....	12
3.1 Feature Extraction.....	12
3.2 Defect Detection Methodology.....	12
4. Comparison Method-2: Texture Defect Detection using Regularity Approach by Dimtry Chetverikov.....	15

4.1 Regularity Approach.....	15
4.2 The Contrast Function.....	15
4.3 Computing Directional Regularity.....	16
4.4 Defining the Regularity Feature Vector.....	18
4.5 The Maximal Regularity.....	19
4.6 Outlier detection.....	19
4.7 Implementation.....	21
5. Comparison Method-3: Texture Defect Detection using GLCM.....	23
5.1 Definition of GLCM.....	23
5.2 Features calculated from GLCM.....	24
5.3 Defect Detection Methodology.....	25
6. Proposed Texture Defect Detection using Entropy.....	27
6.1 Definition.....	27
6.2 Implementation.....	27
7. Automated Texture Defect Detection using Entropy.....	30
7.1 Maximum Count Method.....	30
7.2 Maximum Entropy Method.....	30
7.3 GM Distribution Method.....	30
7.4 Combined GM Distribution and Maximum Count Method.....	30
8. Experimental Results and Discussions.....	32
8.1 Environmental Setup.....	32
8.2 Measures.....	32
8.2.1 Measures for objective comparison of Regularity and Entropy Methods on images with Synthetic Defects	32
8.2.2 Measures of objective comparison of Regularity, GLCM and Entropy Methods using Ncorrect and	

Nfalse on images with complex defects.....	32
8.2.3 CPU Time.....	33
8.3 Results.....	33
8.3.1 Results of Segmentation Approach.....	33
8.3.2 Results of Regularity Method (Chetverikov’s Method).....	35
8.3.2.1 Results of calculating Maximal Regularity in some Images.....	35
8.3.2.2 Results of Images with Synthetic Defects.....	36
8.3.2.3 Results of Images with Complex Defects.....	37
8.3.3 Results of GLCM Method.....	41
8.3.3.1 Results of Images with complex defects.....	41
8.3.4 Results of Entropy Method.....	44
8.3.4.1 Results of Images with Synthetic Defects.....	44
8.3.4.2 Results of Images with Complex Defects.....	46
8.3.5 Results of Automated Entropy Algorithm.....	49
8.3.5.1 Results of Max Count Method.....	49
8.3.5.2 Results of Max Entropy Method.....	53
8.3.5.3 Results of GM Distribution Method.....	57
8.3.5.4 Results of Combination of GM Distribution and Max Count Method.....	60
8.3.5.5 Results of Ideal Case.....	64
8.4 Results of Objective Comparison of Chetverikov’s and Entropy Method on Images with synthetic defects.....	68
8.5 Result of Comparison of all the three methods on images with complex defects.....	69
8.6 Result of Comparison of four Automation Methods.....	70
8. Conclusions and Future Work.....	74
References.....	75

List of Figures

Figure 1.1 Some basic fabric defects.....	2
Figure 1.2 Architecture of a typical fabric inspection system.....	4
Figure 2.1 Creation of GLCM.....	6
Figure 2.2 Sobel operator.....	8
Figure 2.3 Block diagram of Fabric Defect Detection using FFN.....	9
Figure 4.1 Typical contrast functions of a random, weak regular and a regular pattern.....	16
Figure 4.2 Maximal regularity values of different kinds of patterns.....	19
Figure 4.3 Principle of outlier detection. Empty and grey circles are outliers.....	20
Figure 5.1 Concept of GLCM.....	23
Figure 5.2 Offset directions in GLCM.....	24
Figure 8.1 Maximal Regularity value of 4 categories of patterns.....	35

List of Tables

Table 2.1 Haralick's 14 Features calculated from GLCM.....	7
Table 8.1 Result of Segmentation approach.....	33
Table 8.2 Result of Regularity Approach on synthetic images.....	36
Table 8.3 Result of Regularity Approach on complex defects.....	37
Table 8.4 Result of GLCM method on complex defects.....	41
Table 8.5 Result of Entropy Method on synthetic defects.....	44
Table 8.6 Result of Entropy method on complex defects.....	46
Table 8.7 Result of Max count method.....	49
Table 8.8 Result of Max Entropy method.....	53
Table 8.9 Result of GM Distribution method.....	56
Table 8.10 Result of combination of GM distribution and max count method.....	60
Table 8.11 Result of Ideal Case of Variable window size Entropy Method.....	64
Table 8.12 Precision, Recall and F-measure values for Chetverikov and Entropy Method on synthetic defects...	68
Table 8.13 Result of Comparison of Regularity, Entropy and GLCM methods on complex defects.....	69
Table 8.14 Result of Comparison of Max Count, Max Entropy, GM distribution and combination of GM distribution and Max count method.....	70

CHAPTER 1

INTRODUCTION

1. Introduction

In modern industrial manufacturing, product inspection is an important feature. In order to maintain the fabric quality in the textile industry, **automatic texture inspection** is crucial. Before automated inspection, industries were dependent on human visual inspection, thus hiring skilled people for identifying defects in the fabric. This was an inefficient method as well costly. Long hours of inspection would exhaust these people and thus reducing their efficiency in their task of detecting defects. Therefore, for the whole process of inspection to be effective, there is requirement of automated texture defect inspection for cost reduction. The texture defect detection algorithm has to be efficient for an automated inspection system. One of the most intriguing and hardest areas of image processing is complex texture defect detection. Therefore in order to develop an **automatic visual inspection system for texture images**, analysing and studying texture patterns is important. [1].

Texture defects cause 80-84% of the defects in the textile industry [2]. Due to the presence of defects in texture, production houses recover only 40-60% of profit from defective quality products [3]. To avoid such losses due to defects, their identification and prevention is necessary. Therefore, there is very need of an automated inspection system which detects the defects in texture patterns and thus helps in improving the product quality, so that productions houses do not incur losses and customer satisfaction is also met. Apart from this, automated defect detection systems reduce the cost of products of defective quality. Earlier, defect detection was done after a sufficient amount of textile is produced. This amount of textile is removed and is sent for defect detection procedure. A better approach would be to detect the defects in texture during the time of production from the machine and if required inform the person in charge of the machine for repair. And this can be done using the image processing techniques for detecting the texture defects. Therefore, it is quite economical to invest in the automatic texture inspection system when we consider its cost and related benefits.

1.1 Fabric Defects

Yarn quality as well as irregular weaving machine operations generally affect the quality of textile. A low quality yarn will lead to certain type of defects such as width inconsistencies, broken ends, hairiness, etc. Apart from poor yarn quality, defects can also be generated by looming or weaving machines if their surrounding conditions are changed. Change in surrounding temperature, humidity, etc can cause change in the normal working of the weaving machines. This can lead to defects such as double yarn, missing yarn, variation of yarn etc. Variations in the tensile strength of one or more strands of the yarn is generally misread as defect due to poor quality of yarn. There are various categories of defects of texture and a minute change in the looming process can lead to the generation of a new type of texture defect. Following diagram shows some basic types of fabric defects:



(a)Double yarn (b) Missing yarn (c) Broken Yarn (d) Variation of yarn

Fig1.1. Some general fabric defects

1.2 Traditional Inspection

Texture defect detection can be done in two different ways. The first method involves the inspection of the weaving process in which it is continuously overseen for the existence of defects. But as the weaving process is quite complex, this kind of inspection is not followed in the textile production industries. The second method involves inspection of the manufactured textile for the existence of defects. The weaving machines in the textile industry manufacture textile at the speed of 0.2-0.4 meters per minute which is 1-2 meters wide. Due to the slow pace of textile production by weaving machines, the textile inspection process cannot be performed efficiently because a human inspector is bound to get distracted after some time and it is quite difficult to focus for a long time continuously to detect defects. The surrounding environment of textile production [2] near the weaving machines is also relatively hostile. In traditional inspection method, a sufficient amount of textile is removed from the weaving machines and placed on the inspection table. When a defect is detected by the human inspector on the moving textile, he stops the operation, registers the defect and its position, and starts the operation again.[3]

1.3 Automation for Inspection

Due to the high cost and inefficient human visual inspection process, there arises the very need for automating the inspection process. The automated inspection system is quite complicated and requires integration of large number of sub-systems and components[4]. Nickolay [5] have shown that utilizing an automated texture inspection system is quite economical because it leads to reduction in personnel cost and labour. The basic architecture of an automated fabric web inspection system is shown in Fig. 2. As we can see from the diagram, lightning conditions and image acquisition is very important for such systems and that is why there is a series of cameras to scan the texture from every angle. The following are the basic components of such a system: frame grabber, a lighting system, a computer console accommodating processors, and electrical as well as mechanical interfaces for the inspection machine. Huge parallelism is generally employed in image acquisition and defect detection[38].

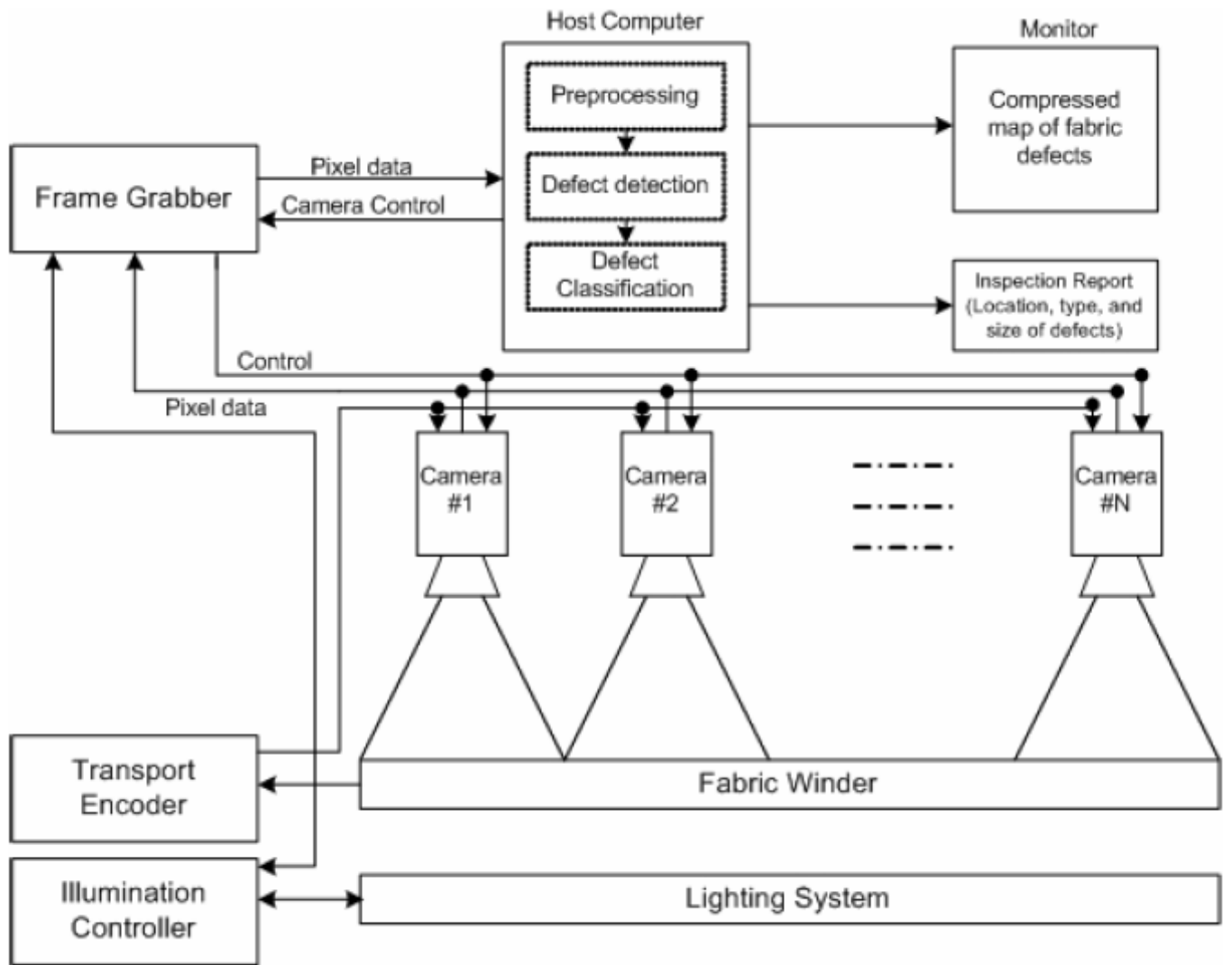


Fig.1.2 Architecture of a typical fabric inspection system

CHAPTER-2

DIFFERENT TYPES OF APPROACHES OF DEFECT DETECTION

2. Different Types of Defect Detection Methods

Various techniques exist for identifying the defects in texture patterns. Some of the commonly used techniques are discussed below:

2.1 Defect Detection using Co-occurrence Matrix Features

For the texture features to be characterized, spatial interactions among pixels in the neighbourhood are used due to the reason that texture is considered a neighbourhood property. In order to study texture analysis, the co-occurrence matrix method is widely used to study texture analysis and this method is also known as spatial gray-level dependence method [37]. The repeated occurrences of several grey level values in a specific direction and distance is the basis on which the method depends on. Generally computation of features of a texture is done through techniques of visual inspection which are automated for images with textures in the spectral or spatial domain. The definition of defect detection is defined in [40] using sub band domain co-occurrence matrices. The co-occurrence matrix is used to derive features of a texture namely entropy, homogeneity, energy, contrast and correlation [6]. Following diagram shows the basic concept of a GLCM formation from a matrix.

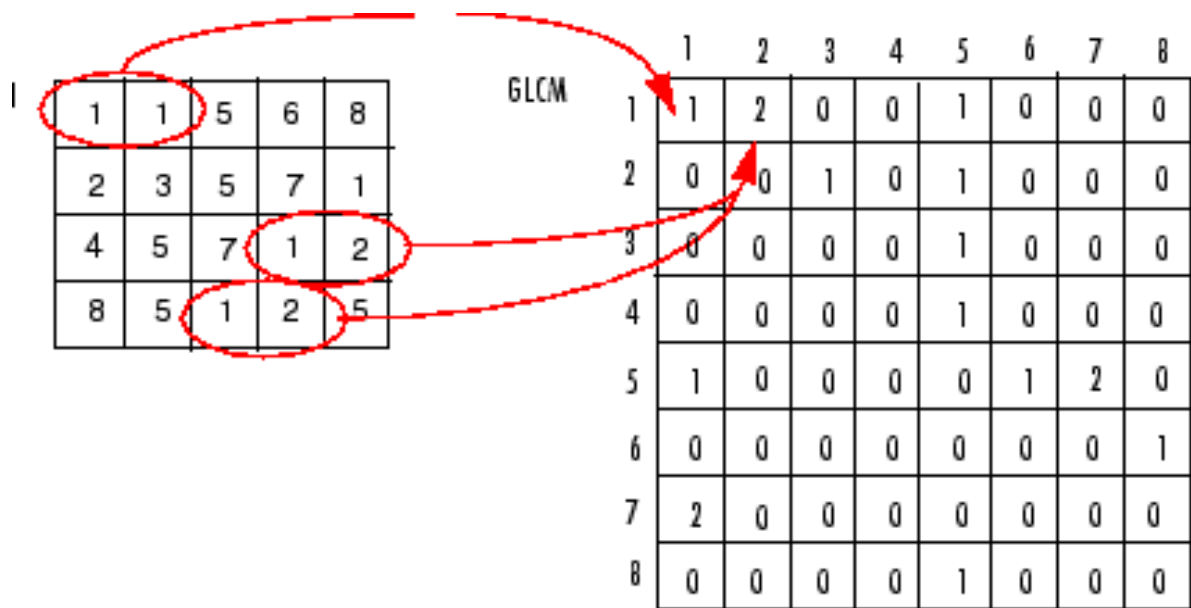


Fig.2.1 Creation of GLCM

Haralick gave 14 texture features in [7], which are computed from GLCM. These features characterize different spatial aspects of a texture and give statistical measure of the different properties of a texture pattern. Following table gives the 14 features which were given by Haralick.

Table. 2.1 Haralick's 14 Features calculated from GLCM

ID	Name	Equation
1	Angular Second Moment (Energy)	$f_1 = \sum_i \sum_j \{p(i, j)\}^2$
2	Contrast	$f_2 = \sum_{n=0}^{N_p-1} n^2 \{ \sum_i \sum_j p(i, j) \mid i-j =n \}$
3	Correlation	$f_3 = \frac{\sum_i \sum_j (i) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
4	Variance (Sum of Squares)	$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j)$
5	Inverse Difference Moment	$f_5 = \sum_i \sum_j \frac{p(i, j)}{1+(i-j)^2}$
6	Sum Average	$f_6 = \sum_{i=2}^{2N_p} i p_{x+y}(i)$
7	Sum Variance	$f_7 = \sum_{i=2}^{2N_p} (i - f_6)^2 p_{x+y}(i)$
8	Sum Entropy	$f_8 = - \sum_{i=2}^{2N_p} p_{x+y}(i) \log\{p_{x+y}(i)\}$
9	Entropy	$f_9 = - \sum_i \sum_j p(i, j) \log\{p(i, j)\}$
10	Difference Entropy	$f_{10} = \text{variance of } p_{x-y}$
11	Difference Variance	$f_{11} = - \sum_{i=0}^{N_p-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$
12	Information Measures of Correlation	$f_{12} = \frac{HXY - HXY1}{\max(HX, HY)}$
13	Information Measures of Correlation	$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$
14	Maximal Correlation Coefficient	$f_{14} = (\text{Second largest eigenvalue of } Q)^{1/2}$

2.2 Defect Detection using Edge Detection

The amount of edge per unit area distribution is considered by the defect detection when the edge detection is used. It is considered a feature important for the detection of any defect in textured images. Other spatial discontinuities along with edges, lines and point defects represent for a texture image how much transitions of gray levels are present. For detection of defects in texture the following features are used [8]-[10]. Conci and Proença [9] used Sobel operator for detection of defects in texture and comparison of results was done with thresholding based methods. In a basic methodology, the transformation of the texture image into a gradient image is done using a set of operators such as Sobel Operator, Prewitt Operator, etc. Now, in order to separate pixels which are defect-free, thresholding is applied on this gradient image. For further separation of pixels which are defective from noise, the image result is dilated with the Structural Element. The blob analysis is finally done in the last step, which labels the connected pixels as single object which is considered to be a defective region in the texture. Use of edge detection with defect detection approaches [8]-[10], [11] is mostly suitable for low resolution plain textured images. Major drawback of this technique occurs when the defects are not isolated correctly due to generation of noise from structure of texture resulting in false detections at a high rate eventually making them less suitable for textile inspection in real time.

Sobel Operators

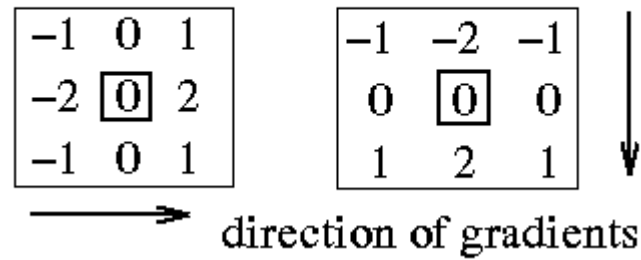


Fig. 2.2 Sobel operator

2.3 Defect Detection using Bi-Level Thresholding

A simple method of Bi-Level thresholding can be used for detection of defects with high contrast. The presence of a texture defect causes the fall or rise of the signal level locally; thus a trough or peak occurs which indicates a defect easily detected by use of thresholding technique. In bi-level thresholding method, a threshold value is found out which can be used to detect a defect in the sense that if the value is greater than threshold, it may be considered as a defect and if it is less than threshold, that region is defect free or vice versa. This kind of defect is detected mainly when a decision threshold is crossed by the signal. Norton-Wayne et al. [12]-[13] made use of this approach for detection of textile defects on moving at one meter per second speed. However, the method was not suitable enough because presence of noise in the image was also considered as a defect. The description of defect detected in fabric is done by Bradshaw [14] and Cho et al. [15] with the use of bi-level thresholding. An improved version of bi-level thresholding is developed in which adaptive thresholding is used for detection of defects with low contrasts in galvanized metallic strips [16]. For detection of defect techniques using bi-level thresholding, the main advantage is the implementation simplicity, but the techniques fail to detect those defects which appear without changing the mean gray intensity level in defect-free areas. Though easy but this technique fails in many textures in which complex defects are there which do not change the gray-level values in the defect-free regions.

2.4 Defect Detection using Neural Networks

One of the fastest and most flexible classifiers is the use of neural networks in fault detection due to their ability to categorize complex data decision boundaries and also due to the fact that they do not require any parameter to be passed upon. In [17], for the segmentation of local textile defects, a new approach using feed-forward neural network (FFN) is described. In this method it is assumed that a defect changes the gray-level distribution of neighbouring pixels. So, using the neighbouring pixels, features of every pixel are calculated and the reduction of feature dimension is done using Principal Component Analysis. And finally the classification of the pixel's feature vectors into defective and non-defective categories is done using FFN. Following is the block diagram of this approach:

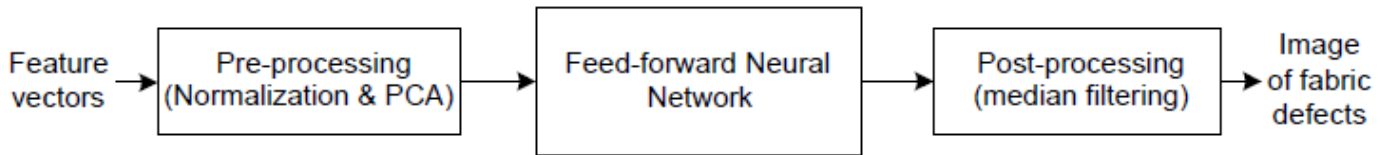


Fig.2.3 Block diagram of a Fabric Defect Detection using FFN

Feature extraction of a pixel is basically based on the gray values of the neighbouring pixels rather than some statistical measures. Though FFN is fastest among all the neural networks but implementing the detection of defect using FFN in real time is costly in terms of. Therefore, a solution which is economic for detection of texture with the use of a neural network which is of linear described in [17], [18]. Hung and Chen used the fuzzy technique and backpropagation neural network together for the classification of eight different fabric defect types along with the fabric which is defect-free[19]. Apart from FFN, Support Vector Machines (SVM) are another simple technique not suffering from local minimum problem usually occurring in FFN. Therefore detection of defects in texture with the use of SVM is proposed in [20]. Large intra-class diversity in a large number of classes of fabric defects remains a major obstacle for the use of SVM [20] and FFN [21], [17] based approaches for the inspection of fabric online.

2.5 Defect Detection using Discrete Fourier Transform

Enhancement of features which are periodic and noise immunity are the two important features of the Fourier transform (FT). The most important feature of FT is that it uses frequency components in describing the texture of an image. The periodically occurring features can be observed quite easily from the degree of frequency components. In order to detect the different types of defects in fabric, fourier transform is used. For extraction of seven significant characteristic parameters, the central spatial frequency spectrum is used for detecting the defects. Warp and weft yarn patterns together form the woven texture image. The yarns are basically 1-D are represented by a set of impulses modulated using profile of one yarn [22]. Because of textured components which are stochastic in nature, the proper localization of the local maxima peaks in the 2-D frequency plane is not done. Due to this reason, Goddard Sari-Sarraf and [23] in order to include frequency components which are local used contiguous concentric rings having constant width. However, use of local frequency components for texture defect identification is described by Chan and Pang in [24]. Tsai and Hu in [25] presented Fourier Models of four different types for defects in texture; missing pick missing end, oily fabric and broken fabric. Fourier features of the real defects in texture with the use of DFT are extracted using these models.

2.6 Defect Detection using Morphological Operations

Use of morphological operations for detection of defects is described in [26]. Firstly histogram equalization of every inspection image is done and to generate a binary image, thresholding of the resultant image is done. In the training phase, optimal size of Structuring Element is determined using the binary image of defect-free texture by using autocorrelation function. Now, in testing phase the size of SE which is optimal is used. Using SE, erosion and dilation operations are applied on each binary test image. The distance between defective pixel results is used for grouping defects into blob defects. The method is limited for practical application as most of the general defects in texture will not be present in the binary image generated by a simple method of thresholding. Datta and Mallik-Goswami [27] in order to detect defects performed laser-based morphological operations. Through this approach, the structure of fabric in optical domain which is periodic is by introduction of Fourier's lens after proper spatial filtering. Thus only on defective aperiodic images the morphological operations are performed. However, the defects are obvious for the experimental results given in [27] and there is no suggestion of an advantage over other less complex approaches which are available.

2.7 Defect Detection using Gabor Filters

The efficiency of methods based on WFT has shown importance of the conjoint analysis of the image texture in both spatial and frequency domains. Due to this reason, features of textures representing the frequency components in regions which are local in the spatial domain of texture pattern have led to a new area of research. The extraction of these texture features from the inspection images can be done through local spatial filtering. For this filtering, the 2-D Gabor filters are ideal in many ways [106]: firstly, they have axial frequency bandwidths and tunable angular, tunable center frequencies, and can achieve optimal joint resolution in frequency and spatial domain. Gabor filter parameters can be easily augmented for a known category of defects to be distinguished. Such texture defect segmentation with the use of optimal Gabor filter is shown in [42]-[44], [45], [49]. The orientation and dimension of local defects which are generated on the textile web varies in a random manner. As a result, a general web inspection system which uses a bank of asymmetric and symmetric Gabor filters are shown in [49], [46]-[47] and [48] respectively. The texture segmentation methodology in human visual system is explained in the retinal adaptations with the sigmoidal shaped nonlinearity and the Real Gabor Functions. As a result, a set of multiresolution and multiorientation RGF, followed by 20 inter-scale and intra- image fusion has suggested fabric defects segmentation [51], [52]. Kumar [50], [18] also demonstrated that the FFT can be used in defect-free texture to calculate the dominant spectral component which can be used for automatic selection of the Gabor filter's center frequencies.

CHAPTER-3

COMPARISON METHOD – 1 : TEXTURE DEFECT DETECTION USING SEGMENTATION PROPOSED BY K.N. SIVABALAN, DR. D. GHANADURAI

3. Defect Detection Using Segmentation

The algorithm which I have implemented for defect detection using segmentation is described in [28.]. Compared to the other algorithms, it is quite fast and simple. This algorithm has two basic steps: the first one is feature extraction in which simple features are extracted and the second one is segmentation in which based on the extracted features, defects are segmented. These steps are quite simple and easy to understand.

3.1 Feature Extraction

Initially the given image is preprocessed and after this, feature extraction is applied on it in order to extract features. The first step is to convert the given image to gray scale image. And then, for each row of the image, we calculate the median value

Consider an image $Img(m,n)$, the median for each row is calculated as follows:

1. *for* $i=1$ to m
 $Median(i)=x_{md}$ from $x_1, x_2, \dots, x_{md}, x_{md+1}, \dots, x_n$
where $md = \frac{n+1}{2}$ when n is odd.

We use the above equation to find the median value of a row in case when the number of elements in the row is odd.

2. *for* $i=1$ to m
 $Median(i)=\frac{x_{md} + x_{md+1}}{2}$ from $x_1, x_2, \dots, x_{md}, x_{md+1}, \dots, x_n$
where $md = \frac{n}{2}$ when n is even

The above equation finds the median value of a row in the case when the number of elements in a row is even.

3.2 Identification of Defected Area

The above calculated median value for each row and greyscale image $I(m,n)$ are used for detecting the defective region. In order to detect the defective region in the image, each pixel intensity is compared with the median value of that row and the previous pixel value, if there is a sudden change in the intensity level, then that pixel is considered to be in the defective region. If this sudden change in the intensity is atmost 60% of the previous pixel value or the median value, then that pixel is identified to be in the defective region.

To identify the defected area, apply the following equation:

- for* $x=1$ to m
for $y=1$ to $n-1$
if $(Img(x,y) < median(x)*0.6$ OR $Img(x,y+1) < Img(x,y)*0.6)$

Img(x,y)=255

else

Img(x,y)=0

Display ***Img(m,n)***, which contains the defected area in white colour and rest is black.

