DETECTION OF DEFECTS ON WARP KNITTED FABRIC SURFACES

Hungampala Ralalage Dimuthu Rangana Wijesingha

(148487X)

Degree of Master of Engineering

Department of Mechanical Engineering

University of Moratuwa

Sri Lanka

December 2018

DETECTION OF DEFECTS ON

WARP KNITTED FABRIC SURFACES

Hungampala Ralalage Dimuthu Rangana Wijesingha

(148487X)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the Master of Engineering in Manufacturing Systems Engineering

Department of Mechanical Engineering

University of Moratuwa

Sri Lanka

December 2018

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Date: ………………………   Signature: …………………………

The above candidate has carried out research for Master's thesis under my supervision.

Dr. A.G.B.P. Jayasekara
Senior Lecturer
Department of Electrical Engineering
University of Moratuwa

Date: ………………………   Signature: …………………………

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. A.G.B.P. Jayasekera for the continuous support during the research period, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of the research and writing of this thesis. I could not have imagined having a better advisor and mentor for this study.

Besides my advisor, I would like to thank Prof. R.A.R.C. Gopura for his encouragement, insightful comments, and advice.

My sincere thanks also goes to the Department of Mechanical Engineering, University Of Moratuwa for the unstinted support and guidance which paved the way for this research.

Furthermore I would like to thank Mr. A. Herath, Assistant Manager Product Development of Trischel Fabrics (Pvt) Ltd for giving his support in collecting fabric samples and capturing images of fabric samples.

Finally, without the support from my family I would not be able to achieve this goal and I am deeply indebted to all of them for their support and bearing with me during the entire course of my research study.

Dimuthu wijesingha
(dimuthuwije@gmail.com)

Abstract

This thesis is concerned with the development of a novel learning algorithm based method for detection of defects on patterned, textured surfaces of warp knitted fabric surfaces using neural networks. The acquired images were subjected to several filtering processes and morphological operations to improve the state of the image and enhance texture details.

The proposed method was developed by considering textural abnormality as a defect. Since the warp knitted fabric surface is a repetitive patterned texture the image was splitted into windows prior to analysis in order to enhance detectability of defects. Also, gray level co-occurrence matrix and local binary pattern were used as the texture models of an image window. Selected set of statistical measurements were used to extract the texture from gray level co-occurrence matrix. Since detection of defects on an image is a binary classification problem an anomaly detection scheme was proposed. This enabled the development of the detection model by learning the feature space of one particular class of problem. A self-organizing map was used to learn the texture patterns on images of the non defective fabric samples. The resultant Euclidian distance of a window from the self-organizing map was used as the measure of similarity to non-defective windows while thresholding the similarity measure by using the maximum value similarity of non-defective windows as the threshold. The proposed anomaly detection scheme enabled detection of defects on particular type of texture.

There were different surface types associated with warp knitted fabrics. Self-organizing map based clustering approach was used to discretize the detection problem according to surface texture type and the intention was to simplify the detection problem and solve it with respect to specific texture. Furthermore, the histogram of the local binary pattern was used for development of compressed self-organizing map to represent the local texture of a window of different surface types.

All the calculations, analysis tasks and development of mathematical models were performed in a matlab environment. The appropriate graphical user interfaces were also developed with the proposed method been applied on images with seven different types of defects on seven surface types. The quality percentage was calculated based on the number of false positives/false negatives of the detection results for the image windows in order to evaluate the validity of the proposed method. The method results quality percentage was in the 80% range during the detection of defects.

***Key words* :** Intelligent learning algorithms, Self-organizing neural networks, defect detection, Statistical texture analysis, Fabrics

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
ANN	Artificial neural networks
BMU	Best matching unit
BP	Back propagation algorithm
GD	Gradient descent method
GLCM	Gray level co-occurrence matrix
GUI	Graphical user interface
k-NN	*k* - nearest neighbor algorithm
KSOM	Kohonen self-organizing map
LBP	Local binary pattern
PC	Personal computer
PCA	Principle component analysis
MLP	Multi-layer perceptron
USB	Universal Serial Bus
W.R.T.	With respect to

CHAPTER 1 INTRODUCTION

Automated visual inspection has applications over a wide variety of industrial surfaces such as textile, wood, steel, ceramics, paper, etc. In the Sri Lankan context, textile surface inspection using automated visual inspection techniques is at a high demand in the textile and apparel industry. Currently, the inspection process of textile surfaces for quality is being performed manually, however due to boredom and inattentiveness of the labor force, the inspection process is not consistent. Thus, the reliability of the manual inspection process is limited while automated surface inspection has become a competency factor for the increase of competitiveness of the end product.

Warp knitting represents the fastest method of producing fabric and have many applications in the Sri Lankan textile industry while an unlimited number of fabric structures could be produced using this method. This research emphasizes defect detection on warp knitted fabric surfaces during automated surface inspection. A defect detection process consists of two independent stages- (i) image analysis which deals with texture representation and extraction and (ii) classification consists of pattern representation, pattern recognition, cluster analysis and decision making. After representing the texture with mathematical parameters the defect detection problem can be considered as a binary classification problem in the domain of mathematical parameters. Warp knitted fabric family consists of a wide variety of fabric structures. On the other hand wide varieties of defect types are associated with warp knitted fabrics. So the classification model should be able to adapt according to new scenarios. Since adaptability closely correlates with the learning capability of a classifier more focus on classifiers with learning capability were explored as can be illustrated later in the literature review. There are two main types of classifiers- (i) supervised classifiers and (ii) un-supervised classifiers. However, one important necessity for supervised classifier is having sets of training samples over the entire visual space for each class of the problem being considered. However, in this scenario it can be quite problematic to have a training sample set of defective class which spans over the entire visual space. Also there can be instances of a defect where nature of the defect may not be pre determinable. On the other hand, good quality samples are readily available and possess a narrow visual space. Since the constraint regarding classification has been faced before, special attention has been focused for the classification stage which is illustrated in the literature review.

This research deals with development of defect detection scheme for detection of defects on patterned, textured surfaces of warp knitted fabric surfaces. This chapter provides overview on various classification techniques available for surface defect detection. Related work in the field of visual surface inspection is provided in section 1.1 where definitions provided by Bishop [32] and Kohonen [31] are used. The overview of this research is presented in section 1.2 followed by thesis organization in section 1.3.

1.1 Related work for visual surface inspection

The visual inspection problem of industrial surfaces could be considered as a texture analysis and classification problem. Texture analysis includes finding the most important visual properties or features of an object that makes it distinguishable from others. Among the texture description methods statistical and signal-processing based approaches have been very popular [15]. However signal-processing based methods are not of interest in this research. Thus, statistical approaches have been reviewed in the literature review section with special attention to the texture classification problem.

1.1.1 *k*- nearest neighbor classifier

k-nearest neighbor algorithm (*k*-NN) is a method for classifying vectors based on classes of closest training vectors in the feature space. Training process of the classifier consists of storing feature vectors with class labels of the training vectors, and the classification of unlabelled vector based on the class labels of its k nearest neighbors by *majority voting*. A distance measure is used to measure the closeness of vectors. The most common distance measure is the Euclidean distance measure given by,

$$d(x, u) = \sqrt{\sum_{i=1}^{n}(x_i - u_i)^2} \tag{1.1}$$

Where x and u are n-dimensional vectors. Readdy *et al.* [1] has presented the fabric defect detection model using *k*-NN as the classification algorithm. They have used gray level co-occurrence matrix (GLCM) and local binary pattern (LBP) as textural models. Since GLCM capture the global texture, LBP captures local texture around pixel this combination of GLCM and LBP with *k*-NN classifier has been applied on various other industrial surface inspection as well [4]. Amet *et al.* [2] uses Mahalanobis distance based *k*-NN classifier for defect detection problem of fabric surfaces. They have decomposed the image using wavelet transform to transform the image into a lower dimensional form. The feature vector was generated from image windows of a partitioned image while GLCM statistics have been used as texture features. Kyllonen *et al*. [3] introduced a combined distance measure for improved wood inspection system. They have extracted the color information using color percentiles and represent using Euclidean distance in a vector space. Texture details were calculated as gray level differences and LBP and have used a log-likelihood distance measure on them. Also, the sum of scaled distances was used as input for the *k*-NN classifier.

1.1.2. Multi-layer perceptron

Multi-layer perceptron (MLP) is the most commonly used form of artificial neural networks (ANN) which is a supervised type feed-forward ANN. A node in MLP acts as mathematical neuron and computes the weighted sum of the input vector at the presence of the bias, and passes this sum through the activation function. This can be expressed as,

$$y_j = f_j(\sum_{i=1}^{n} w_{ji}x_i + \theta_j) \tag{1.2}$$

Where x_i is the i th element of n dimensional input vector. θ_j , f_j are bias and activation function for neuron j respectively. $w_{ji}$is the synaptic weight between j th neuron and input node and y_j is the output of j th neuron. These weighted summations at each node sequentially propagate forward through hidden layers to the output layer where the error calculation takes place. The error E is calculated as a mean square error between the actual output y_j and the desired output d_j,

$$E = \frac{1}{2} \sum_{j=1}^{n_j} \left(d_j - y_j\right)^2 \tag{1.3}$$

Back propagation (BP) algorithm is a general purpose training method for a MLP. The BP algorithm is based on the gradient descent (GD) method in which the weights are updated in such a direction so as to reduce the error. The weight update rule of GD given by,

$$\Delta w_i = -\mu \frac{\partial E}{\partial w_i} \tag{1.4}$$

$$\Delta w_{ji} = \mu\, \delta_j y_i \tag{1.5}$$

Where μ the learning is rate parameter of BP algorithm and δ_j is the local gradient of j th neuron. The error propagating backwards from output layer to input layer and local gradients calculation at each node take place. The BP algorithm calculates the amount of weight adjustment and applies over ANN as a backward pass. This forward propagation and back ward propagation iterates and weight adjustment take place until the stop criterion is met.

Kumar [5] suggest that a fabric defect on an image alters the local neighborhood of the image. He has presented a new approach for segmentation of fabric defects where texture feature extraction was done by applying a macro-window over the neighborhood of each pixel in the image and applied them on the MLP after reducing the dimension using the principle component analysis (PCA). In contrast, Stojanovic *et al*. [6] have proposed a defect classification system by combining MLP with a fuzzy logic grading scheme. They have used the gray level difference method for texture feature generation combined with geometric features generated from the binary processing of images in order to have a more detailed feature vector. Thus, they have concluded that a simple fuzzy logic scheme was sufficient for classification of holes, spots with geometric features. However, for more complicated defects they have applied combined feature vector onto the MLP. Furthermore, Chung-Feng *et al.* [7] have used a MLP to solve fabric defect classification problem. Maximum width, height and the gray level value of the segmented defects were used as input to the MLP. In some literature researchers have used generalization capabilities of MLP. Choi *et al*. [8] have incorporated MLP in to the fabric defect detection process for the purpose of membership function generation for their fuzzy logic based decision process. They have applied normalized area, mean chord of defect on to a MLP alongside with human expert decisions. Membership functions were generated from the generalized and trained MLP. Chang *et al*. [9] used a combined MLP and fuzzy logic approach to solve the complicated uncertainty associated with the cork quality classification problem. They trained the MLP with membership values generated by a fuzzy system instead of crisp values. Mursalin *et al.* [18] have used MLP for defect classification purpose which utilized area, number of objects and the shape factor of the thresholded image as inputs to the MLP. On the other hand, the outputs of the MLP represented holes, fades, scratches and no-fault conditions. Banumathi *et al.* [19] has proposed a defect detection scheme using a segmentation approach for woven fabric defects. They have used statistical measures and the fractals as inputs to the MLP.

1.1.3. Kohonen self-organizing maps

Kohonen self-organizing map (KSOM) maps input vector x_i from continuous higher dimensional space on to a two-dimensional discrete grid of nodes. There are two learning processes associated with KSOM which includes the competitive learning process and the associative learning process. During the competitive learning process all nodes are in competition for the input where x_i compete with nodes in the grid and finds the best matching unit (BMU) in such a manner that it reduces the Euclidean distance between x_i and neuron(w_j)

$$c = \arg\min_j \|x_i - w_j\| \tag{1.5}$$

Here c is the winner or BMU where the corporative learning scheme is there to adjust the weights of neighborhood neurons of winning c th neuron. The neighborhood of c is defined using the symmetric function as a function of distance from winning neuron c. The most common is the Gaussian neighborhood which could be expressed as,

$$h_{jc} = \exp\left(-\frac{d_{jc}^2}{2\sigma^2}\right) \tag{1.6}$$

Here h_{jc} is the neighborhood. d_{jc} is the distance from c and σ is the width of the neighborhood. However the neighborhood should be a monotonically decaying with iteration number (n). The decaying neighborhood could be defined as,

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \tag{1.7}$$

Here σ_0 is the initial width of neighborhood. So the decaying neighborhood can express using the neighborhood function which can be stated as,

$$h_{jc}(n) = \exp\left(-\frac{d_{jc}^2}{2\sigma(n)^2}\right) \tag{1.8}$$

The training process of a KSOM consist of two phases- (i) *coarser training* in which the neighborhood decays to a certain width and weight adaptation takes place (ii) *fine* *training* in which neighborhood is kept constant while weight adaptation takes place. Weight adaptation is takes place in accordance with modified Hebbian learning rule. The modification term $\mu h_{jc} w_j$ is added to prevent over learning of this same pattern and it could be considered as aforgetting term. The modified Hebbian learning rule could be stated as

$$\Delta w_j = \mu(n) h_{jc} (x_i - w_j) \quad (1.9)$$

In this situation,$\mu(n)$ is the decaying learning rate function of Hebbian learning algorithm and Δw_j is the amount of weight adjustment takes place at neuron j. KSOM represents data,set by set of local models or clusters on a two dimensional space. The most common applications of KSOM are solving clustering problems, dimensionality reduction problems and data representation problems. Interestingly, detection of defects of wood surface is one of the extensively addressed problems.

Niskanen *et al*. [10] [11] tried to solve the defect detection problem of wood surfaces by a clustering approach using KSOM. They have introduced the non segmentation approach for detection of defects as well. They have used color percentiles to extract the details of the color histogram and LBP to represents the local texture. They have mapped the features on to a KSOM and set a boundary on the KSOM. It is also interesting to note that Iivarinen *et al.* [14] have used KSOM to solve the classification problem of paper defects. They have extracted shape descriptors, histogram descriptors and GLCM statistics from the segmented defect and these features were applied on to separate KSOMs in order to achieve separate classification results. Consequently, the classification results were combined to reach a final decision. Furthermore, Vapola *et al.* [17] develops a fault detection and identification system for an anesthesia system using KSOM. They have mapped the anesthesia system space onto a KSOM. The purpose was to reduce the system to finite number of KSOM nodes. The state of the unknown condition was identified as the state of BMU. Also, fault conditions were identified by the Euclidean distance to BMUs. Tian *et al*. [12] used a similar approach for anomaly detection of the cooling fan. However, they have considered three BMUs using k-NN algorithm in their approach. Martins et al. [20] has addressed the problem of automated visual inspection of surface defects on rolled steel. In this context, image analysis was performed using Hough Transform in-order to detect lines and spots while PCA has been used to parameterize defects of 32×32 image windows.

In general, researchers have solved the defect detection problem in image textural feature space using either image segmentation approach or image partitioning approach. In fact, GLCM and LBP were the most popular textural models in literature. Since a defect on fabric structure does not cause noticeable variations on gray scale image histogram or first order statistics, they were not considered popular as features in image analysis. Some researchers have proceeded with dimensionality reduction using wavelet theory or the PCA while some have solved the classification problem with supervised classification approaches such as MLP, *k*-NN while others solved it with a clustering approach such as KSOM. In addition, some researchers have solved the binary classification problems of feature space with an abnormality detection approach when faced with constraints to use conventional classification methods.

There were certain constraints on the problem being considered in this context, as it is not possible to percept the defective visual space of a fabric type. A fabric type may have wide verity of defects and there is a difficulty in achieving this variety with practically available defective samples. This may results in incorrect classification model. With its present defects an un-supervised KSOM was considered the preferred option for model the feature space. While reviewing the literature, it clearly showed that binary classification problems which associate difficulties in defining feature space of one particular class have been solved based on abnormality detection schemes. Therefore, the classification approach proposed here defines vector space of good quality samples with a KSOM and proceeds on defect detection with abnormality detection approach. It is important to note that, defect detection problem of images have been dealt by splitting the image in to windows and presenting windows into the classifier. This localization approach magnifies the abnormality presence within the image as well.

On the other hand, several issues must be dealt with when solving the defect detection problem. Some of these issues have minimized variations due to imaging conditions by selecting the most suitable texture features for the representation of the image. This thesis signifies an approach to propose a classification scheme for the purpose of defect detection in different warp knitted fabric surfaces. The validity of the scheme will be proven with images of fabric samples gathered from actual production environments with respect to different fabric surface types. The learning capability of the scheme ensures incorporation of new fabric surface types in to the proposed model. Also, this capability introduces several training phases with respect to different types of fabric surface types and several parameters which require optimization.

1.2 Overview of the Thesis

This research reiterates the defect detection problem of the warp knitted fabric surface while defect detection in different surface textures has been considered too. This can be considered as a set of binary classification problems, however to solve this problem a fixed positioned personal computer camera and lighting arrangement has been employed. There are many textures associated with warp knitted fabrics. On the other hand, there could be a possibility of developing new fabric structures as well. Owing to this, visual space of warp knitted fabric surfaces could be considered as a partially defined one. When considering the textures of the particular type of defect there is a possibility of occurrence with difference in texture. Thus, there is a considerable amount of uncertainty associated with the visual space of fabric defects. This makes the situation deemed by two characteristics: (1) Partially defined feature space (2) uncertainty in feature space.

Since the detection problem has the above mentioned constraints our proposed solution should possess capability in: (1) Dealing with expansion in visual space of non defective fabrics (2) Dealing with uncertainty associated with visual space of defective fabrics. One possible way of allowing the adaptability is by incorporating learning capability in to the classifier. Hence, KSOM for classification task could be proposed in this situation. Secondly, in order to deal with the uncertainty of the visual space of defects, abnormality detection scheme where the classification results do not rely on the visual space of defects have been proposed in this context. Major achievements of this research project are:

1. *Clustering based method to identify surface type of warp knitted fabric:* A KSOM clustering approach with LBP features has been used for this purpose. The objective was to reduce problem space for detection in order to simplify the detection task.

2. *KSOM based abnormality detection method to detect defects in various types of fabric structures:* Based on the clustering decision regarding the fabric surface, a selective set of GLCM statistics of windows were guided to the respective KSOM which is the generic texture model of the non defective windows of the identified fabric type. Based on the similarity of the texture of windows with its best matching local model the abnormality detection scheme has been developed. The defect detection method was validated by experimenting with various types of defects.

3. *Un-supervised training scheme:* The proposed method solves the above mentioned problem with an abnormality detection approach. Defective image windows have being identified based on similarity to the local model and by thresholding from the non-defective space which was developed during the training of the scheme. In fact, the scheme demands only good quality samples for the development of classification model.

