DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DETECTING AND CLASSIFYING FABRIC DEFECTS WITH COMPUTER-VISION ALGORITHMS

by
Fatma Günseli ÇIKLAÇANDIR

September, 2022
İZMİR

DETECTING AND CLASSIFYING FABRIC DEFECTS WITH COMPUTER-VISION ALGORITHMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering, Computer Engineering Program

by
Fatma Günseli ÇIKLAÇANDIR

September, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**DETECTING AND CLASSIFYING FABRIC DEFECTS WITH COMPUTER-VISION ALGORITHMS**” completed by **FATMA GÜNSELİ ÇIKLAÇANDIR** under supervision of **ASSOC. PROF. DR. SEMİH UTKU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Semih UTKU

Supervisor

Assoc. Prof. Dr. Hakan ÖZDEMİR

Thesis Committee Member

Assoc. Prof. Dr. Mehmet Hilal ÖZCANHAN

Thesis Committee Member

Prof. Dr. Ayşegül ALAYBEYOĞLU

Examining Committee Member

Assoc. Prof. Dr. Deniz KILINÇ

Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

At first, I would like to thank my dear family members, who have always been my biggest supporters in everything. I would like to thank to my supervisor Assoc. Prof. Dr. Semih UTKU, Assoc. Prof. Dr. Hakan ÖZDEMİR, and Assoc. Prof. Dr. Mehmet Hilal ÖZCANHAN for their encouragement and contributions. I am grateful to everyone who made this journey easier for me.

Fatma Günseli ÇIKLAÇANDIR

DETECTING AND CLASSIFYING FABRIC DEFECTS WITH COMPUTER-VISION ALGORITHMS

ABSTRACT

Image processing has been employed in a variety of fields since the advent of image processing techniques. One of these fields is textile. The existence of any defect in a fabric is one of the most important factors affecting the quality of the fabric. There are many types of fabric defects that can occur for various reasons. It's critical to figure out what caused the defect and fix it so that it doesn't occur again.

Automation of fabric defect detection has recently attracted a lot of interest in view of the development in artificial intelligence technology in order to be able to discover defects with a high degree of success and to limit the harm to the manufacturer. However, some problems are encountered in this area. Fabric defect detection is a challenging subject since there exist a great number of defects that might result from a variety of issues. Additionally, the restriction of this study is that the Tilda database is one of the limited datasets that contain fabric defect samples and can be accessed in this field.

This thesis focuses on analyzing different feature extraction methods and different classifiers and discussing the advantages and disadvantages of the combinations. Different cases have been created that handle the data sets from different angles and apply different methods. While three different methods (EL, KNN, and SVM) have been tested in the classification stage, different CNN-based approaches (ResNet18, ResNet50, GoogLeNet, and AlexNet) have been tested in the feature extraction stage. The results obtained have been also compared with the results of ResNet18, ResNet50, GoogLeNet, and AlexNet.

Keywords: fabric defect classification, CNN, machine learning, deep learning.

BİLGİSAYAR-GÖRME ALGORİTMALARI İLE KUMAŞ HATALARININ TESPİTİ VE SINIFLANDIRILMASI

ÖZ

Görüntü işleme, görüntü işleme tekniklerinin ortaya çıkmasından bu yana çeşitli alanlarda kullanılmıştır. Bu alanlardan biri de tekstildir. Bir kumaşta herhangi bir kusurun varlığı bu kumaşın kalitesini etkileyen en önemli faktörlerden biridir. Çeşitli nedenlerle oluşabilen birçok kumaş kusuru türü vardır. Kusura neyin neden olduğunu bulmak ve tekrar oluşmaması için düzeltmek çok önemlidir.

Kumaş kusur tespitinin otomasyonu, hataları yüksek derecede başarı ile keşfedebilmek ve üreticiye verilen zararı sınırlandırmak için yapay zekâ teknolojisindeki gelişmeler göz önüne alındığında son zamanlarda büyük ilgi görmüştür. Ancak bu alanda bazı sorunlarla karşılaşılmaktadır. Kumaş kusur tespiti, çeşitli sorunlardan kaynaklanabilecek çok sayıda kusur bulunduğundan zorlu bir konudur. Ayrıca bu çalışmanın kısıtlılığı, Tilda veri tabanının kumaş kusur örneklerini içeren ve bu alanda erişilebilen sınırlı veri setlerinden biri olmasıdır.

Bu tez, farklı öznitelik çıkarma yöntemlerini ve farklı sınıflandırıcıları analiz etmeye ve bu kombinasyonların avantaj ve dezavantajlarını tartışmaya odaklanmaktadır. Veri kümelerini farklı açılardan ele alan ve farklı yöntemler uygulayan farklı durumlar oluşturulmuştur. Sınıflandırma aşamasında üç farklı yöntem (EL, KNN ve SVM) test edilirken, öznitelik çıkarma aşamasında CNN tabanlı farklı yaklaşımlar (ResNet18, ResNet50, GoogLeNet ve AlexNet) test edilmiştir. Elde edilen sonuçlar ResNet18, ResNet50, GoogLeNet ve AlexNet sonuçlarıyla da karşılaştırılmıştır.

Anahtar kelimeler: kumaş kusuru sınıflandırma, KSA, makine öğrenmesi, derin öğrenme.

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

The textile industry, one of Turkey's more established industrial sectors, has had a considerable influence on the country's economy throughout the years in terms of both the employment opportunities it provides and the value it contributes to the production process (Ala & İkiz, 2015). One of the most important subsectors of the textile industry is the woven fabric sector. Additionally, woven fabric exports account for a significant portion of all global textile exports. On the other hand, as in the manufacture of all goods, defects are inevitable in the production of woven fabrics. Woven fabric defects are the variations that impact the fabric's look, modify the fabric's structure, and result in modifications to the region's limits (Dülgeroğlu Kısaoğlu, 2010). The Turkish Standards Institute (TSE) defines fabric defects as "defects in fabrics that can be seen and evaluated due to yarn, auxiliary materials, workmanship, machinery, equipment or working method and spoil the appearance of the fabric" (2005). Therefore, defects on fabrics cause some negative consequences such as customer dissatisfaction and financial loss. If there is a defect in a fabric, it causes 45-65% decrease in price of it (Srinivasan et al., 1992). Traditionally, fabric defect detection is done by human vision. Detection of fabric defects can be done on-line during fabric production or off-line after fabric production (Hanbay & Talu, 2014). Preventing the fabric defects depends on intervening as soon as any defect occurs. While the possibility of intervention during production is eliminated in off-line control, there is a possibility that the error rate increases in on-line control during production. Although human-based control is a common type of control, it has a number of disadvantages (Table 1.1). Personnel who will provide fabric control are selected among trained and experienced people in this field. In the study of Ala and İkiz (2015), production is carried out on 30 weaving looms and the personnel work in three shifts. This creates the need for a large number of personnel in a company. Despite the high costs, the success of finding defects is low.

Table 1.1. The disadvantages of traditional defect control

Disadvantage	Explanation
Loss of human power	There is a loss of manpower due to the employment of trained personnel who are assigned only to find defects.
Waste of time	If the control is done after fabric production, it causes time loss.
Low success rate	It has been reported that the experienced personnel in this field can find 70% of the defects (Dorrity et al., 1995).
Higher cost	In addition to the salaries of the personnel employed for the control process, the cost is high due to the fact that some of the defects cannot be found by the personnel.
High rate of error	At least 30 percent of the defects cannot be found. Personnel worked in this field may be tired and unable to concentrate.

According to the study of Kısaoğlu (2002), fabric defects are divided into four categories: defects in the weft direction, defects in the warp direction, defects on the fabric surface and edge defects. Some defect names belonging to these four types are given in Table 1.2. As a result of our reviews, it is seen that different numbers of fabric defects as well as different groupings of these defects have been reported in the literature. Goldberg (1950) categorized fabric defects into five groups and cited 194 distinct types of defects.

They are divided into six classes and 130 distinct defects in accordance with ISO standards. Automatic defect classification is a challenging endeavor since there is such a wide diversity of fabric types and defect types.

Table 1.2. Defects according to their direction

Defect direction	**Defects**
Defects in the weft direction	Weft bar, dense-loose pick spacing, weft loops, dirty yarn
Defects in the warp direction	Drafting, warp end
Defects on the fabric surface	Fibrous weft, hole, lattice, untwisted yarn, knot, fly
Edge defects	Temple mark, tightloose warp, selvedge fault

Some samples of defects belonging to the types are given in Figure 1.1. Defects in (a) and (b) are the defects in weft direction. Defects in (c) and (d) are the defects in warp direction. Defects in (e) and (f) are on the fabric surface. Defects in (g) and (h) are the edge defects. As can be seen from these samples, some of the defects are obvious, while others are more difficult to understand. Considering that the working personnel are trying to find the fabric defects on the board that flows at a certain speed during production, most minor defects may be overlooked. For this reason, while there are actually many errors, in practice only 40-50 of these defects are recorded on the quality control cards by textile companies (Ala & İkiz, 2015).