CHAPTER-4

COMPARISON METHOD-2: TEXTURE DEFECT DETECTION USING REGULARITY APPROACH BY DIMTRY CHETVERIKOV

4. Texture Defect Detection using Regularity Approach

This section contains the two basic topics for texture defect detection described in [29] and [30]. These papers explain two basic approaches for texture defect detection, i.e regularity and local orientation approach. The first one defines texture defects to be as the regions where regularity falls suddenly, and the second one as disturbances in the dominant orientation. Among these, the first approach is explained in the following sections.

4.1 Regularity Approach

The regularity approach searches for *irregularities*- here *irregularities* are defined as regions where regularity of a texture pattern is considerably lower than its dominant value. For a set of windows, regularity features are computed, then outliers occurring in the regularity feature space are identified as defects. The basic assumption of this approach is that most of the basic region of defect free texture, has some periodicity. The first step of this approach is to compute the regularity feature values for a set of windows spanning the texture image containing the defect. From these regularity values, the one with maximum value is considered to be the maximal regularity, which denotes that this is the dominant regularity value of the given texture . Maximal regularity is shown to be equivalent to the human judgement of regularity and is used to detect defects in the texture. After this, the third step is outlier detection. In this step , outlier points which fall outside the cluster of defect free pattern are detected as defective regions. The algorithm described in this section is called **StrucDef** [29].

In StrucDef , pattern regularity is computed by calculating in-polar coordinates and the periodicity of autocorrelation function. Maximal regularity feature is defined by the feature vector. Maximum value of the *directional regularity* $R(i)$ is called the Maximal Regularity. Regularity feature vector having four features is defined based on $R(i)$. Detailed steps StrucDef algorithm are mentioned below.

4.2 The Contrast Function

Consider a digital image $I(m,n)$ of MXN size and a spacing vector of (dx, dy) , where m is row and n is column. The autocorrelation function of $I(m,n)$ when normalised is defined as:

$$p_{xy}(d_x, d_y) = \frac{1}{S^2} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n)I(m+d_y, n+d_x) \quad (1)$$

Where $k=0,1,2,\dots$

$$S_k = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I^k(m,n) \quad (2)$$

For faster calculation [31], fourier transform is used to calculate autocorrelation as:

$$p_{xy}(m,n) = \text{IFFT}[\text{FFT}[I(m,n)] * \text{FFT}[I(m,n)]] \quad (3)$$

where IFFT is the Inverse FFT.

As the, pattern regularity needs to be computed for different directions ,a polar representation form of the autocorrelation function is most suitable. This representation, $\mathcal{P}_{pol}(\alpha, d)$ is defined for magnitude (spacing) d and an angle α . Given a non-integer location $(\delta x, \delta y)$ in \mathcal{P}_{xy} (dx, dy), $\delta x = d \cos \alpha$; $dy = -d \sin \alpha$; the value in this location is obtained by the linear interpolation of the four neighbouring pixels. Using the interpolation, $\mathcal{P}_{pol}(\alpha, d)$ is calculated on the polar grid (α_i, d_j) , where $\alpha_i = \Delta \alpha \cdot i$; $d_j = \Delta d \cdot j$. The resultant matrix is shown by $\mathcal{P}_{pol}(i, j)$. Generally, $\Delta d = 1$ is set and we let the spacing d to the maximum value of $N/2$: It is supposed that $N/2$ will span over at least two if not more periods of the texture pattern. Usually, we set $\Delta \alpha = 3^\circ$ or 5° as it is task dependent.

The autocorrelation function is normalised: $\mathcal{P} \in [0, 1]$. Mostly, negated version of this function is used and is called autocorrelation based interaction map:

$$M_{pol}(i, j) = 1 - \mathcal{P}_{pol}(i, j) \quad \text{from} \quad M_{xy} = 1 - \mathcal{P}_{xy}(m, n) \quad (4)$$

where $\mathcal{P}_{pol}(i, j)$ is the polar representation of the autocorrelation function.

In an interaction map, the periodicity vectors of the structure are shown to be occurring as dark points. A single row of $M_{pol}(i, j)$ is known as *contrast function*. It shows the variation of contrast along a given direction i with spacing d .

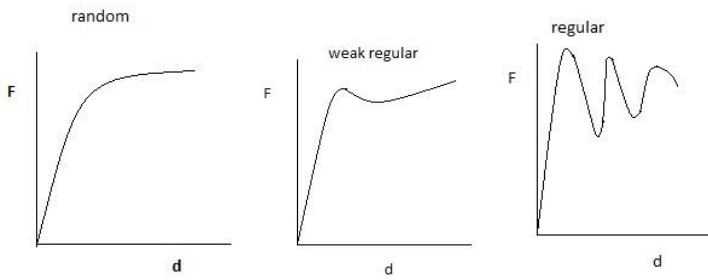


Fig. 4.1 Typical curves of contrast functions of a random, weak regular and a regular pattern

4.3 Computing Directional Regularity

The regularity measure for the direction $i=1, 2, \dots, N_\alpha$ where $N_\alpha=2\pi/\Delta\alpha$ is the number of total directions considered is grounded on contrast function $F_i(d)$. Generally, a structure with periodicity will have a contrast function having deep and periodic minima. For an angle i , the *directional regularity*, $R(i)$ can be defined as:

$$R(i) = [R_{int}(i) \cdot R_{pos}(i)]^2 \quad (5)$$

Where $R_{int}(i)$ is the intensity regularity and it shows how regular is the intensity of the pixels in the pattern. $R_{pos}(i)$ is the position regularity which indicates the periodic nature of the layout of the pixels in the pattern. While examining the periodicity of the pattern, two cases are taken into consideration. The first case occurs when the depths of the global minima decrease monotonically with d . The second is the unusual case which can lead to possible inhomogeneity of the texture pattern, where the monotonicity may not hold. The algorithm involves three procedures as follows:

Procedure 1 : Finding the extrema of $F(d)$

Step 1: Firstly, to remove the noisy extremes, apply the median filter of width 3 to $F(d)$. Original unfiltered function is denoted by $F_0(d)$.

Step 2: Find the extremes of the filtered function $F(d)$, except for the points $d = d_{\max}$ and $d = 0$. Denote the total number of minima and maxima by N_{\min} and N_{\max} , respectively.

Procedure 2 : Computing intensity regularity R_{int} .

Step 1: Calculate the amplitude of each maxima as follows: starting from the maximum, move right till a higher value of maximum is found, then store the lowest minima in this interval, then store the difference between the initial maxima and this lowest minima.

Step 2: Select the maximum (d'_{\max} , F'_{\max}) having the largest amplitude $F'_{\max} - F'_{\min}$ where F'_{\min} is the lowest minimum assigned to F'_{\max} .

Step 3: Resolve F'_{\max} by searching $F_0(d)$ for an even higher value in the ± 2 neighbourhood of d'_{\max} . Set F'_{\max} to the highest value found in the neighbourhood.

Step 4: Similarly, resolve F'_{\min} but this time search for a smaller value.

Step 5: Calculate the intensity regularity

$$R_{\text{int}} = 1 - \frac{F'_{\min}}{F'_{\max}} \quad (6)$$

Procedure 3 : Computing position regularity R_{pos} .

Step 1: If only one minima exists ($N_{\min} = 1$), find the position d_{\max} of the highest maxima and the position $d_1 > d_{\max}$ of the lowest minima after d_{\max} , then set

$$R_{\text{pos}} = 1 - \frac{|d_1 - 2d_{\max}|}{d_1} \quad (7)$$

and stop. Else ($N_{\min} > 1$), select the two lowest minimum values (d_1, F_1) and (d_2, F_2), $d_1 < d_2$, and continue.

Step 2: If there is no minima between d_1 and d_2 (normal case), calculate

$$R_{\text{pos}} = 1 - |1 - 2\gamma| \quad (8)$$

where $\gamma = \frac{d_1}{d_2}$, $0 < \gamma < 1$

Else, (special case), consider also

$$R'_{\text{pos}} = 1 - |1 - 3\gamma| \quad \text{if} \quad 1 - |1 - 3\gamma| \geq 0 \quad (9)$$

else

$$R'_{pos} = 0$$

and select the larger value from R_{pos} and R'_{pos} .

The regularities are normalized so that $R_{int} \in [0,1]$ and $R_{pos} \in [0,1]$. By default $R_{int} = R_{pos} = 0$ if $N_{min} = 0$ or $N_{max} = 0$.

Though median filtering in procedure 1 removes false extrema, but it also shifts the true ones. The correction steps 3 and 4 in procedure 2 reinstate the original values of the amplitude.

4.4 Defining the regularity feature vector

The maximal regularity indicates prevailing direction within the texture pattern which display stronger periodicity. Although affine transformations change the shape of this cyclical function, the order of the significant maxima, their height and number remain quite stable. This is because regularity of a pattern in a principal direction is preserved. Transformations which are affine in nature are able to preserve both periodicity and collinearity. The largest of the maxima called the maximal regularity. It is the dominant feature of the texture pattern and it quantifies the regularity as observed by a human.

Grounded on a sequence of the significant maxima of $R(i)$, a regularity feature vector is defined which are invariant to affine transformations. This affine invariance of these features is due to the transformational properties of the autocorrelation function. The local maximum values of $R(i)$, are denoted by T_k where $k=1,2,3,\dots,k$ is the index of the maximum in the sequence. Only the order of T_k in $R(i)$ is relevant. To select a significant maxima, the regularity is thresholded at a value T_{thr} and remove the low maxima $T_k < T_{thr}$. Earlier it was experienced [3] that $R(i)$ above $R_{thr} = 0.25$ indicated the presence of visually perceived periodicity in the texture pattern. To deal with weak regular patterns there is some relaxation, now R_{thr} and T_{thr} is set to= 0.15 [39].

Thresholded maxima sequences are denoted by $T'_k, k=1,2,\dots,K'$. The feature vector \mathcal{O}_1 consists of four components: the largest value M_R , the mean μ_R , the variance σ_R^2 , and the density of maxima ν , defined as:

$$M_R = \max_k \{T'_k\} \quad \mu_R = \frac{\sum_k T'_k}{K'} \quad (10)$$

$$\sigma_R^2 = \frac{\sum_k (\mu_R - T'_k)^2}{K'(K'-1)} \quad \nu = \frac{K'}{N_\alpha} \quad (11)$$

Since $R(i) \in [0,1]$, $0 \leq \mu_R \leq M_R \leq 1$ having value 0 indicates random and having value 1 indicates a highly periodic texture pattern.

4.5 The Maximal Regularity

Maximal regularity is denoted by M_R and it was first introduced in [30]. It is the most discriminating property to structure of all the four regularity features. It associates both the layout and intensity regularities. It is also reliable with the visually perceived sense of regularity. This feature is able to differentiate between patterns based on their regularity. Random patterns thus have low maximal regularity value and highly periodic patterns have high value of maximal regularity. Following figure shows some maximal regularity values of different kinds of patterns:

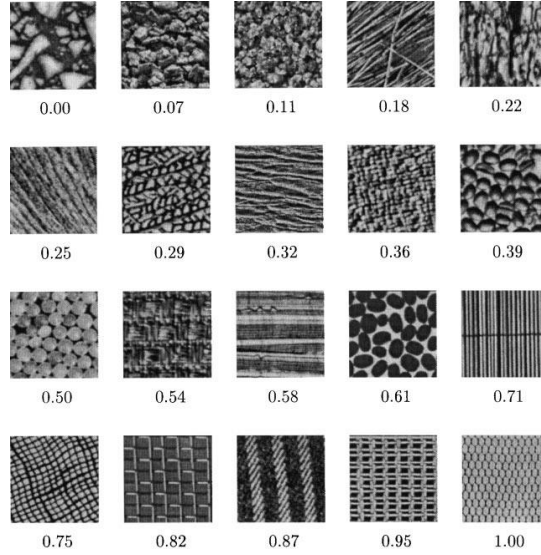


Fig. 4.2 Maximal regularity values of different kinds of patterns as in [2]

In the above figure, patterns are categorized in rows as four groupings according to their maximal regularity: random $[0, 0.25)$, low $[0.25, 0.50)$, medium regularity $[0.50, 0.75)$, high regularity $[0.75, 1.00]$. In each row, the M_R value grows from left to right. The layout and the intensity regularities are combined by the Maximal Regularity. The first pattern being completely random has maximal regularity value to be 0 while the last pattern (mesh) has 1 value as it has perfect periodic and regular structure.

4.6 Outlier Detection

For a set of windows spanning the texture pattern image, we compute the maximal regularity for each window in the set. As the defect is localized in the texture pattern, it is covered by only some windows, but most of the windows span over non-defective area which has some regularity. When this is observed in the feature space, it shows that most of the windows (defect free) form a cluster and the defective windows occur as outliers in the feature space as their distance from the center of the cluster is larger than the radius of the cluster. Regions of sudden high regularity, are called ‘positive outliers’ can also be present.

The basic principle of the algorithm to detect defects is based on outlier detection and robust regression [32]. We denote the regularity feature vectors of the windows by p_i . The regularity feature vectors of every window are represented by points in the feature space. As most of the windows contain defect free pattern, most of them form a cluster. Firstly, the central point p_c of the cluster is found. It is done by finding the median distance of each point from all other points. The one with the minimum value of median is the central point p_c

$$d_{med}(c) < d_{med}(i) \text{ for all } i \neq c \quad (12)$$

where

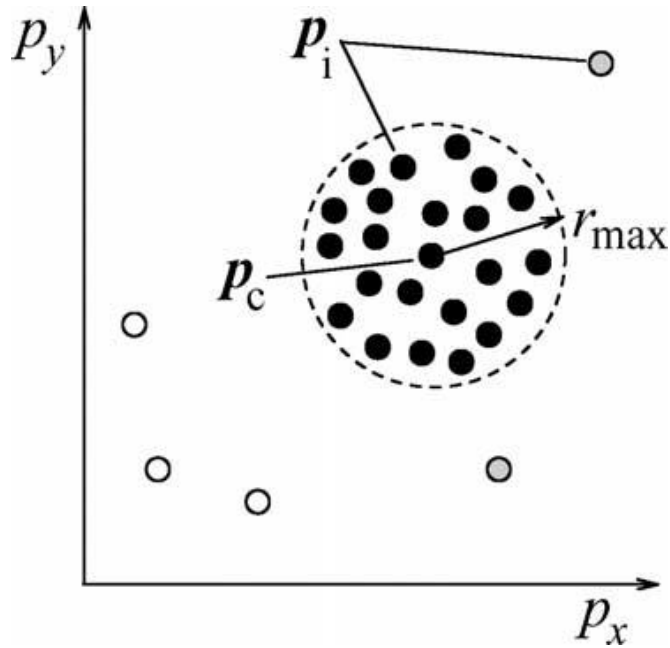
$$d_{med}(i) = \text{median} \|p_i - p_j\| \quad \text{given } i \neq j$$

p_c is the inmost point of the cluster. The distance between a point p_i and the center p_c is called the *radius* r_i of p_i .

$$r_i = \|p_i - p_c\| \tag{13}$$

and denote by r_{mean} the mean radius of the cluster.

A point p_k is then considered as an outlier if its distance from the centre point is greater than the maximum of the radius $r_{max} : r_k > r_{max}$. Based on whether the ground truth is available or not, the value of r_{max} can be found in two different ways. In the case of availability of the ground truth, r_{max} is learned from the distribution of the radius values calculated for the corresponding defect free patterns. In case when the ground truth is not available, it is set as $r_{max} = Kr_{mean}$ where K varies from 2-4. ‘Negative’ outliers i.e windows having low regularity values are only accepted as defects.



1.

Fig. 4.3 Concept of outlier detection. Gray and empty circles are outliers

Apart from thresholding factor K, StrucDef, has only two parameters: the detector window size W and the window slide step S. W is an important parameter to be selected carefully. The window must cover several periods of the pattern. At the same time, it should be as close to the expected defect size as possible. S is much less important. Basically, one has to balance between more reliable detection and higher processing speed. The small number of parameters reflects the simplification of the regularity approach. Any defect is viewed as just a significant decrease in regularity. W and K are only needed to tune the algorithm to the period of structure and the spread of the cluster in the feature space.

4.7 Implementation of Paper

In this project, the above mentioned approach is used to find Maximal Regularity and then using it to detect the defects in texture patterns. Values not given have been assumed. And instead of using Descending Component Analysis in Procedure 1 of calculating Directional Regularity to calculate numbers of maxima and minima, I have used a function called Peakdet which finds number of minima and maxima in a given Contrast Function. It takes a single row of contrast function and threshold value as its input parameters. Results are discussed in the Results section.

CHAPTER-5

COMPARISON METHOD-3:

TEXTURE DEFECT

DETECTION USING GLCM

5. A new statistical approach of defect detection using GLCM.

A new statistical method for defect detection is implemented in order to identify defects of complex nature occurring in texture patterns. The method uses Grey Level Co-Occurrence Matrix in order to derive certain features of texture patterns. Using those features and finding the Euclidean distance between template features and sample features, defects are determined using a threshold value [35].

5.1 Definition of GLCM

Grey-level co-occurrence matrix (GLCM) can be defined as a statistical method of examining texture by considering the spatial relationship of the pixels. GLCM is also known as the grey-level spatial dependence matrix. It is a second order statistics. The GLCM is calculated by counting the number of times a pixel with the intensity value 'i' occur in a specific spatial relationship to a pixel with the value 'j' with different offsets and angles [36].

MATLAB provides a function called "graycomatrix" to calculate the GLCM.

```
glcm = graycomatrix(I);
```

The above command creates a GLCM of Image I.

Following figure shows the concept of GLCM. The figure shows how several values in the GLCM are calculated by graycomatrix in a 4-by-5 image I. Element (1,1) in the GLCM has value 1 because in the image, two horizontally adjacent pixels having values 1 and 1, occur only once. Similarly, element (1,2) in the GLCM contains the value 2 because in the image, two horizontally adjacent pixels having values 1 and 2 occur twice. Thus in a similar way, all the values in GLCM are processed and filled by graycomatrix.

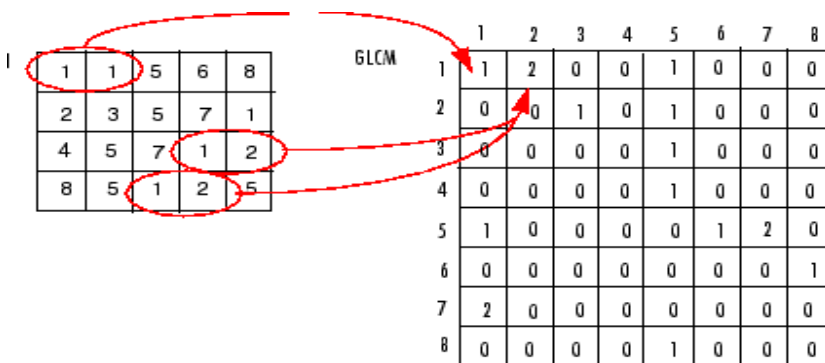


Fig.5.1 Concept of GLCM

The offsets in graycomatrix function explain the pixel relationships of varying distance and direction. Four possible spatial relationships with angles (0°; 45°; 90° and 135°) and distance (D=1) are specified in the following figure:

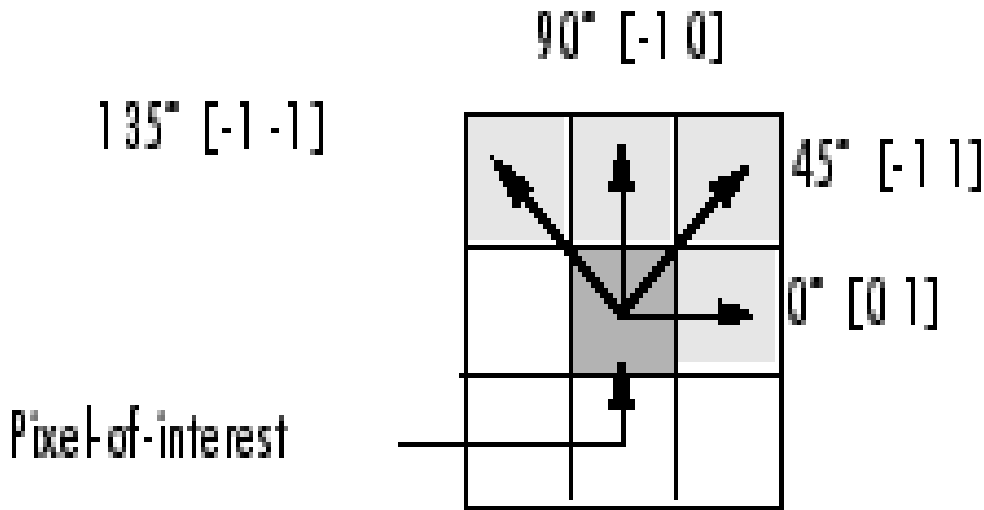


Fig.5.2 Offset directions in GLCM

Similarly, we can find GLCMs in all the eight directions. The size of the GLCM matrix is dependent on the intensity scale of an image. For ex. If the image is of grayscale, then its intensities will vary from 0 to 255. So, the size of GLCM will be 256X256.

5.2 Features Calculated from GLCM

The algorithm evaluates five features from GLCM using which defects are identified in the defective image. Following are features which are evaluated:

- 1) **Contrast:** It gives a measure of the intensity contrast between a pixel and its neighbour over the whole image. For a constant image, contrast is 0.

$$C = \sum_{i,j} |i - j|^2 p(i, j) \quad (14)$$

- 2) **Energy:** Also called Angular second moment, it gives us measure of the homogeneity of an image. Hence it is suitable for detecting disorders in textures. It returns the sum of elements squared in the GLCM. For constant image, 1 is returned.

$$E = \sum_{i,j} p(i, j)^2 \quad (15)$$

- 3) **Entropy:** It is a statistical measure of randomness which can be used to characterize the texture of an image. Also, it gives a complexity measure of the image.

$$Et = \sum_{i,j} p(i, j) * \log(p(i, j)) \quad (16)$$

- 4) **Homogeneity:** It returns a value to measure the closeness of the distribution of values in GLCM to the GLCM diagonal. For a diagonal GLCM, homogeneity is 1.

$$\mathbf{O} = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \quad (17)$$

5) Variance: It gives the measure of the distribution of values around the mean. For a constant image, variance is 0.

$$\mathbf{V} = \frac{1}{L} \sum_{i,j} (p(i,j) - \text{mean})^2 \quad (18)$$

5.3 Defect Detection Methodology

Steps:

Step 1: Input the defective Image of any size.

Step 2: Evaluate the GLCMs of the first 7X7 window in all the eight directions using graycomatrix function of Matlab.

Step 3: Evaluate Grey Level Co-Occurrence Probabilities from GLCMs

Step 4: Calculate Contrast, Energy, Entropy, Homogeneity and Variance for each GLCPs.

Step 5: Evaluate average of all the 5 features and name it Template.

Step 6: For each window of 7X7

Step 6.1: Evaluate average GLCM from all 8 directions

Step 6.2: Evaluate GLCPs from GLCMs

Step 6.3: Calculate Contrast, Energy, Entropy, Homogeneity and Variance for each of the GLCPs.

Step 6.4: Evaluate average of all the 5 features and name it Sample.

Step 6.5: Find the **Euclidean distance between these features of Template and Sample:**

$$\text{dist} = \sqrt{(C_T - C_s)^2 + (E_T - E_s)^2 + (Et_T - Et_s)^2 + (O_T - O_s)^2 + (V_T - V_s)^2} \quad (19)$$

Step 7: If (distance > threshold)

Then defective

Else

Not defective

CHAPTER-6

PROPOSED TEXTURE DEFECT DETECTION USING ENTROPY

6. New Method of Defect Detection using Non-Extensive Entropy

6.1 Definition

The new method of defect detection in texture patterns uses **Non-Extensive Entropy** as an indicator of regularity in order to detect defective regions in a texture pattern. This new Non-extensive entropy is based on **Gaussian Information measure** which ensures that only the relevant and necessary info lying inside the bell shape of the curve will be considered for calculating the entropy [33]. Since the proposed entropy has a **non-additive property**, it is used for the representation of information in the non-extensive systems which contain some amount of regularity or correlation. For the computation of entropy function, grey level co-occurrence probabilities (GLCP) are used. As this new non-extensive entropy is based on Gaussian Information Gain, most of the information is inside the bell curve and the outliers (in this case defects), lie outside $3*\sigma$. Here σ represents standard deviation.

Let random variable $X=\{x_1, x_2, \dots, x_n\}$ has probabilities $P=\{p_1, p_2, \dots, p_n\}$. Assuming complete probability distribution, $i.e. p_i \in [0, 1]$ and for $i=1, 2, \dots, n$, (n is the number of probabilistic experiments).

The information gain on the i th event of X , with associated probability p_i is defined by the Gaussian function as follows:

$$I(p_i) = e^{-p_i^2} \quad (20)$$

Entropy of X can be defined as:

$$H(P) = E(I(p_i)) = \sum_{i=1}^n p_i I(p_i) = \sum_{i=1}^n p_i e^{-p_i^2} \quad (21)$$

6.2 Implementation of Entropy Method

Following are the parameters which are used for this method:

1. **Image Size:** 256X256
2. **Window Size:** 7X7
3. **Step size:** 0

Steps:

Step 1 : Firstly the image is read

Step 2 : Then a 7X7 window is defined.

Step 3 : For each window:

Step 3.1 : GLCM is found.

Step 3.2 : Using GLCM, we calculate GLCP

Step 3.3 : Non-extensive entropy calculated based on GLCP.

Step 4 : IF($\text{abs}(\text{mean-entropy}) > 3 * \text{stdeviation}$)

Then this area is defective (Central pixel of the window is equated to 255)

ELSE

Non-defective

We have used Entropy method in two different ways. The first way is by keeping the window size constant for all the images. We have kept window size to be 7X7 in the first method and we call it Constant Window size Entropy Method. The second way is by keeping the window size variable. Here, we first find the best window size for a particular image which can detect defects in the best possible manner. So, for a certain image, there was a certain window size which detected the defects accurately, this window size may or may not work for other images. So, our first task was to find the window size for each image. We called this second approach as Variable Window size Entropy Method. Constant window size Method is executed on Images with synthetic defects as well as complex defects images. Variable window size Method is executed on complex defect images.

CHAPTER-7

AUTOMATED TEXTURE DEFECT DETECTION USING ENTROPY

7. Automated Defect Detection using Non-Extensive Entropy

Applying Non-Extensive Entropy with a constant window size for all images was not working and did not give satisfactory results. But when tested manually, most of the images were showing good results for a specific window size. So, we used the second approach of Variable Window size Entropy Method. Now the main goal of this approach is to find that specific window size for which best result is coming out. So, to automate the complete algorithm for defect detection using entropy, several methods were applied to find that window size for which defect is correctly detected. These methods are applied on the image using window size 3 to the half of the minimum side of the image. Finally, based on a certain criteria, the best window size is selected and the basic algorithm of defect detection by entropy is run on the image for that particular window size which is found out. The automation process has been defined using best four methods. These method find the most suitable window size for a specific image. They are as follows:

7.1 Maximum Count Method

In this method, the number of values which are greater than 3σ are counted for each of the window size. Finally, that window size for which the maximum number of entropies qualify the 3σ test is selected and considered to be the best window size for which defects will be detected correctly.

7.2. Maximum Entropy Method

In this method, the entropies which qualify the 3σ test are collected and their overall Non-Extensive Entropy is computed. Finally, that window size is selected for which maximum entropy results.

7.3. Gaussian Mixture Distribution Method

In this, method, we try to fit the data of entropy values by a Gaussian Mixture distribution. ‘gmdistribution’ is a class in Matlab. Gaussian Mixture distribution is defined by an object of this class. This model is called mixture model because it consists one or more multivariate Gaussian distributions. Each of the Gaussian component is well-defined by its covariance and mean, and the whole mixture is defined by a vector of mixing proportions. In order to fit a Gaussian mixture distribution model to data, use `gmdistribution.fit`. `fit` is a function of the class `gmdistribution`. Here, the returned object is of struct type. We minimize the entropy of `PComponents` which is one of the components of this object. In the last step, we select that window size for there is minimum entropy of `PComponents`.