4. *Graphical user interface for training and execution of the method:* A graphical user interface (GUI) has been developed for selective execution of images, displaying the feature extractions and training states of the KSOM. The GUI displays the relevant image analysis parameters, KSOM training parameters during the respective stages of the process.

This research project gives rise to solving the defect detection problem in various surface textures and introduces following novelties,

- KSOM clustering based method to identify surface texture type
- Abnormity detection scheme to detect defects on textured patterned surfaces
- Detection scheme which can be trained by only using non-defective samples.

1.3 Organization of the Thesis

Rest of the thesis is organized as follows,

In chapter 2, the background information on problem being considered is provided while discussing nature of the problem and strategies to overcome any constraint of the problem. Finally, a defect detection scheme for the problem has been proposed.

Chapter 2 provides pre-processing methods to convert the image to an enhanced state and thereby ensuring meaningful feature extraction. Several image enhancement techniques as well as texture enhancing methods are suggested alongside sample results.

Chapter 3 deals with texture extraction from the images and thereby providing mathematical measures to represent the image at classification stages. Several texture models alongside statistical measurements to extract the texture have been suggested. Also, an experiment is introduced to identify the validity of particular statistic.

Chapter 4 provides a method to identify the fabric surface type where KSOM is used as un-supervised classifier thereby discretizing and simplify the defect deection problem. Apart from this several modifications are suggested to improve classification accuracy.

In chapter 5, abnormality detection scheme based on similarity measure with best matching local model of the KSOM which is representative of generic model of global texture of non-defective sub-images is proposed for defect detection. A graphical user interface is presented in this part too.

Chapter 6 summarizes the thesis by providing conclusions and directions.

CHAPTER 2 OVERVIEW OF THE METHODOLOGY

This chapter provides an overview of the problem along side with overview of initial development stages of the solution and its practical implementation in the context of the problem. A brief overview on warp knitted surface types in section 2.1 followed by types of defects associated with warp knitted surfaces in section 2.2. The problem definition, objectives of this research explained in section 2.3 alongside with strategies to overcome practical constraints. Finally, an overview of proposed solution is presented alongside with strategy for practical implementation of section 2.4.

2.1 Warp knitted fabrics: An overview

Figure 2.1: Example surface types associated with warp knitted fabrics

Warp knitting is a family of knitting methods which enables the construction of variety of fabric structures while keeping a low stress on yarns. Trischel Fabrics (PVT) Ltd, Thulhiriya is the only warp knitted fabric manufacturer in south Asia which provides fabric solution for intimates and active-wear. Fabric samples for this research were collected from manufacturing environment of Trischel Fabrics (PVT) Ltd under a special permission. Warp knitting methods can be grouped in to two categories. They are (i) *Tricot knit* which enables soft drape fabric production with length wise stretch which is extensively used in manufacturing intimate wear and especially in lingerie manufacturing. (ii) *Raschel knit* enable production of bulky and less stretchable fabrics which are often used as unlined material for jackets. Surface images of different types of fabrics gathered from the manufacturing environment are shown in Figure 2.1.

2.2 Defect types on warp knitted fabrics

Figure 2.2: Example defect types associated with warp knitted fabrics

A fabric defect is an abnormality that hinders its acceptability by the consumer. Defects can be due to defective yarn from the material supplier or processing defects at manufacturing stages. Figure 2.2 shows images of defective warp knitted fabric samples. There are more than seventy types of defects associated with warp knitted fabric manufacturing. However, most of them do not frequently occur. Defects such as pin-hole, snag and stop-line are due to machine faults during the knitting process. A pin-hole is caused by a broken needle, and a stop-line could be caused by a broken needle where loop formation does not take place with respect to the yarn with the broken needle. A Snag is another instance of knitting machine related defects. Figure 2.3 shows common types of defects associated with faults of the knitting machine.

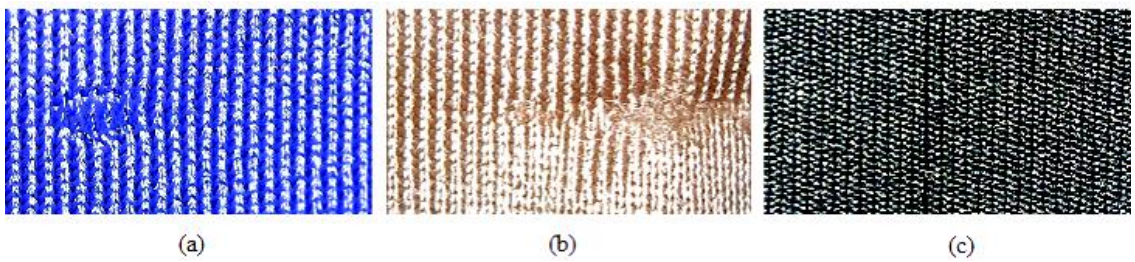


Figure 2.3: Knitting machine faults - (a) pin-hole (b) snagging (c) stop-line

There are some defects due to material faults too. If the yarn package contains thick or thin yarns then the fabric has the same fault as the yarn. End-out is a common defect which is caused by the knitting machine running continuously with a missing end on warp knits. These defects are shown in figure 2.4.

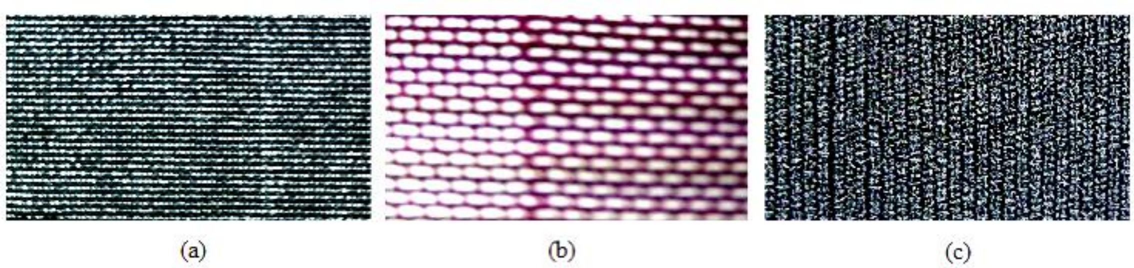


Figure 2.4: Defects associated with yarn faults- (a) thin-yarn (b) thick-yarn (c) end-out

There are some defects which occur due to the contamination problems during dyeing and finishing processes. These contaminants could be either dirt particles or chemical particles where the defects caused by them are dirt marks and chemical marks. Crease marks are another instance of defect caused by variation in the finishing process. These defects are shown figure 2.5.

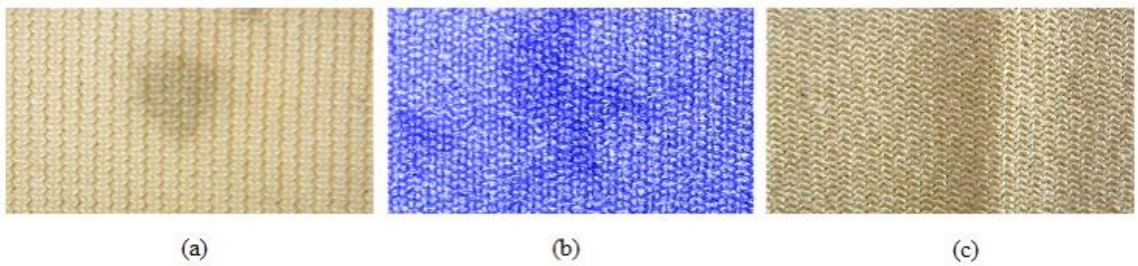


Figure 2.5: Defects associated with chemical processes- (a)dirt (b)chemical-mark (c)crease-mark

2.3 The problem definition and objectives

Defect detection on surface could be treated as a binary classification problem. One class is the defective class and other class is the non-defective class. Since multiple types of fabric surface types and multiple types of defects are being considered in this context, it could be classified as a set of binary classification problems. Each warp knitted fabric surface type has a unique texture. However, a minor variation in texture could take place between samples as well as within the sample. Since the development of a new fabric structure is an introduction of a new texture the current visual space of a non-defective class is incomplete in nature. On the other hand, each defect type consists of texture which is uncertain in nature. The texture of a defect may vary greatly between samples. However, there is difficulty in having defective samples to represent a majority portion of visual space of a particular defect. Since it cannot pre determine the nature of the next instance of particular defect, the visual space of a defective class is associated with uncertainty.

The problem is being considered in this thesis deals with several issues and strategies are suggested to overcome those which can be mentioned as follows,

- The input space for this problem is partially defined and expansion of input visual space could take place with the introduction of new fabric structures. In this thesis, a learning based approach is suggested to enable the expansion of the input space to suit new occurrences of fabric types.
- The task of defining the visual space of defects for a particular fabric type could be a difficult task. This is due to the difficulties in finding defective samples to represent the entire visual space of defects. Thus, abnormality detection based on a similarity to KSOM which is the generic global texture model of non-defective image windows has been proposed to overcome this issue.
- The defect detection in a texture surface is a much complicated task when compared with plain surfaces. Since the abnormalities are more prominent at localized window level images are splitted into windows during pre-processing.
- Due to the constraints associated with the problem, the defect detection model should be based on the texture feature space of non-defective image windows. Hence, KSOM is used to model the feature space of non-defective image windows.
- There is uncertainty associated with image textures which results in reducing the suitability of similarity measure which is the Euclidean distance between BMU and texture of image window. Therefore, the k-NN based BMU calculation strategy has been proposed to overcome the issue.
- The Final objective of this thesis is to develop a GUI to enable the interaction during experimental stage with the proposed system while discussing limitations and issues.

2.4 The proposed solution and implementation

A classifier with learning capability seemed to be much suitable due to the nature of this particular classification problem. However, due to the limitations in having defective samples the solution should also be an un-supervised classification approach based anomaly detection. Eventually, once the classification criterion has been learnt, the scheme would be able detect whether an image region is defective or non-defective.

The schematic diagram of the proposed solution is shown in figure 2.6. The texture features extracted from the captured image would be considered as the inputs and defect/non-defect classification as the output. It consists of a personal computer camera, a lighting system with an image processing unit and set of classifiers which are executed using a computer. The surface type identification results and defect/non-defect classification results would be visualized by the computer monitor. The Image processing unit is used for three main purposes. They are: - (i) Pre-processing of images (ii) Filtering and image enhancement operation (iii) Texture feature extraction.

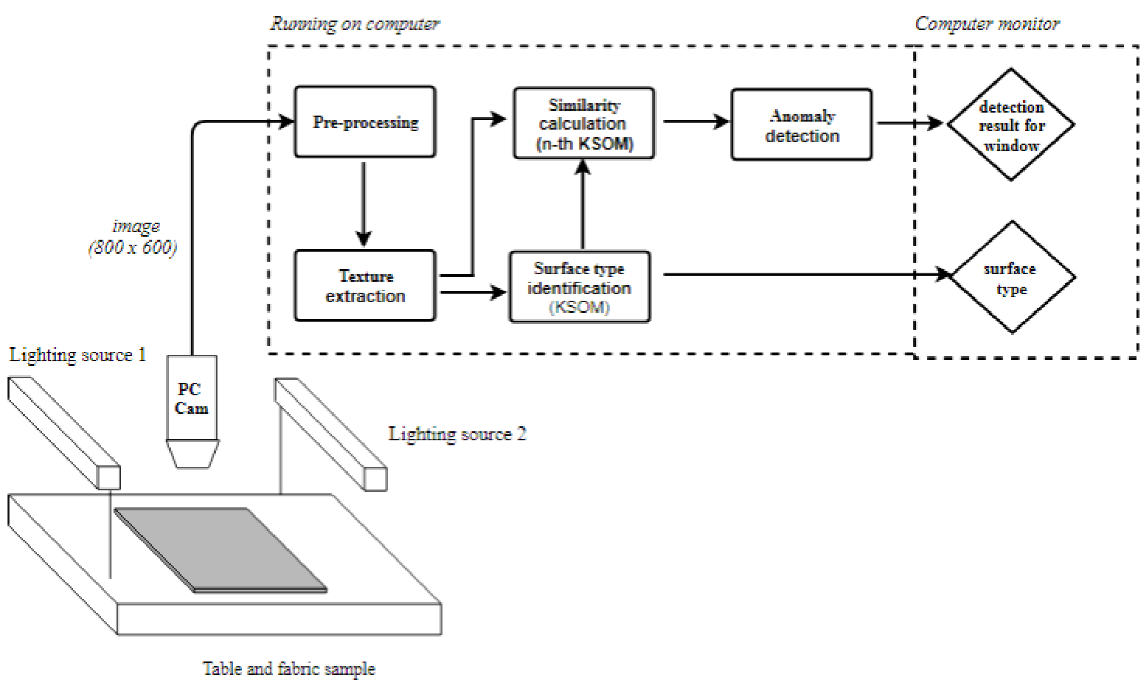


Figure 2.6: Schematic for implementation of the proposed solution

The actual setup for the experiment is shown in figure 2.7. A USB 2.0 PC camera was used to capture the image of the fabric panel in the portable network graphic format with a dimension of 1280 x 720 pixels. Also, all image analysis tasks were performed in MATLAB environment. The classification algorithms were implemented using MATLAB environment as well. The SOMPAK package has been used for the construction and training purposes of the KSOM while GUIs for training of KSOM, executing a input image and feature extraction from images have been designed using the MATLAB in order to facilitate the interaction during experimental stage.

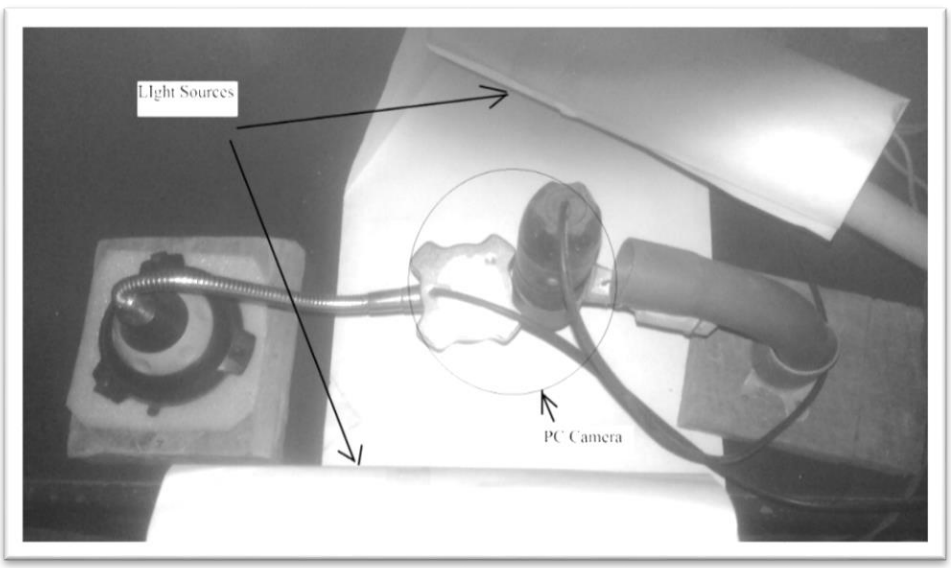


Figure 2.7: Set-up for capturing images of fabric samples

CHAPTER 3 PRE PROCESSES FOR IMAGE ENHANCEMENT

Constant and homogeneous lighting conditions are essential components of a visual inspection system. However, keeping such conditions is a challenging task. This results in acquired images which lack brightness, contrast and illumination. On the other hand acquired images consist of noise generated by the camera. Consequently, there was a requirement of converting the image to an improved state which was better suited for analysis. Image enhancement is an essential component of image pre-processing. The aim of image enhancement is an improvement of the image that suppresses unwilling distortions or enhances image features important for further processing. Examples include contrast and edge enhancement, noise filtering, sharpening and texture enhancement. The enhancement process itself does not increase the inherent information content in the data. It emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application dependent. The Most prominent image enhancement techniques are- (i) Contrast stretching (ii) Noise filtering (iii) Histogram modifications.

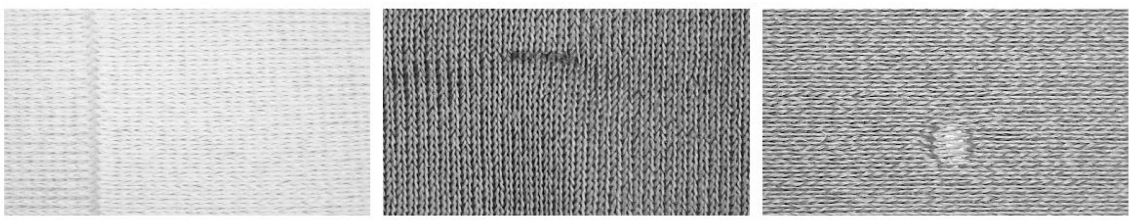
The focus of this chapter is to propose an image enhancement strategy in order to enable detailed texture extraction in later stages. While doing so correcting for contrast and illumination should be considered as a very important task. On the other hand sharpening and improvement of texture while minimizing noise should also be take into consideration. The proposed image enhancement strategies in this chapter are based on definitions provided by Gonzalez *et al.* [29].

Images of fabric samples which consist of three different defect types were used in order to demonstrate the effect of each pre-processing stage on the image. The defect types on these images are, (1) end-out (2) snag (3) pin-hole as shown in figure 3.1.

(a) (b) (c)

Figure 3.1: Images of defective fabric samples with (a) end-out (b) Snag (c) pin-hole

A color image consists of three channels: - Red, Green and Blue. During the experiment it has been observed that more important details regarding the texture are present on the green channel while the red channel consists most of the noise details. Thus, the green channel has been selected to represent the image (Figure 3.2).



(a) (b) (c)

Figure 3.2: Green channel of images of defective samples

A lighting system cannot assure a constant and homogeneous lighting over the fabric panel due to un-controllable environmental parameters and limitations in the inspection set-up. Hence, there was a requirement for correcting illumination and contrast of the captured images.

3.1 Gama transformation

In gamma transformation each input intensity value of the image has being raised to power of gamma(γ). The map between input s and output γ is given by,

$$s = r^{\gamma} \tag{3.1}$$

The images acquired from the camera usually represents a histogram, shifted towards brighter pixels and represents a narrow range of gray-levels which results in eliminating important texture and intensity details of the image. Thus, in order to perform contrast stretching on acquired images, gamma transformation with $\gamma = 5$ has been applied and example results are shown in figure 3.3.

(a) (b) (c)

Figure 3.3: Gama transformed images with (a) end-out (b) snag (c) pin-hole

3.2 Homomorphic filter

Intensity of a particular pixel in a captured image is a function of both illumination and reflectance where illumination is a property of lighting conditions and reflectance is a property of object. Homomorphic filtering is a method applied on logarithmic domain to overcome the multiplicative noise model of illumination and reflectance.

$$\ln\big(I(x,y)\big) = \ln\big(L(x,y)\big) + \ln\big(R(x,y)\big) \tag{3.2}$$

In this context I represent pixel intensity while L and R represent illumination and reflectance components of intensity. Also, (x, y) are the coordinates of the pixel. A homomorphic filter based on1^{st} order high-pass filter has been applied and example results are shown in figure 3.4.