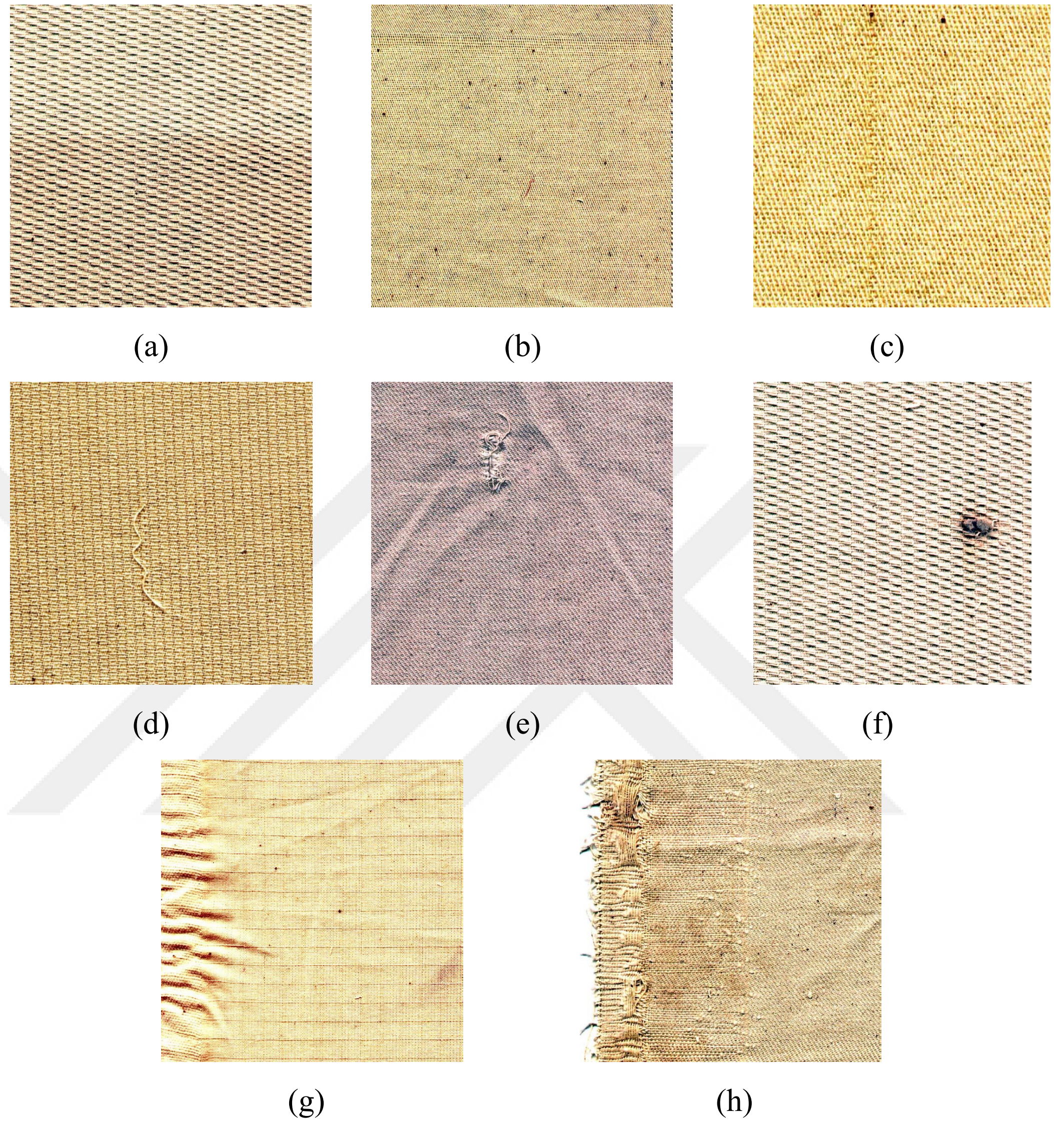


Figure 1.1. Fabric defect samples (a) weft bar (b) dense-loose pick spacing (c) drafting (d) warp end (e) hole (f) fly (g) tight-loose warp (h) selvedge fault (Kısaoğlu, 2002)

Ala & İkiz (2015) have encountered 3211 defects in 140.062 meters of fabric in their study. According to the study, broken warp, stopping mark, warp stack, weft stack, half wrong lift, double pick, missing warp, missing weft, tight-loose warp, and wrong lift are the top 10 most common defect types (Table 1.3). The ten most common defects constitute 94.74% of the total number of defects.

Table 1.3. Most common defect types

Defect Type	Direction	Number of Defects	Percentage of Defects
Broken warp	Warp direction	1542	48.02
Stopping Mark	Weft direction	492	15.32
Warp stack	Warp direction	279	8.69
Weft stack	Weft direction	221	6.88
Half wrong lift	Weft direction	191	5.95
Double Pick	Weft direction	104	3.24
Missing warp	Warp direction	69	2.15
Missing weft	Weft direction	65	2.02
Tight-loose warp	Warp direction	40	1.25
Wrong Lift	Weft direction	39	1.21
SUM		**3042**	**94.74**

Since the formation of these defects can be caused by raw materials, machinery or human beings, it is in the benefit of the manufacturer to determine the type of defect and take action accordingly (Barış and Ozek, 2019). Therefore, traditional defect control cannot fully meet the expectations considering the disadvantages mentioned above. Over the last two decades, many studies have been conducted in the field of automatic fabric control to avoid them (Rasheed et al., 2020). Automatic detection has a number of benefits, including lessening the loss of human power, cutting down on the time and expense needed for control, providing more precise results, and recording the detection process in order to catch future defects (Kısaoğlu, 2002). Some of the automated systems in this field are merely designed to identify problems; others classify them after doing so. Studies to categorize defects are only doing so for certain defect kinds due to the vast array of defect types that might exist in textiles.

The objectives of this study can be listed as follows:

1- To perform fabric defect control, which is traditionally done using human power, with high accuracy rate via computer vision.

2- To classify defect types using image processing and deep learning methods

3- To design a lower-cost automated system with higher success, instead of a costly control process with 70% success in the best case.

4- To save the manufacturer, who wants to perform the control process automatically, from foreign dependency and to meet with a much more affordable domestic system.

5- Textile industry is one of the leading sectors in Turkey (Uyanık & Çelikel, 2019). It is obvious that launching a domestic product in this field will make significant contributions to our country's economy.

The organization of this thesis consists of eight chapters. In the first chapter, frequently encountered fabric defects, the types of these defects and the reasons for the search for an automatic system are mentioned. In the second chapter, a detailed literature review is included. Studies using different feature extraction methods and studies using different classifiers are examined separately. The third chapter deals with data mining. The fourth chapter contains information about the dataset. In the fifth chapter, the methods used for feature extraction (deep learning based approaches, and the proposed system) in the study are discussed. The sixth chapter consists of the bacground for classification methods. While the seventh chapter includes experimental results, the eighth chapter contains explanations about the findings of the study and future studies.

CHAPTER TWO

LITERATURE REVIEW

Considering that each picture is a matrix, it is costly to take the entire matrix as input due to the size of the matrices. For this reason, feature extraction methods are used to reduce the cost in image classification studies. In the literature, it is seen that different feature extraction methods have been used before using a classifier. Gray level co-occurrence matrix (GLCM), Local Binary Patterns (LBP), Principal Component Analysis (PCA), Independent Component Analysis (ICA) are among the commonly used feature extraction methods.

Yıldız et al. (2016) have proposed a system that classifies the fabric defects using the input images obtained from a thermal camera. K-Nearest Neghbor has been used for recognition of the defects using the features extracted via Gray level co-occurrence matrix (GLCM). In this study, four types of defect (hole, tear, nep, foreign yarn) have been tried to be recognized. Experiments show that the system achieves the classification with 96% success rate.

An algorithm has been developed using the combination of wavelet theory and co-occurrence matrix in the study of Latif-Amet et al. (2000). Images are divided to non-overlapping subframes. Frames are classified as defected-non-defected with the help of the features extracted from each frame. Classification is carried out with a success of up to 90.78%.

Zuo et al. (2012) have divided the images to non-overlapping windows and they use texture enhancement method to discriminate the defected area from the background. Features of the defected area is extracted using gray level co-occurrence matrice, and they are classified using euclidean distance classifier. The experiments have been performed on 19 fabric images with the five defect types (broken end, mispick, netting multiples, slack end, thick bar). According to the experiments, they obtain higher accuracy rate (88.79) when non-local means algorithm is used for denoising before gray level co-occurrence matrice. 67.30% success has been achieved when using the method without denoising.

Hamdi et al. (2016) identify the patterns of patterned fabrics and split them into blocks of equal sizes. Features are extracted from images using GLCM. Defected blocks are identified using Euclidean distance. Experiments have been performed on a fabric database. There are three types of pattern in the database (dot, box, and star). They obtain high accuracy rates according to the experimental results.

Ngan et al. (2005) have developed WGIS (wavelet preprocessed golden image subtraction) method. The experiments have been tested on 60 patterned images (30 non-defected, 30 defected). The success rate is 96.7%

Features are extracted using four scale dyadic wavelet decomposition in the study of Lambert & Bock (1997). Following that, a neural network is used to classify these features. They underline that the speed of the wavelet transform gives the devised approach less temporal complexity than the other methods.

Yang et al. (2004) compare six different methods based on wavelet transform. They have been tested on 900 sample images (466 defected, 434 non-defected). Defected samples contains eight fabric defect type. Discriminative feature extraction using adaptive wavelet has been the best methods among the others. It successfully classifies the images with a 95.8% accuracy rate.

Kang et al. (2013) use the combination of wavelet transform and neural network. Accuracy rate for recognition the defects is above 90%. It is stated that the system developed is not sufficient for smaller fabric defects.

Sabeenian et al. (2011) have developed a system that detects and classifies the fabric defects. The features are extracted from the images (stored in the database) using Multi Resolution Combined Statistical and Spatial Frequency (MRCSF). These features are compared with the test images to decide whether there is a defect in the test image. Nearest neighborhood algorithm is used for defect detection phase. Then, defect type is found. However, it is not specified which method is used at classification stage. Experiments have been performed on two different fabric type: normal fabrics and silk fabrics. In the developed system, 85% success has been achieved in classifying the defects of normal fabrics, while silk fabric defects have been classified with 80% success.

Tajeripour et al. (2008) use the modified version of Local Binary Patterns (LBP) to reduce the complexity. A defect-free and patterned image is divided into frames. LBP is applied to each frame and the reference property vector is calculated. A suitable threshold is determined for these defect-free fabrics. In the test image divided into the frames, the defective frame is determined by the threshold.

Li et al. (2019) develop an algorithm based on the combination of gabor and histogram of oriented gradients (HOG). Gabor-HOG has been used for feature extraction. Then, defective regions of the fabrics have been detected using low-rank decomposition. Their experiments have been performed for the dot-patterned fabrics, the star-patterned fabrics, and the box-patterned fabrics. The algorithm detects the defective regions correctly according to the experimental results.

In the study of Ananthavaram et al.(2012), it is aimed to find defects in patterned fabrics by combining Regular Bands (RB) and Independent Component Analysis (ICA) methods. According to the tests, it is concluded that better results have been obtained when histogram equalization method is applied. It is indicated that the proposed approach is suitable for real-time applications.

Martinez-Leon et al. (2016) develop a system to detect the defects in patterned fabrics using the entropy feature calculated from histogram differences and totals. The image is converted to an entropy image where the defects have lower values, and a simple threshold is used to determine if the fabric is defective or not. In this study, test have been performed to detect the defects of broken end, thick bar, thin bar, hole, multiple netting and 96.1% success has been achieved.