7.4. Combined Method of Gaussian Mixture Distribution and Maximum Counts:

This method combines the third and the first method to select the best window size. Firstly, the dummy window size id found out by Gaussian Distribution Method. In the second step, we count the number of entropy values which are falling outside 3σ for window sizes which are in +10 and -10 vicinity of the dummy window size. Finally, from this new range of window sizes, we select that window size, which has the maximum number of entropies outside 3σ .

CHAPTER-8

EXPERIMENTAL RESULTS AND CONCLUSION

8. Experimental Results and Conclusion

8.1 Environmental Setup:

The following system configuration has been used while conducting the experiments:

Hardware Configuration

Processor	Intel® Core™ i5-2450M CPU
Clock Speed	2.50GHz
Main Memory	4GB
Hard Disk Capacity	500GB

Software Configuration

Operating System	Windows 7
Software Used	MATLAB 7.9.0 (R2009b)

We calculate the performance of our algorithms on images from Brodatz Database as well as on images from the link : <http://athos.vision.sztaki.hu:8080/strucdef/strucdef.html>

8.2 Measures

8.2.1 Measures for objective comparison of Regularity and Entropy Methods on images with Synthetic Defects

In this section, both the methods are evaluated and an objective comparison between them is used to draw conclusions which one is a better method. Both methods are compared by evaluating three parameters: **Precision**, **Recall** and **F-measure**.

$$\text{Precision} = \frac{\text{relevant_pixels} \cap \text{retrieved_pixels}}{\text{retrieved_pixels}}$$

OR

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

High **precision** means that an algorithm gave more important results than less important ones.

$$\text{Recall} = \frac{\text{relevant_pixels} \cap \text{retrieved_pixels}}{\text{relevant_pixels}}$$

OR

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

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

8.2.3 Measures of objective comparison of Regularity, GLCM and Entropy Methods using $N_{correct}$ and N_{false} on images with complex defects

While subjective comparison shows that Regularity is a better method in detecting defects of complex nature in texture patterns than Entropy and GLCM based method, but an objective comparison is better way to judge comparison between both the methods.

So, we have used, 2 parameters for objective comparison of all the three methods:

1. **$N_{correct}$** : no. of pixels which are correctly detected. This is the total number of pixels which are having 255 intensity value at the same position in mask of defect detected image and ground truth. (Here, defective region is considered to be white in the mask image).
2. **N_{false}** : no. of pixels which are incorrectly detected. This is the total number of pixels which are having 255 intensity value in the mask of defect detected image and 0 intensity value in the ground truth. (Here, defective region is considered to be white in the mask image).
3. **$N_{correct}:N_{false}$** : ratio of $N_{correct}$ and N_{false} . Higher the value of this ratio, better is the detection of defect in the defective image.



8.2.4 CPU Time

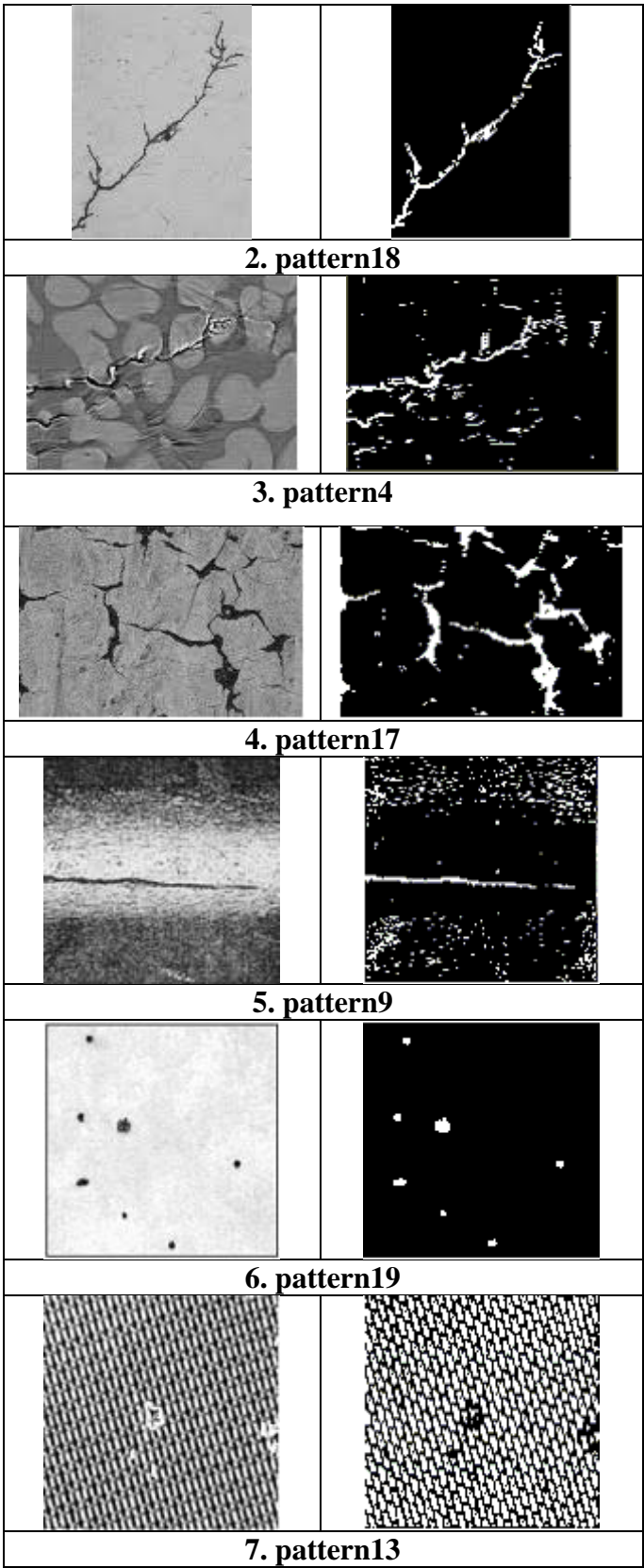
CPU time is the amount of time taken by a CPU for processing instructions of a computer program. The CPU time is generally measured in clock ticks or seconds. We have computed the CPU time in seconds for the running program by using the standard MATLAB commands “tic toc”. Thus the CPU time computed using the tic toc commands on the Variable window size Entropy Method is 6493.557952 seconds for a single image.

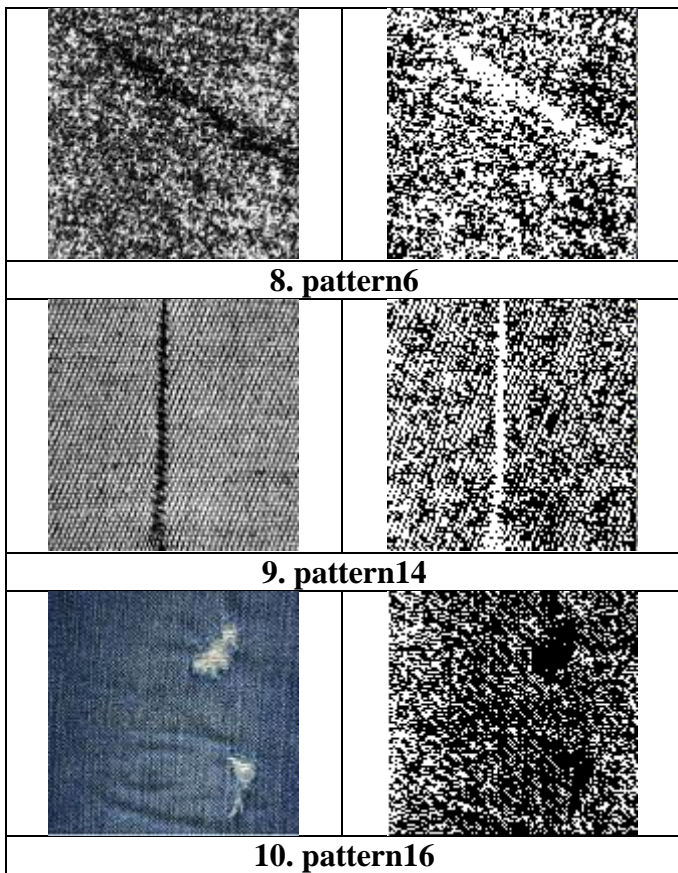
8.3 Results

8.3.1 Results of Segmentation Approach

Table 8.1 Results of Segmentation approach

Defective Image	Defect Detected Image
	
1. pattern19	





8.3.2 Results of Regularity Method (Chetverikov's Method)

8.3.2.1 Results of calculating Maximal Regularity on four Images

Images shown here represent each of the four categories, i.e random, low, medium and high regularity value images.

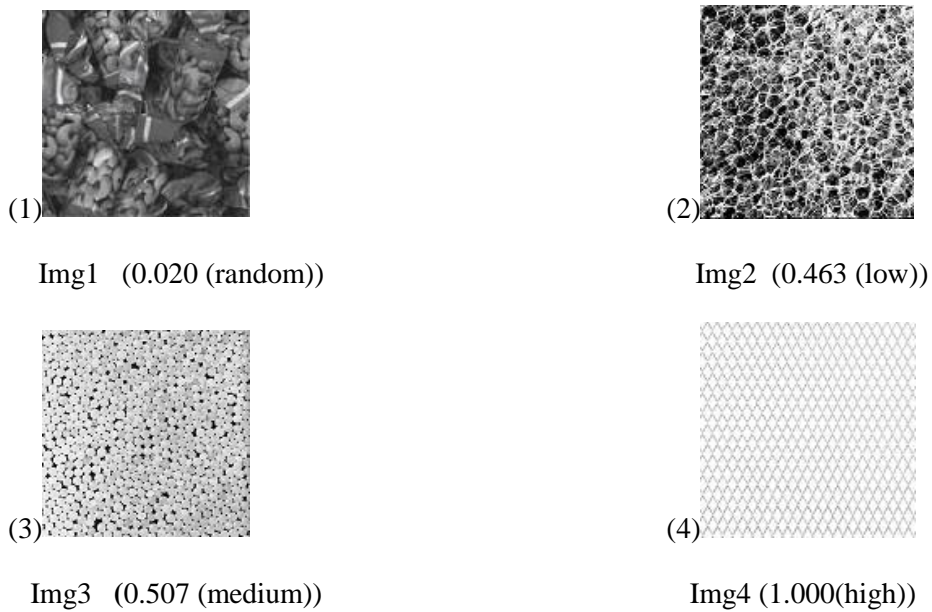
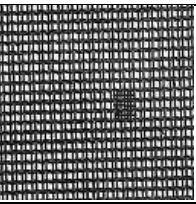
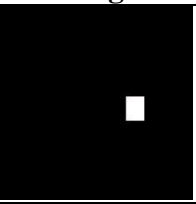
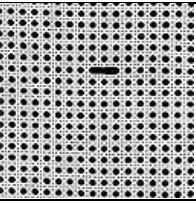
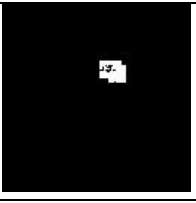
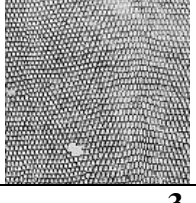
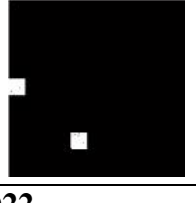
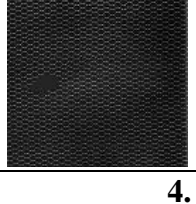
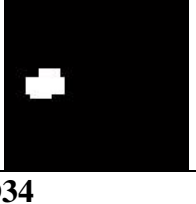
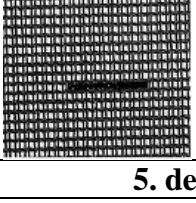
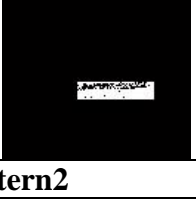


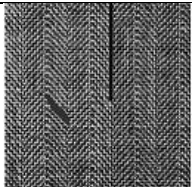

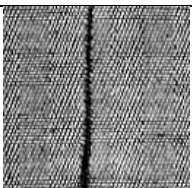


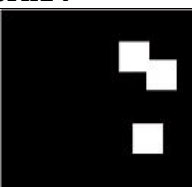
Fig. 8.1 Maximal Regularity value of 4 categories of patterns

8.3.2.2 Results of Images with Synthetic Defects

In this test, eight 256 X 256 pixel size brodatz/textile images are used [4]. The textures contain more or less obvious imperfections. For each pattern, the threshold value is set manually. The window should be such that it includes at least two periods of the pattern. Generally the size should be at least 40 X 40 pixels. In this project window size is taken as 64 X 64. There is another parameter called step size which denotes the number of steps (pixels) to be left when the window is moved to next position. Step size is used to balance between more reliable detection and higher processing speed. In my implementation , I have taken step size to be 5. In these images defects are created synthetically, Following are some of the brodatz/textile images with the defect detected identified:

Table 8.2 Result of Regularity Approach on synthetic images

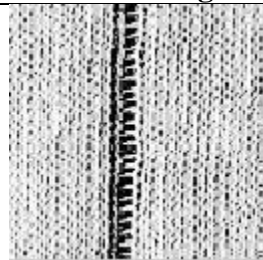
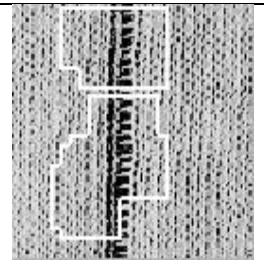
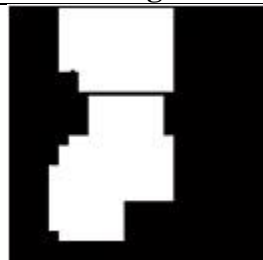

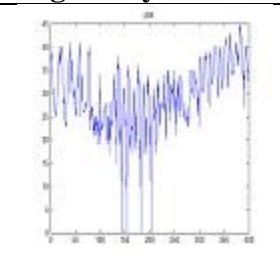

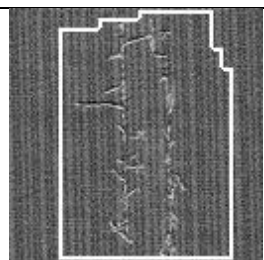
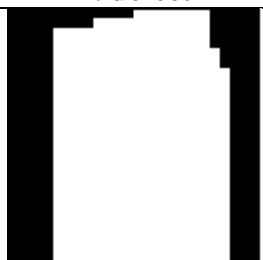

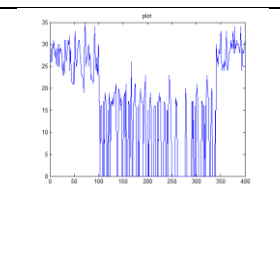
Defective Image	Defect Detected Image
	
1. D20	
	
2. brodatz100	
	
3. brodatzD22	
	
4. brodatzD34	
	
5. defect_gpattern2	

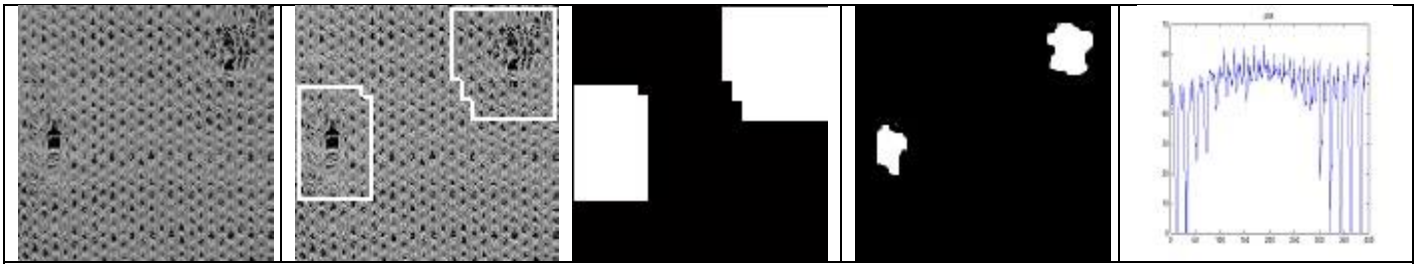
	
6. defect_gpattern3	
	
7. defect_pattern14	
	
8. defect_pattern16	

8.3.2.3 Results of Images with Complex Defects

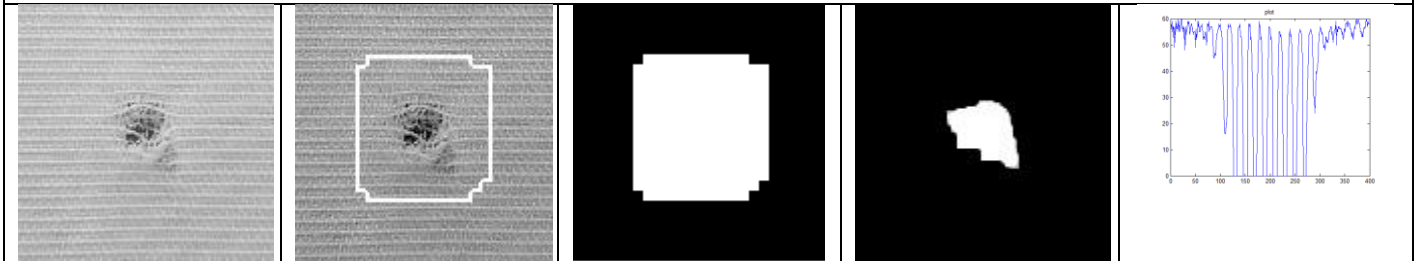
,Chetverikov’s method is applied to such natural defects in order to evaluate its efficiency on natural or more complex defects which are generally found in textile.

Table 8.3 Result of Regularity Approach on complex defects

Defective Image	Defect Detected	Mask Image	Ground Truth	Regularity Plot
				
1. defect1				
				
2. defect2				



3. defect3



4. defect4



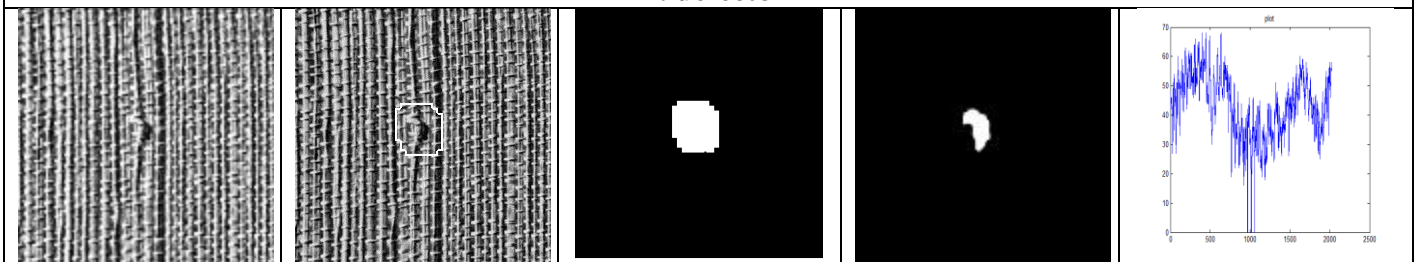
5. defect6



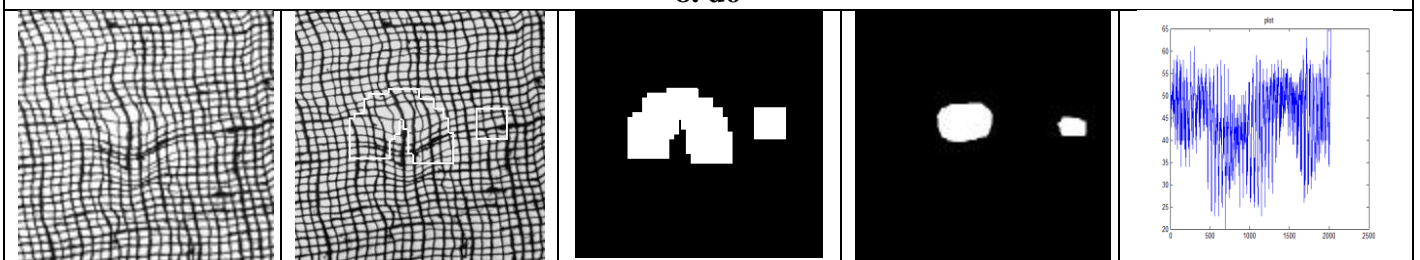
6. defect7

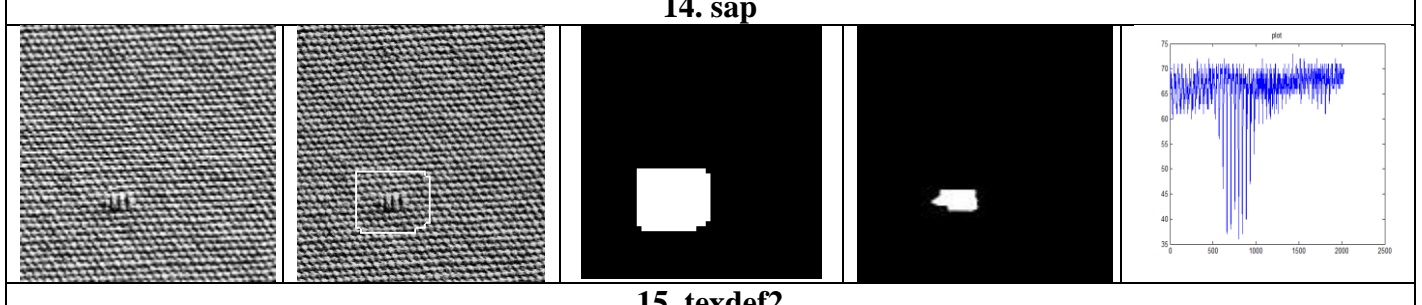
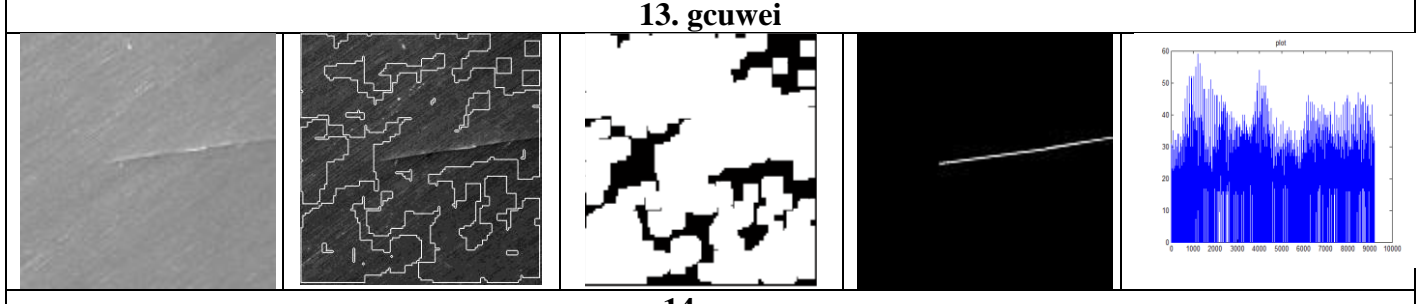
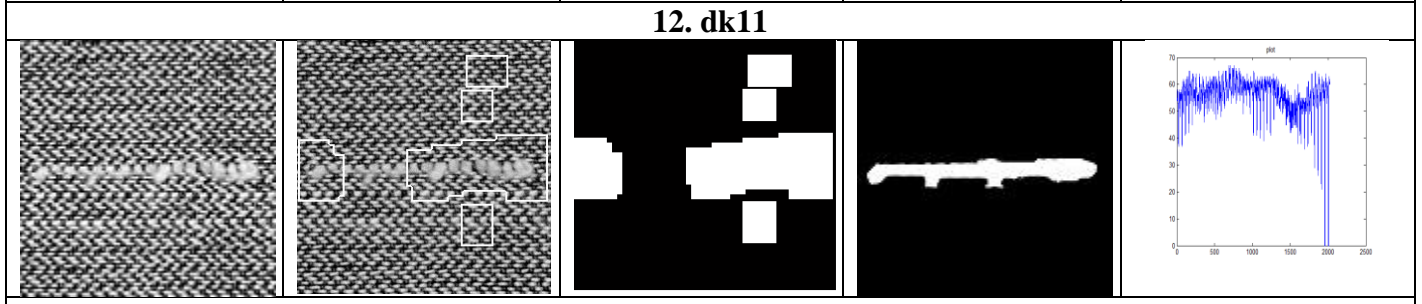
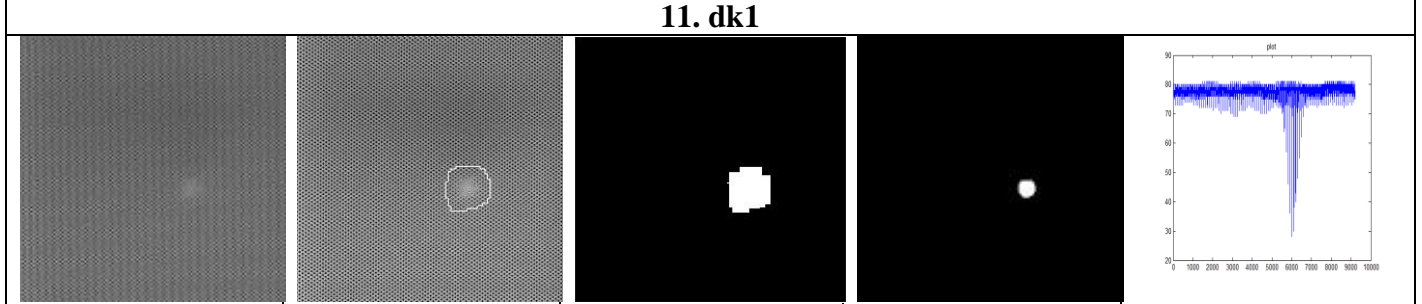
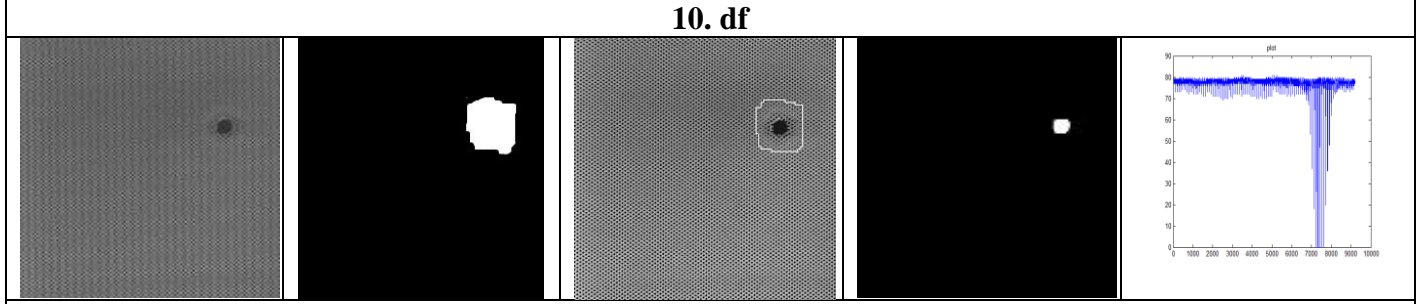
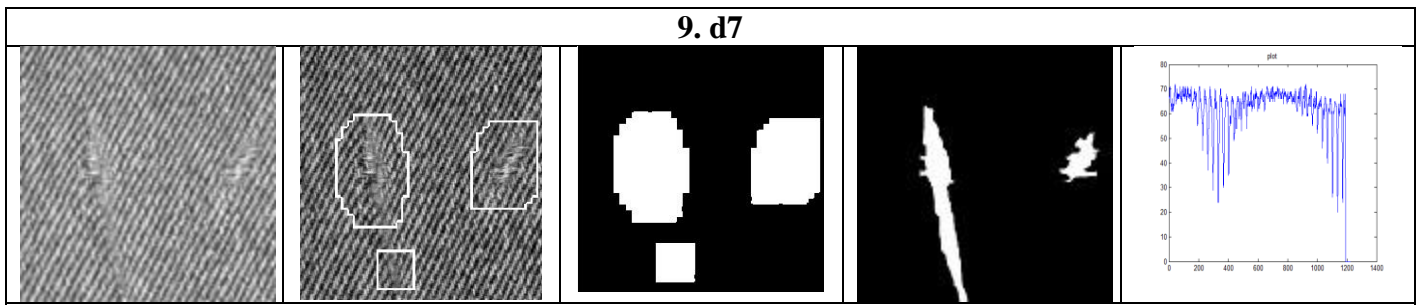