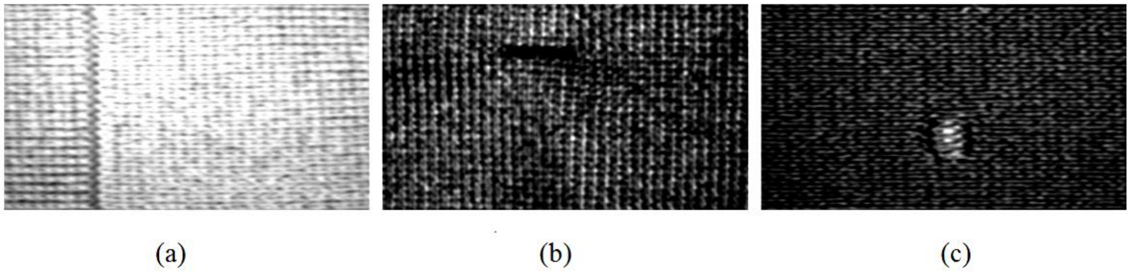


Figure 3.4: Corrected images for illumination with (a) end-out (b) snag (c) pin-hole

The corrected image incorporated noise and there was a requirement for enhancing texture within the image. A high pass filtering has been applied on the image in order to sharpen the texture and a morphological gradient operator has been applied for the enrichment of the texture.

3.2.1 High-pass filter

A high pass filter tends to retain the high frequency information while reducing the low frequency information. High-Pass filtering is generally used for sharpening purposes. The Butter worth high-pass filter is a method of filtering operaes in a frequency domain. The transfer function for n^{th}order Butterworth high-pass filter with cutoff frequency of D_0 can be defined as,

$$\mathrm{H}(\mathrm{x},\mathrm{y}) = \frac{1}{1 + [\mathrm{D}_0/\sqrt{\mathrm{x}^2 + \mathrm{y}^2}]^{2\mathrm{n}}} \tag{3.3}$$

In this context $(\mathrm{x},\mathrm{y})$are the pixel coordinates. Furthermore, 2^{nd}order filter with cut-off 40 has been applied and the results are shown in figure 3.5.

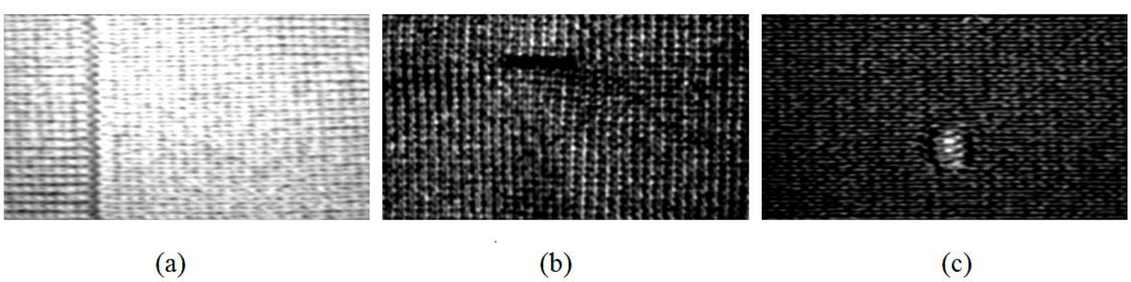


Figure 3.5: High-pass filtered images with (a) end-out (b) snag (c) pin-hole

3.2.2 Morphological operations

In grey-scale morphology an image is represented by a scalar function $f(x, y)$ with (x, y) as the pixel coordinates. Structuring element is a set B that determines the neighborhood relation of pixels as basic operations in mathematical morphology operate on these two sets. Grayscale $dilation \oplus$ replaces the grey value of the image $f(x, y)$ by its supremum within a mask defined by B as shown below:

$$(f \oplus B)(x, y) := \sup \{ f(x - x', y - y') \,|\, (x', y') \epsilon B \tag{3.4}$$

Greyscale $erosion \ominus$ replaces the grey-value of the image $f(x, y)$ by its infimum within a mask defined by B as shown below:

$$(f \ominus B)(x, y) := \inf \{ f(x + x', y + y') \,|\, (x', y') \epsilon B \tag{3.5}$$

These operations form the basis of many other processes in mathematical morphology. $opening \circ$ And $closing \cdot$ operations could be defined respectively as below:

$$f \circ B = (f \oplus B) \ominus B \tag{3.6}$$

$$f \circ B = (f \ominus B) \oplus B \tag{3.7}$$

Furthermore $tophat(h_t)$, $bottomhat(h_b)$ and $Gradient(G)$ operations could be defined respectively as below:

$$h_t = f - (f \circ B) \tag{3.8}$$

$$h_b = (f \cdot B) - f \tag{3.9}$$

$$G = (f + h_t) - h_b \tag{3.10}$$

In this approach gradient morphological operator with a *disk* structuring element has been applied on the filtered images in order to strengthen the texture details. Some results are shown in figure 3.6.

(a) (b) (c)

Figure 3.6: Morphologically processed images

3.2.3. Histogram equalization

Histogram equalization is a method of enhancing the contrast of the image by modifying and redistributing image histogram. The process transforms the probability distribution of intensities to a uniform distribution. The normal histogram equalization process applies the same transformation over the entire image. In contrast local histogram equalization calculates histograms for each local region and performs the equalization process separately (Figure 3.7).

(a) (b) (c)

Figure 3.7: Histogram equalized images

Since an abnormality is more prominent in local stage, the acquired images have been splitted into sub-images. The choice of the window size depends on two factors - (1) The degree of abnormality results in the window due to defect over the non-defective window and (2) The degree of representativeness of the repetitive texture unit of the surface by the window. In addition, the size of the sub-window created on the image, should be greater than the size of the repetitive unit of fabric texture. In this context, the size of the local window has been chosen to be 100 × 150.

CHAPTER 4 EXTRACTION OF TEXTURE FEATURES

Texture is defined in literature as *"repetitive arrangement of a unit pattern over a given area"* [21]. Representation of texture by mathematical means is an important step in image processing. Primarily, statistical texture analysis methods deal with the distribution of grey levels in a texture. The most commonly used statistical texture analysis methods are gray level co-occurrence matrices, grey level run-length matrices and autocorrelation function. Gray level co-occurrence matrix has been used in this context as the global texture model of an image window and a local binary pattern as the local texture model of an image window. Also a set of statistics were used to extract the global texture from the gray level co-occurrence matrix. Haralick *et al.* [26] has introduced seventeen statistics on gray level co-occurrence matrix. During the section 4.1 the values produced by each of these statistics on image windows have been visually observed in order to decide the validity of a statistic to represent the texture of image window during classification stages. In fact, a histogram was used to represent the local binary pattern as described in section 4.2.

Since the proposed solution is a non-segmentation approach the image has been partitioned into 100×150 sized windows during the pre-processing stage. Each of these windows has been considered as a sub-image. Since a defect cause a texture abnormality, three main types of texture abnormalities have been noticed in fabric surfaces which resulted from end-out, snag and pin-hole. These textural abnormalities are presented in figure 4.1.

Since an image is being segmented into sub-images there should be defective sub-images as well as non-defective sub images. The objective of this chapter is to identify statistical texture features of sub-images which distinguish defective sub-images from non-defective sub-images.

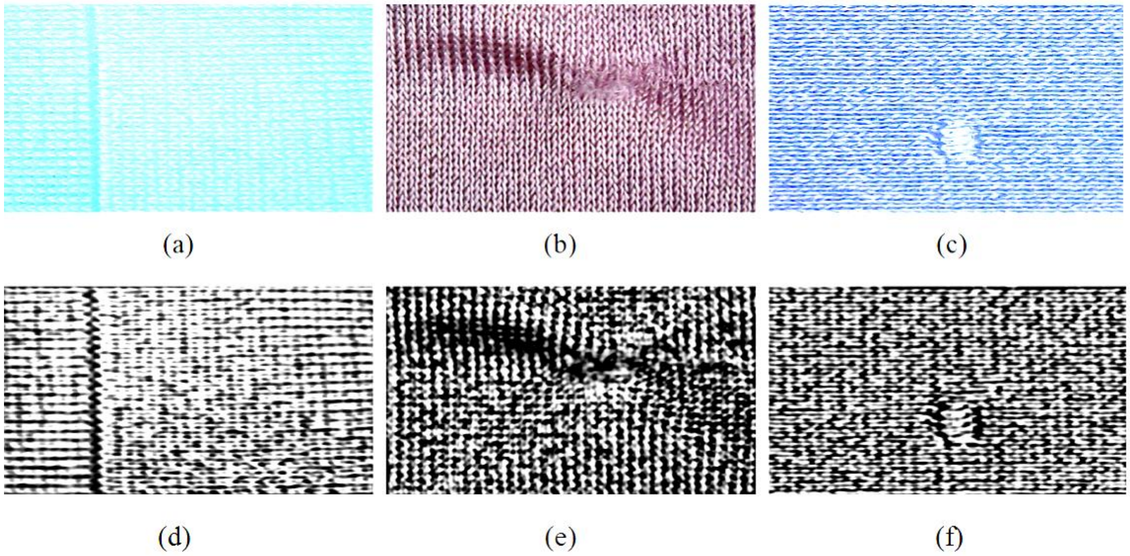


Figure 4.1: Original and pre-processed images of main types of texture abnormalities

4.1 Global texture model: Gray level co-occurrence matrix

The gray level co-occurrence matrix (GLCM) is a basis for extracting second order statistical texture features. A GLCM is a G × G matrix which approximates the joint probability distribution of pixel pairs. Here G refers to the number of gray levels in the image. However, the image is generally being quantized into G gray levels prior to calculation of GLCM. GLCM is a function of the angular relationship and distance between the neighboring pixels. A matrix element $P\ (i, j \mid d, \theta)$ contains the second order statistical probability values for changes between gray levels i and j at a particular displacement distance d and at a particular angle θ.

In fact, there is a more feasible method of defining GLCM by incorporating distance values which is using in mathematical formulation. That is the matrix element $P(i, j \mid \Delta x, \Delta y)$ is the relative frequency with which two pixels, separated by a pixel distance $(\Delta x, \Delta y)$ occur within a given neighborhood, one with intensity i and the other with intensity j .Given $M \times N$ neighborhood of an input image containing G gray levels from 0 to G-1, let $f(m, n)$ be the intensity at sample m, line n of the neighborhood. Then,

$$P(i,j|\Delta x,\Delta y) = \frac{1}{(M-\Delta x)(N-\Delta y)} \sum_{n=1}^{N-\Delta y} \sum_{m=1}^{M-\Delta x} A \tag{4.1}$$

Where,

$$A = \begin{cases} 1 & \text{if } f(m,n) = i \text{ and } f(m+\Delta x, n+\Delta y) = j \\ 0 & \text{elese where} \end{cases}$$

In this proposed approach, the GLCM is developed with respect to the 100×150 sized pre-processed sub-images processed. Since Quantization of image into sixteen gray levels is often sufficient for discrimination of textures [22], the same number level in GLCM construction was selected. GLCMs were calculated with respect to every sub-image using off-set of 0°. Since d with 1-8 range gives best results, $d = 8$ was considered here. On the other hand the ordering of pixel pairs was not important in this scenario as symmetric GLCM was considered [27].

GLCM is a mathematical structure which contains the information regarding image texture in terms of frequencies of pixel pair occurrences. The GLCM itself is a feature. However the common approach is to extract the information by means of statistical measures where certain researchers have used raw GLCM as a texture feature [23]. Haralick *et al.* [26] have suggested sixteen statistical measures which could be performed on GLCM. Even though these features contain information regarding the textural characteristics of the image it is hard to identify which specific textural characteristic is represented by each of these features. The following notations have been used to simplify the definitions of statistical measures.

G is the number of gray levels used and μ is the mean value of P. μ_x, μ_y, σ_x and σ_y are the mean and standard deviations of p_x,P_y and $p_x(i)$ is the i th entry in the marginal-probability matrix obtained by summing the rows of $P(i,j)$.

$$P_x(i) = \sum_{j=0}^{G-1} p(i,j)$$

$$P_y(i) = \sum_{i=0}^{G-1} p(i,j)$$

$$\mu_x = \sum_{i=0}^{G-1} ip_x(i)$$

$$\mu_y = \sum_{j=0}^{G-1} jp_y(j)$$

$$\sigma_x^2 = \sum_{i=0}^{G-1} (p_x(i) - \mu_x(i))^2$$

$$\sigma_y^2 = \sum_{j=0}^{G-1} (p_y(j) - \mu_y(j))^2$$

$$\text{for } k = 0,1, \dots, 2(G-1)$$

$$p_{(x+y)}(k) = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} p(i,j) \qquad i + j = k$$

$$\text{for } k = 0,1, \dots, (G-1)$$

$$p_{(x-y)}(k) = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} p(i,j) \qquad |\, i - j \,| = k$$

The following equations provide the definitions of GLCM statistics proposed by Haralick *et al.* [26]. Additionally two cluster parameters proposed by Soh *et al.* [25] were also considered.

Autocorrelation (AUTO.):

$$\mathrm{AUTO.} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ij\ p(i,j) \tag{4.2}$$

Contrast (CONT.):

$$\mathrm{CONT.} = \sum_{n=0}^{G-1} n^2 \{ \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(i,j) \} \tag{4.3}$$

Contrast is a measure of local intensity variation and will favor contributions from $P(i,j)$ away from the diagonal, $i.e.\ i \neq j$

Correlation (CORR.):

$$\mathrm{CORR.} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i\ \times j\} \times P(i,j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \tag{4.4}$$

Correlation is a measure of gray level linear dependence between the pixels at the specified positions relative to each other.

Cluster Prominence (CPRO.):

$$\mathrm{CPRO.} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^4 \times P(i,j) \tag{4.5}$$

Cluster Shade (CSHA.):

$$\mathrm{CSHA.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\left\{i + j - \mu_x - \mu_y\right\}^3 \times P(i,j) \tag{4.6}$$

Dissimilarity (DISS.):

$$\mathrm{DISS.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}|i - j|p(i,j) \tag{4.7}$$

Energy (ENE.):

$$\mathrm{ENE.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p(i,j)^2 \tag{4.8}$$

The energy measure is also popularly known as angular second moment. It is the measure of textural uniformity of pixel pair distribution. It detects the disorder in texture.

Entropy (ENR.):

$$\mathrm{ENTR.} = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) \times \log(\, P(i,j) \tag{4.9}$$

Entropy measures the disorder or complexity of an image. Complex texture tends to have high entropy also inhomogeneous scenes have low first order entropy. It is important to note that a homogeneous scene also has high entropy which is inversely correlated to energy.

Homogeneity (HOMO.):

$$\mathrm{HOMO.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{p(i,j)\}^2 \tag{4.10}$$

A homogeneous scene will contain only a few gray levels, giving a GLCM with only a few but relatively high values of$p(i,j)$. Thus, the sum of squares will be high.

Maximum probability (MAXP.):

$$\mathrm{MAXP.} = \max_{i,j} P(i,j) \tag{4.11}$$

Sum of squares: Variance (SUMSQVAR.):

$$\mathrm{SUMSQVAR.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-\mu)^2 p(i,j) \tag{4.12}$$

This feature puts relatively high weights on the elements that differ from the average value of $P(i,j)$.

Sum average (SUMAVE):

$$\mathrm{SUMAVE.} = \sum_{i=0}^{2G-2} i p_{x+y}(i) \tag{4.13}$$

It measures the average of the gray level within an image.

Sum of Variance (SUMVAR):

$$\text{SUMVAR.} = \sum_{i=0}^{2G-2} \{i + \sum_{i=0}^{2G-2} P_{x+y}(i)\log(P_{x+y}(i))\}^{2} p_{x+y}(i) \tag{4.14}$$

This is a measure of average heterogeneities in the image.

Sum Entropy (SUMENT.):

$$\text{SUMENT.} = -\sum_{i=0}^{2G-2} P_{x+y}(i)\log\left(P_{x+y}(i)\right) \tag{4.15}$$

This is a measure of average of disorders in an image.

Difference variance (DIFFVAR.):

$$\text{DIFFVAR.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-\mu)^2 P_{x-y}(k) \tag{4.16}$$

Difference entropy (DIFFENT.):

$$\text{DIFFENT.} = -\sum_{i=0}^{G-1} P_{x+y}(i)\log\left(P_{x+y}(i)\right) \tag{4.17}$$

Local Homogeneity (LHOMO.):

$$\text{LHOMO.} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{1}{1+(i-j)^2}p(i,j) \tag{4.18}$$

This is a measure of homogeneity. It is more sensitive to the presence of any near diagonal elements in the GLCM.