Sakhare et al. (Sakhare et al., 2015) compare the performances of six approaches (statistical approach, morphological approach, Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Wavelet Transform, and Gabor Filter). Input images are divided into four parts. The mean of these parts is used to identify the defective part. The existence of a hole in the defect is first investigated. If the defect is not a hole, the remaining sorts of defects are examined for classification. Testing has been done to classify four different types of defects in the study. The experiments indicate that FFT has produced the best result. DCT has also shown to be the second-most effective technique.

Mottalib et al. (Mottalib et al., 2015) use Bayesian classifier in their study. After segmenting of the defect from a fabric image, four features are extracted from the defect window segmented. These features are height of defect window, width of defect window, height to width ratio of defect window, and number of defective regions. Five fabric defects have been aimed to detect (color yarn, vertical missing yarn, horizontal missing yarn, hole, spot). They have used 70 of 128 input images for training phase and the rest for testing phase. Non-defected images have been classified with the success rate of 100%. In total, 99.19% success rate has been obtained.

Ozkaya et al. (Ozkaya et al., 2018) have used thresholding, HSV transformations, and morphological opening/closing operations to detect the faults in fabrics. They have proposed an on-line fault detection system. Stains, rips and tears, pencil scratches have been tried to detect in their study.

Fabric images have been examined using morphological filters in the study of Mak et al. (2009). First, the features are extracted from non-defective images using Gabor wavelet network. Then, the input image is compared with these extracted features. If the image has the same background as any non-defective image previously introduced into the system, whether there is a defect in the image is investigated. In the study, experiments have been carried out on offline and real-time systems. 97.4% success has been achieved in offline systems and 96.7% success in real-time systems.

Chan et al. (Chan & Pang, 2000) have used central spatial frequency spectrum to classify the defects of double yarn, missing yarn, webs or broken fabric, and yarn densities variation using seven features extracted from the images. They have obtained very good results in this study.

The difference of offset Gaussian (DOOG) filter has been used in the study of Bagkur (2013). Testing phase has been carried out on 32 images. The system cannot find only one defect in a fabric image containing six different errors, but it correctly detects and recognizes the remaining five defects.

The studies performed by Faouzi et al (2014), Hamdi et al. (2018), Thorave & Biradar (2014), Campbell et al. (1999) are the studies using unsupervised machine learning systems. Faouzi et al. (2014) use fuzzy c-means (FCM) to the parameters of straight-line ratio, ratio of dark areas, and gap ratio. Fabrics with the defects of missing warp, missing weft, oil stains, and holes have been used for experiments. Hamdi et al. (2018) have developed a system that automatically defects the defects in patterned fabrics without any training steps. Images are divided into non-overlapping blocks after the determination of periodic patterns in fabric images. K-Means algorithm is applied to median values of blocks for classifying them as defected or non-defected. The success rate of the system reaches 95%. Thorave & Biradar (2014) remove the noise effect using median filter, and apply the K-Means algorithm. Their algorithm has low computation time. In this study, they obtain a success rate of 96%. Model-based clustering method is used to detect linear defects in knitted and woven fabrics in the study of Campbell et al. (1999). Bayes Information Criterion (BIC) based model has been selected. The developed method gives the best and effective result compared to the methods of thresholding, binary image rotation, and background subtraction.

Although there are many studies for the fabrics of plain, woven and knitted, there are very few studies done for patterned and colored patterned fabrics. Oni et al. (2018) have examined the studies for detection of these types of fabrics in the literature. Seven features have been extracted from the images using frequency spectrum analysis in the study of Yu et al. (2005). Back propogation neural networks have been used for color classification. In the study of Li et al. (2012), energy-based local binary pattern has been used for the images in L*a*b* space. Habib et al. (2013) use back propogation in their study. Autocorrelation function and gray level co-occurrence matrix have been used in the study of Zhu et al. (2015).

Abdellah et al. (2014), Dongli et al. (2013), Ghosh et al. (2011) have been developed systems using support vector machines (SVM).

In the study of Abdellah et al. (2014), necessary parameters are obtained by using genetic algorithm to reach the most suitable SVM classifier also in case of a small number of sample data in hand. A classification is performed with the SVM technique using the geometric properties of the defects. The images with the defects of missing yarn, spot, hole, and oil stains have been used for experiments. Defects have been recognized with the success rate of 94.84%.

Dongli et al. (2013) combine gabor filter and SVM. Optimal parameters for SVM have been selected using genetic algorithms. The developed system is effective in identifying and classifying common monochrome cloth defects. The success rate for identifying defects is up to 94%.

Numerous studies in the area of fabric defect inspection use neural networks (NN) (Kuo & Lee, 2003; Huang & Chen, 2001; Kumar, 2003; Jmali et al., 2014; Rebhi et al., 2015; Büyükkabasakal, 2010; Çelik et al., 2014; Hanbay et al., 2015; Hanbay et al., 2017; Yu et al., 2005; Habib et al, 2013; Behera & Mani, 2007).

Kuo & Lee have used a feedback neural network using three features extracted from the defects (maximum length, maximum width and grey level of defects) (Kuo and Lee, 2003). Classifier has been trained with four classes: weft lacking, warp lacking, hole, oil stain. Success rate of classification for the defects of weft lacking and warp lacking is up to 95%, and it is up to 100% for the defects of holes and oil stains.

Huang & Chen have developed a neural network based fuzzy system (Huang and Chen, 2001). 144 gray level images have been used for classification of eight types of defects (double ends, double picks, missing end, missing pick, hole, light filling bar, cobweb, oil stain). They compare the performances of neural network and fuzzy version of neural network. The performance of fuzzy neural network is superior to neural network according to their experiments.

Kumar has used a feedforward neural network to classify the defects in twill and plain-woven fabrics (Kumar, 2003). Three defect types (mispick, netting multiplies, thin bar) for twill woven fabrics and four defect types (double-weft, thin bar, broken ends, slack pick) for plain woven fabrics have been tried for recognition.

Jmali et al. have used a single layer neural network to classify the defects of warp threads, weft threads, oil stains, and hole (Jmali et al., 2014). 45 input images have been used for experiments. Test have been performed using the regression curve. They have obtained high performance from the developed system.

Rebhi et al. use back-propogation neural network using the five features extracted from discrete cosine transforms (DCT) of H-images (homogeneity images obtained from input images) (2015). These features are vertical energy, horizontal energy, diagonal energy, energy mean, and energy standard deviation. They use a dataset provided by a textile industry in Tunisia. There are 89 images (13 non-defected, 76 defected) in the dataset.

Büyükkabasakal (2010) has been aimed to recognize the defects in the fabrics by extracting the feature vectors of images using principal component analysis (PCA). Defects have been classified using neural networks. In the study, the system has been trained for the defects of warp leakage, warp tip, abrage and weft leakage. In the testing phase, 40 images have been used includig these defects. According to the experiments, a success of 83% has been achieved.

In the study of Çelik et al.(2014), a system that detects fabric defects in offline and real-time systems is recommended using linear filters and morphological processes. Tests have been performed for five different defect types (missing warp, missing weft, dirty yarn, hole and knot). The defects detected have been classified using an feedforward neural networks with an average success rate of 96.3%. In this study, higher success rate has been obtained real-time systems than offline systems.

Hanbay et al (2015) use the methods of co-HOG (Co-occurrence Histograms of Oriented Gradients), wavelet transform and gray level co-occurrence matrix to extract the features and use artificial neural network to train the system in their study. 9165 test images (3242 defected, 5923 non-defected) have been used for the experiments. When using the wavelet transform, defects are classified with a 90% success rate. Also, it is seen that the cost has decreased considerably.

Hanbay et al. (2017) apply neural networks to the seven features extracted from the images using fourier frequency spectrum. Experiments have been performed for both off-line and on-line systems. The number of sample fabrics used in the experiments is 11.000. The system recognizes non-defected fabrics and fabrics with the needle defects with 100%

success rate. It recognizes the lycra defects with 86% success rate, while it recognizes the yarn defects with 92% success rate.

The methods based on CNN have become popular in recent years.

Zhu et al. (2020) optimize DenseNet which is a CNN algorithm. They combine the new method with a new hardware for fabric defect detection.

Karlekar et al. (2015) use wavelet decomposition and different preprocessing operations to obtain segmented defect.

Chang et al. (2018) develop a new method for patterned fabrics. Fabrics are divided into lattices including periodic patterns. Then, the lattice containing the defect is detected.

Wei et al. (2019) make a combination of compressive sampling theorem with CNN. The new method is more effective compared to traditional methods and performs well in small data sets.

Wang et al. (2018) develop a CNN based system which have two major parts. One is global frame classification part. It classifies the image samples using background features. The other is sub-frame detection part. The part checks whether each sample contains defected areas or not. The second part uses output of the first part for checking operation.

Zhao et al. (2020) develop a CNN model based on visual long-short-term memory (VLSTM).

Guan et al. (2019) use VGG (Visual Geometry Group) model for CNN. Simon & Uma (2020) extract the features using CNN and perform the classification using SVM. In this study, SVM is compared to DenseNet201, ResNet50, ResNet101, Inceptionv3 and AlexNet.

Şeker et al. (2016) use autoencoder algorithm as a deep learning algorithm to detect the fabric defects. It is the first study which uses transfer learning in the area of fabric defect. They aim to increase the feature extraction achievement. 88% accuracy rate has been obtained in this study which classifies the fabric images as defected-non-defected. Classification success rate is higher in the defect types of holes and stains while it is lower in other defects.

There are some challenges in CNN. For this reason, there are studies developed to solve the challenges of CNN-based studies. CNN algorithms have long execution times. Therefore, some studies aim to shorten this period (Liu et al., 2018; Wei et al., 2018). Wei et al. (2018) suggest VGG based RCNN to speed up the detection process. CNN algorithms cannot be successful for small sample sizes. So, developing CNN algorithms in this area has been aimed in some studies. Li et al. (2019) develop Wide-And-Compact Network (WACNet). Wei et al. (2019) develop CS-CNN (compressive sampling) theorem. They make a comparison between the performances of CS-CNN, CNN, KNN, multi-layer perceptron (MLP) and SVM. In the study of Şeker (2018), this disadvantages has been overcome by using transfer learning. A pre-trained model AlexNet is used. Fabrics are tried to be divided into two different classes as defective and defect-free. There is not any preprocessing operation performed before deep learning in some studies, while learning process is performed after preprocessing operations in others (Jing et al., 2019).