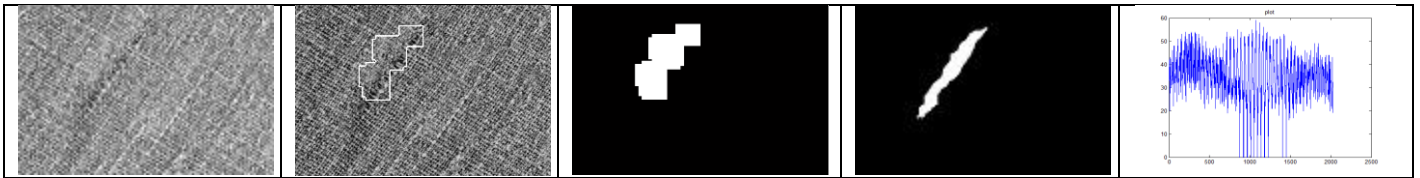
7. defect8



8. d6



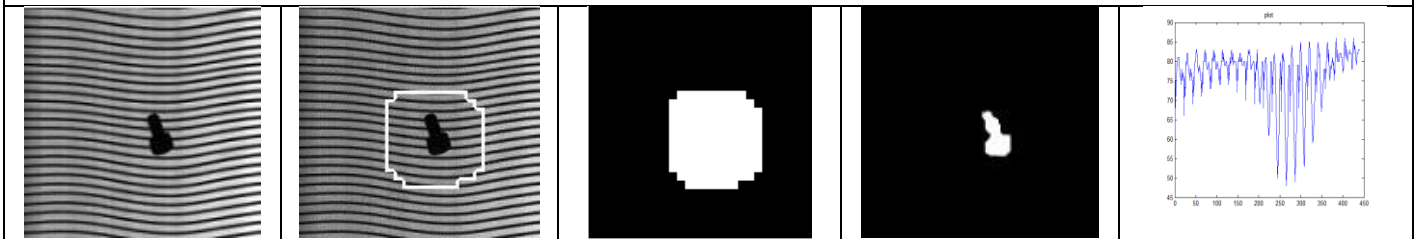




16. textile0



17. textile1



18. x



19. YY1



20. d1

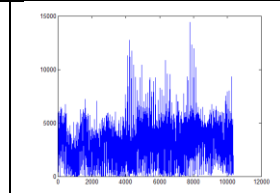
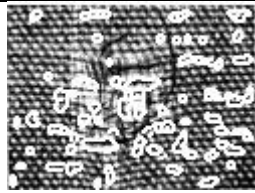
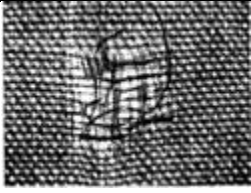
8.3.3 Results of GLCM Method

8.3.3.1 Results of Images with complex defects

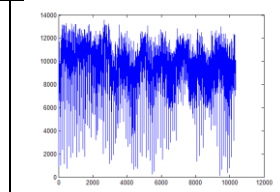
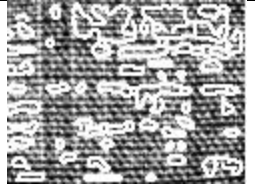
Table 8.4 Result of GLCM method on complex defects

Defective Image	Defect Detected	Mask Image	Ground Truth	Euclidean Distance Plot
1. defect1				
2. defect2				
3. defect3				
4. defect4				

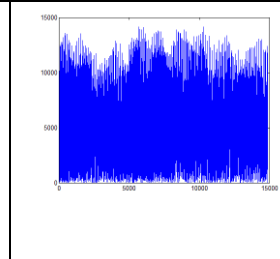
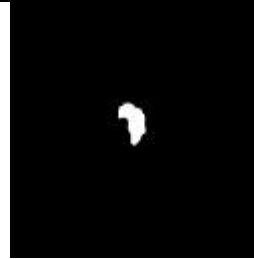
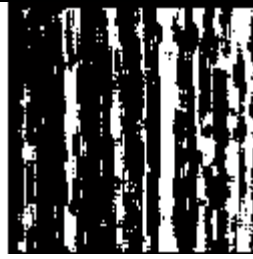
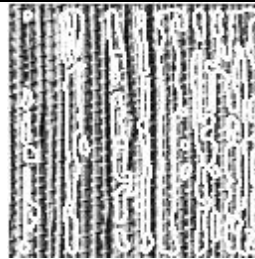
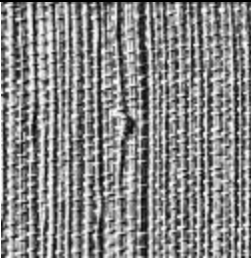
5. defect6



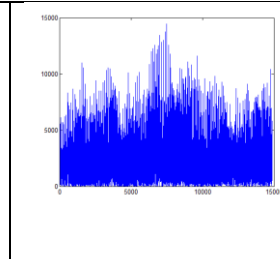
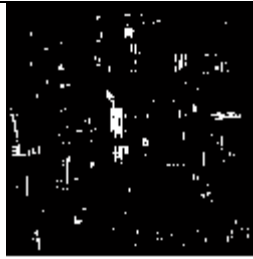
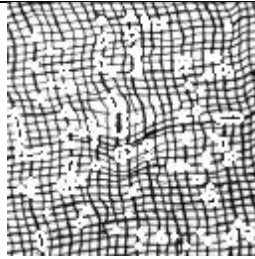
6. defect7



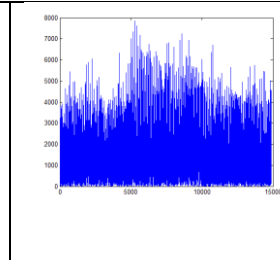
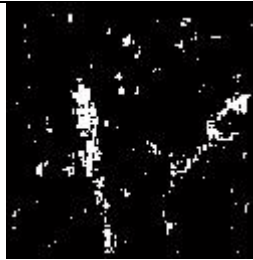
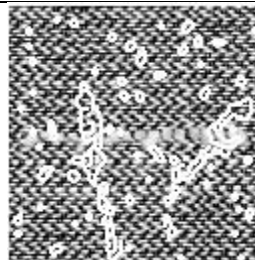
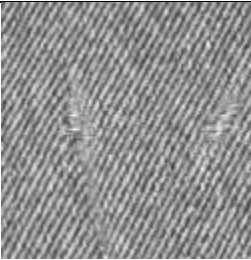
7. defect8



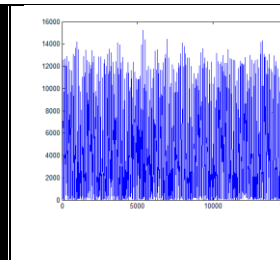
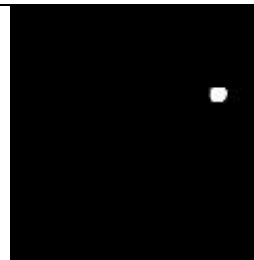
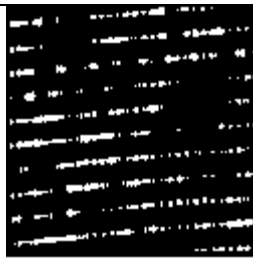
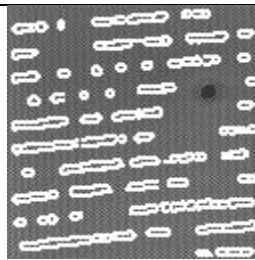
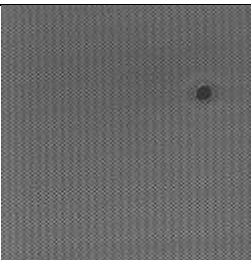
8. d6



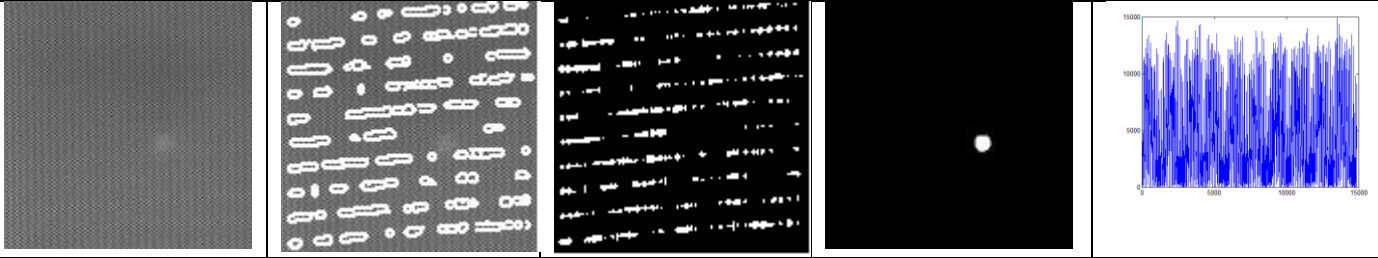
9. d7



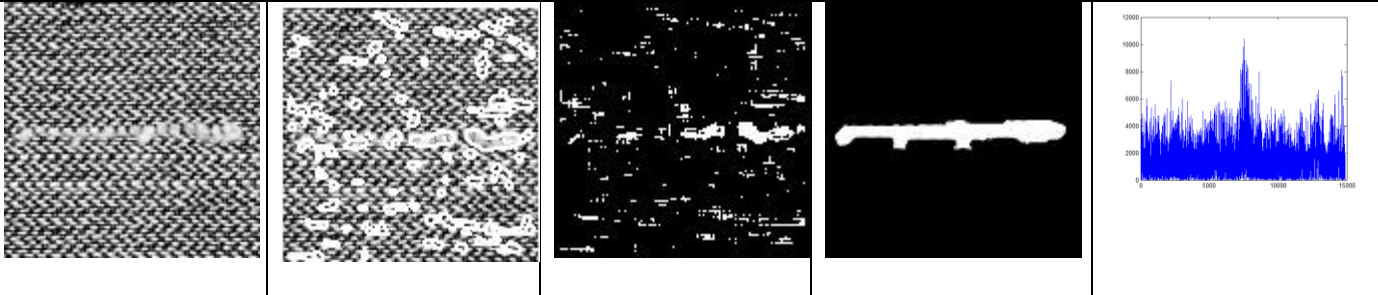
10. df



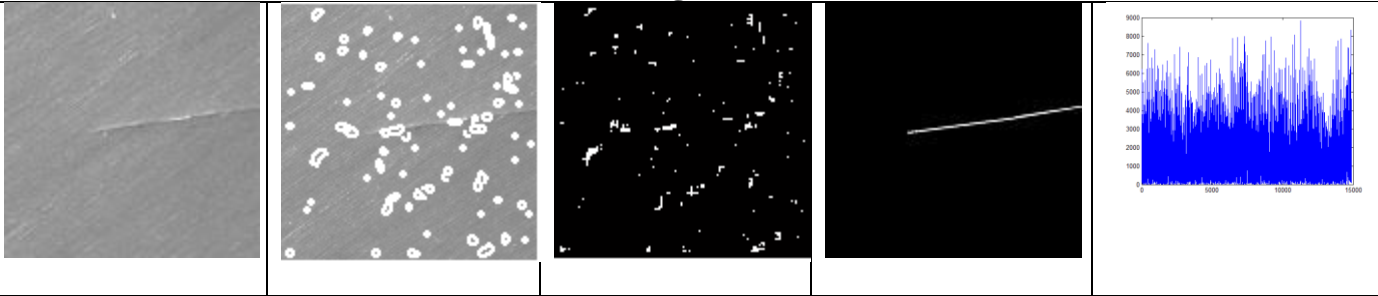
11. dk1



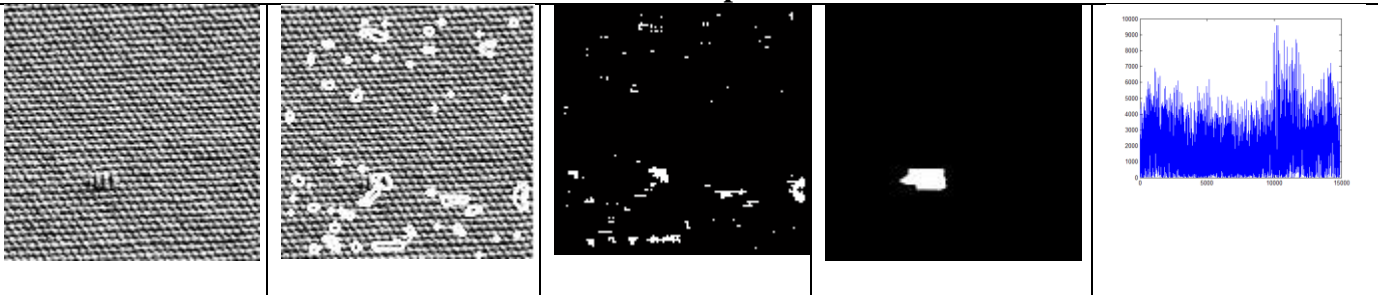
12. dk11



13. guwei



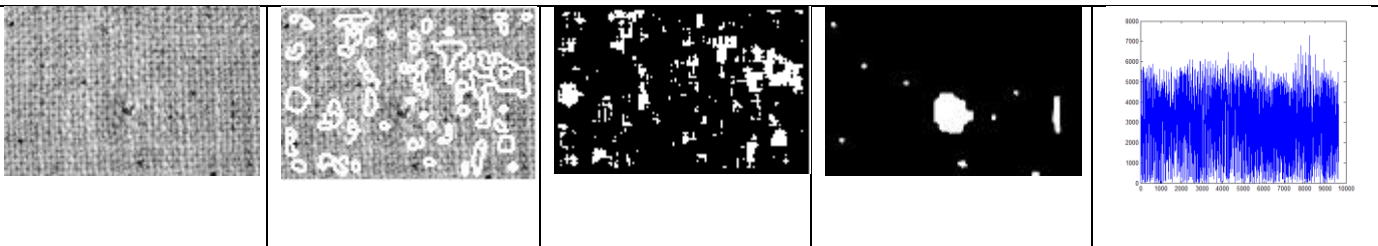
14. sap



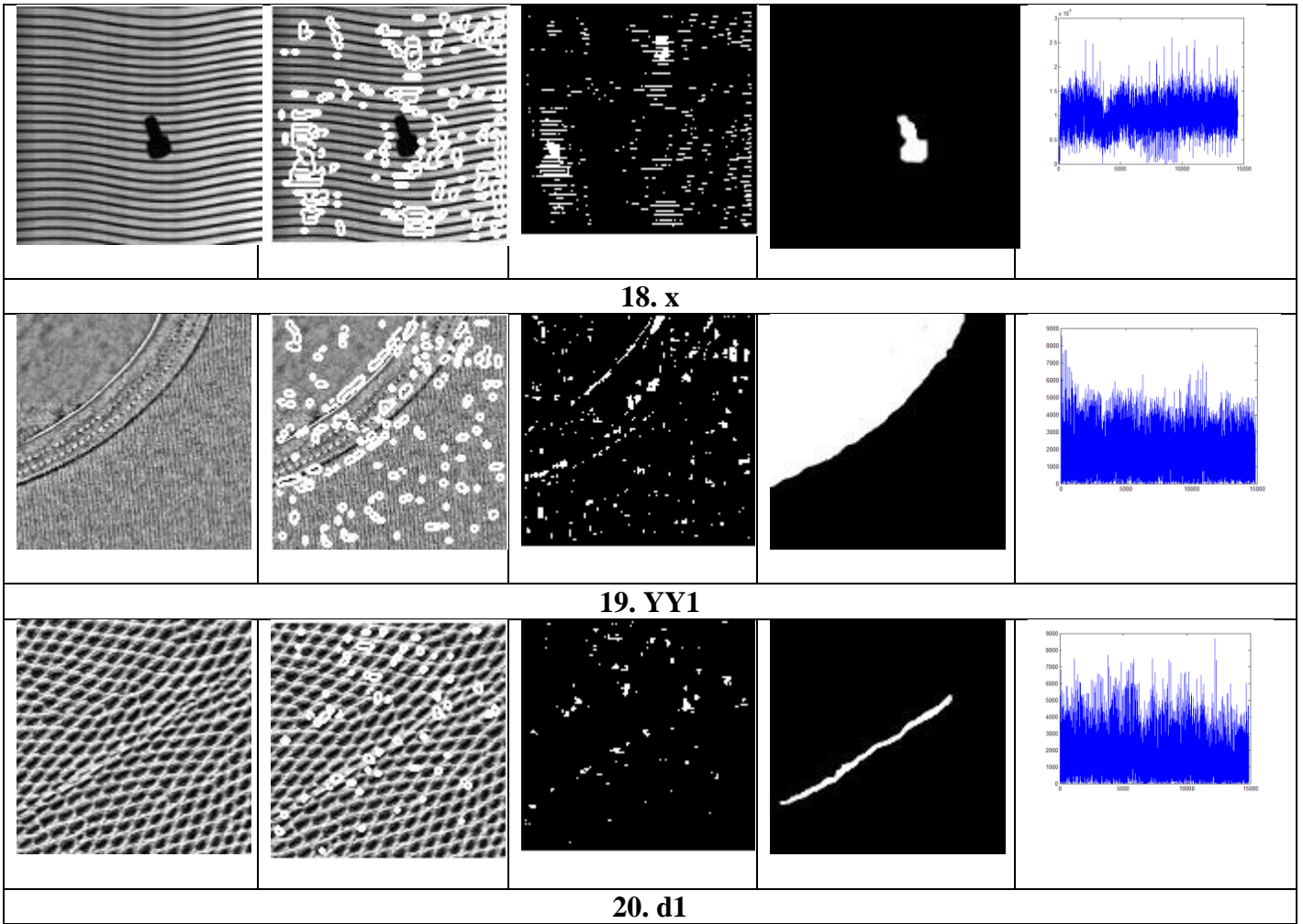
15. texdef2



16. textile0



17. textile1

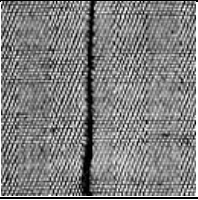



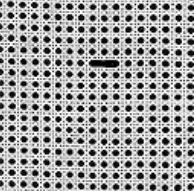
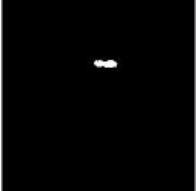
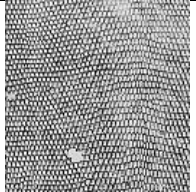

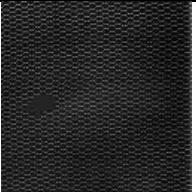
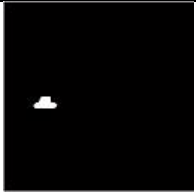
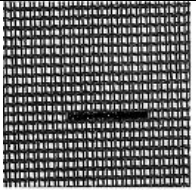

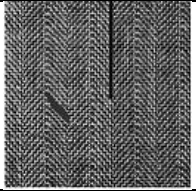
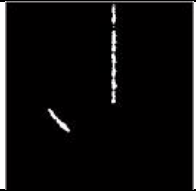
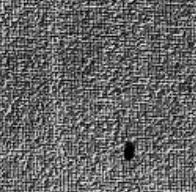
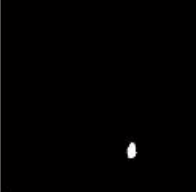
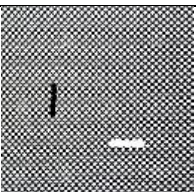
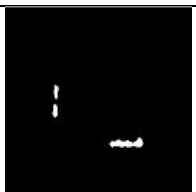
8.3.4 Results of Entropy Method

8.3.4.1 Results of Images with synthetic defects

Here, images with synthetic defects are used for execution.

Table 8.5 Result of Entropy Method on synthetic defects

Defective Image	Defect Detected Image
	
1. defect_pattern14	

	
2. brodatz100	
	
3. brodatzD22	
	
4. brodatzD34	
	
5. defect_gpattern2	
	
6. defect_gpattern3	
	
7. defect1_gpattern1	
	
8. defectmesh21	

8.3.4.2 Results of Images with complex defects

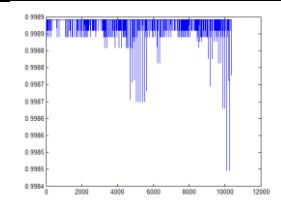
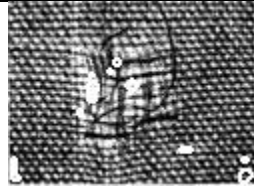
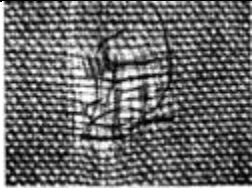
Here, Entropy Method, which uses a new Non-Extensive Entropy as a regularity measure is used to detect defects in images having complex defects.

Here, the images used are of varying size, so the program is modified in such a way so that image of any size can be run.

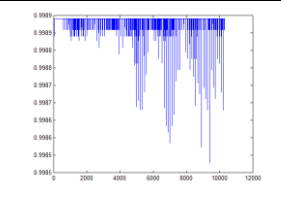
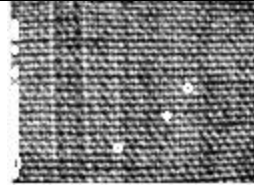
Table 8.6 Result of Entropy method on complex defects

Defective Image	Defect Detected	Mask Image	Ground Truth	Entropy Plot
1. defect1				
2. defect2				
3. defect3				
4. defect4				

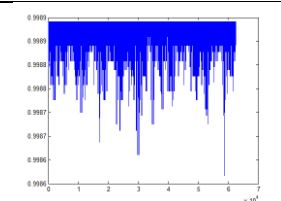
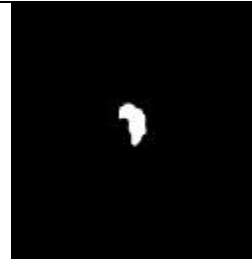
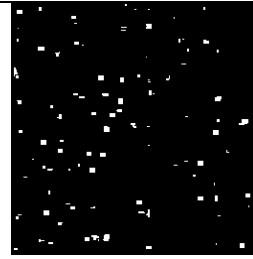
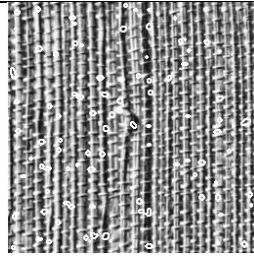
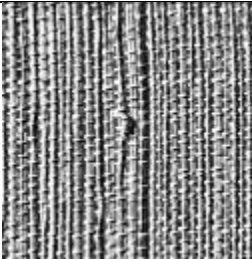
5. defect6



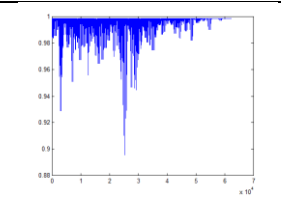
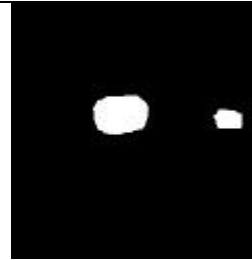
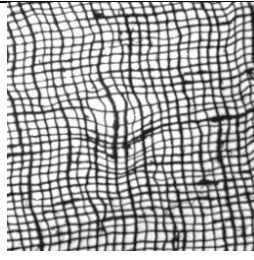
6. defect7



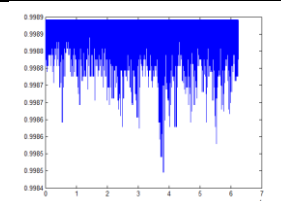
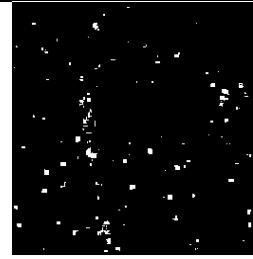
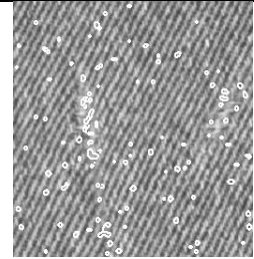
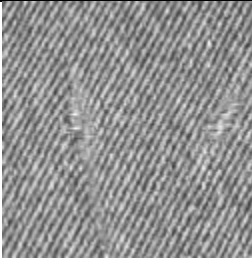
7. defect8



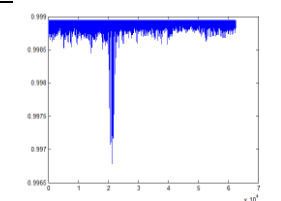
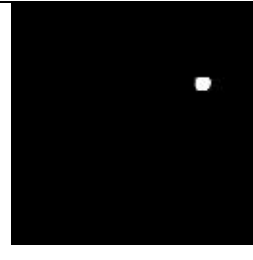
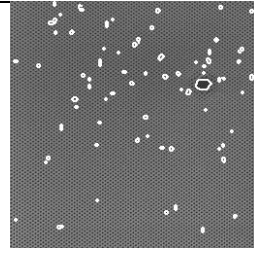
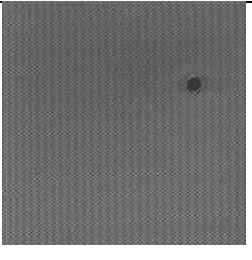
8. d6