The purpose of performing statistical measurements on GLCM is to use the resultant values during the classification sages as representative of the texture of sub-image. However, every statistic may not be useful for classification. An experiment is used to determine the suitability of each statistic for classification. Distinguishability is the useful property for classification. The degree of distinguishability provided by each statistic between defective sub-image and non-defective sub-image is the determining factor for suitability. The objective of the experiment was to identify the GLCM statistics which results abnormal values at the defective sub-images which make defective sub-images distinguishable from non-defective sub-images. In this approach GLCM were constructed with respect to every sub-image of an image using values specified earlier. For this experiment images with three main types of texture abnormalities have been considered. Eighteen statistical measures have been calculated for every window of an image. The value results from statistical measurements from the GLCM has been developed as window with same dimensions as the sub-image and presented in a manner that it align with the sub-image while have been presented with a *'hot'* color scheme for the purpose of comparison. This experiment provides an insight in selecting the statistics to represent the texture. The observations of this experiment are presented in Table 4.1. The degree of distinguishability provides by each statistic between defective sub-image and non-defective sub-image was determined by visual examination of the results. Based on the results of the experiment selected statistics are sum of squares: variance, sum average, sum variance, sum entropy, contrast, correlation, entropy, cluster prominence, cluster shade and dissimilarity**.**

TABLE 4.1: RESULTS OF EXPERIMENT CONDUCTED TO DETERMINE THE SUITABILITY OF STATISTICS

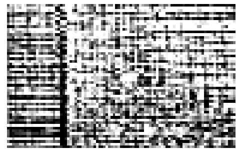
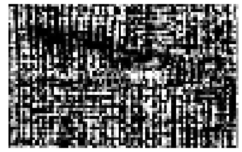
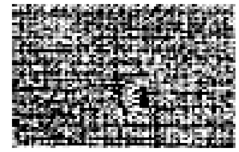
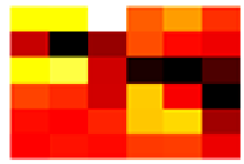
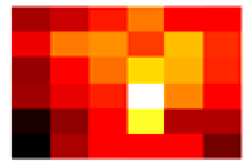
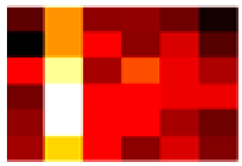
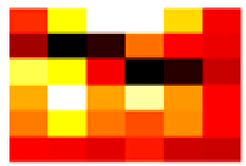
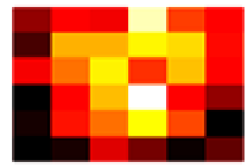
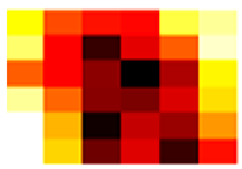
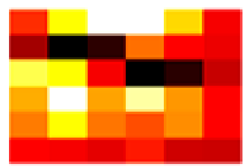
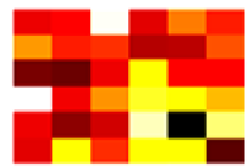
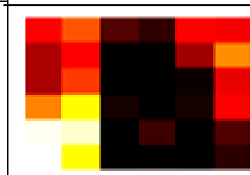
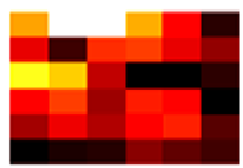
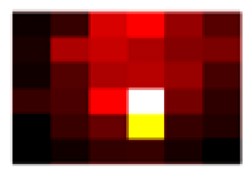
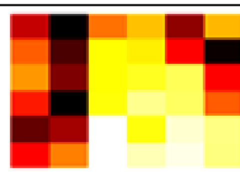
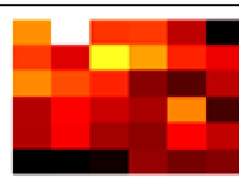
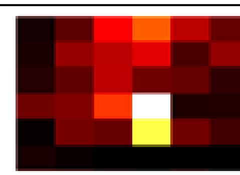
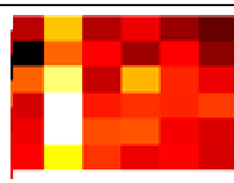
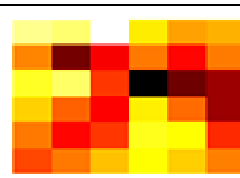
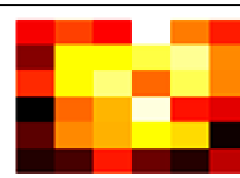
Name of the statistic			
Autocorrelation			
Contrast			
Correlation			
Cluster Prominence			
Cluster Shade			
Dissimilarity			

TABLE 4.1: RESULTS OF EXPERIMENT CONDUCTED TO DETERMINE THE SUITABILITY OF STATISTICS

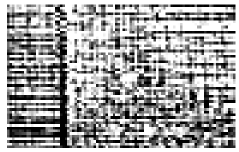
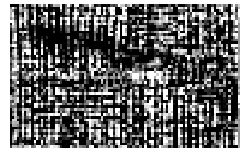
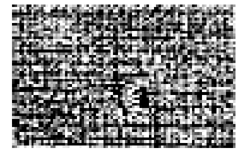
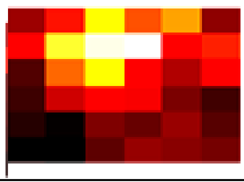
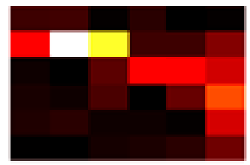
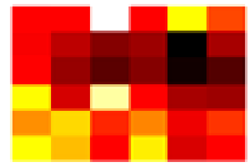
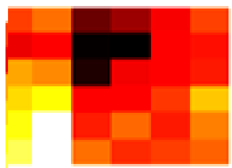
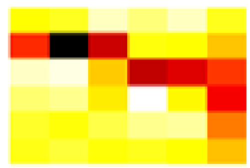
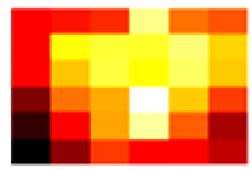
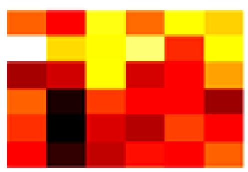
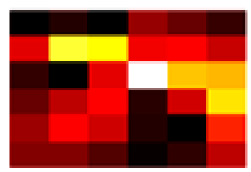
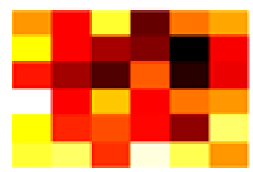
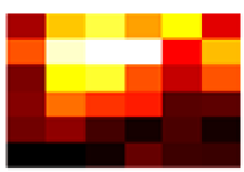
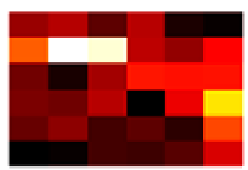
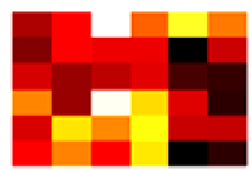
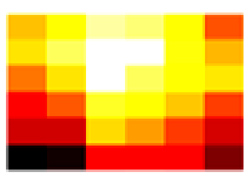
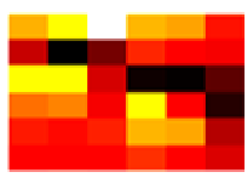
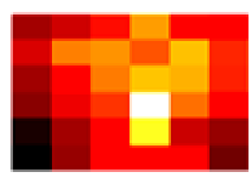
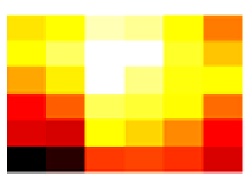
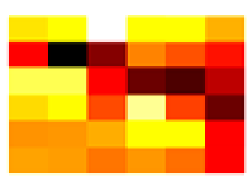
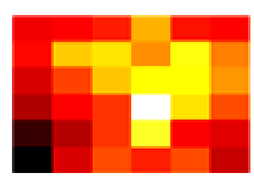
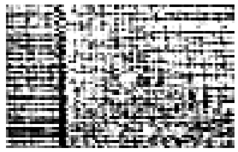
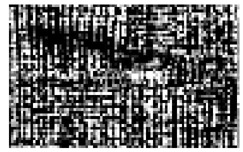
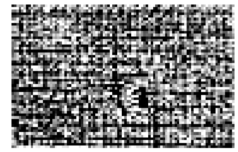
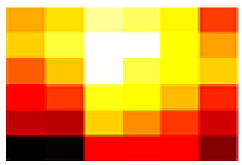
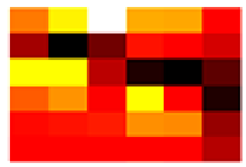
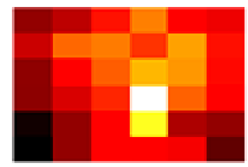
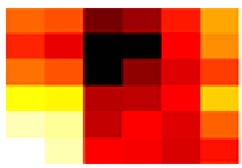
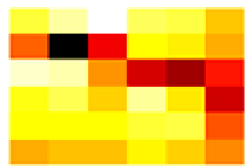
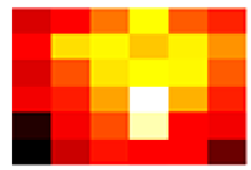
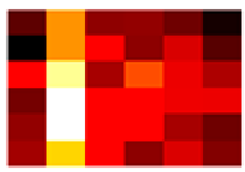
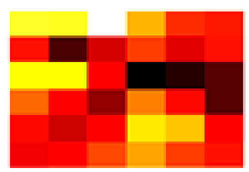
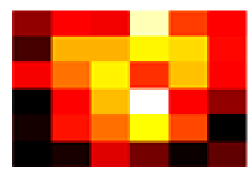
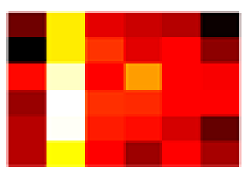
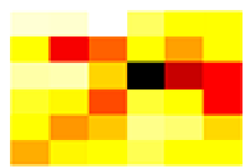
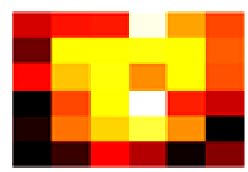
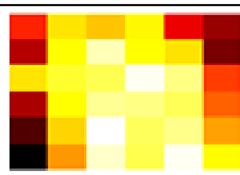
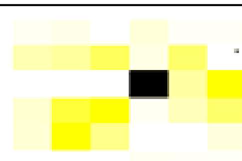
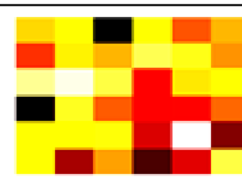
Name of the statistic			
Energy			
Entropy			
Homogeneity			
Maximum probability			
Sum of squares: Variance			
Sum average			

TABLE 4.1: RESULTS OF EXPERIMENT CONDUCTED TO DETERMINE THE SUITABILITY OF STATISTICS

Name of the statistic			
Sum of Variance			
Sum Entropy			
Difference variance			
Difference entropy			
Local Homogeneity			

.

4.2 Local texture model: Local binary pattern

The local binary pattern (LBP) is a texture primitive used for description of a local texture. In this approach the local neighborhood is being thresholded at the gray value of the center pixel into a binary pattern by the LBP operator.

Ojala *et al.* [30] has extended the definition of LBP to suit neighborhood of different sizes, by capturing features at different scales. With the introduction of circular neighborhoods the pixel values allow any radius and number of pixels in the neighborhood. In this context the image texture has been considered as a two-dimensional phenomenon characterized by two orthogonal properties, spatial structure and contrast. The texture T has been represented as a local neighborhood of a monochrome texture image as the joint distribution of the gray levels of P image pixels as,

$$T = (g_c, g_0, \dots, g_{p-1})$$

Where gray values g_c correspond to the gray value of the center pixel of the local neighborhood and g_p $(p = 0, \dots, p-1)$ correspond to the gray values of p equally spaced pixels on a circle of radius R that form a circularly symmetric neighbor set.

Joint difference distribution is a highly discriminative texture operator. It records the occurrences of various patterns in the neighborhood of each pixel in a P-dimensional histogram. After introducing gray level invariance by means of signed difference, the joint difference distribution could be expressed as follows,

$$T \approx t(s(g_0 - g_c), s(g_1 - g_c), \dots, s(g_{p-1} - g_c))$$

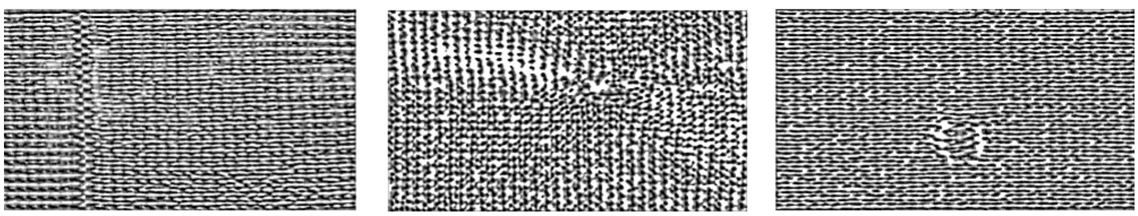
Where,

$$s(x) = \begin{cases} 1; & x \geq 0 \\ 0; & x < 0 \end{cases}$$

The Joint difference distribution could be transformed into $LBP_{P,R}$ number that characterizes the spatial structure of the local image texture

$$LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p \tag{4.19}$$

The $LBP_{P,R}$ operator is an excellent measure of the spatial structure of local image texture. It detects local structures (e.g. edges, lines, spots, flat areas), whose underlying distribution is estimated by a number. In this context this property enables LBP operator to capture surface texture of fabric at micro level and development of generic texture model of surface texture. $LBP_{32,16}$operator has been applied on the pre-processed images in figure 4.1 to demonstrate the nature of local texture model developed by a LBP operator in this context. The results are shown in figure 4.2. Since directional details were minimally important, a weighted mask on binary word generation was not applied during thresolding the neighborhood by the LBP operator.



(a) (b) (c)

Figure 4.2: Local binary pattern (a) end-out (b) snag (c) pin-hole

In this proposed approach, the LBP is calculated with respect to 100 × 150 sub-images by applying an $LBP_{32,16}$ operator. Thereafter a histogram has been plotted from LBP with respect to the sub-image to use as the feature during classification stages. Since, the $LBP_{32,16}$ operator has been used the histogram consists of 32 distinct values and results in vector with dimension of 32. Eventually, this vector would be used as a texture descriptor.

CHAPTER 5 MEHOD FOR IDENTIFICATION OF SURFACE TYPE

The previous chapter discusses the strategy associated with extraction and modeling of texture. This was carried out by modeling local texture using LBP extracting local texture using the histogram. On the other hand global texture details of sub-images have been extracted by applying selected set of statistical measurements on GLCM. The intention of this chapter is to use LBP histogram details for identification of fabric surface while examining the possibility of using LBP features for defect detection.

There had been several attempts on automatic identification of fabric surface type by many researchers. Li *et al.* [22] has suggested a learning vector quantification based classification scheme to identify woven fabric surfaces. They have concluded that the proposed scheme accurately classifies plain weave fabrics, twill weave fabrics and satin weave fabrics. Liqing *et al.* [21] has introduced a method to identify the woven fabric surfaces based on image decomposition using a winner filter. Chung *et al.* [25] has suggested a method based on learning vector quantification to identify fabric surfaces.

Since a narrowed down feature space simplify the classification task the purpose of fabric surface type identification is to restrict the feature space to feature space of a specific fabric type for the defect detection purpose. A KSOM based clustering method has been proposed in this context for fabric surface identification task. The idea is to represent the visual space of different fabric types as a collection of local models as a local model represents similar instances of fabric surfaces in visual space. Since the problem discussed in this thesis concerns *abnormity detection* on texture pattern space consists of texture patterns of different surface types there is a need in reducing the pattern space as it could be simplified the problem. It is important to note that fabric identification is the step towards simplifying this defect detection task.

This chapter is organized as follow:

In the following section a description on scope of the problem has been provided while proposed KSOM model is presented in section 5.1 followed by different stages of development of KSOM. The possibility of developing an abnormality detection criterion is examined and results will be presented in the next chapter.

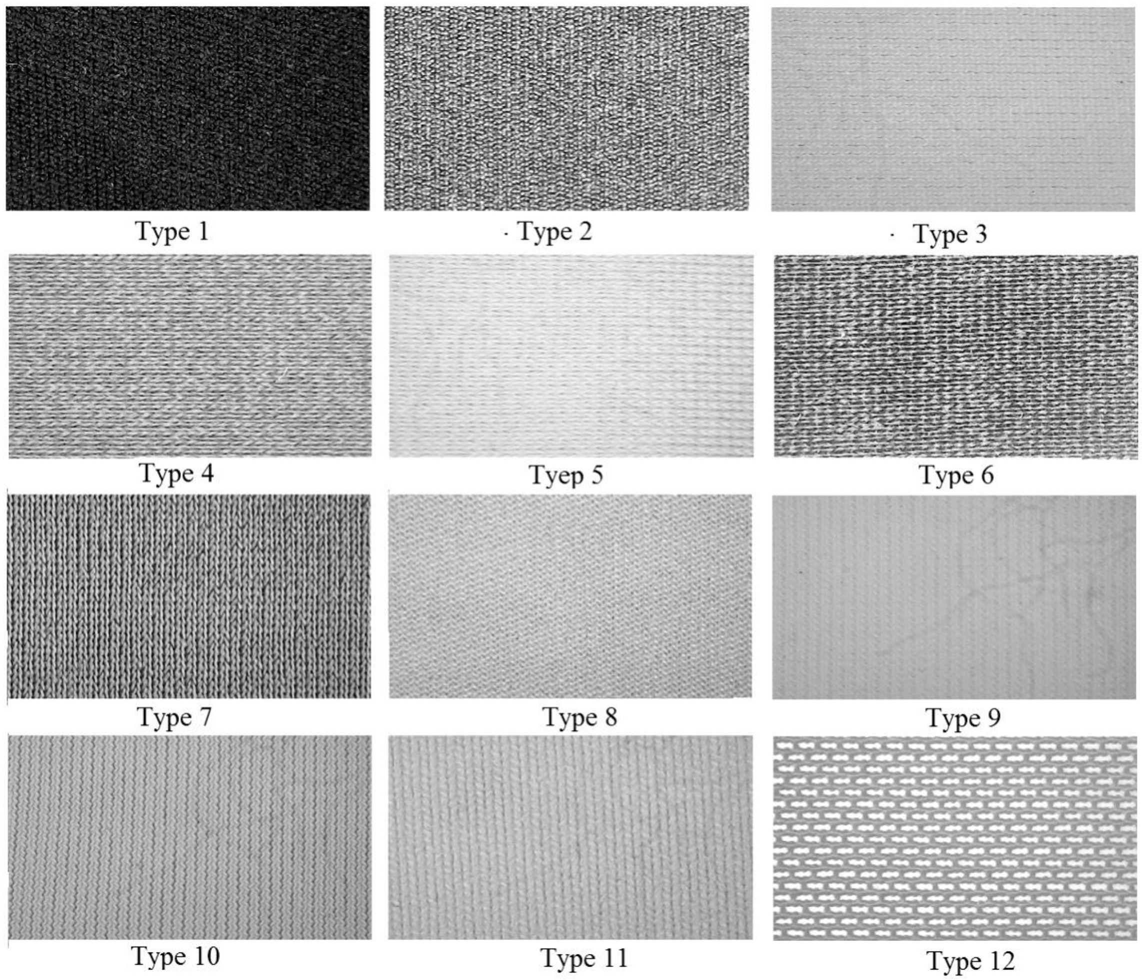


Figure 5.1: Catalogue of fabric types using for the construction of KSOM

5.1 KSOM model for surface type identification

The clustering approach has been used in this context to cluster the surface visual space followed by identification of fabric type using a clustered map where the clustered map has been generated with respect to twelve fabric types. A catalogue of fabric types used in constructing the KSOM has been illustrated in figure 5.1. Four images from each fabric type have been utilized for training purposes. Since the image was subdivided into

150 × 100 sub-images, there were a total of 144 sub-image samples for each fabric type. The histogram feature has been used to extract the details of LBP. Even though LBP represents local texture of fabric surface, the extent of the change a defect causes on the LBP histogram could not be anticipated. Figure 5.2 shows the GUI which has been used for feature extraction during the training stage. The aim of the GUI was to provide state of processing and a basic overview on pre-processing and types of features extracted from images. It should be noted that a method to change the parameters were not provided relevant with pre-processing or feature extraction in this context.