In the studies conducted, two main factors that cause recognition defects are mentioned (Xin et al., 2009). The first is the quality of the images. An environment where light is reflected and motor vibration affects success. The second is structurally large curves or overlaps in the fabric.

New classification approaches are tested on the most known datasets for example Iris, Adult, Wine, and Breast Cancer Wisconsin … (Uci Repository). This makes it easier to compare with the other studies developed. Thus, which method is more effective can be said easily. However, there are two known databases called as Parvis (Italian Textile Institute) and Tilda (Workgroup on Texture Analysis of DFG) existing in the area of automatic fabric defect control. The Parvis database is private, and the Tilda database is now free. Due to lack of public and free datasets for fabric defects, it is seen that different data sets are used in each study. In addition, since most of the data sets used in the studies have few examples, the effectiveness of the developed systems are discussed. A fabric database is proposed in the study of Silvestre-Blanes et al.(2019). The database consists of 245 images (105 defected, 140 non-defected). There is 12 different defect types. All images are captured with the size of 4096x256, and they are converted to the size of 256x256. The database is available on the Internet.

In this study, the public part of the Tilda dataset is used. The studies that will be mentioned next are among the studies using Tilda dataset. While some of these studies distinguish the images as defected/un-defected, some perform classification according to defect types. In addition, some of these studies use not only the Tilda dataset, but also other datasets. The findings shown below are merely the results they acquired using the Tilda dataset.

Gabor wavelets and Principal Component Analysis (PCA) have been used in the study of Basturk et al. (Basturk et al., 2007). Gabor wavelets have been used for feature extraction, and PCA is used to reduce the dimension of features extracted. Experiments are performed for four defect types (e1, e2, e3, e4) of the Tilda dataset. It is obtained that defects have been detected accurately.

Salem & Nasri (2011) compare the performances of local binary pattern and gray level co-occurrence matrix. SVM has been used as a classifier to detect the defects such as un-defected, unrelated corpus, broken end, hole, kink, oil satin, missing weft. According to the experiments, local binary patterns give more effective results in terms of time and accuracy.

In the study of Murino et al. (2004), features have been extracted from the images using the methods of histogram, co-occurrence matrix and shape descriptor. They are classified by SVM. Databases of Parvis (1117 elements) and Tilda (1333 elements) have been used for experiments. Defects in Parvis database are recognized with a 99.11% success rate while the defects in Tilda database are recognized with a 92.87% success rate.

Jing et al. (2015) use the distance matching function to determine the frequency of repetition of the pattern in the patterned fabric, and two properties calculated using the regular band are determined as fabric defects. Defects that disrupt the regularity of the fabric can be detected by this method. In this study, fabrics containing the defects of broken end, thick bar, thin bar, hole, multiple netting, and knots in the Tilda database have been used and the defect detection rate is 96.5% on average.

Kure et al. (2017) investigate homogeneity in fabric images. They use local neighborhood analysis to measure homogeneity. Experiments have been performed for six defect types (holes, slack end, loose weft, drop stitch, broken end, missing plush loop) in Tilda database. A comparison between wavelet transform, gabor transform and the system developed has been made. According to the experiments, the cross validation accuracy of the system is higher than the others (96.40%). The disadvantage of the study is that it only classifies the images according to whether there are defects or not.

Başıbüyük et al. (2008) have achieved 97% success by applying particle filtering in c1 group of Tilda dataset. Images are divided into sub-windows. Using randomly selected un-defected images, AR coefficients are calculated.

Bissi et al. (2013) use gabor filter bank and principal component analysis (PCA), and test the performance using the parts of c1-r1 and c1-r3 of Tilda. This study, with more than 98% success.

After partitioning the images into blocks, feature vectors extracted from each block are used in a regression based method which is named PG-LSR in the study of Cao et al. (2017).

Liu et al. (2019) use ELM (Extreme Learning Machine) method after extracting the features from segmented defects in fabrics. The accuracy of the system they have developed is 94.5%.

Jing et al. (2019) use convolutional neural network (CNN) after division the images to patches. 97.48% classification accuracy rate has been achieved for Tilda. They choose six classes of Tilda (un-defected, Holes, Carrying, Scratch, Stain, and Knots).

Jeyeraj et al. (2019) use a transfer learning based CNN algorithm called AlexNet. They obtain high accuracy rate (96.55%).

Sezer et al. (2007) tries to classify the Tilda dataset using Independent Component Analysis (ICA). In the study, while relatively better results were obtained in c1-r1 and c1-r3 than Tilda's other groups, the result of c3-r3 was not shared due to poor success. This system is very sensitive to external factors.

To detect defects on fabrics with complicated textures, Qu et al. (2016) developed a system based on dual-sale over-complete dictionary. Experiments using Tilda and their database belongs to them showed that this system performed well (96.5%). The disadvantage makes it unsuitable for online fabric assessment. In the study that used the classifiers of KNN and SVM to compare the performances of optimized and non-optimized Haralick parameters, it was discovered that optimized Haralick parameters outperformed (99.00%) non-optimized Haralick parameters (Chandra et al., 2016).

Kaynar et al. (2017) make a comparison between Local Binary Pattern (LBP) and Gray Level Co-occurrence Matrix (GLCM). Artificial Neural Network (ANN) is used for classification after feature extraction is performed using one of LBP and GLCM.

Makaremi et al. (2018) compare several classifiers (SVM, Multilayer Perceptron (MLP), Adaboost, KNN) to the modified version of LBP (MLBP). The study includes the results of a 596-image dataset developed by integrating four independent data sets. It is concluded that the MLP algorithm gives better results (97.31%).

A system that uses a pyramid histogram of edge orientation gradients (PHOG) and a support vector machine (SVM) is recommended in the study of Cuifang et al. (2020). In this study, in which the effect of different block sizes is also examined, it is seen that the detection rate increases as the size increases. They extract features using pyramid histogram of oriented gradients (PHOG) and perform classification using support vector machine (SVM). The performance of PHOG is superior to the performances of scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG).

When looking at the studies using the Tilda data set, it is clear that the analyzed studies attempted to identify patterned or non-patterned materials within themselves. Among the studies examined, there was no study that performed classification by bringing together the patterned and un-patterned fabrics in the Tilda data set. In this study, patterned and un-patterned fabrics were brought together and the fabric samples were tried to be separated as defected/un-defected.

CHAPTER THREE

DATA MINING

3.1 The Definition of Data Mining

Data is meaningless unless it is processed. Data mining is a technique for converting raw (unprocessed) data into information. Thanks to data mining, previously unknown valid and applicable information is obtained from the data stack (Baykal, 2006). It is a multidisciplinary tool consisting of fields such as statistics, machine learning and database management (Jackson,2002). It is not a solution in itself; it is a tool that aids in the decision-making process and gives the knowledge needed to address the problem (Baykal, 2006).

3.2 Data Mining Application Areas

Data mining has applicability in various fields such as banking, economy, health, security for various reasons. Examples in these fields are given below:

- Database analysis,
- Risk analysis such as optimizing service delivery,
- Decision support systems,
- Market research such as identification of similarities between customers,
- Rapid diagnosis and treatment in the health sector

3.3 Data Mining Models

In this study, data mining models are examined under four main headings according to their functions (Figure 3.1).

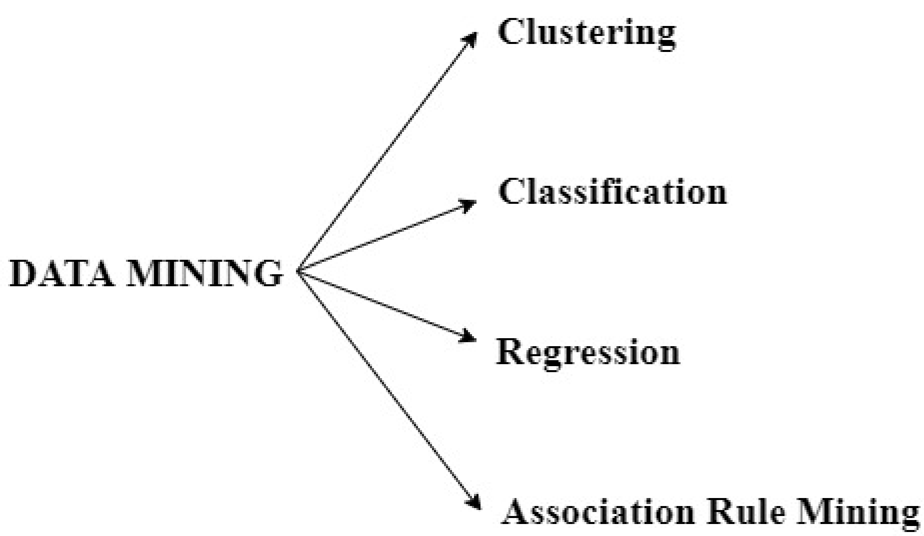


Figure 3.1 Data mining models

3.3.1 Clustering

Clustering is the presence of similar elements in the same cluster and dissimilar elements in different clusters. In clustering, there are no clusters initially. Clusters are found based on the data.

More data exists every day compared to the previous days. If they can be evaluated, more data means more opportunities. Having the correct clusters depends on the clustering algorithm which is used (Figure 3.2).