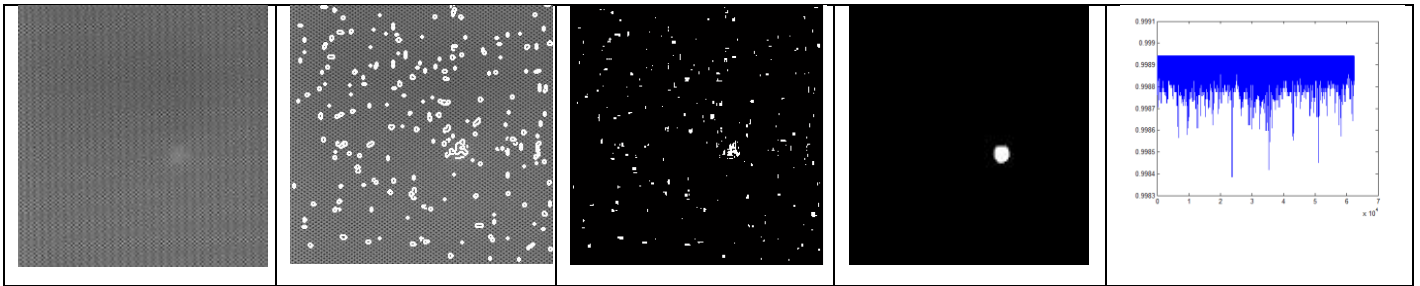
9. d7



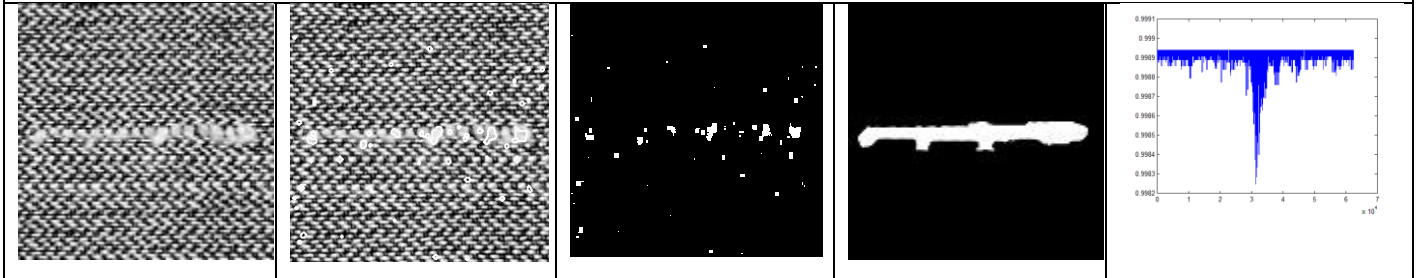
10. df



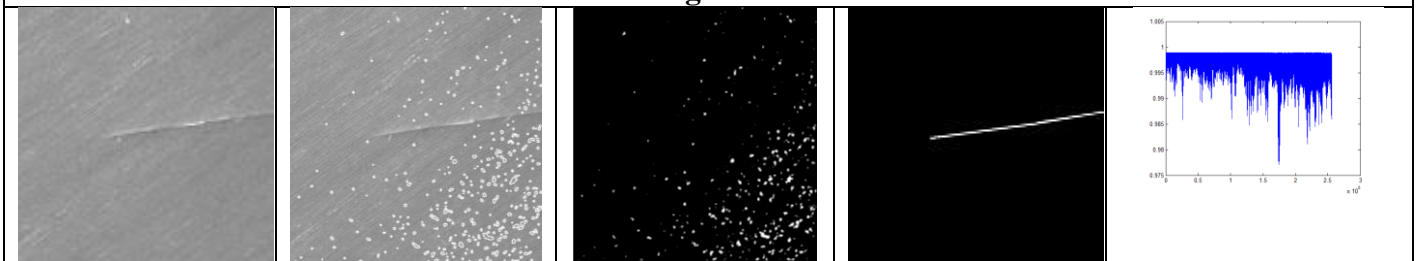
11. dk1



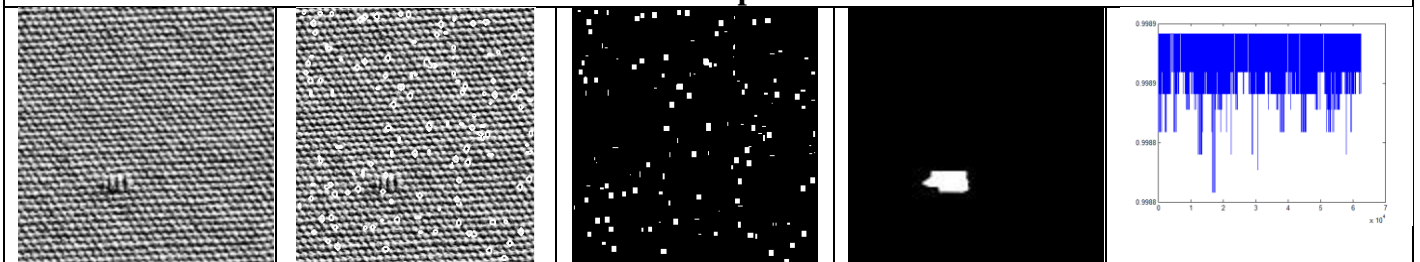
12. dk11



13. guwei



14. sap



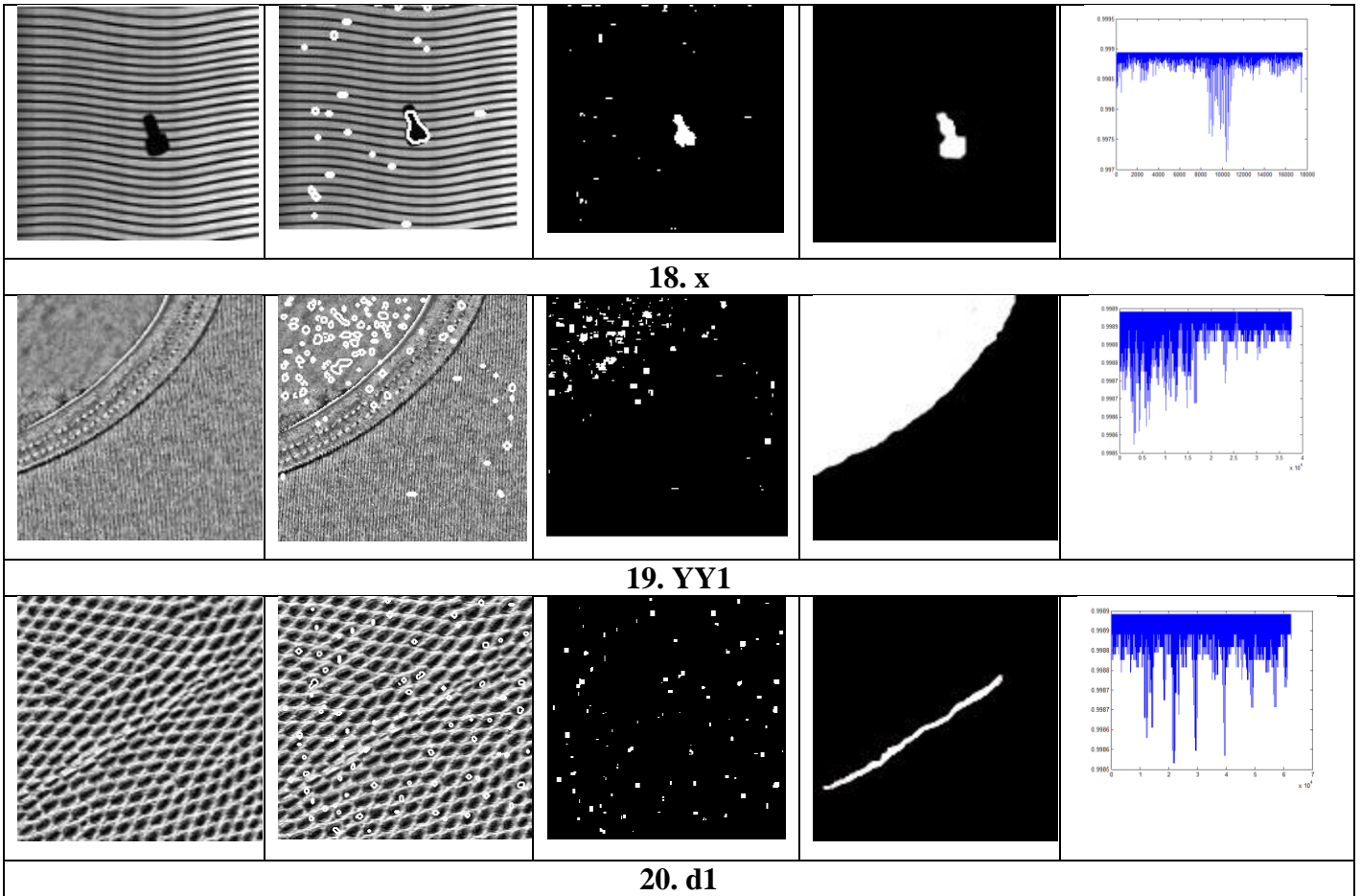
15. texdef2



16. textile0



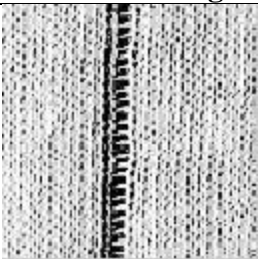
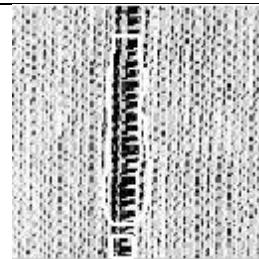
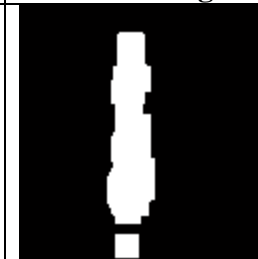

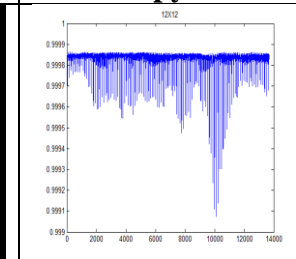
17. textile1

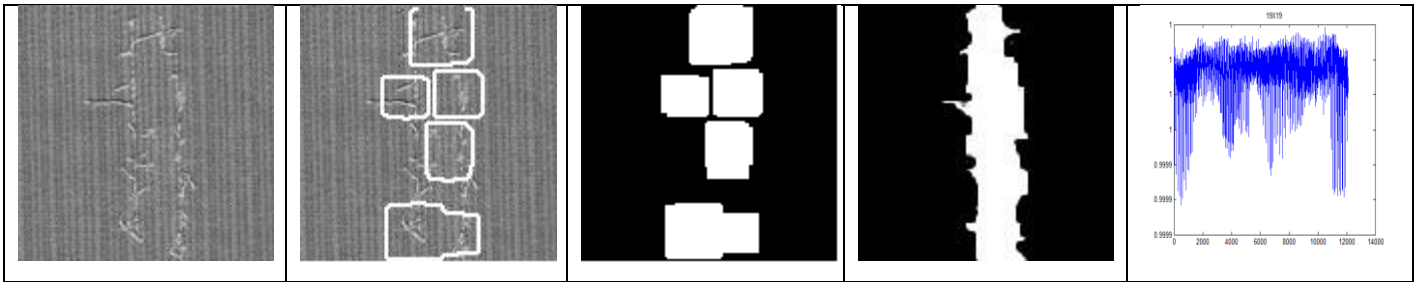


8.3.5 Results of Automated Entropy Algorithm

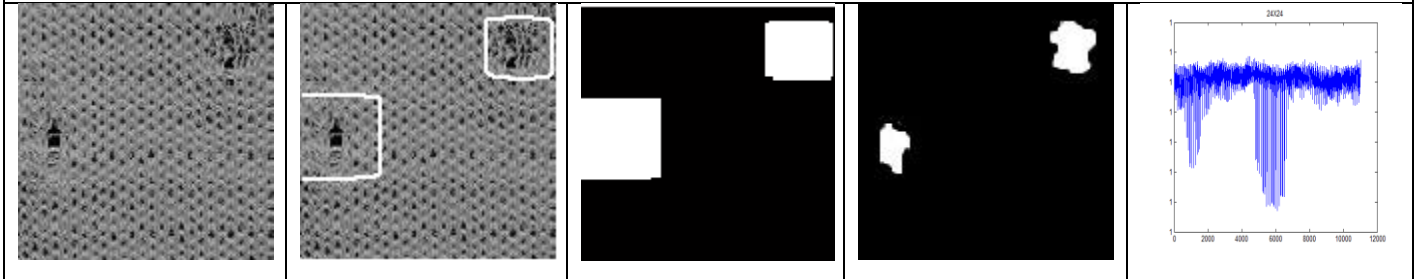
8.3.5.1 Results of Max Count Method

Table 8.7 Result of Max count method

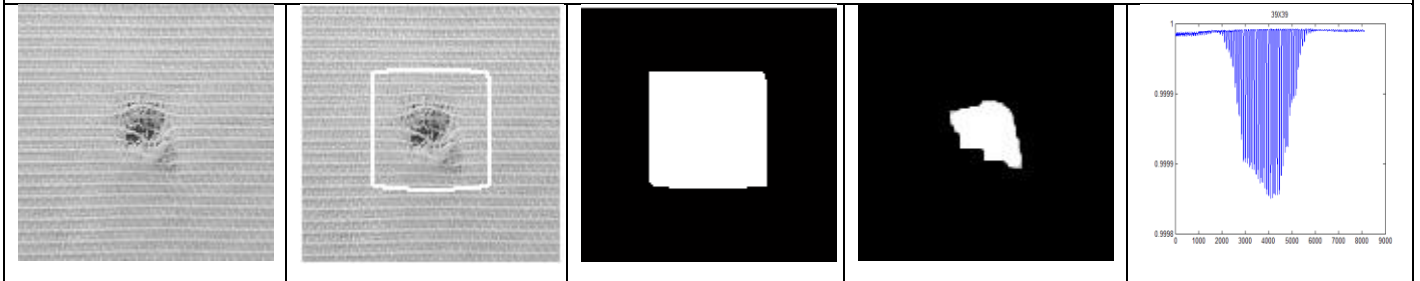
Defective Image	Defect Detected	Mask Image	Ground Truth	Entropy Plot
				
1. defect1				



2. defect2



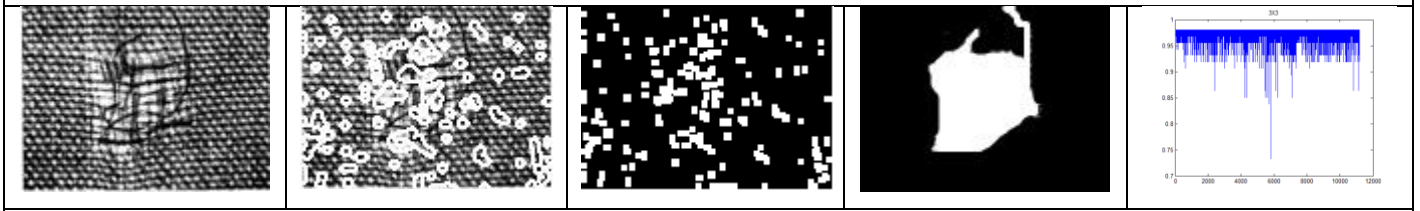
3. defect3



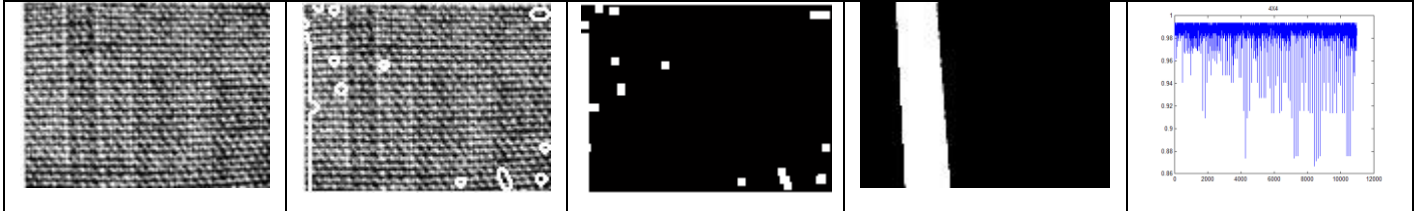
4. defect4



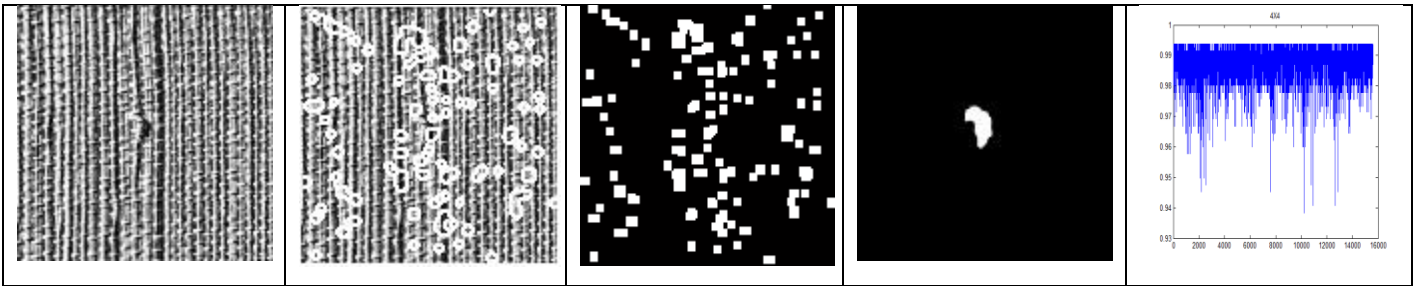
5. defect6



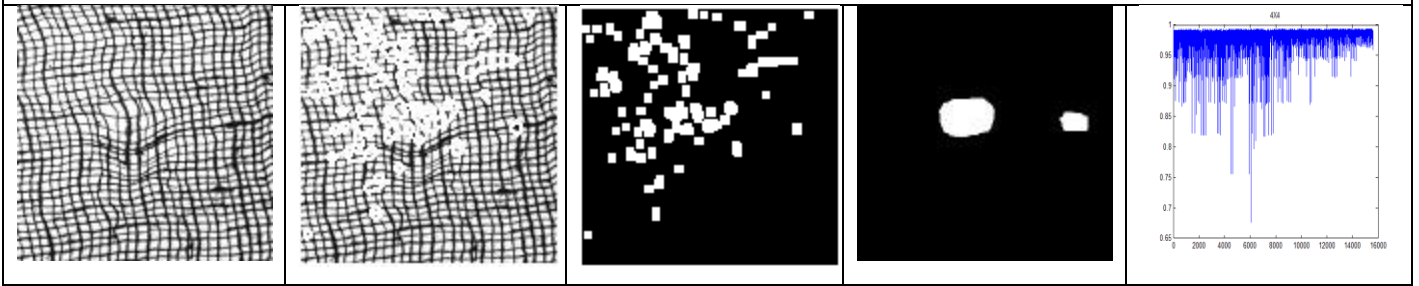
6. defect7



7. defect8



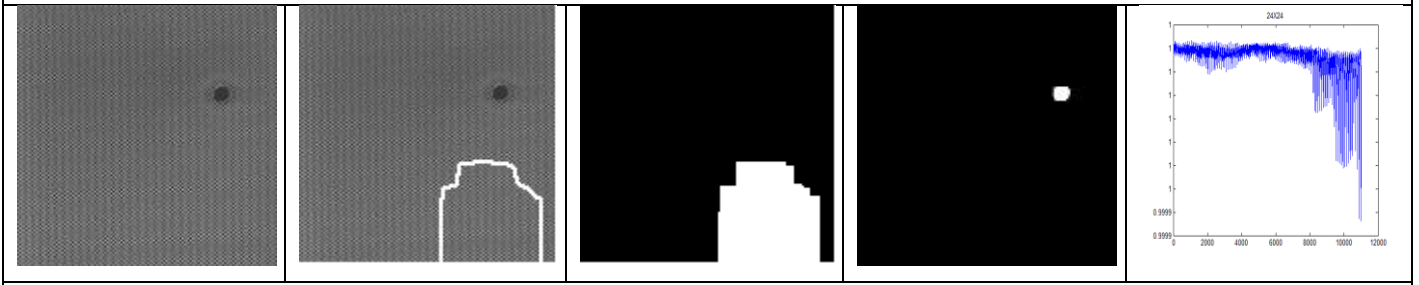
8. d6



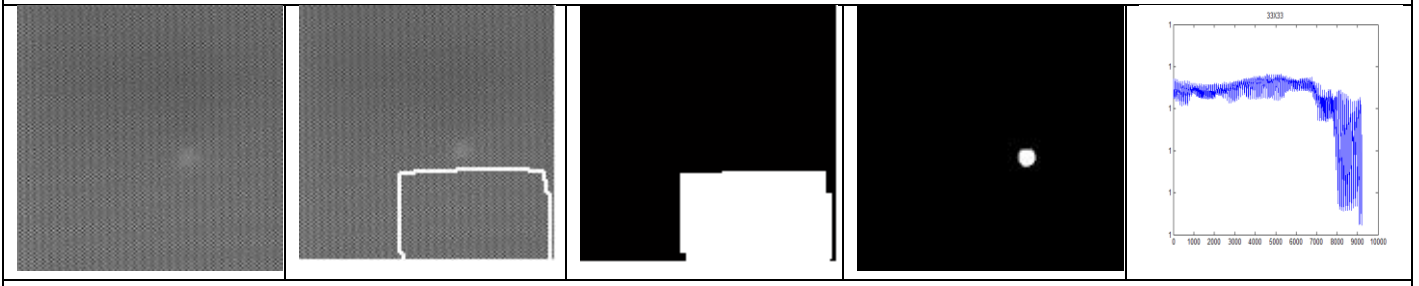
9. d7



10. df



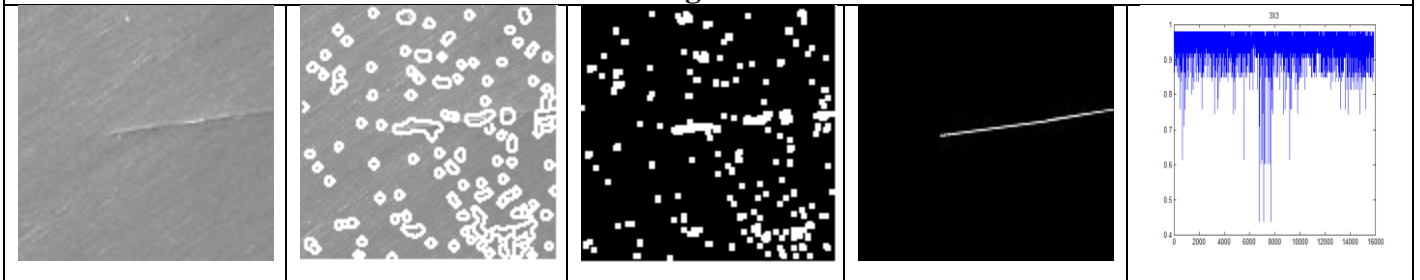
11. dk1



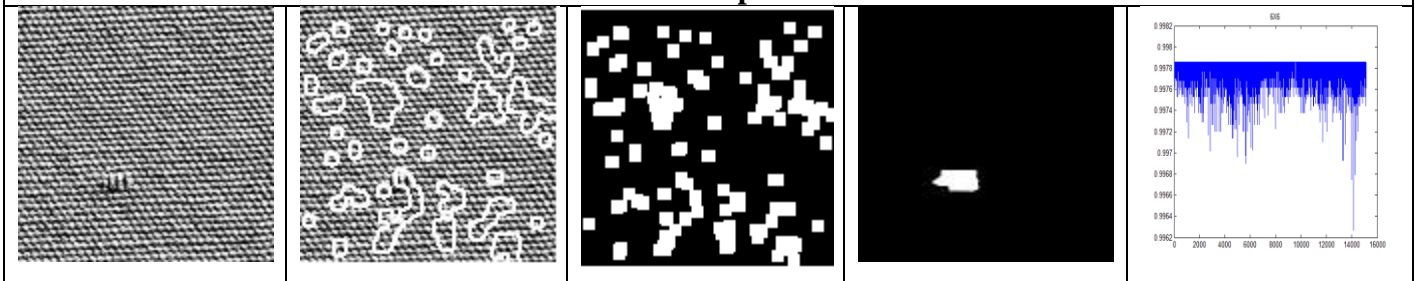
12. dk1



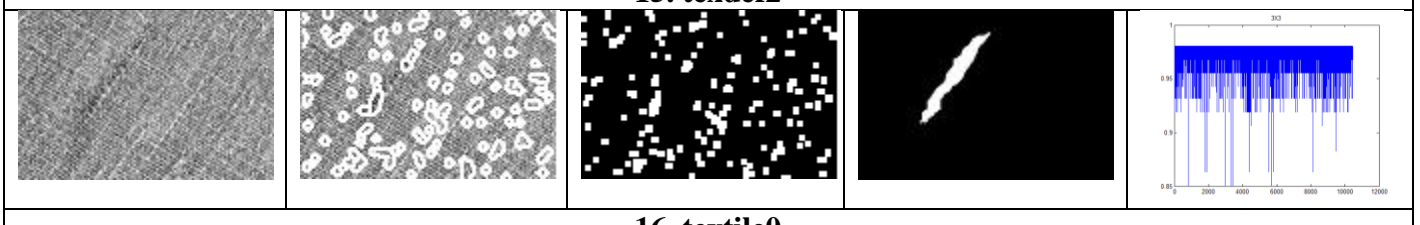
13. guwei



14. sap



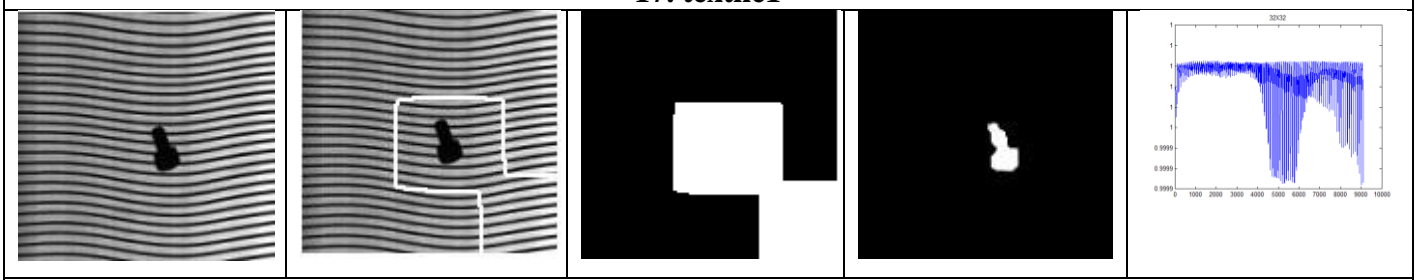
15. texdef2



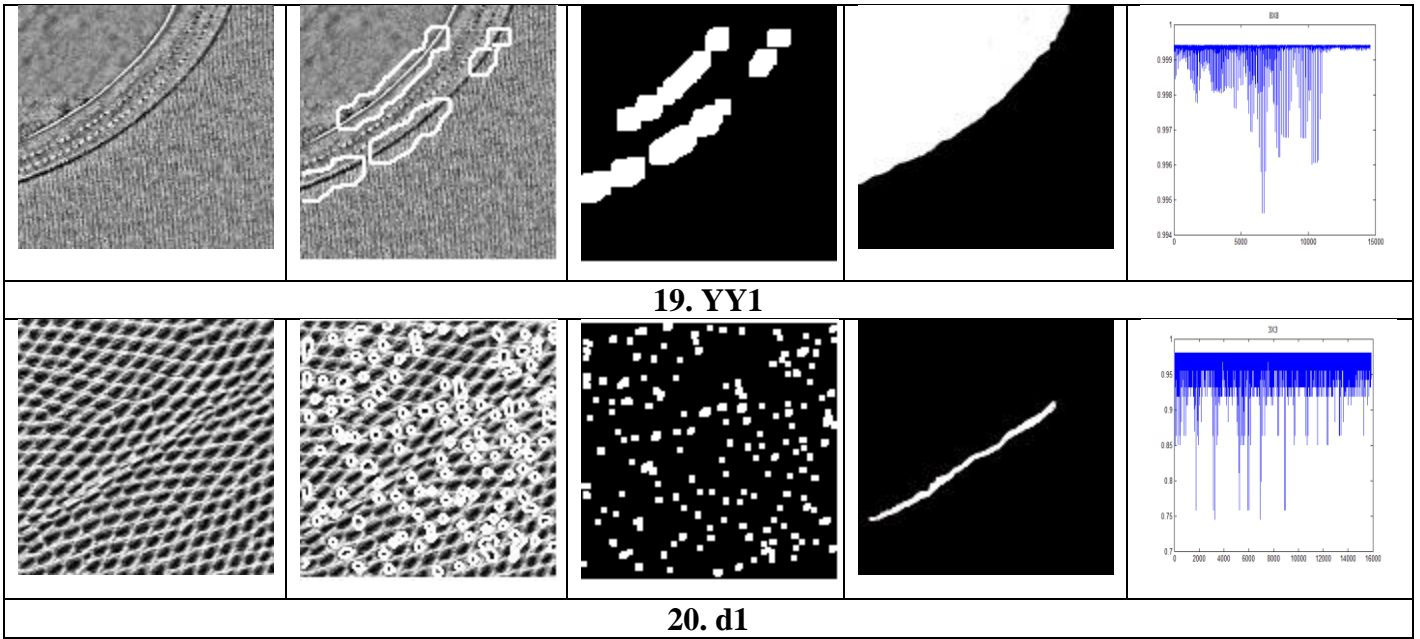
16. textile0



17. textile1

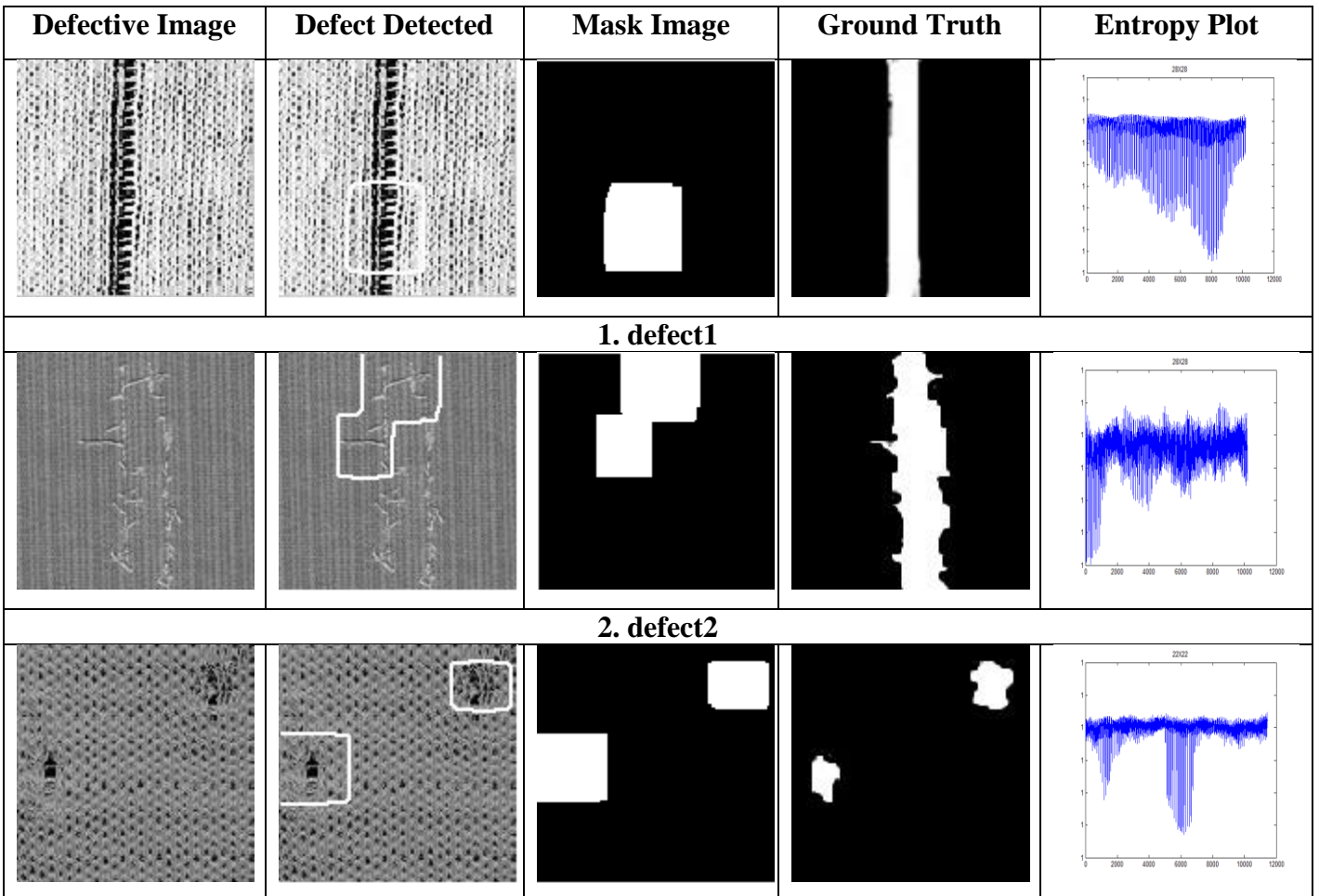


18. x

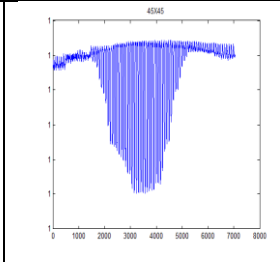
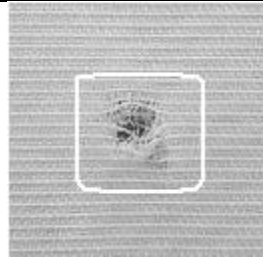
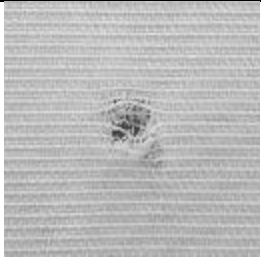