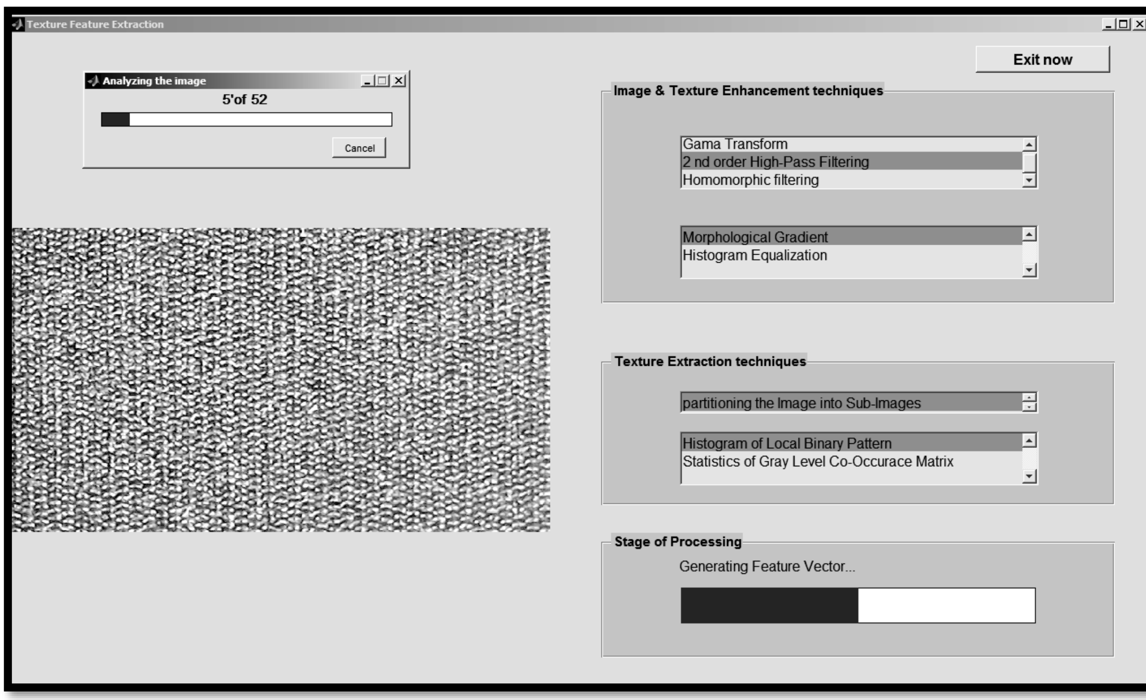


Figure 5.2: GUI for feature extraction stage during construction of KSOM

The surface type identification is a classification problem, thus the intention is to solve it with a KSOM clustering approach. The selection of KSOM for the clustering task is due to its ability to produce local models. However, natural variations and varying lighting conditions may result in variation in illumination and texture within the same sample image and between different sample images. Since texture variation between experimental instances is not a defective state, this scenario could be considered as different texture states within the same texture space. This scenario results in expansion of textural feature space with respect to a particular surface type. Thus, representation of texture space as a collection of local models in which a node represents the locally similar texture situations might be the best way to deal with this scenario.

In this context, the input is the histogram of the LBP of sub-image. The output is the discrete map which provides the basis for fabric type identification. The section 1.2 in chapter one provides an essential computation processes relevant to KSOM. Based on that KSOM is developed using following structural and training parameters

Network architecture and Training Parameters:

A 20×20 Hexagonal grid of nodes has been used as the lattice structure for KSOM in this context. The training of KSOM has been done with a batch computation process where weight adaptation takes place once the system goes through the entire set of feature vectors. Gaussian neighborhood has been used during the training process. The training process of a KSOM consists of two stages which are *coarser training* and *fine training*. During the coarser training phase global ordering of the nodes takes place where the coarser training phase is set to reduce the radius of the neighborhood from 4 to .5 within 30000 iterations. During the finer training phase weight adaptation takes place while the neighborhood is kept constant. In this phase the correct and fine local weight adaptation takes place and the training phase runs for 2000 iterations while Gaussian neighborhood is kept constant at .5. The training process has been carried out in off-line mode and there are other parameter calculations and storing of data associated as shown in figure 5.3. The GUI in figure 5.4 is used for this purpose.

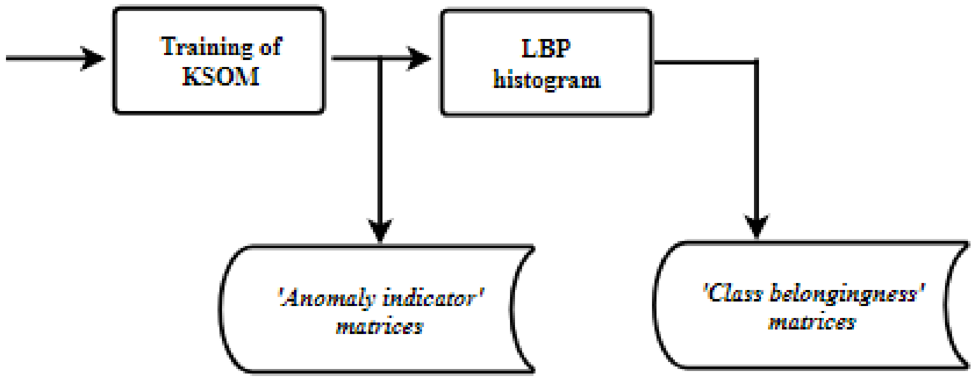


Figure 5.3: Flow diagram for training of KSOM related processes

Labeling phase:

Once the training process is completed the next stage is labeling of nodes. The labeled cluster map was generated by the training samples into the KSOM. Here, instead of using only one BMU, three nearest neighbors have been considered to develop more generalized criteria for fabric type identification in later stages. Identification of the nearest neighbors is performed in accordance with the *k*-nearest neighbor algorithm. The Figure 5.5 illustrates the activated regions of the lattice with respect to each fabric type.

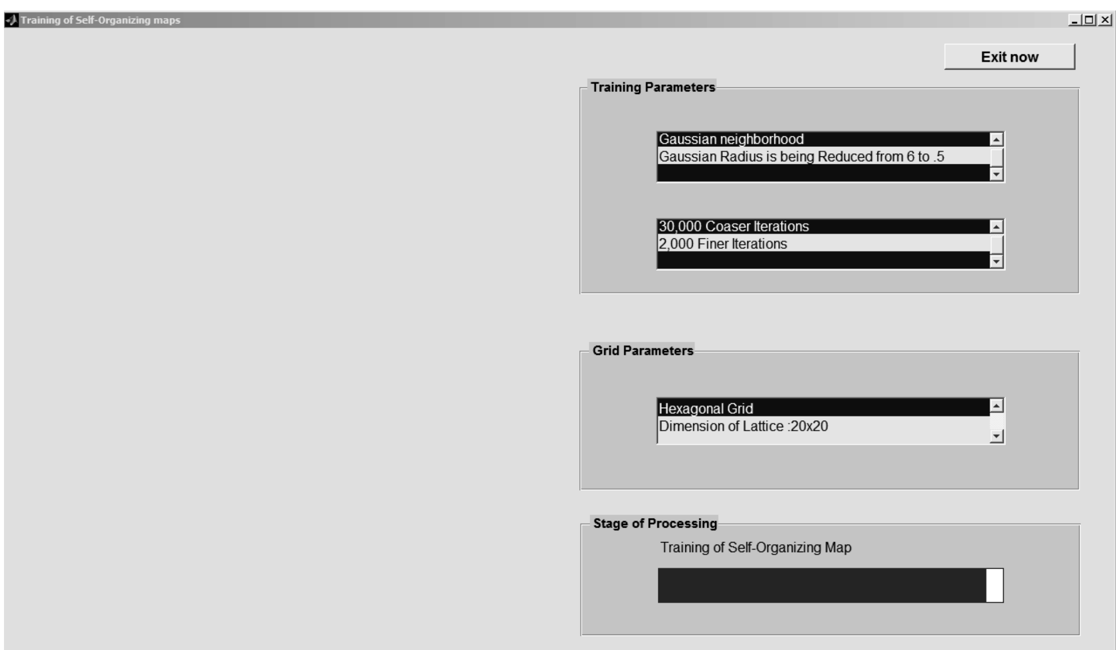


Figure 5.4: GUI for training of KSOM

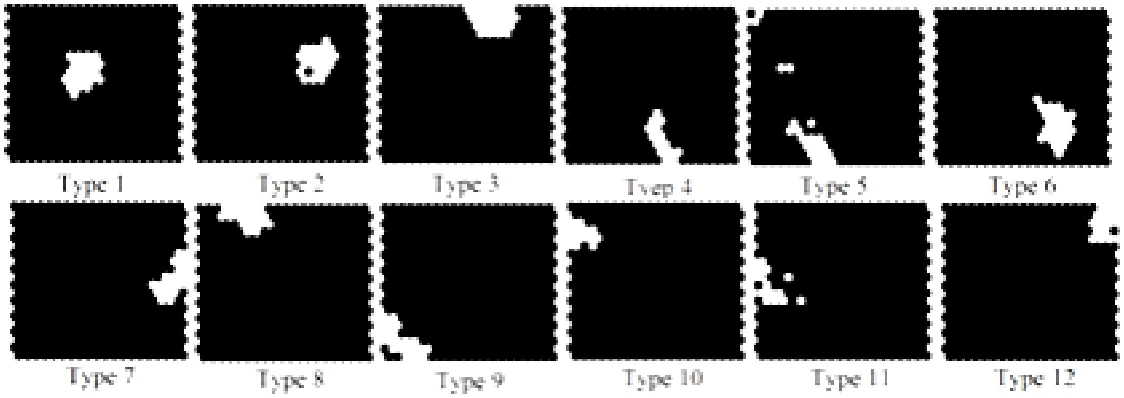


Figure 5.5: Activated regions in KSOM lattice

Based on the activated nodes of the lattice with respect to each surface type compressed map has been generated using Bayesian decision rule in order to represent the class distributions. Figure 5.6 shows the labeled compressed map generated at the end of the training stage and presented in *hsv(13)* color scheme.

A parameter called *'class belongingness'* has been suggested to represent the belongingness of a node to a particular class. Thus, the proposed *class belongingness* of a node to a particular class could be calculated as follows,

$$\text{Belogingness of node k to class i } = \frac{\text{number of hits of class i on node k}}{\text{Total number of hits on node k}} \qquad (5.1)$$

Every node in the grid is associated with a *class belongingness* vector. As a result, the combinations of these vectors produce matrix which have been stored to use in the fabric type identification task. There was another task which was incorporated in to the training phase. The intention was to enforce the defect detection problem towards an abnormality detection criterion. The initial step towards that goal is the examination of the possibility of proposing a defect detection criteria based on local texture abnormalities. Presentation of a sub-image into the KSOM results in three nearest neighbors from the proposed *k*-NN method. The similarity of a sample to BMU is measured in terms of Euclidian distance. The total Euclidian distance resultant from three nearest neighbors after presentation of a test sample to KSOM has been considered as the measurement of '*abnormality indicator'*. These values of abnormality indicators of each sample with respect to each fabric type have been stored in order utilized for detection tasks the up-coming stages as shown in flow diagram of figure 5.3.

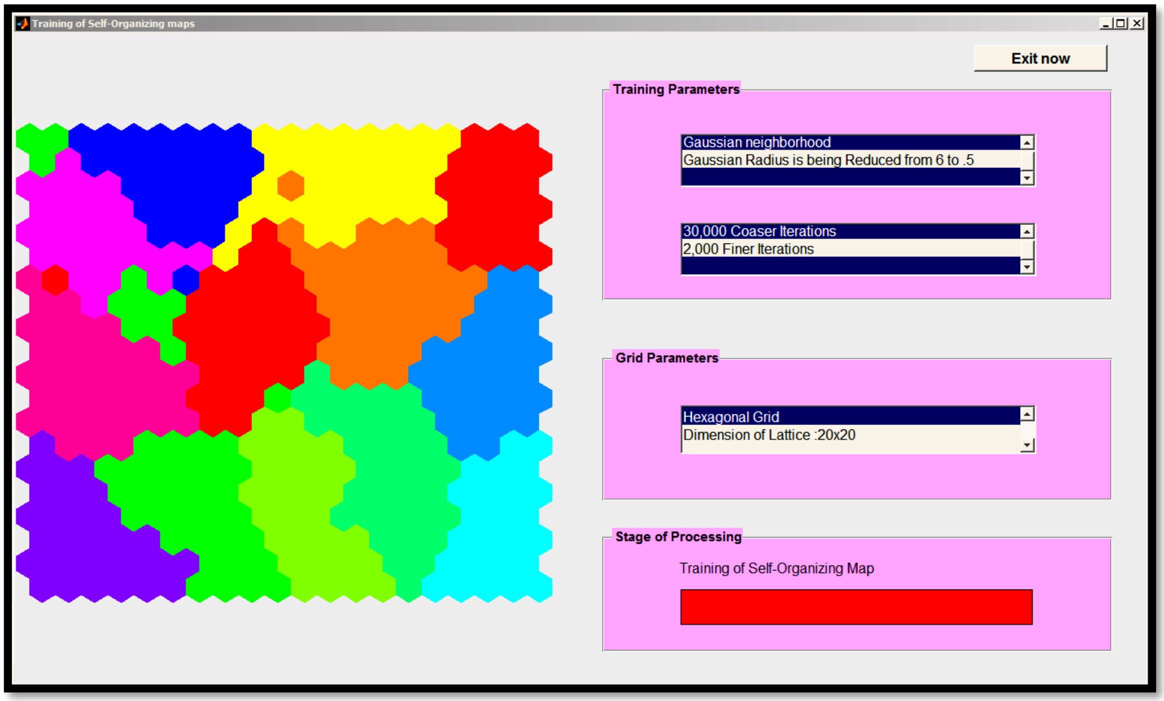


Figure 5.6: Labeled KSOM at the end of training

The Testing Phase:

Even though KSOM operate in sub-images level, the clustering decision should be on an entire image. In this context, the interest is on a combined final decision regarding the surface type, thus an integrating strategy has to be proposed to have a combined final decision. The *class belongingness* vectors of BMUs results from each sub-image are added together. The class with highest *class belongingness* was considered as the class of the image. The proposed scheme is illustrated in figure 5.7.

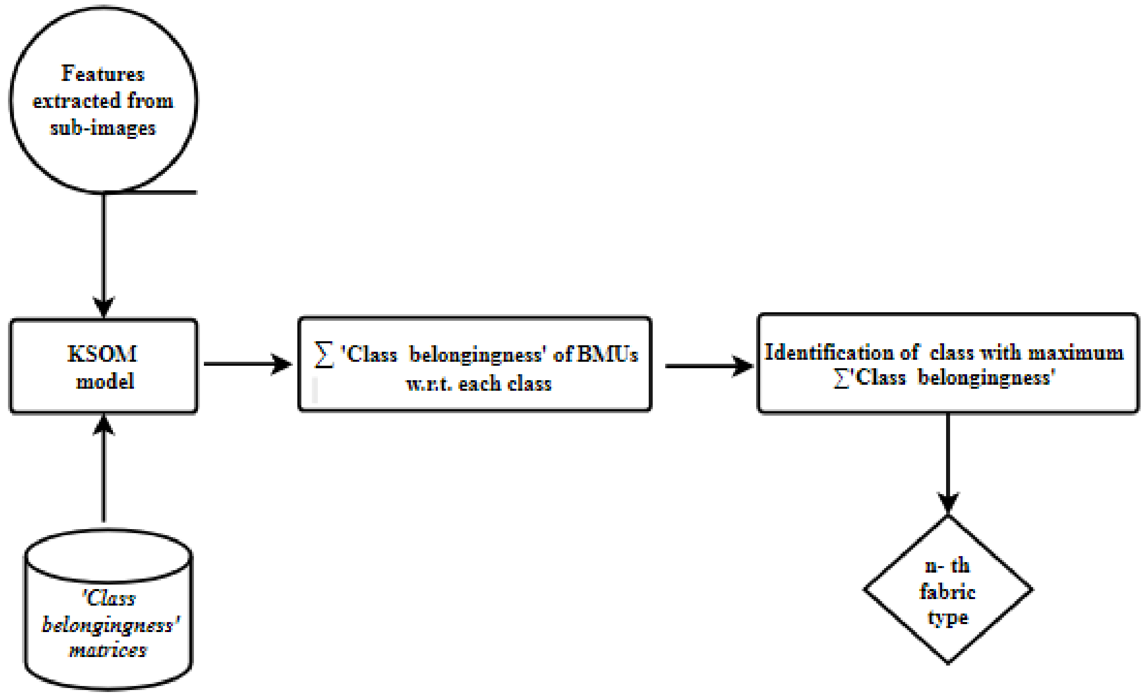


Figure 5.7: Flow diagram of proposed scheme for fabric type identification

The BMU calculation is performed with respect to each feature vector was extracted from sub-images. A winner among all lattice nodes is identified based on a minimum similarity. The similarity between x_i and neuron(w_j) can be measured using Euclidean distance which is given by,

$$c = arg \min_{j} \|x_i - w_j\|^2 \tag{5.2}$$

Here c is the index of neuron having minimum Euclidean distance. Each node in the lattice associated with *class belongingness* vector. Since the input image was divided in to thirty six sub-images during the pre-processing stage, the presentation of an image results in thirty six *class belongingness* vectors. In the proposed scheme, the summing up *class belongingness* vectors of all thirty six BMUs and the index of the class with maximum *class belongingness* is the surface type of the fabric. This criterion could be expressed for a KSOM with n number of nodes and associated with clustering of m number of fabric types which could be illustrated as follows.

$$c = arg\max_{j} \sum_{i=1}^{n} u_{ij} \tag{5.3}$$

Here $j = 1,2, \ldots m$ and c is the label of the class. The scheme has been tested during the experimental sage by using the labeled KSOM in Figure 5.8 with 88 images for final decision which included images from every fabric surface type. In fact, the fabric type identification with labeled KSOM gave 100% positive results.

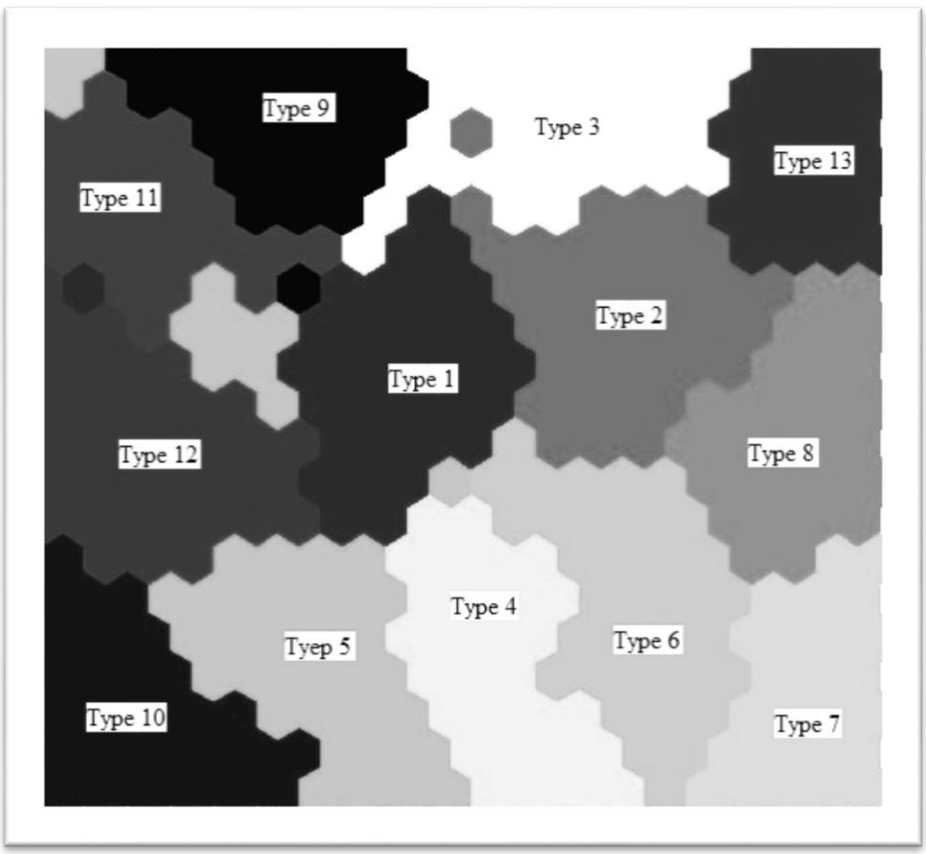


Figure 5.8: Developed KSOM for the surface type identification

CHAPTER 6 DETECTION OF DEFECTS

Monadjemi [13] use the term *'textural abnormality'* to refer to all possible defects in texture surface. In the previous chapter a KSOM based approach was presented for surface type identification of fabric. However, the initial intention was to develop the approach towards an abnormity detection criterion. This possibility has been examined in the previous chapter and the flow diagram is presented in figure 6.1. Since the results were not encouraging, another approach was explored in this situation. However, the identification of surface type itself has certain significance in research perspective as well. On the other hand it simplifies the defect detection task by reducing the problem space of fabric surfaces in to problem space of specific surface type. In this chapter the intention is to propose a modified version of the abnormity detection criterion.