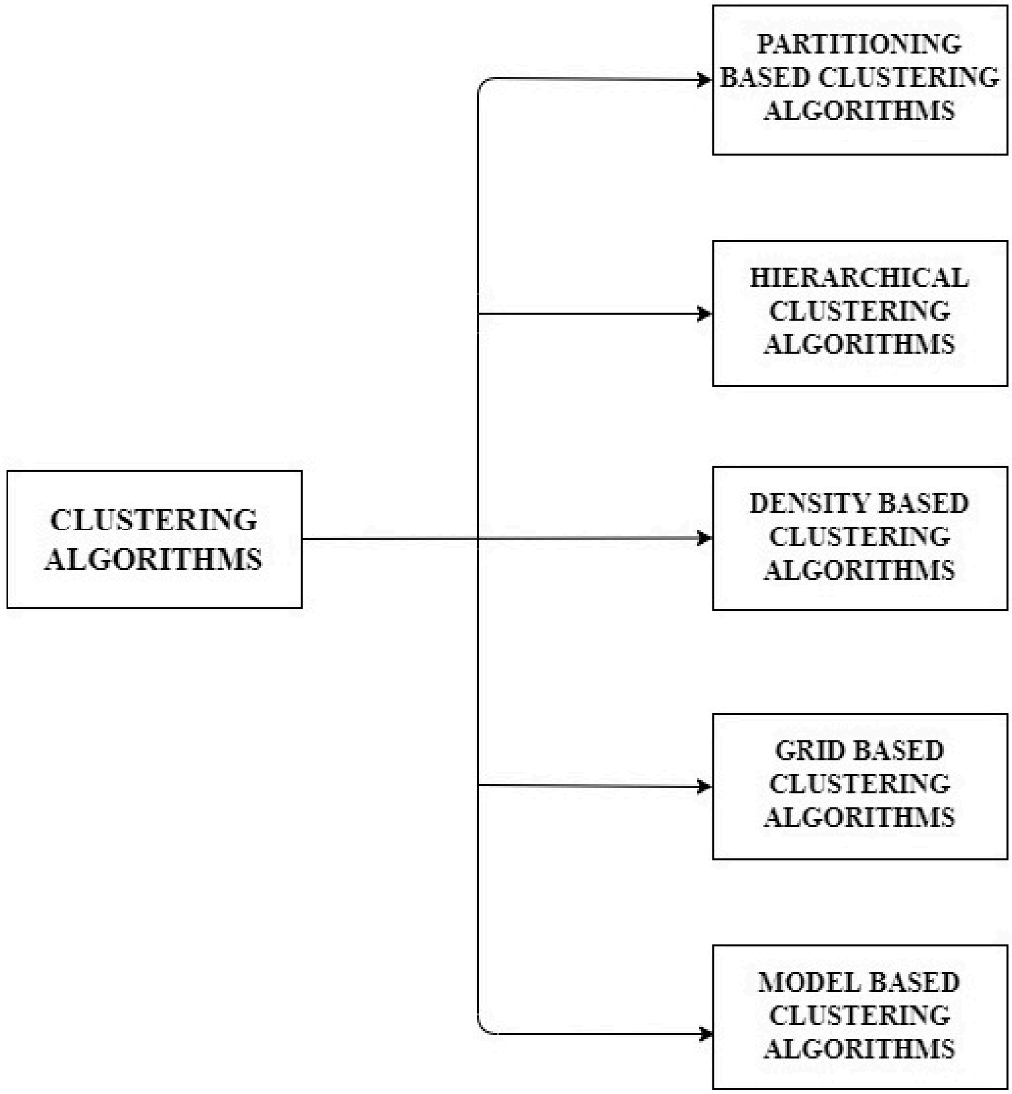


Figure 3.2 Clustering algorithms

There are a variety of clustering algorithms, which are grouped into five classes based on the methodologies they employ. The partitioning-based clustering algorithms are the first group. Initially, one cluster encompassing all objects is handled in this group. Objects are repeatedly grouped into clusters, from the roots to the leaves. K-Means (MacQueen, 1967), K-Medoids (Kaufman & Rousseeuw, 1987), and K-Modes (Huang, 1998) are the most prominent partitioning-based clustering techniques.

The second group is hierarchical clustering. The structure is based on a tree. It may be divided into two types: agglomerative and divisive. The structure is integrated from the leaves to the root in an agglomerative approach. The structure is partitioned from the roots to the leaves in a divisive approach. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang et al., 1996), CURE (Clustering Using REpresentatives) (Guha et al., 2001), ROCK (RObust Clustering using linKs) (Guha et al., 2000), and Chamelon (Karypis et al., 1999) are some well-known hierarchical based clustering algorithms.

The density-based clustering is the third option. Objects are divided into three categories: core, border, and noise. For each object, the neighborhood is taken into account. Unlike previous algorithms, they can find clusters of various forms. DBSCAN (Ester et al., 1996) and OPTICS (Ordering Points To Identify the Clustering Structure) (Ankerst et al., 1999) are two of the most prominent density-based clustering algorithms.

Grid-based clustering is the fourth group. The grid layout is used to create clusters (Sajana et al., 2016). In grid-based algorithms, time complexity is unrelated to the quantity of data. As a result, clustering algorithms of this type are quick. STING (STatistical INformation Grid) (Wang et al., 1997), CLIQUE (CLustering In QUEst) (Agrawal et al., 2005), and WaveCluster (WAVElet based CLUSTER) (Sheikholeslami et al., 1998) are the most common grid-based clustering algorithms.

Model-based clustering is the fifth and final group. Some techniques are used to associate data items with one another. This type of algorithm employs two techniques: neural networks and statistical methodologies. The EM (Expectation-Maximization) technique is the most widely used model-based clustering algorithm (Dempster et al., 1977).

3.3.2 Classification

One of the data mining models is classification. A predefined set which is labelled exists in a classification problem. When a new sample comes in, it is estimated to which class this sample belongs. Here, classification algorithms are examined under five main headings, which are k-nearest neighbors, support vector machines, decision tree, naïve bayes, and neural networks (Figure 3.3). However, there are also classification algorithms other than those mentioned here.

The main purpose of k-nearest neighbor (KNN) is to classify new objects using the trained data. The k parameter specifies the number of data to evaluate while finding a new sample's class label. The label of the sample is assigned to the class in which the majority of the k closest neighbors to the sample are.

Support vector machine algorithms (SVM), is capable of sorting data into linear in two-dimensional space, planar in three-dimensional space, and hyperplane in multidimensional space (Cortes and Vapnik, 1995). Many hyperplanes may exist between the classes. This method aims to find the best hyperplane that divides the classes.

A decision tree is a decision structure that has the shape of a tree. It does classification in which the training procedure is done inductively from known classes of sample data (Su and Zhang, 2006). A decision tree is a structure that is used to divide vast volumes of data into little groups of data using simple decision-making stages.

Naive Bayes classifier is another algorithm of classification task. It is based on Bayes theorem. The Bayes theorem considers two occurrences (X and Y). P(X|Y) is the probability of event X occurring when the event Y occurs (3.1). P(Y|X) is the probability of event Y occurring when the event X occurs. P(X) is the probability of event X occuring. P(Y) is the probability of event Y occuring.

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.1}$$

Artificial neural networks (ANNs) are classification algorithms in which learning, one of the basic functions of the human brain, is used. The inspiration for artificial neural networks is the human brain. The challenge of mathematically modeling the human brain's learning process has shown how the human brain learns.

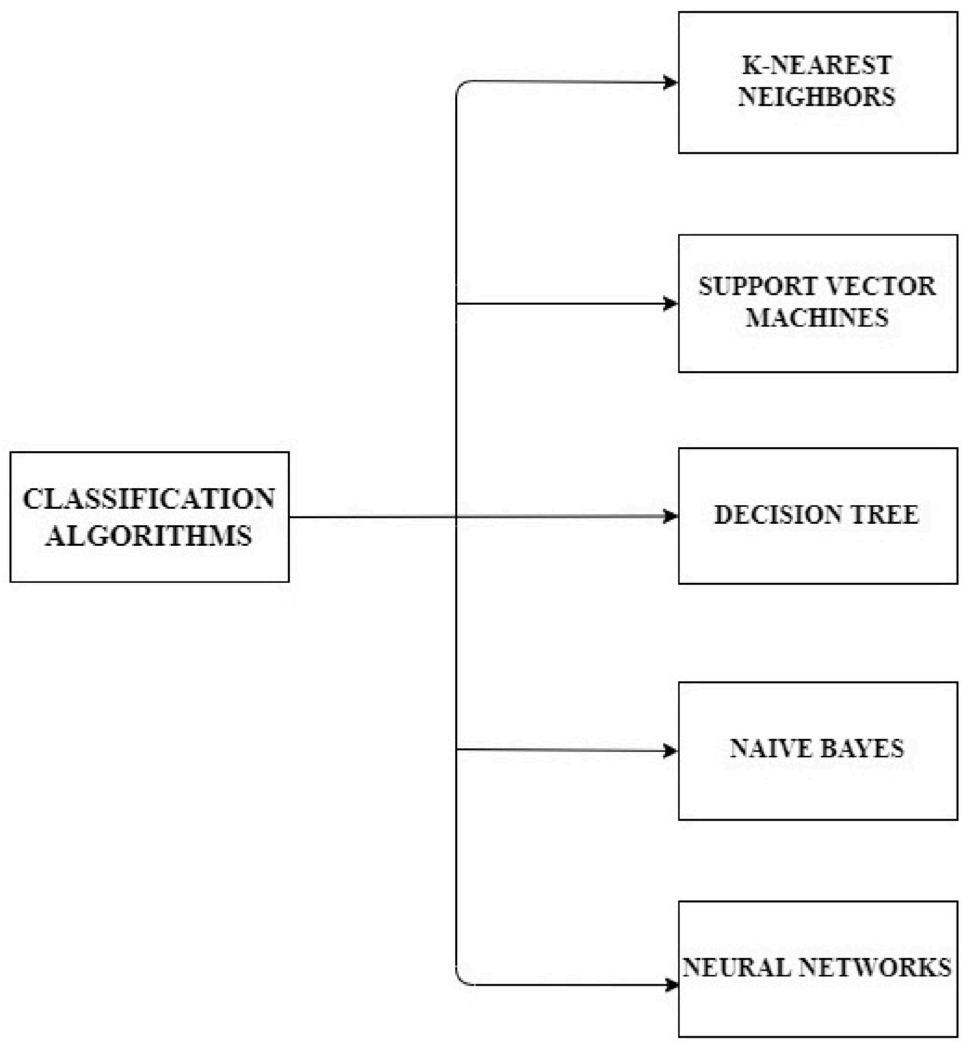


Figure 3.3 Classification algorithms

3.3.3 Regression

Regression is a data mining model in which the target variables of test samples are estimated using a prediction model built by a training set. It is similar to classification. Regression is used to predict continuous values, whereas classification is used to predict categorical values (Han et al., 2011).

3.3.4 Association Rule Mining

Another data mining model is association rule mining (ARM) which seeks to find frequently recurring patterns (co-occurrences), correlations, or intriguing relationships between variables in a big collection of data using certain interestingness criteria.