8.3.5.2 Results of Max Entropy Method

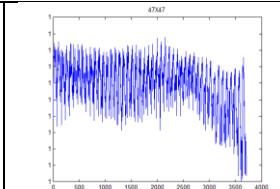
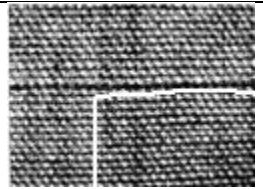
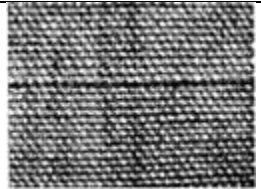
Table 8.8 Result of Max Entropy method



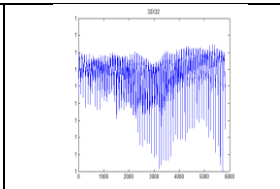
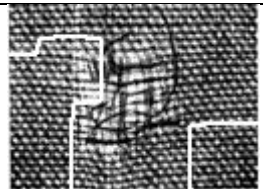
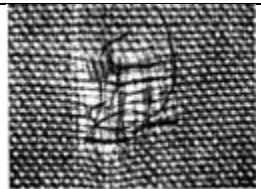
3. defect3



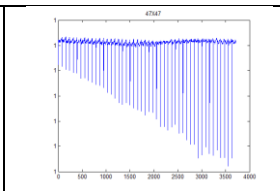
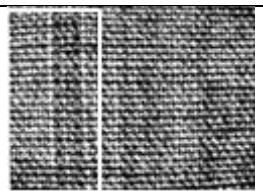
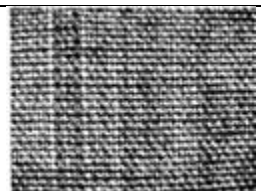
4. defect4



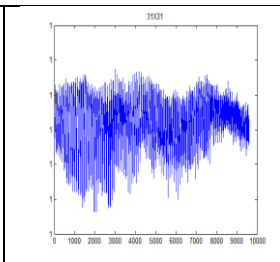
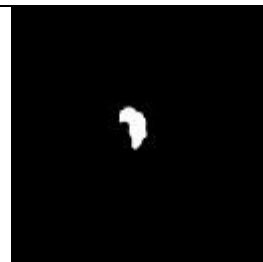
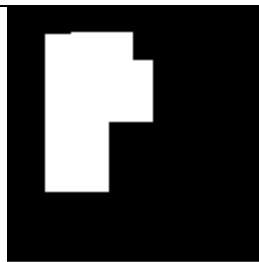
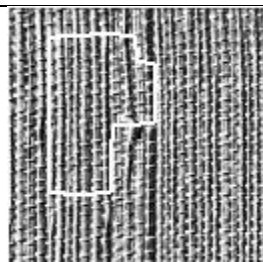
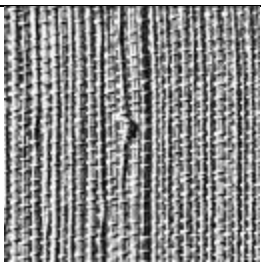
5. defect6



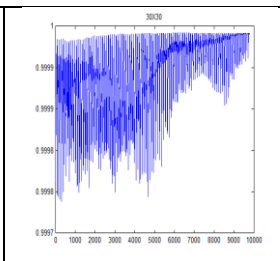
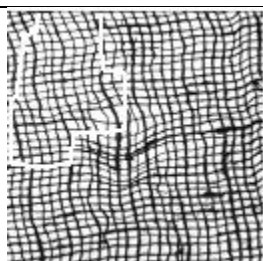
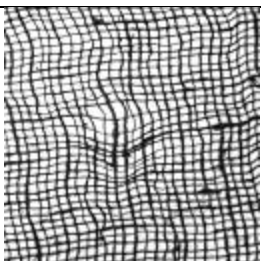
6. defect7



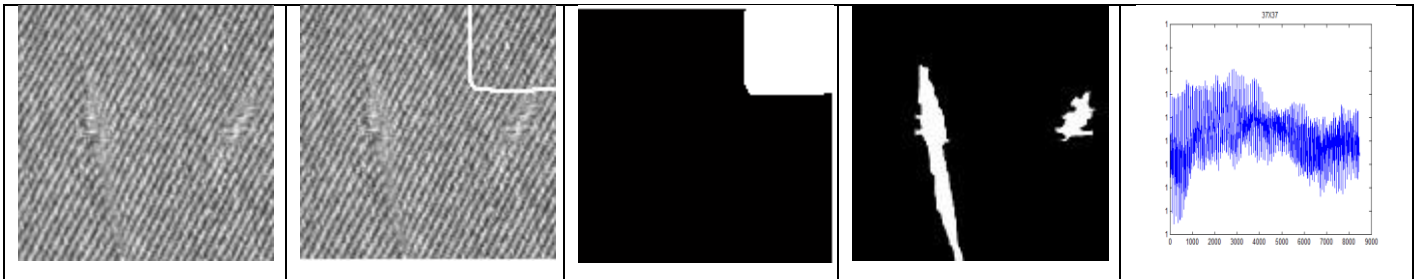
7. defect8



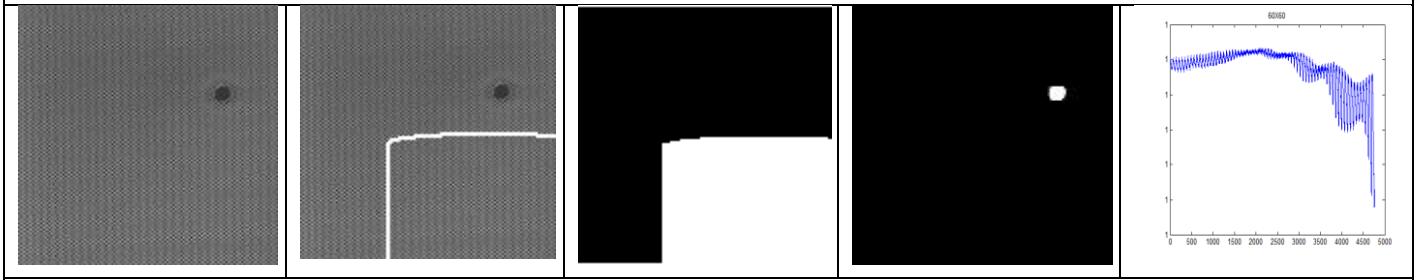
8. d1



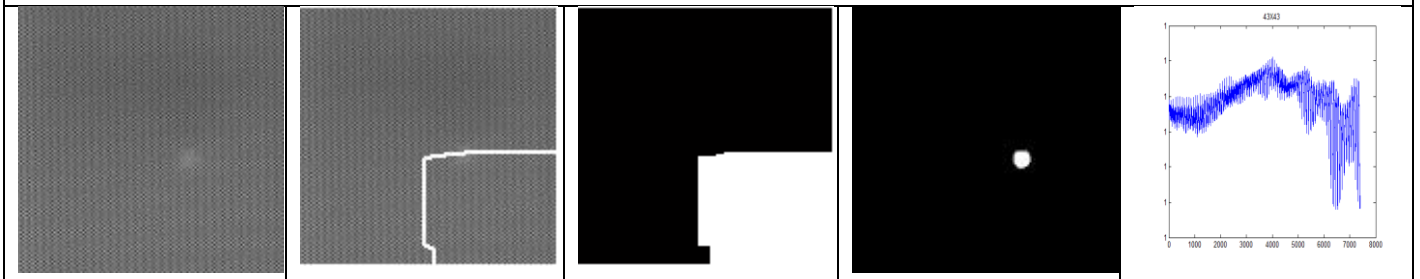
9. d7



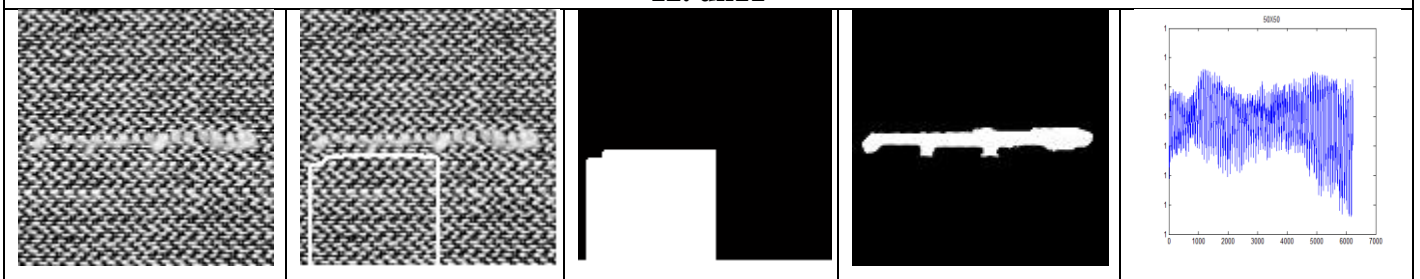
10. df



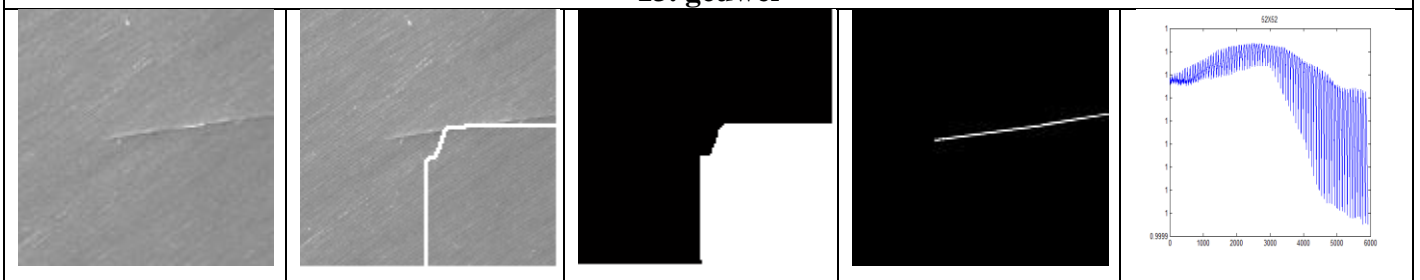
11. dk1



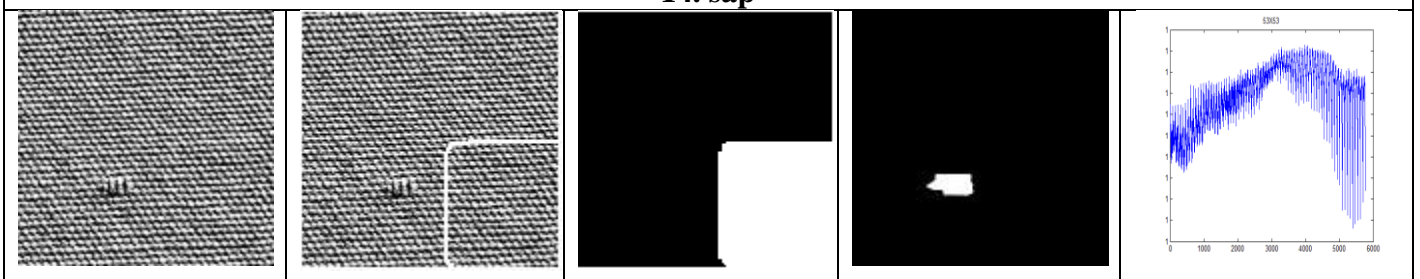
12. dk11



13. gcuwei



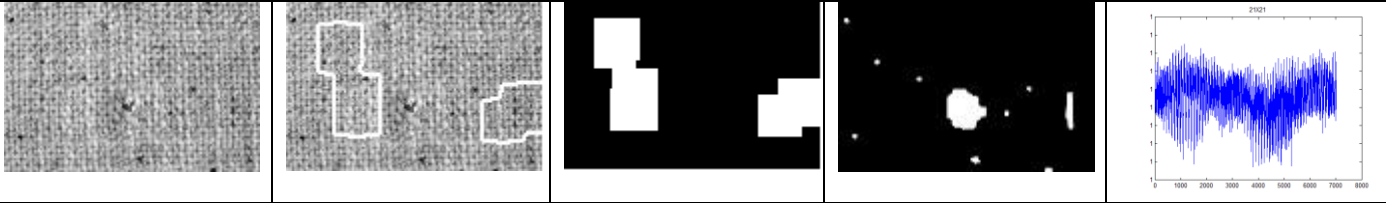
14. sap



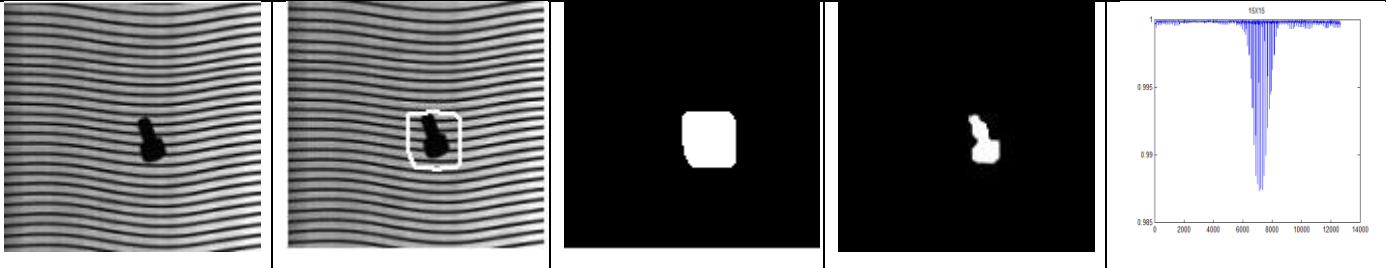
15. texdef2



16. textile0



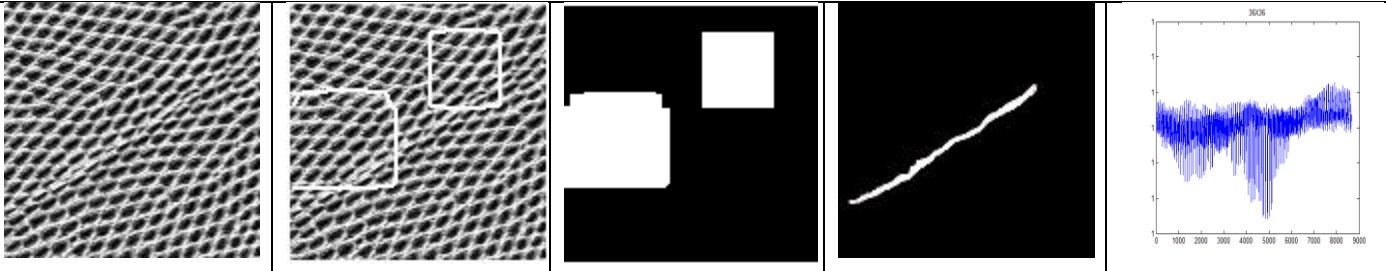
17. textile1



18. x



19. YY1

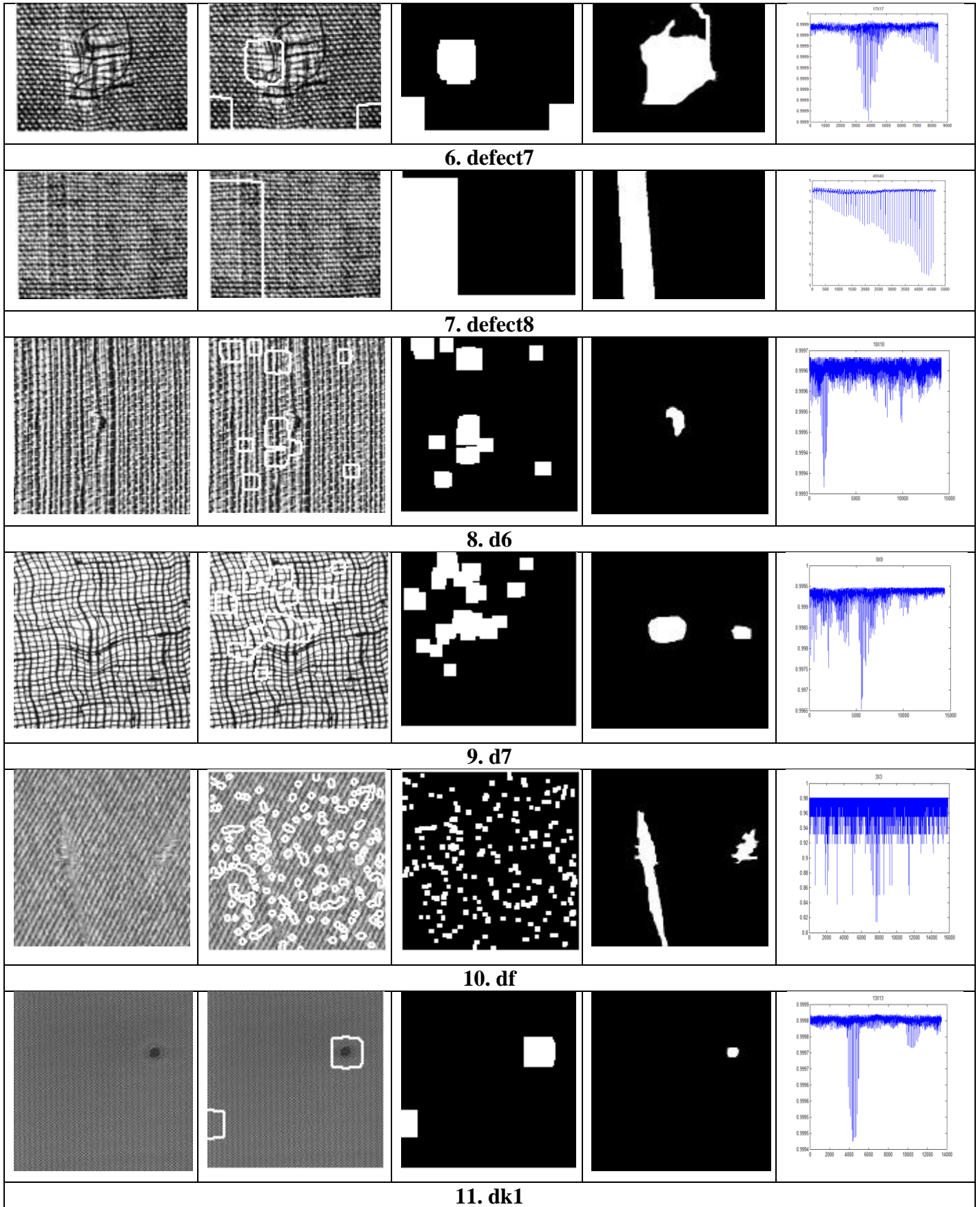


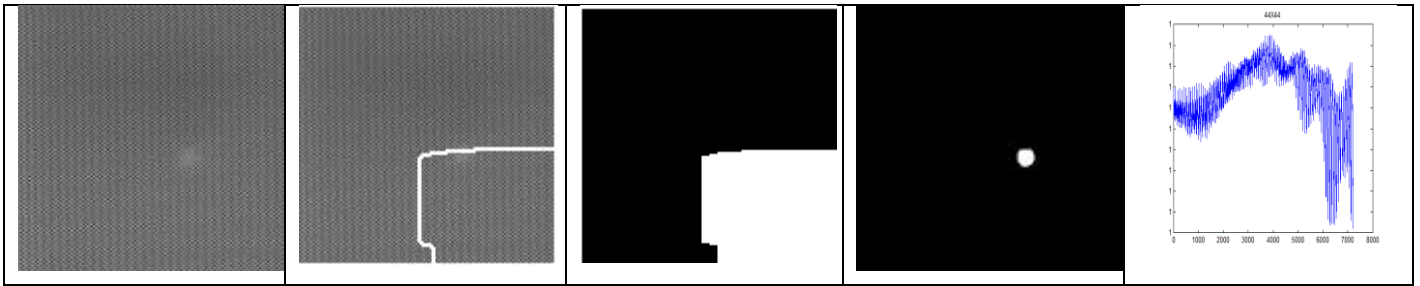
20. d1

8.3.5.3 Results of GM Distribution Method

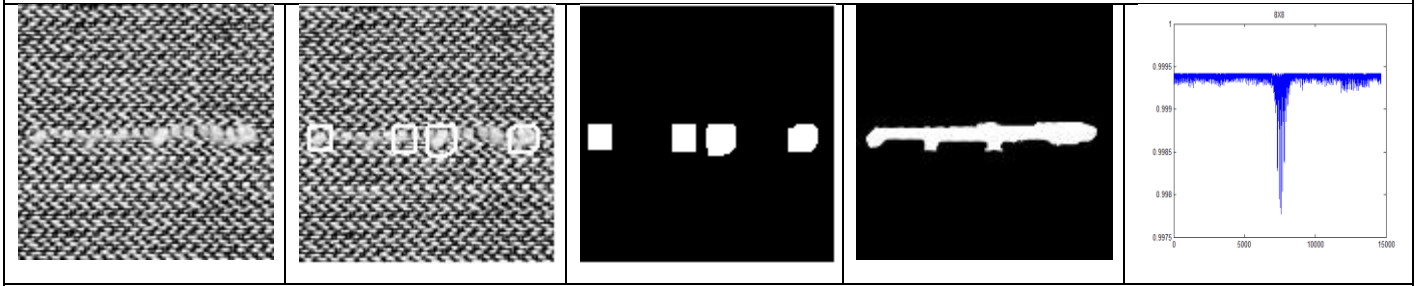
Table 8.9 Result of GM distribution method

Defective Image	Defect Detected	Mask Image	Ground Truth	Entropy Graph
1. defect1				
2. defect2				
3. defect3				
4. defect4				
5. defect6				