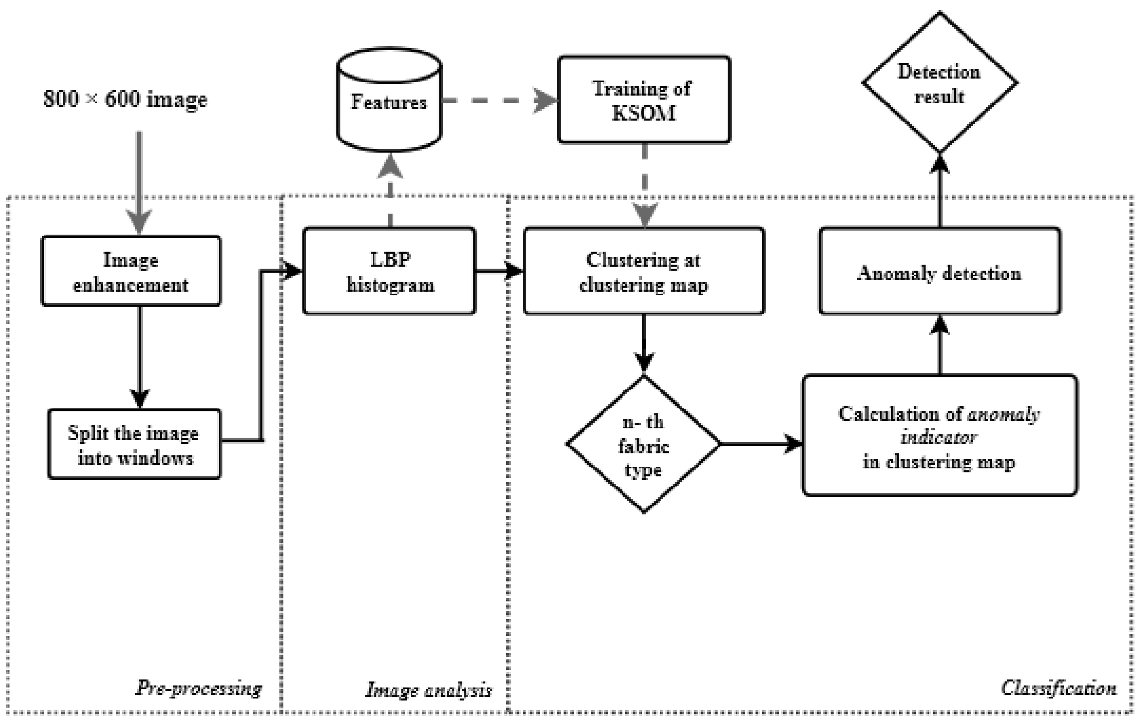


Figure 6.1: Flow diagram of initially developed detection scheme

In this chapter, the objective is to suggest a defect detection scheme based on texture with the following modifications:

1. Instead of defining the surface texture with the LBP histogram feature, the texture was defined using statistics of GLCM which are introduced in section 4.1 of chapter 4 and separate KSOM were developed to represent the texture space.
2. Developing a new abnormality detection criterion for detection of defects in feature space mapped with GLCM statistical features.

The proposed defect detection system can be illustrated using the following flow diagram as shown in figure 6.2.

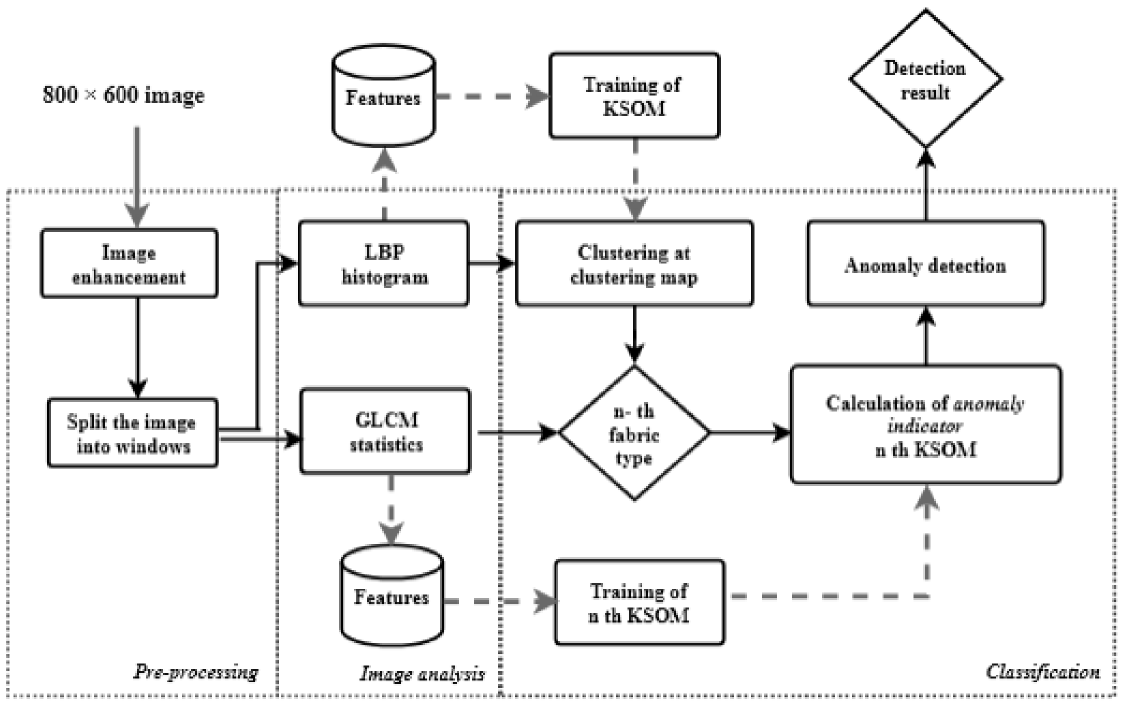


Figure 6.2: Flow diagram of the final detection scheme

Since histogram is a first order statistical model of LBP values of image window it will only represent the combinational effect of all the LBP values on image window. Even though a defect causes change in the local texture it may not represented by the LBP histogram of an image window. However, defect detection using KOM model of local texture was not successful in this context. In this case the new approach of defining the texture of an image window was taken into consideration which defines the texture in terms of GLCM statistics as described in chapter 4. The new approach also includes defining the generic global texture model non-defective image windows using KSOM.

This chapter is organized as follows. In the following section, the proposed KSOM model has been presented. The strategy for detection of defects is proposed in section 6.2 alongside with implementation and experimental results.

6.1 The Proposed KSOM Model

Since the surface type has been identified, the next step would be to proceed with the perspective of the surface texture of the fabric. The approach to perform statistical measurements on GLCM to extract texture of a sub-image was described during chapter 4. A set of trained KSOM models were developed using these measures and used as the generic model of texture of non-defective sub-images of specific surface type. Set of KSOM models to represent the texture was developed as follows,

Grid structure and training parameters:

A 1×10 Hexagonal array of nodes has been used as the grid structure while the training has been done with the batch computation process using a Gaussian neighborhood. In this case, the coarser training phase is set to reduce the radius of the neighborhood from 6 to .5 with 20,000 iterations. During the finer training, a weight adaptation takes place while neighborhood is kept at a constant level. The finer training phase runs for 1000 iterations while the Gaussian neighborhood is kept constant at .5 level.

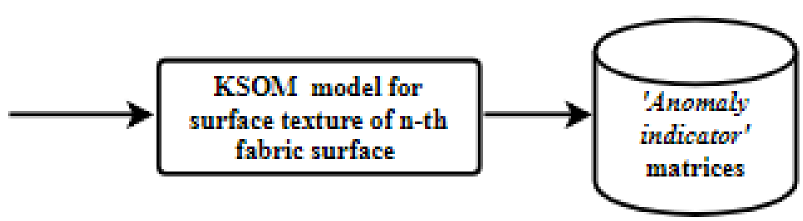


Figure 6.3: Flow diagram for associated processes of KSOM training

Training stage:

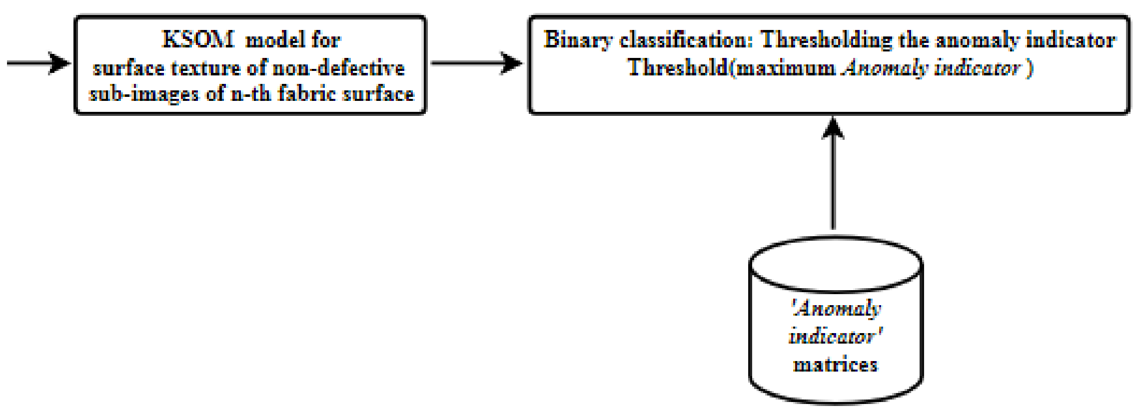
Separate KSOMs have been developed on the proposed grid structure for each fabric surface type using training parameters mentioned. Four images from each fabric type were used for the training purpose of each KSOM. Since the classification tasks are performed with respect to sub-image, an image set resulted in 144 training samples. Since the introduction of a sample into the SOM results in an *abnormality indicator* as described in chapter 5, a database which consists of *abnormality indicator* with respect to each sample was developed during the training of KSOM as illustrated in figure 6.3.

6.2 The method for detection of defects and implementation

The similarity of a sample to BMU is measured inters of Euclidian distance as described in equation 5.2 of scion 5.1. Presentation of a sub-image into the KSOM results in three nearest neighbors from the proposed *k*-NN method and sum of the Euclidian distance of the three nearest neighbors was considered as the *abnormality indicator*. Presence of an abnormal condition is identified using threshold classification which is defined by,

$$f(y_j) = \begin{cases} 1 & \text{if } y_j < T_i \\ 0 & \text{elese where} \end{cases} \quad ; \quad T_i = \max_j(y_j) \qquad (6.1)$$

In this context T_i is the threshold for i-th fabric type. y_j is the *abnormality indicator* of j-th sample.



6.4: Flow diagram for the defect detection strategy

The defect detection strategy is implemented in MATLAB environment. The GUI facilitates the selection of image form require directory as per the experimental requirement. It also provides the stage of processing of the input image. Figure 6.5 shows the GUI while algorithm performs the image analysis.

Figure 6.5: GUI for implementation of defect detection scheme

Figure 6.6 shows the presentation of final decision at the end of classification. The final decision includes the binary mask generated on the input image using the detection results on the sub-images alongside with surface type identification results. Figure 6.6 shows the example results achieved with implementation of the proposed defect detection method. The detections of seven defect types on seven different surface types are considered in this context. The defects considered in this context are stop-line, pin-hole, end-out, snag, thick-line, dirty, marks and crease. Example results achieved during the experimented are presented in table 6.2.

Figure 6.6: Presentation of final detection results on GUI

Table 6.1 illustrates the process of extending the binary classification results of defect detection into binary image and applying it onto the original image as a binary mask.

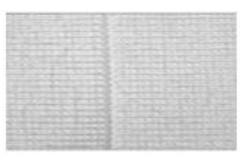
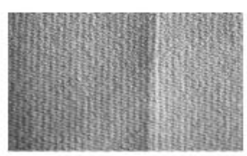
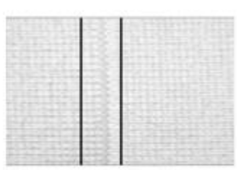
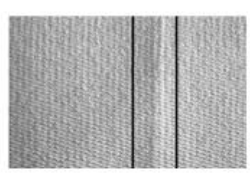
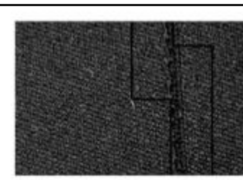
	End-out	**Crease**	**Stop-line**
In-put image			
Detection results			
Application of detection results as binary mask on to image			

TABLE 6.1: ILLUSRATION OF GENERATION OF DETECTION RESULTS

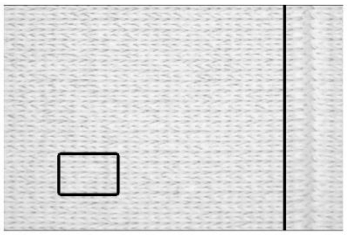
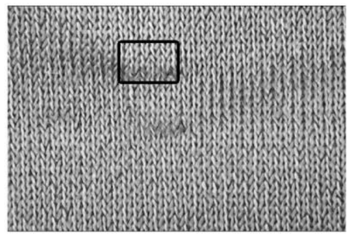
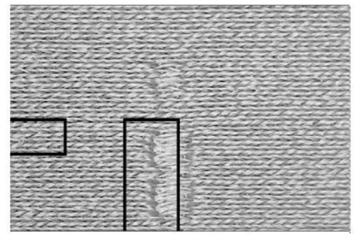
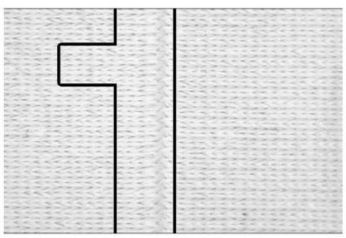
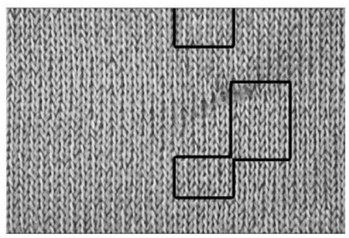
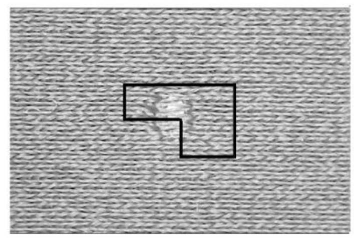
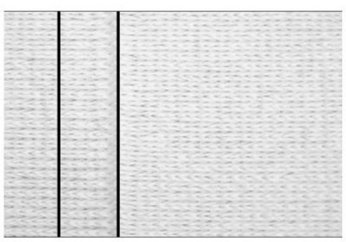
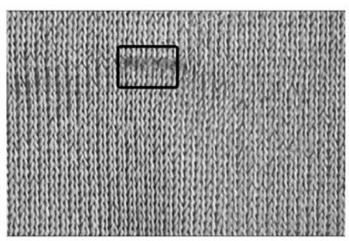
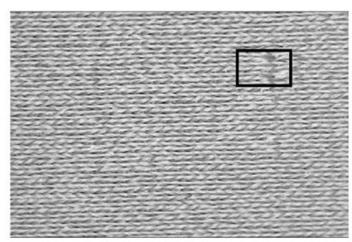
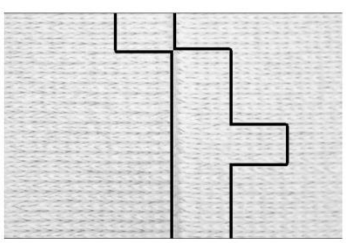
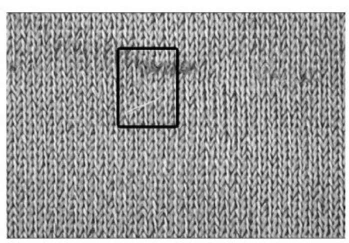
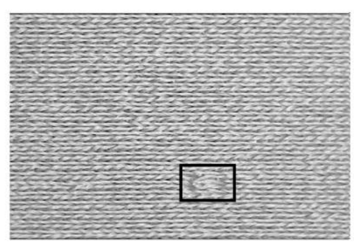
END-OUT	SNAG	PIN-HOLE
		
		
		
		

TABLE 6.2: DETECTION RESULTS OF THE EXPERIMENT

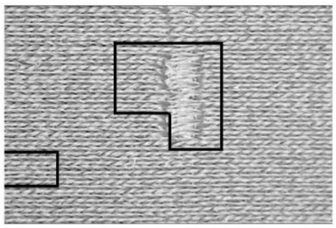
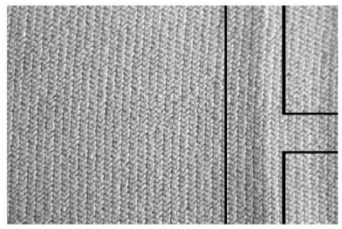
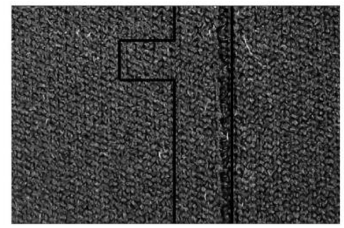
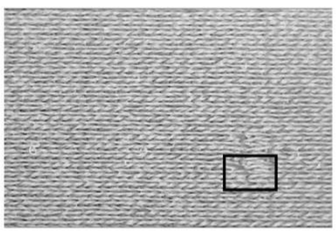
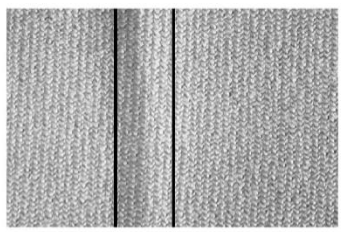
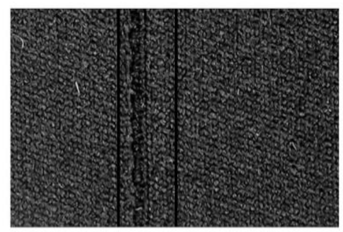
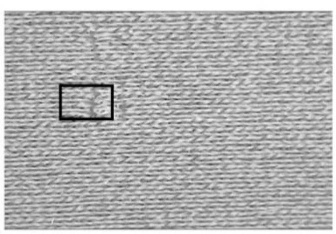
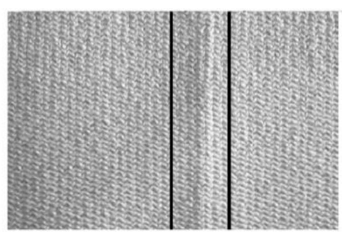
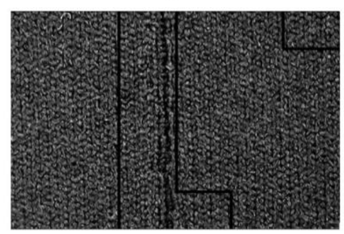
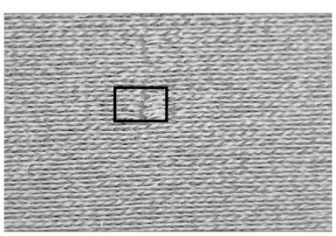
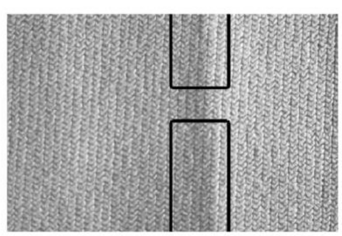
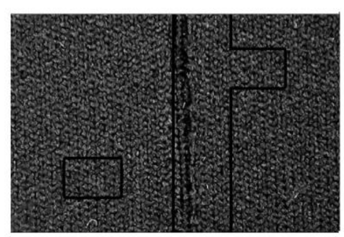
PIN-HOLE	CREASE	STOP-LINE
		