Two steps are followed when finding association rules (Zaki, 1999):

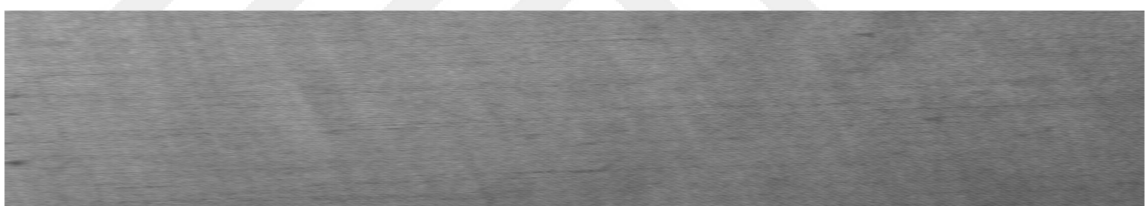
1- Frequently repeated items are found. Each of them is repeated at least as many times as the minimum number of supports required.

2- Items that are regularly repeated produce strong association rules. These rules must have a minimum level of support and confidence.

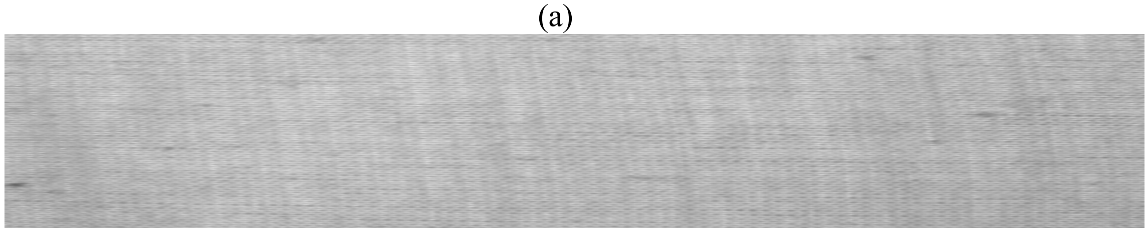
CHAPTER FOUR

DATASET

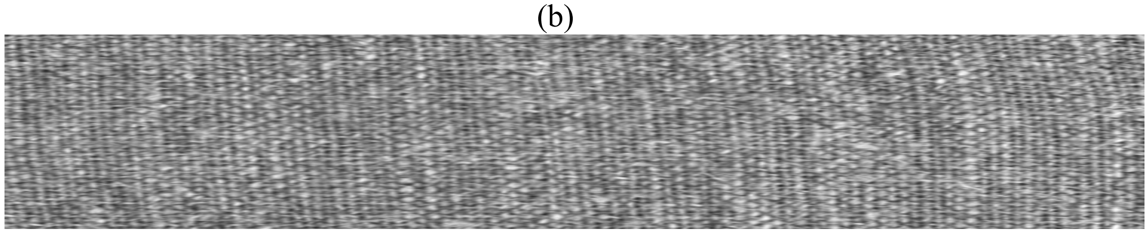
The study is tested on the public part of the Tilda dataset. Tilda dataset have images with 768×512 pixels (Figure 4.1). Images have been resized before processing operations. The dataset consists of two folders such as cd1 and cd2 (Figure 4.2). Both folders consists of four groups (Table 4.1). While four groups in cd1 (c1r1, c1r3, c2r2, c2r3) consist of un-patterned fabric samples, the other four groups in cd2 (c3r1, c3r3, c4r1, c4r3) contain patterned samples. Each group is divided into eight subdirectories (e0, e1, e2, e3, e4, e5, e6, e7), each of which contains 50 samples. Defect-free samples are found in e0, while samples with various types of defects are found in other subdirectories. The un-defected fabric is the fabric in which the texture repetition is not broken.



(a)

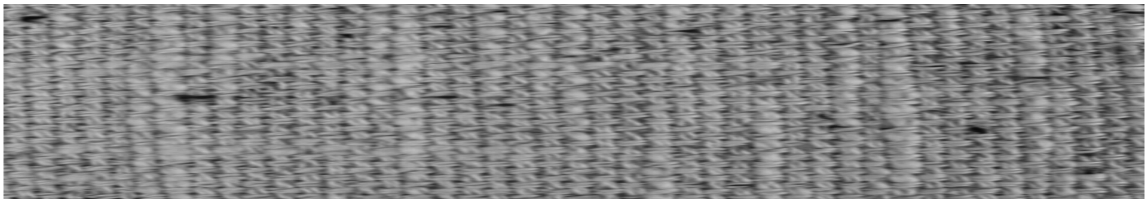


(b)

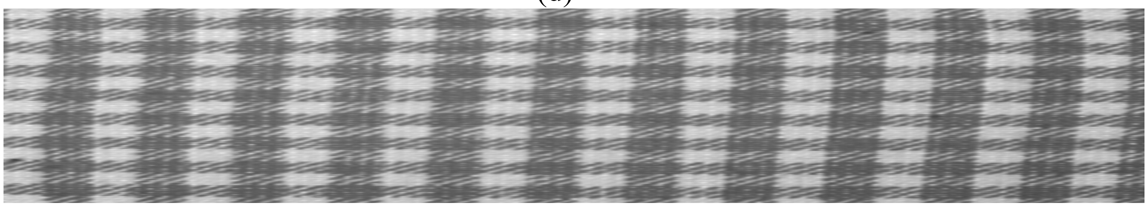


(c)

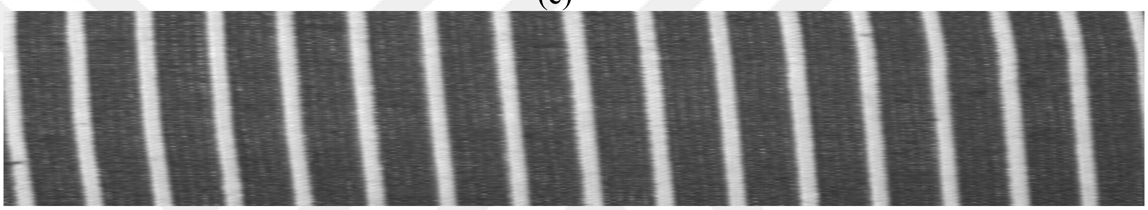
Figure 4.1 Tilda fabric samples (a) c1r1 (b) c1r3 (c) c2r2 (d) c2r3 (e) c3r1 (f) c3r3 (g) c4r1 (h) c4r3



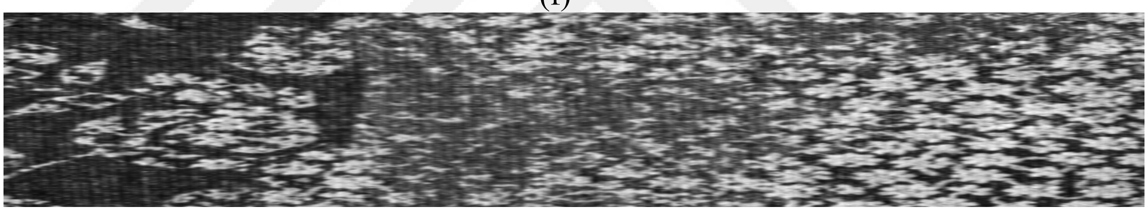
(d)



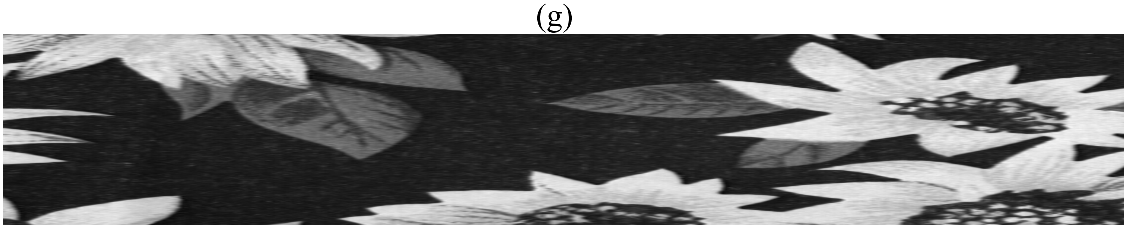
(e)



(f)



(g)



(h)

Figure 4.1 Continues

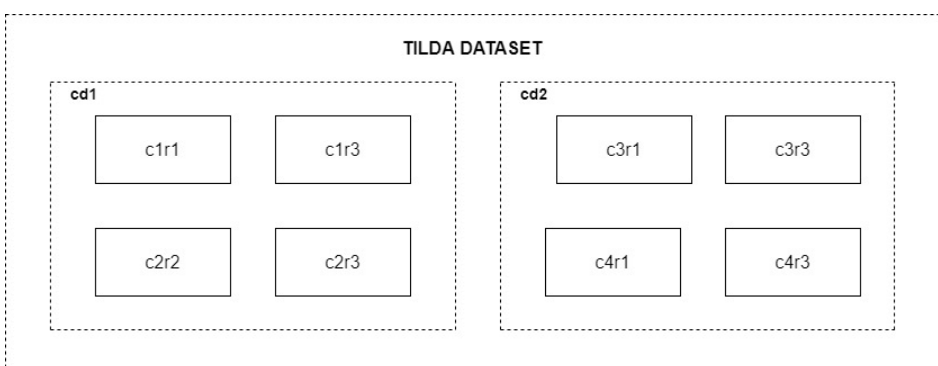
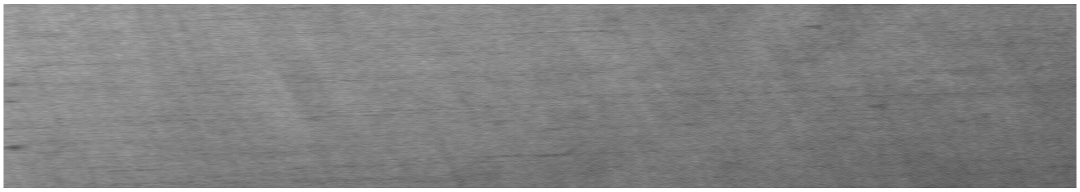