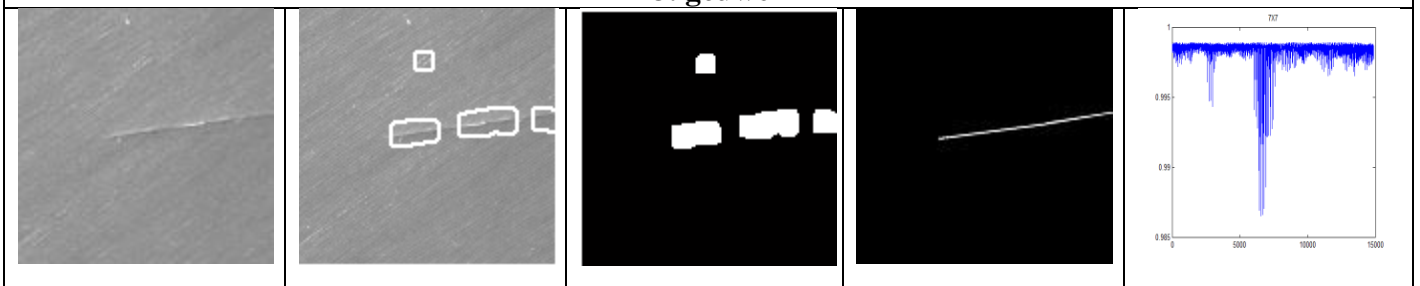




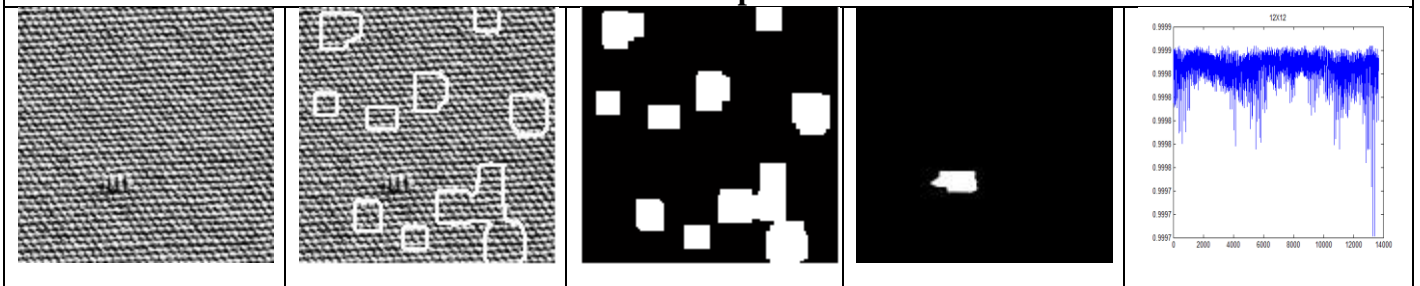
12. dk11



13. gcuwei



14. sap



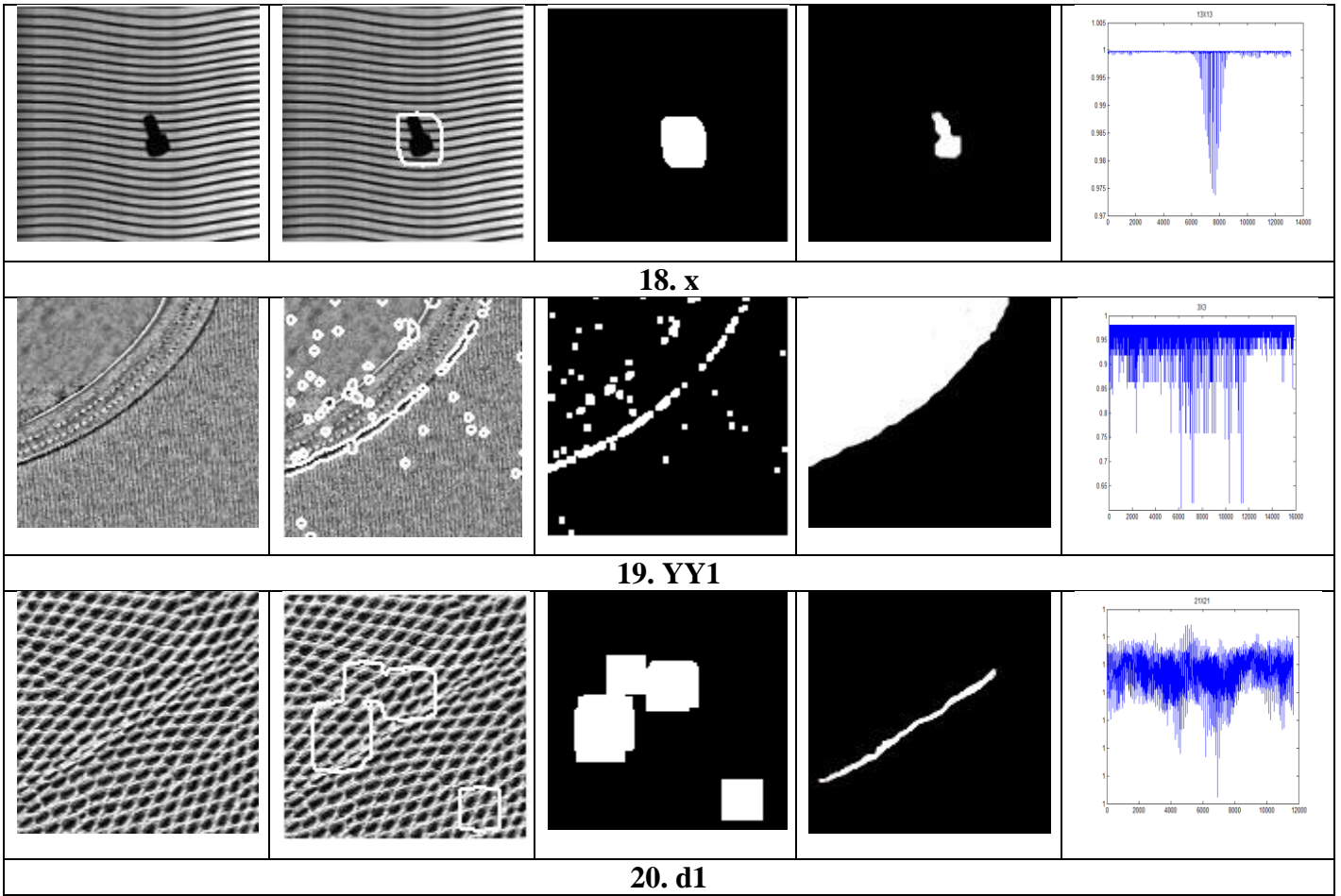
15. texdef2



16. textile0

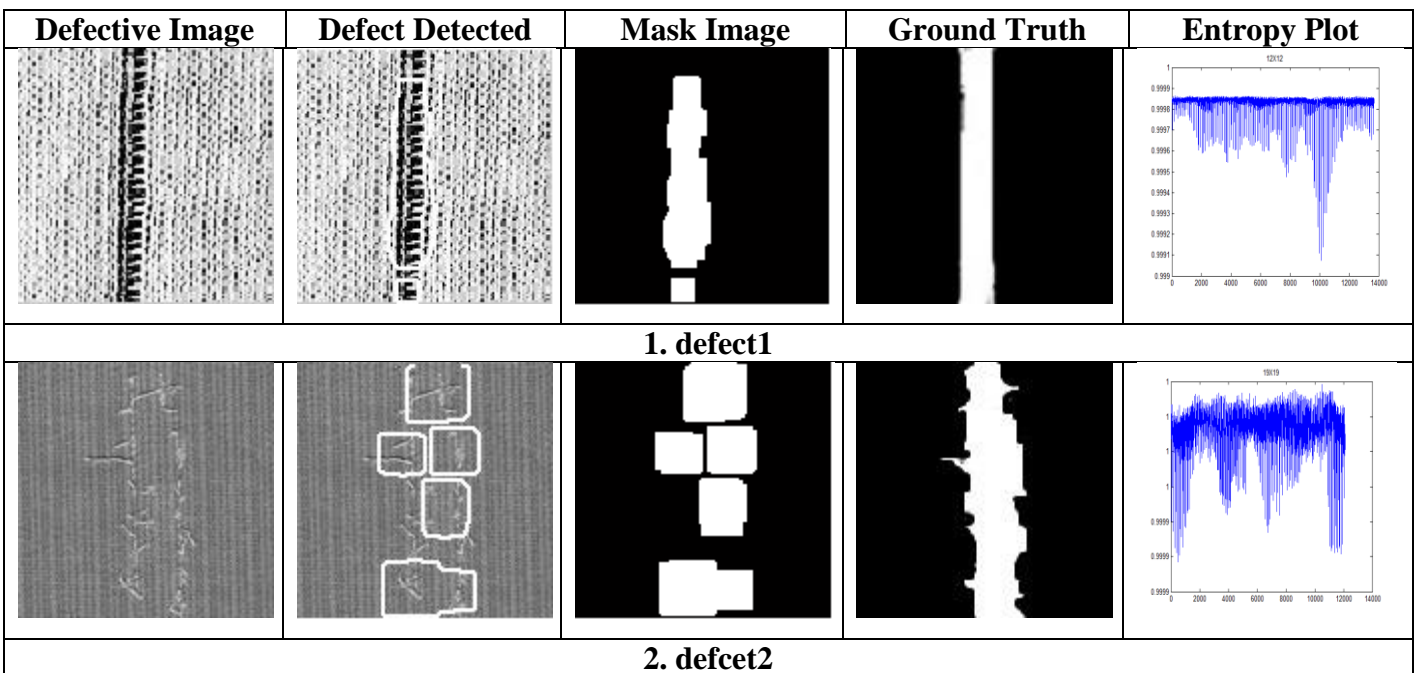


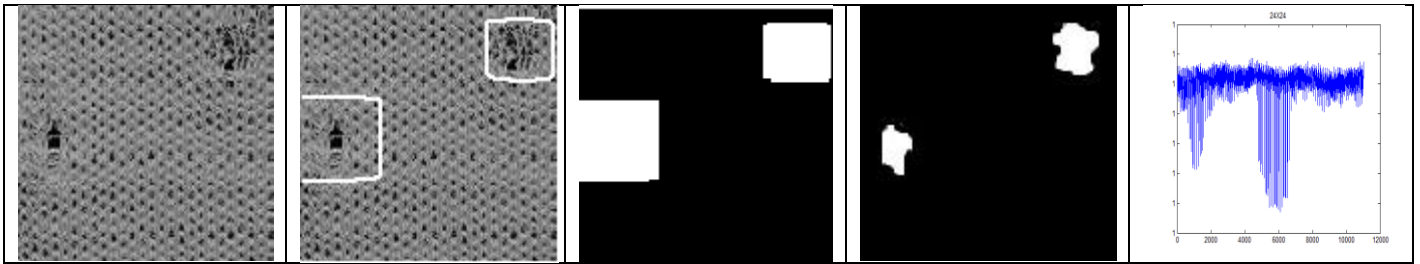
17. textile1



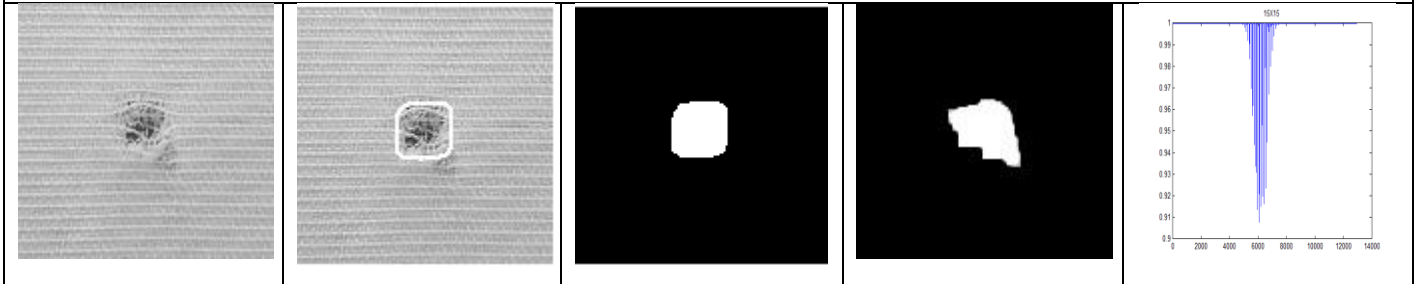
8.3.5.4 Results of Combination of GM Distribution and Max Count Method

Table 8.10 Result of combination of GM distribution and max count method





3. defect3



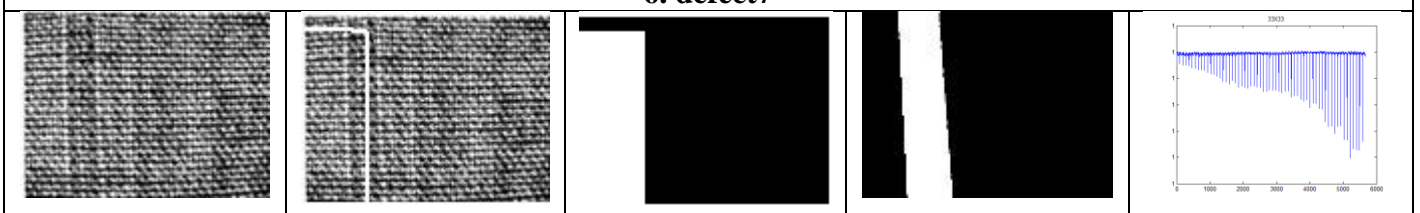
4. defect4



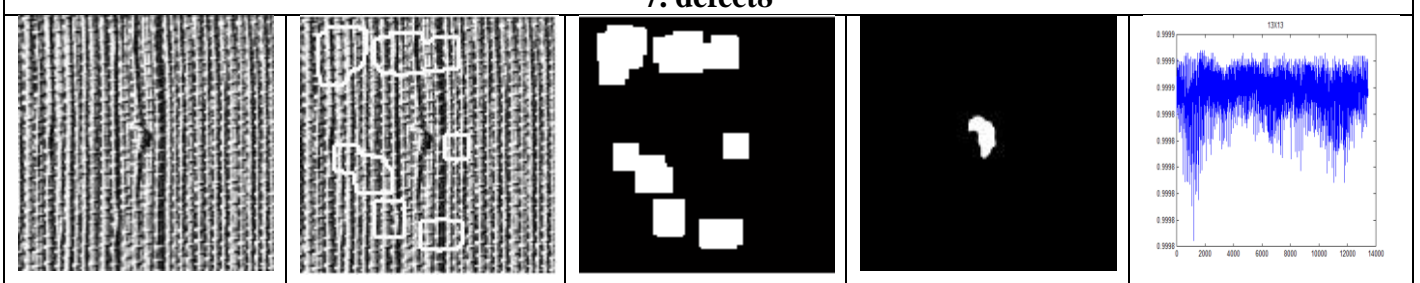
5. defect6



6. defect7



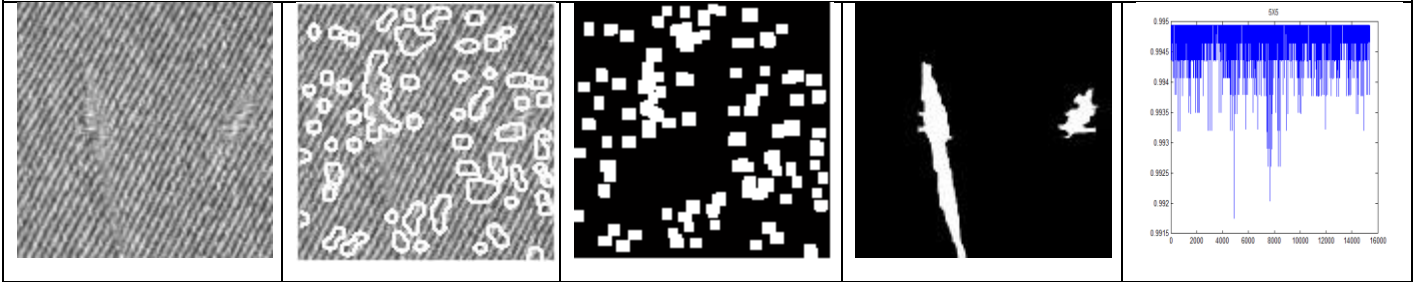
7. defect8



8. d6



9. d7



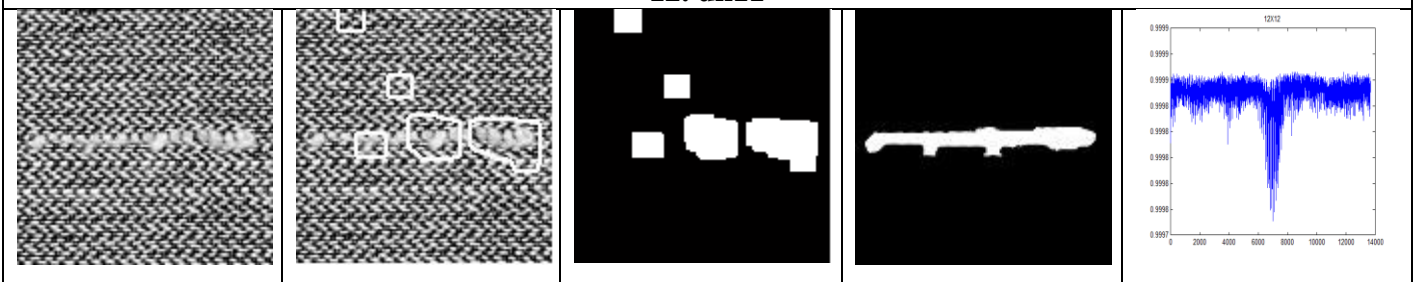
10. df



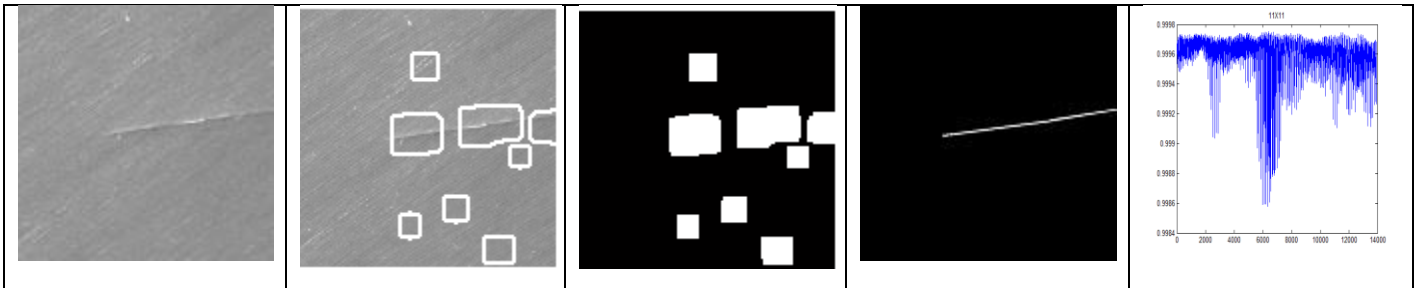
11. dk1



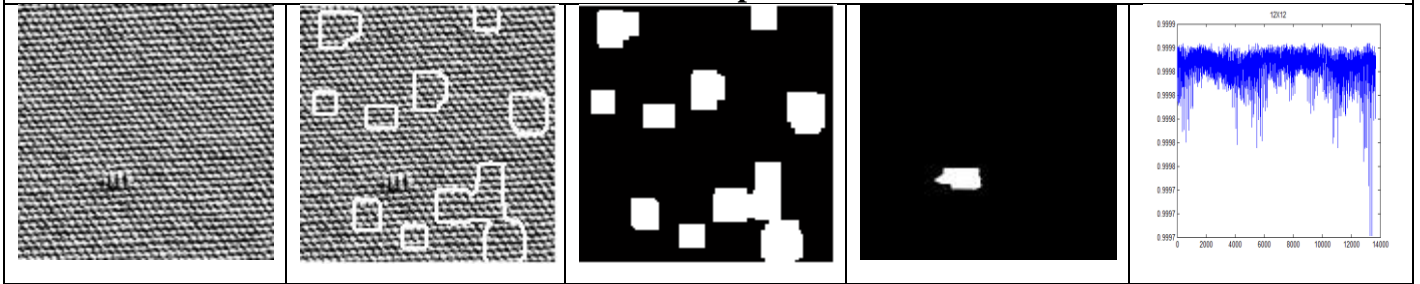
12. dk1



13. guwei



14. sap



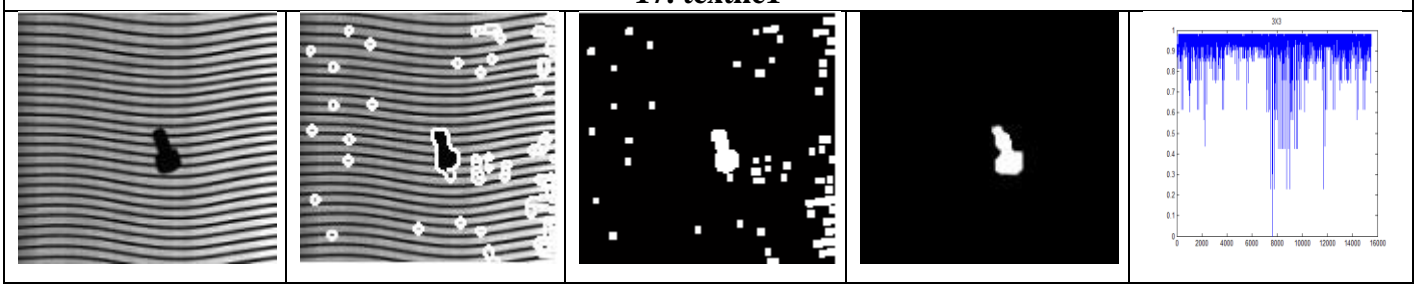
15. texdef2



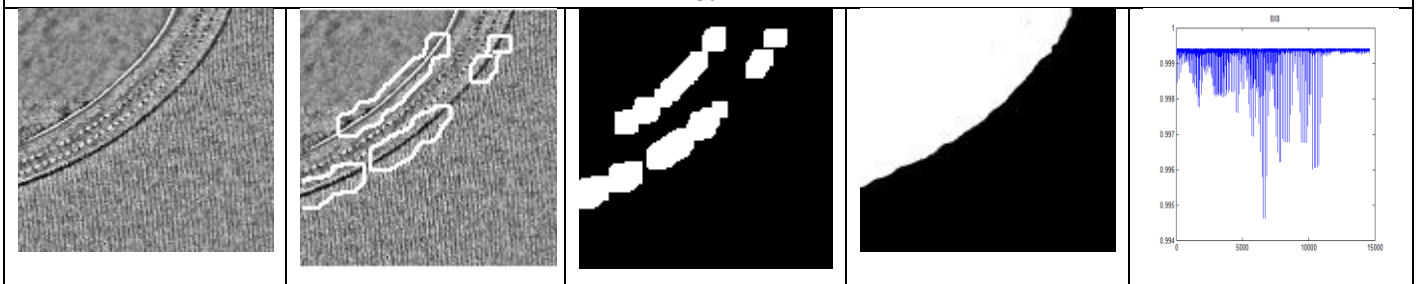
16. textile0



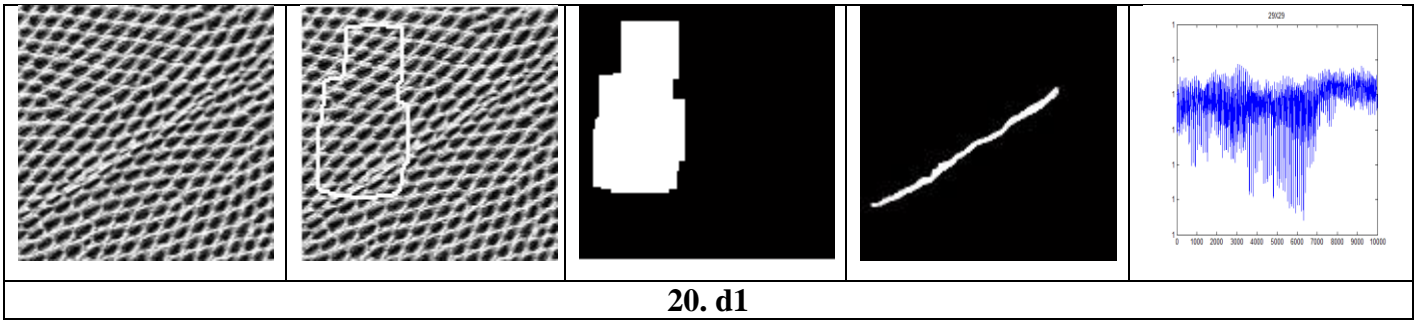
17. textile1



18. x



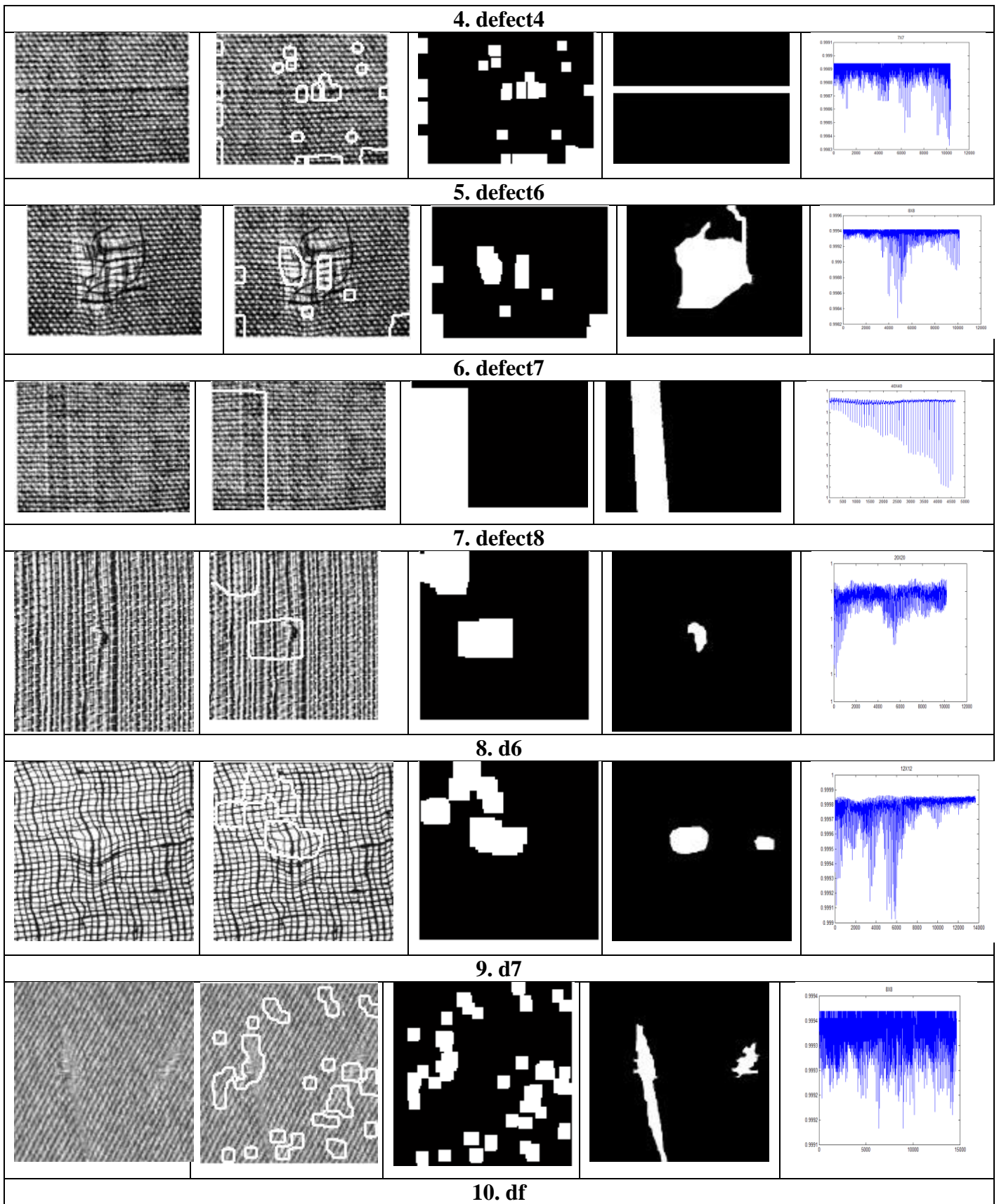
19. YY1

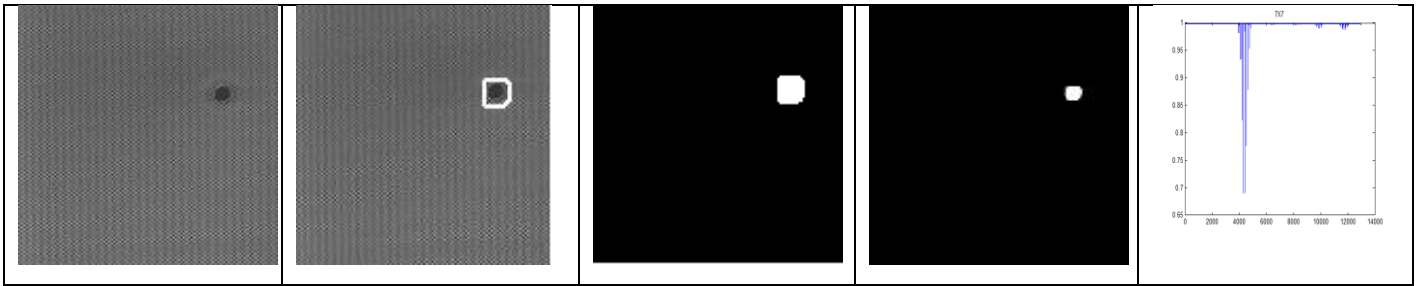


8.3.5.5 Results of Ideal Case

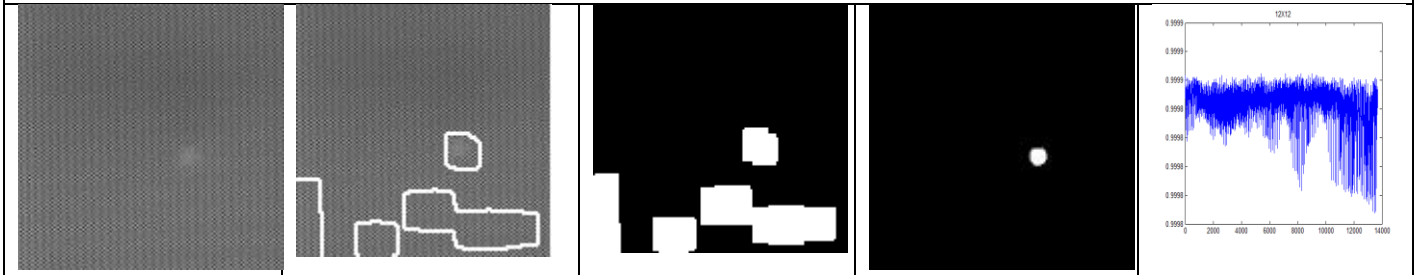
Table 8.11 Result of Ideal Case of Variable window size Entropy Method