		
		
		

TABLE 6.2: DETECTION RESULTS OF THE EXPERIMENT

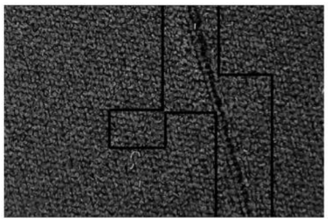
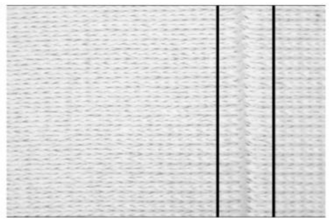
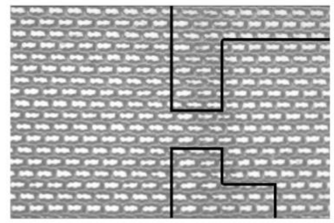
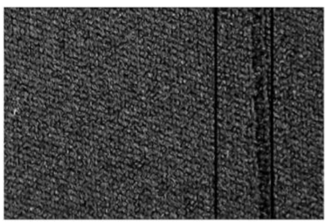
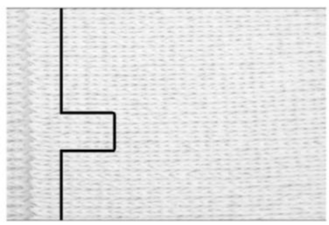
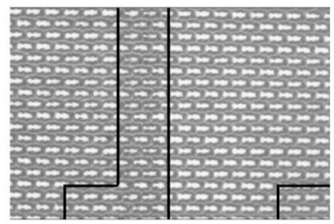
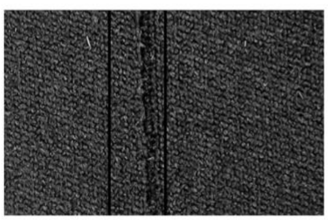
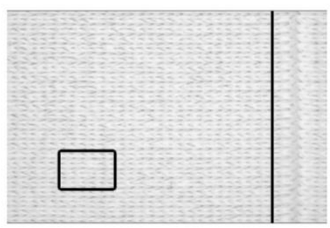
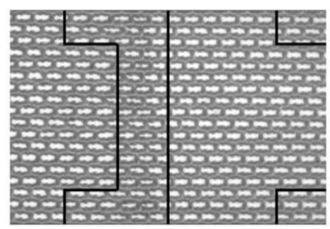
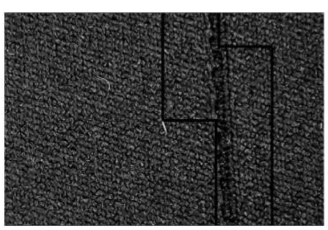
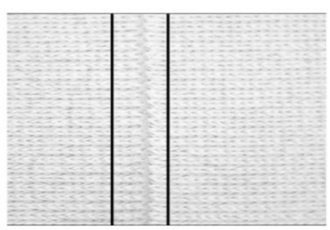
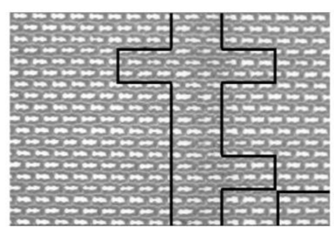
STOP-LINE	**END-OUT**	**THICK-LINE**
		
		
		
		

TABLE 6.2: DETECTION RESULTS OF THE EXPERIMENT

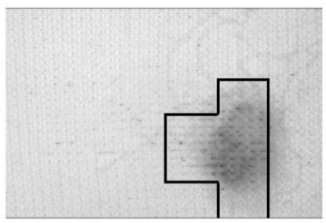
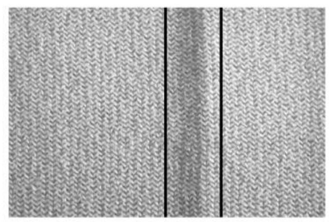
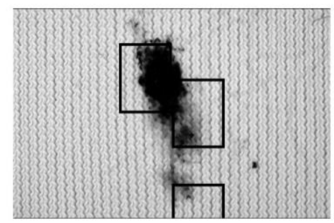
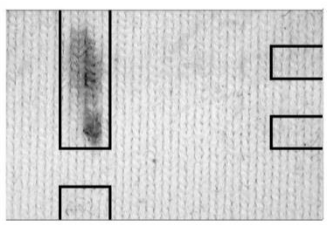
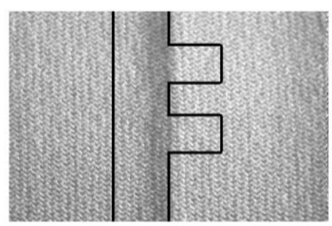
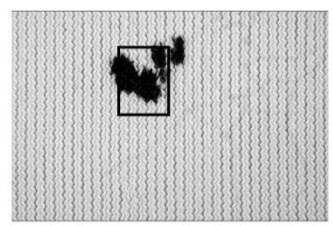
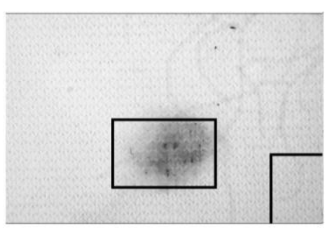
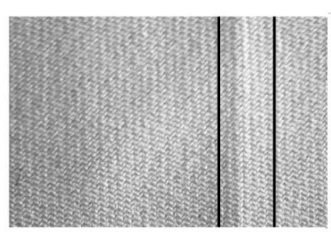
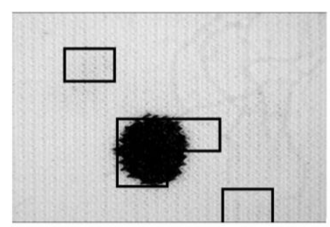
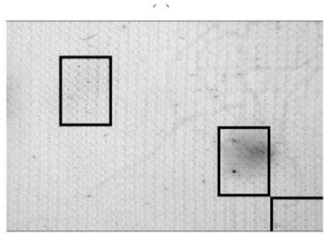
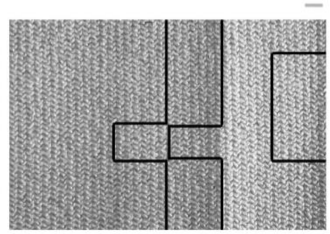
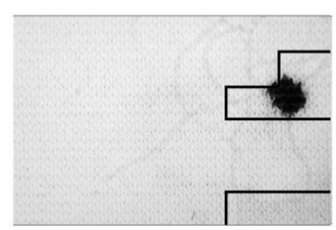
DIRTY MARK	CREASE	CHEMICAL-MARK
		
		
		
		

TABLE 6.2: DETECTION RESULTS OF THE EXPERIMENT

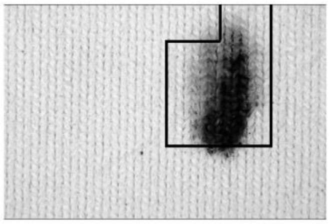
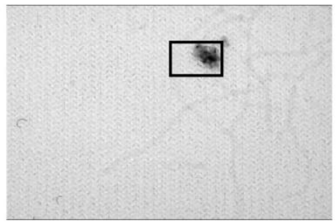
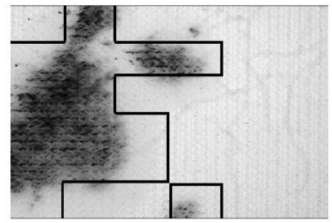
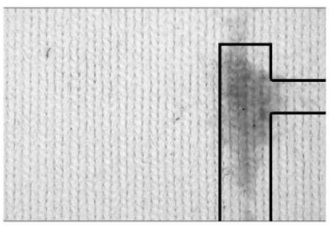
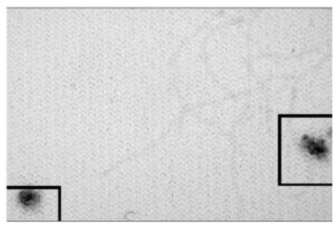
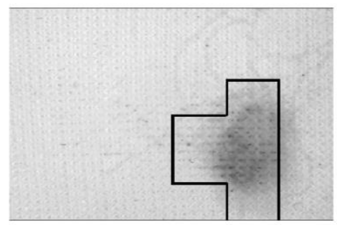
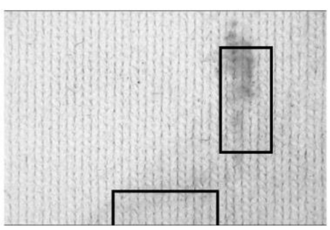
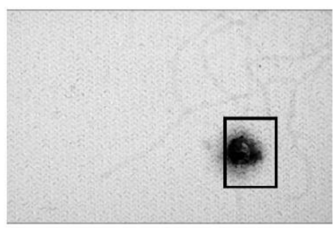
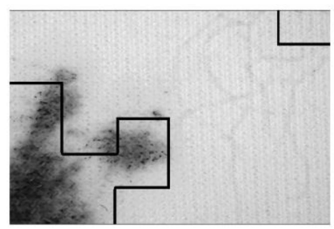
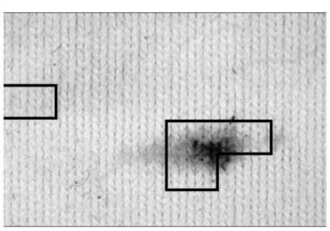
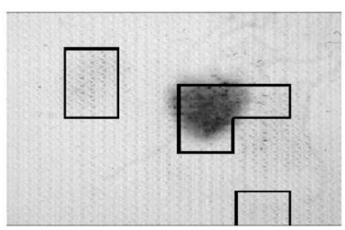
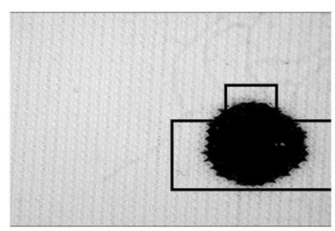
‘MARKS’	***‘MARKS’***	***‘MARKS’***
		
		
		
		

TABLE 6.2: DETECTION RESULTS OF THE EXPERIMENT

TABLE 6.3: RESULTS OF ACCURACY ASSESMENT OF PROPOSED METHOD

Defect type	**Number of windows**				**DP**	**QP**
	TP	*TN*	*FP*	*FN*		
Stop-line	102	447	15	8	92.73	81.60
Pin-hole	29	526	14	6	82.86	59.18
End-out	98	495	14	4	96.08	84.48
Snag	5	135	2	2	71.43	55.56
Dirty mark	55	329	6	6	90.16	82.09
Oil mark	19	120	4	1	95.00	79.17
Thick-line	56	216	6	10	84.85	77.78
Crease	71	351	6	2	97.26	89.87

The detection process includes binary classification of sub-image as defect or non-defect. These results were compared with the state of the sub image percept by visual observation and were categorized into four types: (1) True positive (TP) (2) True negative (TN) (3) False positive (FP) (4) False negative (FN). For performance evaluation, the detection percentage (DP) (equation 6.2) and Quality Percentage (QP) (equation 6.3) are computed with the help of those values:

$$\mathrm{DP} = \frac{100 \times \mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6.2}$$

$$\mathrm{QP} = \frac{100 \times \mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{6.3}$$

88 experiments are performed on images and 3168 detection results were achieved. Table 6.3 shows the detection percentage and quality percentage of the detection process with respect to each defect type which were occurred on different surface types.

CHAPTER 7 CONCLUSIONS

This chapter concludes the research on detection of defects on textured pattered surface of warp knitted fabric surfaces by providing the summary of results achieved during the research. The thesis could be summarized as follows:

1. A set of processes for enhancing the texture has been proposed in this context. Due to the characteristics of the problem several pre-processing stages were introduced in this situation. Thus, Gama correction has been used for the purpose of contrast stretching followed by Homomorphic filtering for correcting illumination. A morphological gradient operator has been applied on the filtered image for the purpose of enriching the texture details followed by histogram equalization. Since a non-segmenting scheme for detection of defects has been proposed the images have been divided into 100×150 sub-images at the end of the pre-processing stage. However modifications may require during practical implementation.

2. The local binary pattern (LBP) provides a better representation of the local texture of an image. The detail of LBP has been extracted using the histogram feature. We explored the possibility of developing *abnormality detection* using histogram feature. Since the results were not encouraging another approach was considered. However, it results in accurate results during surface type identification. The gray level co-occurrence matrix (GLCM) is a method of representing global texture of image. Seventeen statistics have been considered to extract the details of GLCM. Since distinguishability is the necessary characteristic in solving classification problems, a comparative analysis has been performed on values given by each statistic on each sub-image in order to identify the statistics which distinguish defective sub-images from non-defective sub-images. Ten statistics which possesses this quality were identified and used for extraction of texture.

3. Since defect detection on multiple textures were considered, a Kohonen self-organizing map (KSOM) was trained using LBP histogram of sub-images were

used to identify the surface type of the fabric. Three nearest neighbors have been used during the labeling of KSOM. Since the identification of surface type has been taken place in sub-image level a method based on *statistical mode* have been used to integrate clustering decision of the surface type of the sub- image. In fact, the method showed 100% accuracy in identification of fabric structure during 88 experiments which were performed to validate the proposed scheme.

4. After identification of the surface type the problem is reduced to defect detection on single textures. Since abnormality detection is a way of dealing with uncertainty of feature space of one particular class during binary classification, a KSOM based abnormality detection method has been proposed. During this approach a set of KSOM has been developed with respect to each fabric surface type to use as the generic texture model of non-defective sub-images. The abnormality detection was based on the value of the *abnormality indicator* between input and the nearest neighbors. The sum of Euclidean distances between an input and three nearest neighbors was considered as the *abnormality indicator.* Abnormality detection was performed by thrsholding the *abnormality indicator*. The experimental results showed that method results in detection percentage in the range of 80%.

This proposed method is a generic method which may apply in the context of detection of unpredictable textural abnormalities on of textured surface. The applicability of the method has been proved w.r.t. warp knitted surfaces. Research attempts on defect detection on fabric surfaces which include texture pattern are scarce. On the other hand very few researchers have been explored the applicability KSOM for texture classification. This is the first research attempt to solve the defect detection problem of patterned surfaces using abnormality detection and statistical texture analysis. The proposed surface type identification approach enables the integration of multiple types of surface texture types in to the method as well. However, there is a need of exploring the ways of improving the method by selecting fair set of statistics to represent texture, optimizing the image window size, etc. to have improved classification results.

REFERENCES

[1] R. O. K. Reddy, B. E. Reddy and E. K. Reddy, "Classifying the similarity and defect fabric textures based on gray level co-occurrence matrices and binary pattern schemes," *International Journal of Information Engineering and Electronic Business*, vol. 5, pp. 25-33, Nov. 2013.

[2] A. L. Amet, A. Ertüzün and A. Erçil, "An efficient method for texture defect detection: subband domain co-occurrence matrices," *Image and Vision Computing*, vol. 18, pp. 543-553, May 2000.

[3] J. Kyllönen and M. Pietikäinen, "Visual inspection of the parquet slabs by combining color and texture," in *proc. of the IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, 2000, pp. 187-192.

[4] I. Novak and Z. Hocenski, "Texture feature extraction for a visual inspection of ceramic tiles," in *proc. of the IEEE International Symposium on Industrial Electronics*, Dubrovnik, Croatia, 2005, pp. 1279-1283.

[5] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognition*, vol. 36, pp. 1645-1659, July 2003.

[6] R .Stojanovic, P .Mitropulos, C .Koulamas, S .Koubias, G .Popoulos and G .Karayanis, "Automated detection and neural classification of local defects in textile web," in *Proc. of the Int. Conf. on Image Processing And Its Applications*, Manchester, United Kingdom, 1999, pp. 647-651.

[7] C.J. Kuo and C. Lee, "A back-propagation neural network for recognizing fabric defects" *Textile Research Journal*, vol. 73, pp. 147-151, Feb. 2003.

[8] S. H. Jeong, H. T. Choi, S. R. Kim, J. Y. Jaunng and S. H. Kim, "Detecting fabric defects with computer vision and fuzzy rule generation part II: Defect identification by a fuzzy expert system, "*Textile Research Journal*, vol. 71, pp. 561-573, June 2001.

[9] J. Chang, G. Han, J.M. Valverde, N. C. Griswold,J. F. Duque-Carrilloand E. S.´anchez-Sinencio," Cork quality classification system using a unified image processing and fuzzy-neural network methodology," *IEEE Transactions on Neural Networks*, vol. 8, pp. 964-974, July 1997.

[10] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, pp. 275-285, March 2003.

[11] H. Kauppinen, H. Rautio and O. Silvén, "Non-segmenting defect detection and SOM based classification for surface inspection using color vision," in *proc. of the 1999 int. conf. on Polarization and Color Techniques in Industrial Inspection,* Munich, Germany, 1999.

[12] J. Tian, M. H. Azarian, and M. Pecht, "Anomaly detection using self-organizing maps based k-nearest neighbour algorithm," in *Proc. of the Int. Conf. of the Prognostics and Health Management Society*, 2014.

[13] A.Monadjemi, "*Towards Efficient Texture Classification*," Ph.D. dissertation, University of Bristol, Bristol, 2004.

[14] J. Iivarinenand A. Visa, "An adaptive texture and shape based defect classification," in *Proc. of the Int. Conf. on Pattern Recognition*, Brisbane, Australia, 1999, pp.117-122.

[15] A. Kumar, "Computer vision-based fabric defect detection: a survey," *IEEE Transactions on Industrial Electronic*s, vol. 55, pp. 348 - 363, Jan 2008.

[16] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electronic Letters on Computer Vision and Image Analysis*, vol. 7, pp. 1-22, June 2008.

[17] M. Vapola, O. Simula, T. Kohonen and P. Meriläinen, "Representation and identification of fault conditions of an anaesthesia system by means of the self-organizing map," in *Proc. of the Int. conf. on Artificial Neural Networks*, Sorrento, Italy, 1994.

[18] T. E. Mursalin, F. Z. Eishita and A. R. Islam, "Fabric defect inspection system using neural network and microcontroller," *Journal of Theoretical and Applied Information Technology*, vol. 19, pp. 560-570, June 2006.