Figure 4.2 Tilda dataset

Table 4.1 About Tilda

Un-patterned Fabrics		Patterned Fabrics	
Group	**Number of Images**	**Group**	**Number of Images**
c1r1	400 (50 un-defected, 350 defected)	**c3r1**	400 (50 un-defected, 350 defected)
	Classes		**Classes**
	e0, e1, e2, e3, e4, e5, e6, e7		e0, e1, e2, e3, e4, e5, e6, e7
c1r3	400 (50 un-defected, 350 defected)	**c3r3**	400 (50 un-defected, 350 defected)
	Classes		**Classes**
	e0, e1, e2, e3, e4, e5, e6, e7		e0, e1, e2, e3, e4, e5, e6, e7
c2r2	400 (50 un-defected, 350 defected)	**c4r1**	400 (50 un-defected, 350 defected)
	Classes		**Classes**
	e0, e1, e2, e3, e4, e5, e6, e7		e0, e1, e2, e3, e4, e5, e6, e7
c2r3	400 (50 un-defected, 350 defected)	**c4r3**	400 (50 un-defected, 350 defected)
	Classes		**Classes**
	e0, e1, e2, e3, e4, e5, e6, e7		e0, e1, e2, e3, e4, e5, e6, e7
SUM	**1600 (200 un-defected, 1400 defected)**	**SUM**	**1600 (200 un-defected, 1400 defected)**

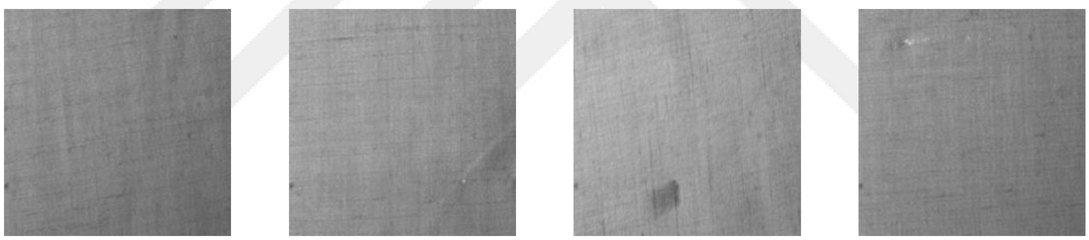


(a)

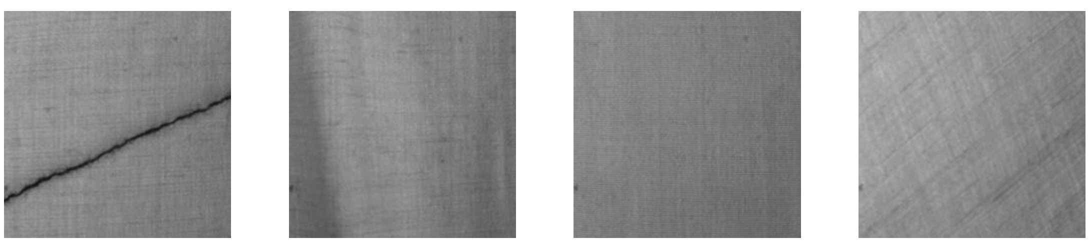
(b)

Figure 4.3 Resizing from (a) 768×512 to (b) 224×224

Resized samples for each group are given in figures 4.4-4.11.



(a) **(b)** **(c)** **(d)**



(e) **(f)** **(g)** **(h)**

Figure 4.4 Resized samples of c1r1 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

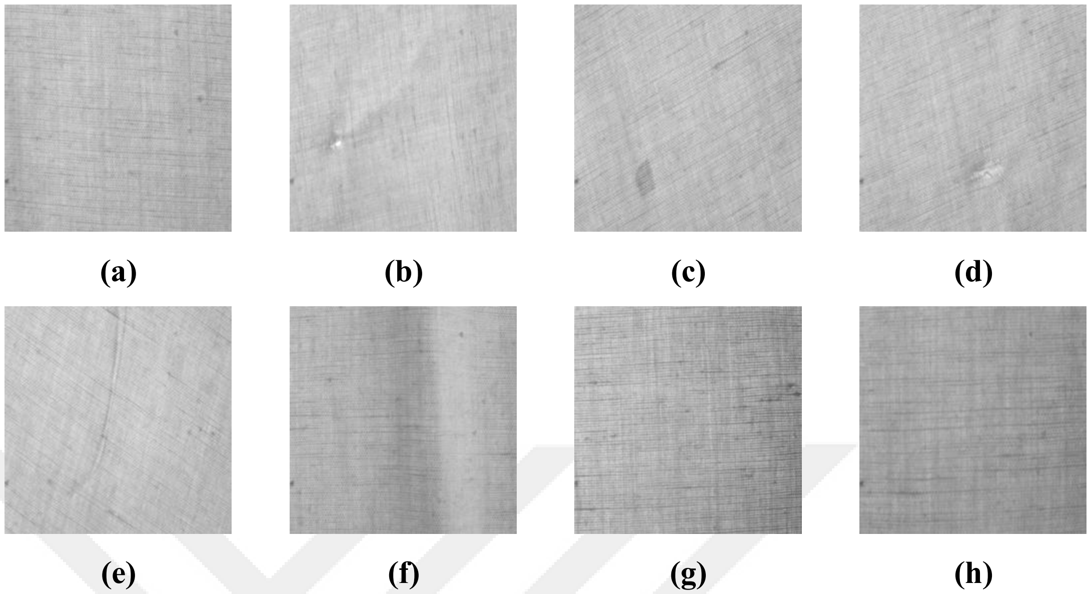


Figure 4.5 Resized samples of c1r3 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

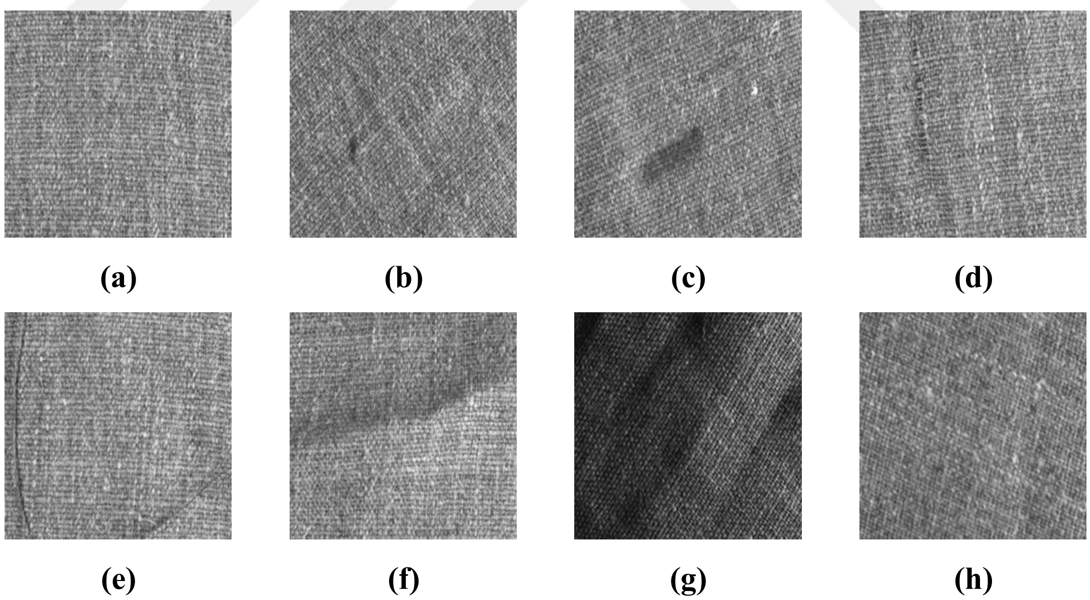


Figure 4.6 Resized samples of c2r2 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

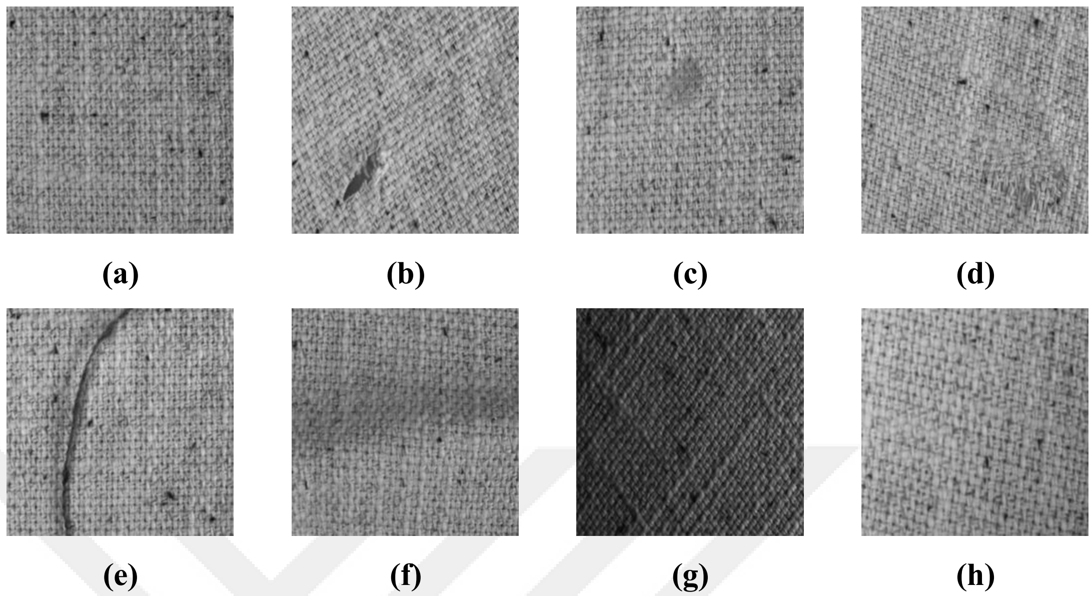


Figure 4.7 Resized samples of c2r3 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