Defective Image	Defect Detected	Mask Image	Ground Truth	Entropy Plot
1. defect1				
2. defect2				
3. defect3				





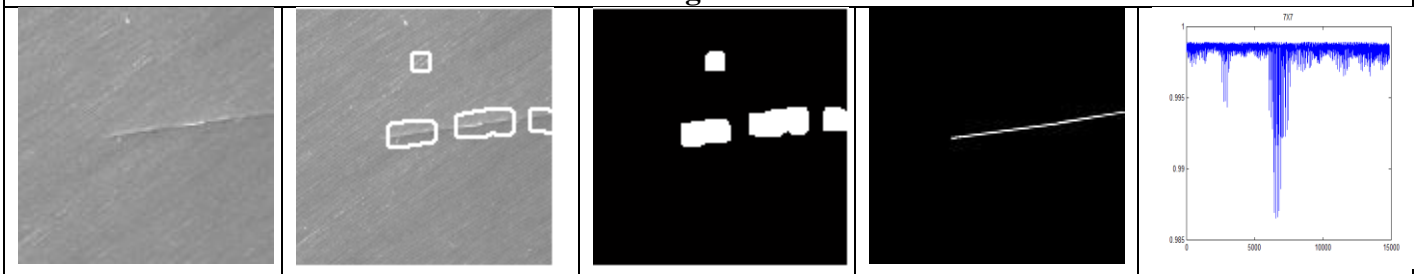
11. dk1



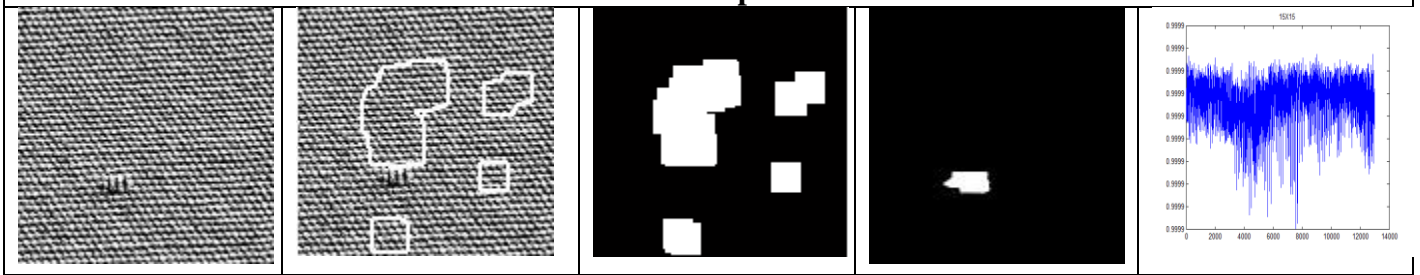
12. dk11



13. guwei



14. sap



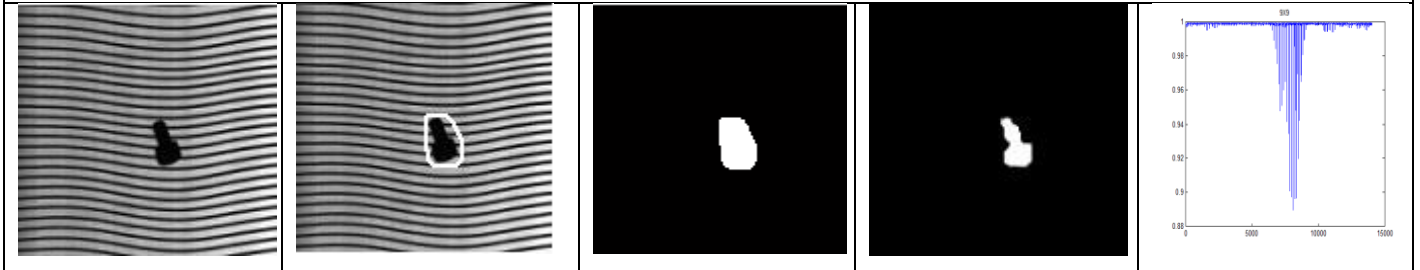
15. texdef2



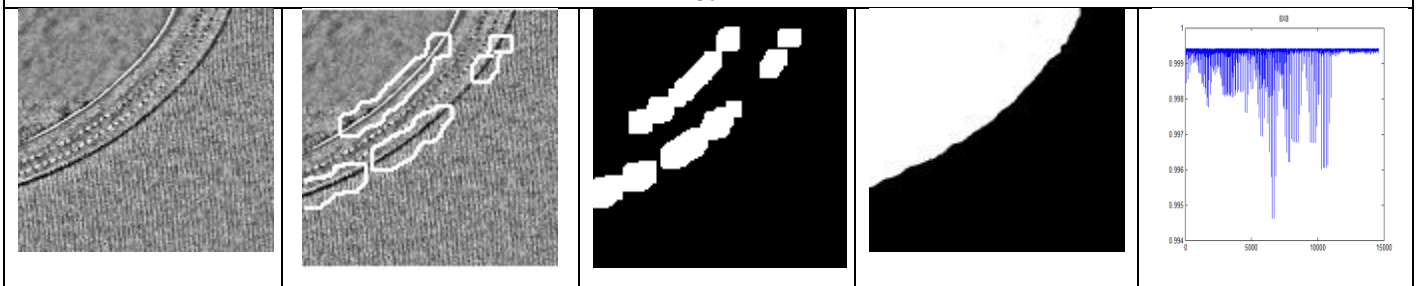
16. textile0



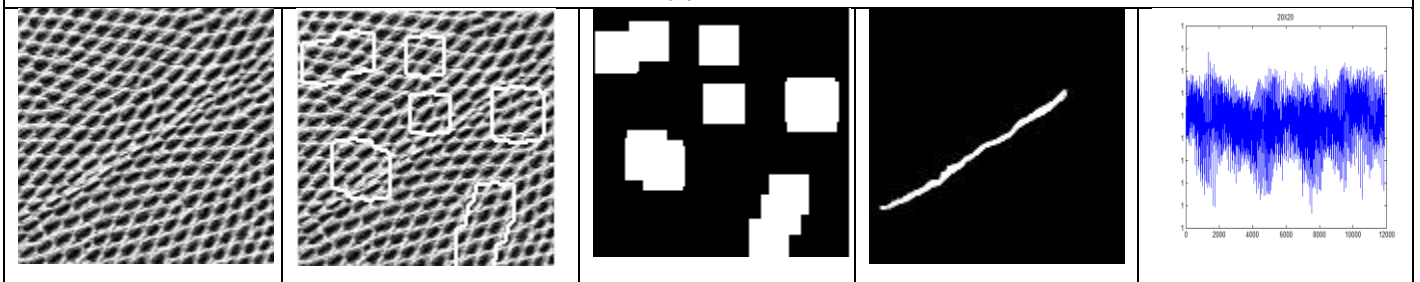
17. textile1



18. x



19. YY1



20. d1

8.4 Results of Objective Comparison of Chetverikov's and Entropy Method on Images with synthetic defects

The following table shows comparison of Precision, Recall and F-measure for both the methods of defect detection:

Table 8.12 Precision, Recall and F-measure values for Chetverikov and Entropy Method on synthetic defects

S.no	Image Name	CHETVERIKOV'S METHOD			ENTROPY METHOD		
		Precision	Recall	Fmeasure	Precision	Recall	Fmeasure
a)	brodatzD22	0.068103117	1	0.127521614	0.942028986	0.902777778	0.921985816
b)	defect_gpattern2	0.096057681	1	0.175278515	0.204402516	0.631067961	0.308788599
c)	defect_gpattern3	0.070895522	1	0.132404181	0.252539913	0.783783784	0.381997805
d)	brodatzD34	0.44950849	0.954459203	0.611178615	0.228658537	0.815217391	0.357142857
e)	defect_brodatz100	0.090681676	1	0.166284404	0.171521036	0.92173913	0.289222374
f)	defect1_gpattern1	0.046959826	1	0.089707026	0.098039216	0.948275862	0.177705977
g)	defect_pattern14	0.802308078	0.975	0.880264244	0.975967957	0.906947891	0.940192926
h)	defectmesh21	0.127283271	1	0.225823045	0.183870968	0.850746269	0.302387268

From the table we can conclude that Chetverikov's Method performs better than Entropy Method in the case where window size is kept same for all images for Entropy Method.

8.5 Result of Comparison of all the three methods on images with complex defects

Table 8.13 Results of Comparison of Regularity, Entropy and GLCM methods on complex defects

S.no	Image Name	Regularity Method	Entropy Method	GLCM Method
		Ncorr:Nfalse	Ncorr:Nfalse	Ncorr: Nfalse
1	defect1	0.3881	3.4062	2.860294118
2	defect2	0.3875	0.0854	0.106086957
3	defect3	0.1628	0.039	2.96
4	defect4	0.1999	0	0.636363636
5	defect6	0.1225	0	5.153846154
6	defect7	2.940705128	1.41509434	0.690340909
7	defect8	0.62138853	0	0.0625
8	d6	0.4223	0.0425	0.005348837
9	d7	0.3527	0.2869	0.274314214
10	df	0.2976	0.4806	1.050541516
11	dk1	0.1087	0.4246	0
12	dk11	0.1903	0.1042	0
13	gcuwei	0.2519	2.037	0.469635628
14	sap	0.007	0	0.063973064
15	texdef2	0.1969	0.0101	0.161392405
16	textile0	0.5432	0.0285	0.053225806
17	textile1	0.3259	0.0762	0.005637773
18	x	0.0988	1.7142	0.003487358
19	YY1	22.1096	12.1917	0.93062201
20	d1	0	0.0348	0.213675214

The coloured row indicates the best method for each image.

As we can see that, there are majority of yellow rows, which again indicates that Chetverikov's Method is better among all the three methods.

8.6 Result of Comparison of four Automation Methods

Table 8.14 Result of Comparison of Max Count, Max Entropy, GM distribution and combination of GM distribution and Max count method

Image Name	Max Count Method		Max Entropy Method		GM Distribution Method		Combination of GM Distribution and Max Count		Ideal Case	
	Ncorr:Nfalse	Window size	Ncorr:Nfalse	Window size	Ncorr:Nfalse	Window size	Ncorr:Nfalse	Window size	Ncorr:Nfalse	Window Size
defect1	2.96082474	12	0.647993311	28	5.6121673	9	2.96082474	12	5.6121673	9
defect2	1.43227848	19	0.595072647	28	0.723655122	25	1.43227848	19	1.43227848	19
defect3	0.35993821	24	0.451842275	22	0.149507305	34	0.35993821	24	0.57356608	20
defect4	0.30056497	39	0.277083333	45	Inf	5	5.45132743	15	5.45132743	15
defect6	0.16073969	3	0	47	0	17	0.1414966	7	0.1414966	7
defect7	0.53164557	3	0.067610903	32	0.488284911	17	0.85714286	8	0.85714286	8
defect8	0.03130435	4	0.851203501	47	0.950659782	40	0.61893896	33	0.95065978	40
d6	0.02990654	4	0.022884673	31	0	10	0	13	0.07937262	20
d7	0.19141324	4	0.070807853	30	0.211281544	9	0.25397727	12	0.25397727	12
df	0.14110825	5	0	37	0.134890372	3	0.14110825	5	0.15839416	8
dk1	0	24	0	60	0.074074074	13	0	23	0.47272727	7
dk11	0	33	0.011835948	43	0.012903226	44	0	34	0.02565147	12
gcuwei	0.65846995	12	0.001130902	50	1.8828125	8	0.65846995	12	1.39082278	9
sap	0.03304348	3	0.0065037	52	0.09557945	7	0.04082613	11	0.09557945	7
texdef2	0.02340953	6	0	53	0	12	0	12	0	15
textile0	0.06743869	3	0.055535521	42	0.088039867	17	0.07001239	18	0.16938998	14
textile1	0.13761929	9	0.035429142	21	23.375	8	0.13761929	9	23.375	8
x	0.06354335	32	0.488612836	15	0.650137741	13	0.22366864	3	1.3038674	9
YY1	1.94412607	8	Inf	42	1.617721519	3	1.94412607	8	1.94412607	8
d1	0.0168204	3	0.043407488	36	0.054189085	21	0.04430985	29	0.03107489	20

The coloured row indicates the best method for each image.

Starting from the Segmentation approach, let us discuss the results of each method. Segmentation method works good only for images with dark coloured defects. As it can be seen from the Table 8.1, the algorithm detects defects clearly only in first six images. In the last four images, the defects are not detected properly. This is because, the algorithm assumes defects to be present only in the darker region which may not be the case always.

Fig. 8.1 shows the Maximal Regularity values for four images ranging from random to the most regular image. As discussed in Chapter 4, Maximal Regularity quantifies the human visual perception of regularity, so accordingly, Img1 which is random in nature shows very low value of Maximal Regularity. Img4 which is most regular shows maximum value of Maximal Regularity. Maximal Regularity values is further used to identify defects in images.

Results of Chetverikov's Regularity Method on synthetic defect images can be seen in Table 8.2 and on complex defect images in Table 8.3 respectively. This method works good for both types of images, but defects are not identified precisely as there is large non defective area is also shown to be as a defect.

Next, we can see results of Constant window size Entropy Method on synthetic defect images and on complex defect images in Table 8.4 and Table 8.5 respectively. The results of Table 8.4 are much better in terms of precision than results of Table 8.2. But the results of Table 8.5 are not good as compared to the results of Table 8.3.

Next, we come to the results of GLCM Method applied on complex images. As we can see from the Table 8.6, textures like defect1, defect6 and df show good results but most of the images do not show satisfactory results.

Next four tables discuss the four methods which are used in the Variable window size Entropy Method. Subsequent paragraphs discuss their results and comparison.

As we can see, that, there are majority of orange rows in the Table 8.14 which indicates that if we use Variable window size Entropy Method for defect detection, then to find the best window size for a certain image, we can use GM Distribution Method. The table also shows the ideal case results of Variable window size Entropy Method. These results have been manually calculated.

As we can see from the Table 8.12, Constant window size Entropy Method has better precision values than Chetverikov's Method (Regularity Method) in case of synthetic defect images. As it can be seen from the results of Table 8.4, defects are very accurately detected by Entropy Method than Regularity Method by Chetverikov in Table 8.2.

Though Entropy Method wins in the case of synthetic images but if we apply the same method to complex defect images, the Entropy Method fails for most of the images. Texture patterns like defect2, defect3 and defect4 from Table 8.5 can be seen failing the Entropy Method. The defects are not identified correctly. Although, a small amount of positive result can be seen in images like defect1, defect7, gcuwei and x. This failure results in the poor results of Entropy Method in Table 8.13, where we can see that Chetverikov's Regularity Method has majority of yellow rows amongst the three methods of defect detection, Chetverikov's Regularity methods stands best.

Now, that the Constant window size Entropy Method was not working properly, we decided to follow a second approach in which different textures work best for different window sizes This is because the defect size is different in different images. So, if one window size works best for one image it may not work best for another image. If the defect is small, a small window size is suitable for that image but if the defect is large, then a large window size is required to detect that defect. So, we implemented the second approach called Variable window size Entropy Method. Initially, we tested the texture images manually to find the best window size for which it was giving best results. This process was done manually by selecting a window size and then executing the program for that window size. The results of finding the best window size manually for all the images can be seen in the Table 8.11 and it is termed as the Ideal Case results. Now, if we compare the results of Table 8.3 and Table 8.11, then we can see, that except for few images, Entropy method can be seen working much better than Chetverikov's Regularity Method. There is good accuracy and precision in the results of Entropy Method as compared to Regularity Method.

But as we can see, there was a last step to automate this Variable window size Entropy Method and it was to find the best window size which can provide the best result for a given image. This window size which was to be found out should match the window size found in Ideal Case and if not match then, it should be as close as possible to the window size found in Ideal Case. So, to find out the best window size, four methods have been employed which are discussed in Chapter 7. The results of these four methods have been shown in Table 8.7, Table 8.8, Table 8.9 and Table 8.10. And Table 8.14 gives the objective comparison of all these four methods. Among these four methods, GM Distribution has majority of coloured rows (orange), which indicate that it is the best method amongst all the

four methods to find the most suitable window size for a given image. Now if we compare the ratio of N_{corr} and N_{false} for GM Distribution Method from Table 8.14 with the ratio of N_{corr} and N_{false} of Regularity Method from Table 8.13, then we can see, that except for few images, our results are comparable with the results of Regularity Method. For some images like defect1, defect2, defect8, gcuwei, sap, textile1, x and d1, our results are much better than Regularity Method. For images like defect3, d7 and dk1, our results are comparable to that of results of Regularity Method. Although, for images like defect6, texdef2, dk11 and textile0 Regularity results are better. All this was not in the case when we kept the window size constant for Entropy Method. So, it becomes clear that Variable window size Entropy Method is much better than Constant window size Entropy Method and if we use GM Distribution Method for finding optimum window size, our results have better precision and accuracy as compared to Chetverikov's Regularity Method.

CHAPTER-9

CONCLUSION AND FUTURE WORK

9. Conclusion and Future Work

Though Constant Window size Entropy Method did not give good results, but every image shows a very good result for a certain window size. That means using the Variable window size Entropy Method gives much better results than Chetverikov's Regularity Method for some common images. To find this, four automation methods are used to find the best window size and from among these four methods, the best method is selected in the Variable window size Entropy Method. Each of these automation methods of defect detection by entropy runs good on some images but not uniformly on all the images. Among these, GM Distribution is found to work best on most of them. Though Variable window size Entropy Method using GM Distribution to identify the window size gives better results than Regularity Method in many cases, there are still few drawbacks for which future work can be done to improve this method and make it the best method for texture defect detection. There are cases of certain images where defects are not identified by any of the four automation methods. Even GM Distribution does not give good results in certain cases. Also, it does not find the exact matching window size as in the Ideal Case. So, the future scope of this work would be to find a single best method which gives good results on all the images. There are some images which have very small sized defect, this method was not able to identify those defects. And also, we can workout a better way to identify a window size which matches with the window size found in Ideal Case. So, these are the few aspects on which future work can be done in this direction to improve it so that it can detect all the types of defects with good precision and accuracy.

References

1. Ajay Kumar, "Computer Vision-based Fabric Defect Detection: A Survey"
2. J. Huart and J.-G. Postaire, "Integration of computer vision on to weavers for quality control in the textile industry," Proc. SPIE 2183, pp. 155-163, Feb. 1994.
3. J. L. Dorrity and G. Vachtsevanos, "On-line defect detection for weaving systems," Proc. IEEE Annual Technical Conf. Textile, Fiber, and Film Industry, pp. 1-6, May 1996.
4. Ryan G. Rosandich, Intelligent Visual Inspection, Chapman & Hall, London (U.K.), 1997.
5. B. N. Nickolay and H. Schmalfuß, "Automatic fabric inspection – utopia or reality?," Mellind Textilberichte, vol. 73, pp. 33-37, 1993.
6. L. H. Hoffer, F. Francini, B. Tiribilli, and G. Longobardi, "Neural networks for the optical recognition of defects in cloth," Opt. Eng., vol. 35, pp. 3183-3190, Nov. 1996.
7. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification" , IEEE Trans. On Systems, Man and Cybernetics, 1973, pp.610-621.
8. W. J. Jasper and H. Potapalli, "Image analysis of mispicks in woven fabrics," Text. Res. J., vol. 65, pp. 683-692, 1995.
9. A. Conci and C. B. Proença, "A computer vision approach to textile inspection," Text. Res. J., vol. 70, pp. 347-350, Apr. 2000.
10. J. S. Lane, "Textile fabric inspection system," US Patent No. 5,774,177, Jun. 1998.
11. B. R. Abidi, H. Sari-Sarraf, J. S. Goddard, and Martin A. Haunt, "Facet model and mathematical morphology for surface characterization," Scientific Literature Digital Library, <http://citeseer.ist.psu.edu/284961.html>
12. L. Norton-Wayne, M. Bradshaw, and A. J. Jewell, "Machine vision inspection of web textile fabric," Proc. British Machine Vision Conf., Leeds (U.K), pp. 217-226, Sep. 1992.
13. L. Norton-Wayne, M. Bradshaw, and C. Sandby, "Machine vision for the automated inspection of web materials," Proc. SPIE 1989, pp. 2-13, 1993.
14. M. Bradshaw, "The application of machine vision to the automated inspection of knitted fabrics," Mechatronics, vol. 5, no. 2/3, pp. 233-243, 1995.
15. C-S. Cho, B-M.Chung and M-J. Park, "Development of real-time vision-based fabric inspection system," IEEE Trans. Industrial Electronics, vol. 52, no. 4, Aug. 2005.

16. L. Macaire and J. G. Postaire, "Flaw detection on galvanized metallic strips in real-time by adaptive thresholding," Proc. SPIE 2183, pp. 14-23, 1993.
17. A. Kumar, "Neural network based detection of local textile defects," Pattern Recognition, vol. 36, pp. 1645-1659, 2003.
18. A. Kumar, Automated defect detection in textured materials, Ph.D. Thesis, Department of Electrical & Electronic Engineering, The University of Hong Kong, May 2001.
19. C.-C. Hung and I.-C. Chen, "Neural-Fuzzy classification for fabric defects," Text. Res. J., vol. 71(3), pp.220-224, (2001).
20. A. Kumar and H. C. Shen, "Texture inspection for defects using neural networks and support vector machines," Proc. Intl. Conf. Image Process., ICIP-2002, pp. 353-356, Rochester, New York, Sep. 2002.
21. D. Rohrmus, "Invariant web defect detection and classification system," Proc. IEEE Conf. Computer Vision & Pattern Recogn., vol. 2, pp. 794-795, Jun. 2000.
22. R. N. Bracewell, The Fourier transform and its applications, Boston: McGraw-Hill, 3rd ed., 2000.
23. H. Sari-Sarraf and J. S. Goddard, "On-line optical measurement and monitoring of yarn density in woven fabrics," Proc. SPIE 2899, pp. 444-452, 1996.
24. C. H. Chan and G. Pang, "Fabric defect detection by Fourier analysis," IEEE Trans. Ind. Appl., vol. 36, pp. 1267-1276, Sep/Oct. 2000.
25. I. S. Tsai and M. C. Hu, "Automated inspection of fabric defects using an artificial neural networks," Text. Res. J., vol. 66, pp. 474-482, Jul. 1996.
26. Y. F. Zhang and R. R. Bresee, "Fabric defect detection and classification using image analysis," Text. Res. J., vol. 65, pp. 1-9, Jan. 1995.
27. B. Mallick-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," Text. Res. J., vol. 70, pp. 758-762, Sep. 2000.
28. K.N. Sivabalan, Dr. D. Ghanadurai, "Detection of defects in digital texture images using segmentation" , International Journal of Engineering Science and Technology Vol. 2(10), 2010, 5187-5191.
29. D. Chetverikov, A. Hanbury, "Finding defects in texture using regularity and local orientation", Pattern Recognition 35 (2002) 2165 – 2180.
30. D. Chetverikov, "Pattern regularity as a visual key", Image and Vision Computing 18 (2000) 975–985.
31. I. Pitas, Digital Image Processing Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 1993.

32. P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection*, Wiley Series in Probability and Mathematical Statistics, Wiley, New York, 1987.
33. Seba Susan, Madasu Hanmandlu, "A non-extensive entropy feature and its application to texture classification" *Neurocomputing* 120(2013)214–225
34. Mahajan P.M., Kolhe S.R. and Patil P.M., "A review of automatic fabric defect detection techniques", *Advances in Computational Research*, ISSN: 0975–3273, Volume 1, Issue 2, 2009, pp-18-29.
35. Dan Popescu, Radu Dobrescu, and Maximilian Nicolae, "Texture Classification and Defect Detection by Statistical Features", *International Journal Of Circuits, Systems And Signal Processing*
36. Biswajit Pathak, Debajyoti Barooah, "Texture Analysis Based On The Gray-Level Co-Occurrence
37. Shervan Fekri Ershad, "Texture Classification Approach Based on Combination of Edge & Co-occurrence and Local Binary Pattern", *Int'l Conf. IP, Comp. Vision, and Pattern Recognition | IPCV'11 |*
38. R. W. Connors and C. A. Harlow, "A Theoretical comparison of texture algorithms," *IEEE Trans. Patt. Anal. Machine Intell.*, no. 3, pp. 204-222, May 1980.
39. D. Chetverikov, Pattern regularity as a visual key, *Image Vision Computing* 18 (2000) 975–985.
40. A. L. Amet, A. Ertüzün, and A. Erçil, "Texture defect detection using subband domain co-occurrence matrices," *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 205-210, Apr. 1998.
41. A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 55-73, Jan. 1990. 41. I. Tsai, C. Lin, and J. Lin, "Applying an artificial neural network to pattern recognition in fabric defects," *Text. Res. J.*, vol. 65, pp. 123-130, Mar. 1995.
42. A. Kumar and G. Pang, "Defect detection in textured materials using Gabor filters," *Proc. 35th IEEE/IAS Annual Meeting, Rome (Italy)*, pp. 1041-1047, Oct. 2000.
43. A. Bodnarova, M. Bennamoun, and S. J. Latham, "A constrained minimisation approach to optimise Gabor filters for detecting flaws in woven textiles" *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Process., Brisbane (Aus.)*, ICASSP'00, vol. 6, pp. 3606-3609, Jun. 2000.
44. A. Bodnarova, M. Bennamoun, and S. J. Latham, "Textile flaw detection using optimal Gabor filters," *Proc. 15th Intl. Conf. Pattern Recognition, Brisbane (Aus.)*, vol. 4, pp. 799-802, Sep. 2000.
45. A. Bodnarova, M. Bennamoun, and S. J. Latham, "Optimal Gabor filters for textile flaw detection," *Pattern Recognition*, vol. 35, pp. 2973-2991, 2002.
46. J. Escofet, R. Navarro, M.S. Millan, and J. Pladelloreans, "Detection of local defects in textiles webs using Gabor filters", *Proc. SPIE 2785*, pp. 163-170, Jun. 1996.
47. J. Escofet, R. Navarro, M.S. Millan, and J. Pladelloreans, "Detection of local defects in textiles webs using Gabor filters", *Opt. Eng.*, vol. 37, pp. 2297-2307, Aug. 1998.
48. C. Beirão and M. Figueiredo, "Defect detection in textile images using Gabor filters," *Proc. ICIAR'2004, Lecture Notes in Computer Science, Springer Verlag*, vol. 3212, 2004.
49. A. Kumar and G. Pang, "Defect detection in textured materials using Gabor Filters," *IEEE Trans. Ind. Appl.*, vol. 38, no. 2, pp. 425-440, Mar. 2002.

50. A. Kumar, "Automated inspection of textured web materials using real Gabor functions," Proc. 2nd SPIE Intl. Conf. Image & Graphics, ICIG 2002, Hefei (China), pp. 59-62, Aug. 2002.
51. A. Kumar and G. Pang, "Defect detection system for quality assurance using automated visual inspection," U.S. Patent No. 6, 753, 965, Jul. 2004.
52. A. Kumar and G. Pang, "Fabric defect segmentation using multichannel blob detectors," Opt. Eng., vol. 39, no. 12, pp. 3176-3190, Dec. 2000.