[19] G. M. Nasira1 and P.Banumathi, "Automatic defect detection algorithm for woven fabric using artificial neural network techniques," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 2620-2624, Jan. 2014.

[20] L. A. O. Martins, F. L.C. P´adua and P. E. M. Almeida, "Automatic detection of surface defects on steel using computer vision and artificial neural networks," in *Proc. of the int. conf. on IEEE Industrial Electronics Society*, Glendale, USA, 2010, pp. 1081-1086.

[21] L. Liquing, T. Gia and X. Chen, "Automatic recognition of the fabric structure based on digital image," *Indian Journal of Textile research*, vol. 33, pp. 388-391, Dec. 2008.

[22] P.F. Li, J. Wang, H.H. Zhang and J.F. Jing, "Automatic woven fabric classification based on support vector machine," in *proc. of the Conf. on Automatic Control and Artificial Intelligence*, Xiamen, China, 2012.

[23] K. Shiranitaa, T. Miyajimab and R. Takiyamac, "Determination of the meat quality by texture analysis," *Pattern Recognition Letters*, vol. 19, pp. 1319-1324, Dec. 1998.

[24] M. S. Hitam, M. Y. H. Muraslan, M. M. Deris and M. Y. M. Saman, "Image texture classification using gray level co-occurrence matrix and neural network," *Europen journal of sciencetific Research*, vol. 75, pp. 591-597, Nov. 2005.

[25] C. Feng ,J. Kuo and C. Tsai, "Automatic recognition of fabric nature by using the approach of texture analysis" *Textile Research Journal*, vol. 76, pp. 564-579, May 2006.

[26] R. M. Haralick and K. Shanmugam, "Textural features for the image classification" *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610-621, Nov. 1973.

[27] A. S. Malek, "*Online fabric inspection by image processing technology*," Ph.D Thesis, Doctoral School, University of Haute Alsace, Haute Alsace, 2012.

[28] l. soh and c. tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE transactions on geoscience and remote sensing*, vol. 37, pp. 780-795, Mar. 1999.

[29] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd ed. New Jercy: Prentice-Hall, 2006.

[30] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray scale and the rotation invariant texture classification with local binary patterns" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971-987, Jul. 2002.

[31] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52-65, Jan. 2013.

[32] C.M. Bishop, *Pattern Recognition and Machine Learning*, Berlin:Sringer, 2006.

APPENDIX A MATLAB SCRIPTS

A.1 Pre-processing of images

```
rgbImage = imread(fullFileName);
```

Green channel

```
greenChannel = rgbImage(:, :, 2);
GG=greenChannel;
```

Gama correction

```
I=double(GG)/255;
G=power(I,5);
```

Homomorphic filter

```
im=im2double(G);
fftlogim=fft2(log(im+0.01));
f=butterhp(im,10,1);
c=fftlogim.*f;
h=real(ifft2(c));
h1=exp(h);
f1=abs(h1);
fm=max(f1(:));
homo=(f1/fm);G=homo;
```

Morphological gradient

```
stE1 = strel('disk',25);
Top=imtophat(homo,stE1);
Bot=imbothat(homo,stE1);
EnIm = (K+Top)-Bot;
```

Histogram equalization

```
hist = adapthisteq(EnIm);
```

Butterworth High-pass

```
function [ out ] = butterhp(im,d,n)
h=size(im,1);
w=size(im,2);
[x,y]=meshgrid(-floor(w/2):floor(w-1)/2, -floor(h/2):floor(h-1)/2);
out =1./(1.+(d./(x.^2+y.^2).^0.5).^(2*n));
end
```

A.2 Texture extraction

Loop for splitting the image:

```
t1 = 100; t2 = 150;
A1 = zeros(1, (600/t1)-1);
A2 = zeros(1, (900/t2)-1);

for k1 = 0 : (600/t1)-1
    A1(k1+1) = t1*k1+1;
end

for k2 = 0 : (900/t2)-1
    A2(k2+1) = t2*k2+1;
end

va=zeros(t1,t2);n=0;

for x=1:600
    for y=1:900
        if (any(x==A1) && any(y==A2))

          %feature extraction from window

        end
    end
     waitbar(i/600,2);
end
```

Local texture model (LBP):

```
%inside spliting loop
n=n+1;
bTemp = b(i:i+t1-1, j:j+t2-1);
nFiltSize=32;nFiltRadius=16;
filtR=generateRadialFilterLBP(nFiltSize, nFiltRadius);
val = efficientLBP(d, 'filtR', filtR, 'isRotInv', false,  'isChanWiseRot', false);
va = efficientLBP(bTemp, 'filtR', filtR, 'isRotInv', true,  'isChanWiseRot', false);
vall{n}=mat2cell(va);
```

Extraction of histogram of local texture model:

```
for xx=1:28 %Fabric type identification training loop

 vall{n}=mat2cell(va);% LBP cell array of image windows

for ii=1:(600/t1)*(900/t2)

hitb=cast(cell2mat(vall{ii}),'double');% cell>mat & casting to double / ii th half of
a2 = unique(hitb);%unique elements in hit b
nb = [a2,histc(hitb(:),a2)];%histogram based on unique elements
nb(1,:)=[];%empty first colum
qa = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,3
aInd = ismember(qa,nb(:,1));% binery vector indicate qa element is in nb
xf = zeros(length(qa),2);
xf(:,1) = qa;% 1 st row of xf = qa
xf(aInd,2) = nb(:,2); % add respective count to 2 nd row of xf based on 1 & 0 of nb
xf(not(aInd),2) = NaN;% NAN to nb's 0
xf(isnan(xf))=0;% 0 to xf's NAN
KKK=xf(:,2);% 2nd coloum of xf
vec{ii}=KKK;% cell array of histogram of ii th split

end
```

Development of GLCM and statistical measurements:

```
%in side splliting loop
n=n+1;
bTemp1 = b1(x:x+t1-1, y:y+t2-1);
glcm2 = graycomatrix(bTemp1,'NumLevels',16,'offset', [0 8], 'Symmetric', true);
out1 = GLCM_features(glcm2,0);

v13=out1.autoc; v14=out1.contr; v15=out1.corrm;v16=out1.corrp; v17=out1.cprom;v
v19=out1.dissi; v110=out1.entro; v111=out1.homom; v112=out1.homop; v113=out1.ma
v115=out1.savgh; v116=out1.svarh; v117=out1.senth; v118=out1.dvarh; v119=out1.d

vat =[v114 v115 v116 v117 v16 v14 v110 v17 v18 v19];
vallt{n}=mat2cell(vat);
```

A.3 Development of KSOM

Surface type wise data base for KSOM training:

```
fvec = (cat(2, vec{:}))'; % vector of all ii parts of image
XX1{xx} = fvec; % cell array of  xx th image

switch xx % Condition to selct image surface type from image folder
   case num2cell(1:4)
        a=1;
   case num2cell(5:8)
        a=2;
   case num2cell(9:12)
        a=3;
   case num2cell(13:16)
        a=4;
   case num2cell(17:20)
        a=5;
   case num2cell(21:24)
        a=6;
   case num2cell(25:28)
        a=7;
end

fvecc = [a.*(ones(size(fvec,1),1)) fvec]; % add additional row(a valu
XXX{xx} = fvecc;% put fvecc to xx th cell
waitbar(xx/28,hf2,sprintf('%1d''of 28',xx))
end
```

Development and training of KSOM:

```
feat=vertcat(XX1{:});% all xx cells to one matrix (xx*ii halfs)
feature = feat(randperm(size(feat,1)),:); % randomize vectors coloum vise
X=feature;
N=20;
lattice = 'hexa';
neigh = 'gaussian';
radius_coarse = [6 .5];
trainlen_coarse = 30000;
radius_fine = [.5 .5];
trainlen_fine = 1000;
smI = som_lininit(X,'msize',[N N],'lattice',lattice,'shape','sheet');
smC = som_batchtrain(smI,X,'radius',radius_coarse,'trainlen',trainlen_coarse, 'ne
sm = som_batchtrain(smC,X,'radius',radius_fine,'trainlen',trainlen_fine, 'neigh',
M = sm.codebook;
norms2 = sum(M.*M,2);
save('Mc.mat','M');
save('norms2c.mat','norms2');
```

Data base generation for labeling of KSOM:

```
featuree=vertcat(XXX{:});
Xc=featuree; % vector with fabric type
B = repmat(featuree(:,1),1,(900/t2)*(600/t1));%type matrix
Xc(:,1) = []; % empty fabric type detail
hits = zeros(400,7);

for u=1:((600/t1)*(900/t2))*xx;
X1 = Xc(u,:)';
Y(:,u) = norms2 - 2*M*X1;
[YY,YYY]=sort(Y);
for i=1:3
    cc=YYY(i);% Nearest 3 naighbours(3 BMU)
switch B(u,i)% put to bins based on type and BMU
   case 1
        hits(cc,1) = hits(cc,1) + 1;
   case 2
        hits(cc,2) = hits(cc,2) + 1;
   case 3
        hits(cc,3) = hits(cc,3) + 1;
   case 4
        hits(cc,4) = hits(cc,4) + 1;
   case 5
        hits(cc,5) = hits(cc,5) + 1;
  case 6
        hits(cc,6) = hits(cc,6) + 1;
  case 7
        hits(cc,7) = hits(cc,7) + 1;
end
end
end
```

3-NN identification

Utilizing 3-NN for Labeling of KSOM

Labeling and coloring of KSOM:

```
colormapigray =  colormap('gray');
for i = 1:7;subplot(1,2,1);
hc1=som_cplane(sm, hits(:, i));
set(hc1,'edgecolor','none');
pause(3);
end

nodelabels = zeros(400,1);
for i=1:length(nodelabels);
[C,c] = max(hits(i,:));
nodelabels(i) = c;
end

colormapigray2 =  colormap('hsv');
subaxis(1,2,1, 'Spacingvert', 1, 'Padding', 0, 'Margin', 0);
hc2=som_cplane(sm, nodelabels);
set(hc2,'edgecolor','none');
```

Calculation of 'class belongingness':

```
out=sum(hits,2);
mem = bsxfun(@rdivide, hits, out);
mem(out==0) = 0;
mem(hits==0) = 0;
save('memc.mat','mem');
```

Execution of KSOM and integration of clustering decisions:

```
fvec = (cat(2, vec{:}))'; % vector of all ii parts of image
X=fvec;
decm=zeros(36,13);

for  u=1:36;
X1 = X(u,:)';
Y = norms2a - 2*Ma*X1;
[C,c] = min(Y);
decm(u,:)=mem(c,:);
end

[su,deci]=max(sum(decm,1));
fprintf('fabric is type %d .\n',deci);
```

Associated processes during training of KSOM:

```
featuree=vertcat(XXX{:});
Xc=featuree; % vector with fabric type
B = repmat(featuree(:,1),1,(900/t2)*(600/t1));
Xc(:,1) = []; % empty fabric type detail

for uu=1:((600/t1)*(900/t2))*xx;
X11=Xc(uu,:)';
d1(uu,:)=norms2 - 2*M*X11;
end
d2=d1';
[BB,II] = sort(d2);
DD=BB((1:3),1:xx*(900/t2)*(600/t1));
DI=II((1:3),1:xx*(900/t2)*(600/t1));
E=sum(DD);

index = zeros(xx*(900/t2)*(600/t1),7);

for u=1:((600/t1)*(900/t2))*xx;

switch featuree(u,1)
   case 1
       index(u,1)=E(u);
   case 2
       index(u,2)=E(u);
   case 3
       index(u,3)=E(u);
   case 4
       index(u,4)=E(u);
   case 5
       index(u,5)=E(u);
   case 6
       index(u,6)=E(u);
   case 7
       index(u,7)=E(u);
end

end

save('indexc.mat','index');
```

Calculation of *'anomaly indicator'*

Storing the *anomaly indicators* results during training in database

A.4 Detection of Defects using KSOM

Using the KSOM for detection of defects:

```
X = (cat(2, vec{:}))'; % vector of all ii parts of image

for uu=1:((600/t1)*(900/t2));
X11=X(uu,:)';
d1=norms2 - 2*M*X11;
d2(uu,:)=d1;
end
d3=d2';
[BB,II] = sort(d3);
DD=BB((1:3),1:(900/t2)*(600/t1));
DI=II((1:3),1:(900/t2)*(600/t1));
Edd=sum(DD);
```

Calculation of *anomaly indicators* w.r.t. all the Sub-images

```
max1=max(index(:,1));
min1=min(index(:,1));
std1=nanstd(index(:,1));
thre = (Edd >max1-3*std1);
```

Thresholding process

Development of binary mask into binary decision:

```
re=(reshape(thre,[(900/t2),(600/t1)]))';
fdeci = sum(re(:));

imshow2 = reshape(repmat(reshape(re',1,[]),t2,1),[],size(re,1))';
imshow3 = reshape(repmat(imshow2(:)',t1,1),[],size(imshow2,2));

maskedimage1 = b1;
maskedimage1(~imshow3) = 0;

Dedge=imshow3;
s=strel('disk',8,0);
Fedge=imerode(Dedge,s);
imshow4=1-(Dedge-Fedge);
rgbImage =imread(fullFileName);
jkm=imresize(rgb2gray(rgbImage),[600 900]);
maskedimage2 = jkm;
maskedimage2(~imshow4) = 0;
```

A.5 Development of graphical user interfaces

GUI for execution of input images:

```
M = struct2array(load('M4.mat'));
norms2 = struct2array(load('norms24.mat'));
index=struct2array(load('index4.mat'));

DlgH = figure(1);
set(1,'units','normalized','outerposition',[0 0 1 1]),set(1,'ToolBar','none');
set(1,'MenuBar','none');
set(1,'Name',sprintf('Fabric defects detection'),'NumberTitle','off');

obj1 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'delete(gcbf)',
'FontSize',11.5,'FontWeight','bold');

p1 = uipanel(DlgH,'Title','Input directory panel','Position',[.015 .81 .4 .12],'FontSi
,'BackgroundColor',[255/255 153/255 255/255]);

p2 = uipanel(DlgH,'Title','Input panel','Position',[.015 .07 .4 .12],'FontSize',11,'Fo
[255/255 153/255 255/255]);

p3 = uipanel(DlgH,'Title','Decision','Position',[.58 .81 .4 .12],'FontSize',11,'FontWe
[255/255 153/255 255/255]);

p4 = uipanel(DlgH,'Title','Stage of Processing','Position',[.58 .07 .4 .12],'FontSize',
[255/255 153/255 255/255]);

textH1 = uicontrol(1,'Style','edit','String','C:\Users\dimuthu\Desktop\type4','Positio

textH2 = uicontrol(1,'Style','edit','String','t.png','Position',[40 90 290 20],'FontSi

obj2 = uicontrol('Position',[340 80 170 40],'String','Continue','Callback','uiresume(g

uiwait(gcf);

myFolder = strcat( get(textH1, 'string')) ;
jpgFileName = strcat( get(textH2, 'string'));
fullFileName = fullfile(myFolder, jpgFileName);
```

GUI for extraction of features:

```
myFolder = 'C:\Users\dimuthu\Desktop\images';
DlgH2 = figure(2);
set(2,'units','normalized','outerposition',[0 0 1 1]),set(2,'ToolBar','none');
set(2,'MenuBar','none');
set(2,'Name',sprintf('Feature Extraction'),'NumberTitle','off');

objdel1 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'dele
'FontSize',11.5,'FontWeight','bold');

p11 = uipanel(DlgH2,'Title','Filtering','Position',[.52 .61 .45 .32],'FontSize',
,'BackgroundColor',[255/255 153/255 255/255]);

p21 = uipanel(DlgH2,'Title','Features and Blocks','Position',[.52 .32 .45 .22],'
[255/255 153/255 255/255]);

p31 = uipanel(DlgH2,'Title','Stage of Processing','Position',[.52 .1 .45 .18],'E
[255/255 153/255 255/255]);

lb11 = uicontrol('Style','listbox',...
                'String',{'Gama Transform by Gama=5) ','Butterworth 1st order HE
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 600 400 60],'FontSize',11.5);

lb21 = uicontrol('Style','listbox',...
                'String',{'close With disk of radius 6 ','TopHat & BottomHat wit
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 500 400 60],'FontSize',11.5);

lb31 = uicontrol('Style','listbox', 'String',...
                {'Devide Image into 100*150 regions ',''},...
                'Position',[750 350 400 25],'FontSize',11.5);

lb41 = uicontrol('Style','listbox', 'String',...
                {'LBP Histogram ','Co-Occurace matrix statistics','Laws features
                'Position',[750 270 400 60],'FontSize',11.5);

hf1 = waitbar(0,'Generating Feature Vector...');

titleHandle1 = get(findobj(hf1,'Type','axes'),'Title');
set(titleHandle1,'FontSize',12,'HorizontalAlignment','right')

cf1 = get(hf1,'Children');
set(cf1,'Parent',2);
set(cf1,'Units','Normalized','Position',[.59 .15 .31 .05]);
close(hf1);

hf2 = waitbar(0,'1','Name','Extracting features',...
            'CreateCancelBtn',...
            'setappdata(gcbf,''canceling'',1)');

titleHandle2 = get(findobj(hf2,'Type','axes'),'Title');
set(titleHandle2,'FontSize',12,'FontWeight','demi')
```

GUI for training of KSOM:

```
DlgH = figure(1);

set(1,'units','normalized','outerposition',[0 0 1 1]),set(1,'ToolBar','none');
set(1,'MenuBar','none');
set(1,'Name',sprintf('Fabric defects detection'),'NumberTitle','off');

obj12 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'delete(gcbf)','Po
'FontSize',11.5,'FontWeight','bold');

p12 = uipanel(DlgH,'Title','Filtering','Position',[.52 .61 .45 .32],'FontSize',11,'FontWeig
,'BackgroundColor',[255/255 153/255 255/255]);

p22 = uipanel(DlgH,'Title','Features and Blocks','Position',[.52 .32 .45 .22],'FontSize',11
[255/255 153/255 255/255]);

p32 = uipanel(DlgH,'Title','Stage of Processing','Position',[.52 .1 .45 .18],'FontSize',11,
[255/255 153/255 255/255]);

lb12 = uicontrol('Style','listbox',...
                'String',{'Size of SOM Lattice ','Lattice Stucture ',''},...
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 600 400 60],'FontSize',11.5);

lb22 = uicontrol('Style','listbox',...
                'String',{'Neighborhood Function-gaussian ','Radius of Coarse Training -[6
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 500 400 60],'FontSize',11.5);

lb32 = uicontrol('Style','listbox', 'String',...
                {'Iterations(coarse)-10000 ','Iterations(fine)-1000'},...
                'Position',[750 300 400 55],'FontSize',11.5);

h = waitbar(0,'Training of Self-Organizing Map');

titleHandle2 = get(findobj(h,'Type','axes'),'Title');
set(titleHandle2,'FontSize',12,'HorizontalAlignment','right');
cc = get(h,'Children');
set(cc,'Parent',1);
set(cc,'Units','Normalized','Position',[.59 .15 .31 .05]);
close(h);
```