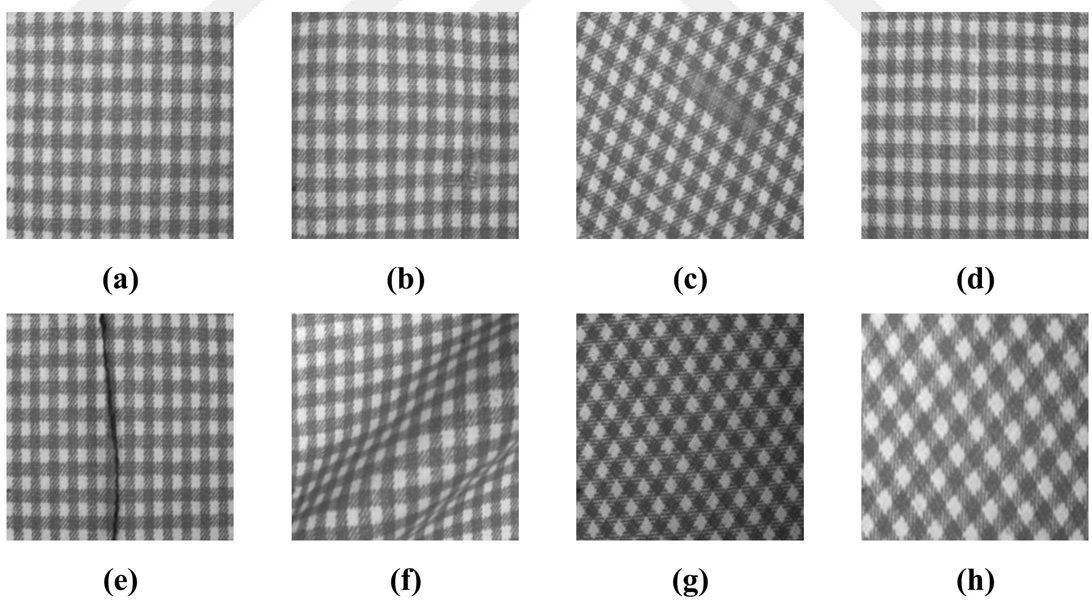


Figure 4.8 Resized samples of c3r1 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

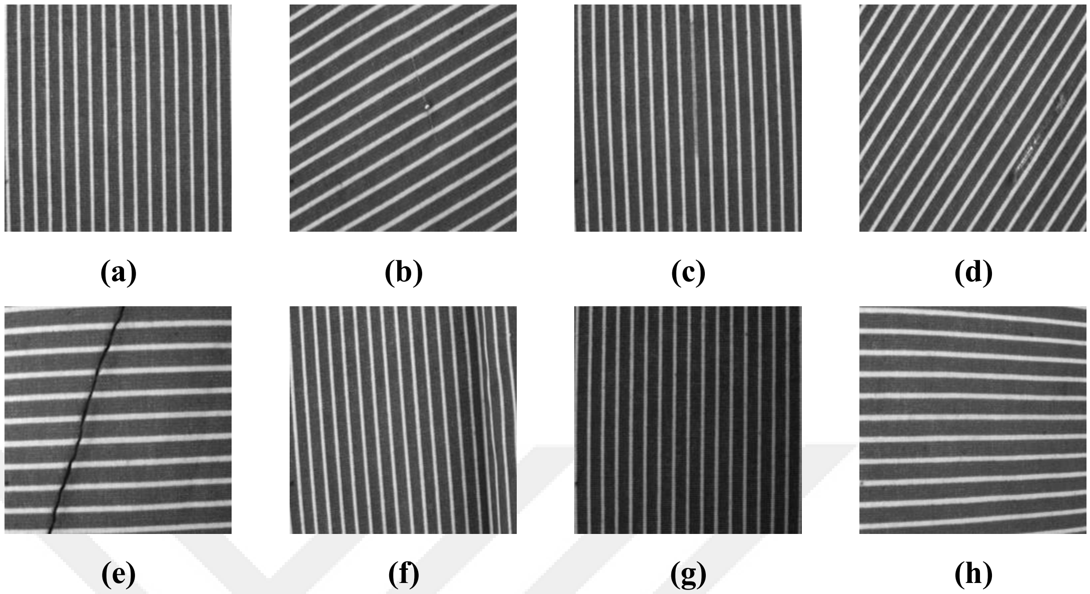


Figure 4.9 Resized samples of c3r3 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

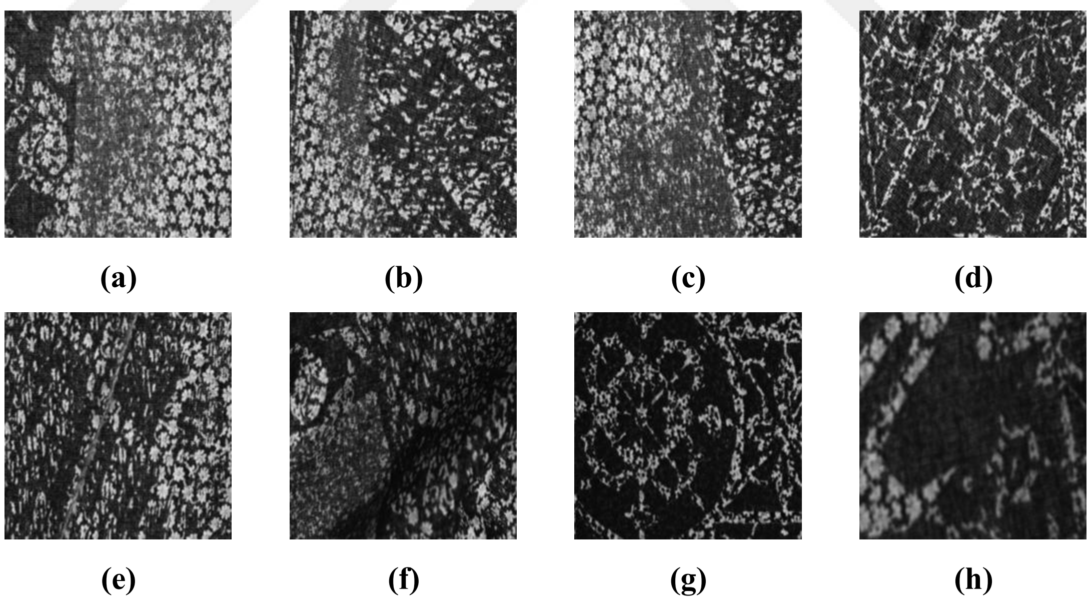
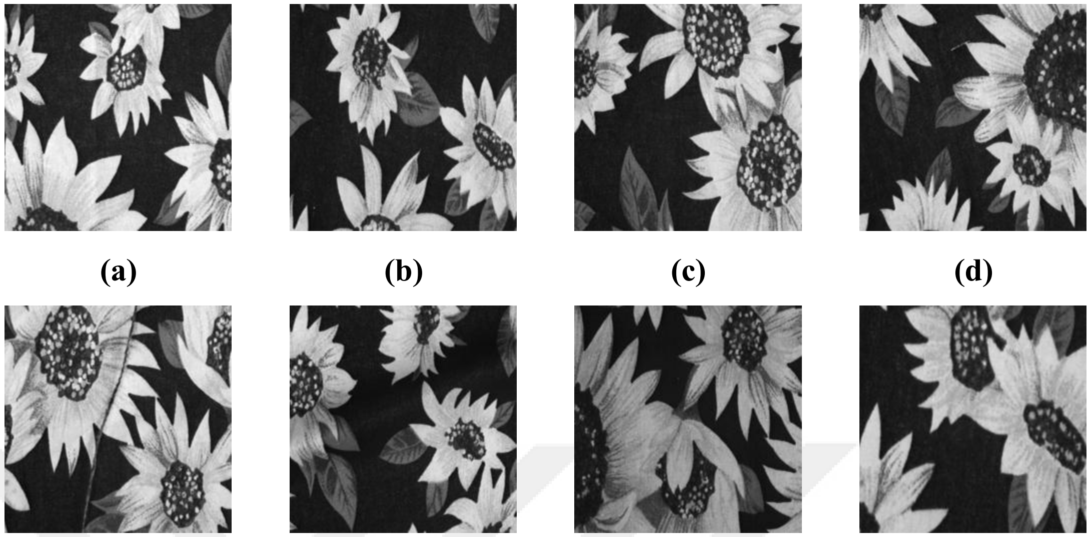


Figure 4.10 Resized samples of c4r1 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.11 Resized samples of c4r3 (a) e0 (b) e1 (c) e2 (d) e3 (e) e4 (f) e5 (g) e6 (h) e7

CHAPTER FIVE

BACKGROUND FOR FEATURE EXTRACTION

5.1 Deep Learning Based Feature Extraction

Deep learning is a sub-group of machine learning and is a multi-layered approach (Türkoğlu et al., 2021). In deep learning, raw data is given to the network and both feature extraction and learning process are performed by using all images given as input. Deep learning uses many layers of nonlinear processing units for feature extraction and conversion.

Deep learning has been used in several research areas in the literature. Robotics, image processing, video processing, signal processing, object recognition, and the military sector are just a few industries that apply deep learning. For feature conversion and extraction in deep learning, several layers of nonlinear processing units are employed. The output from each previous layer is used by each subsequent layer as input.

5.1.1 Convolutional Neural Network (CNN)

By increasing the number of layers in artificial neural networks, a variety of deep learning architectures have been developed. Among them is the Convolutional Neural Network (CNN) (Figure 5.1).

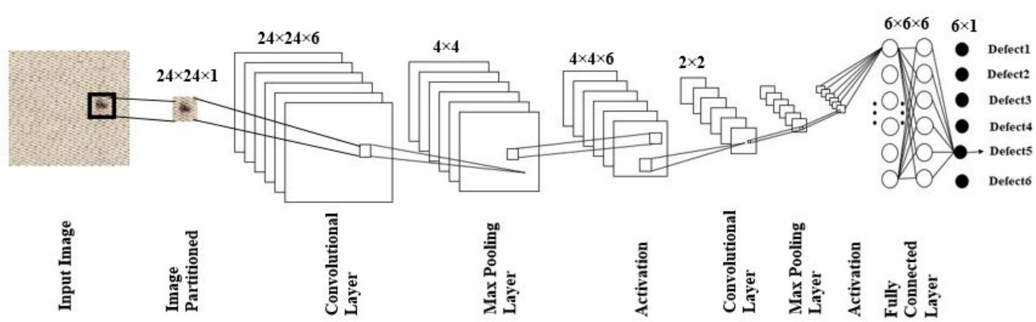


Figure 5.1 A CNN architecture with seven layers for six classes

CNN is a multilayer artificial neural network model created specifically for computer vision applications. It is an approach that provides higher performance compared to other classification methods. On the other hand, CNN's drawback is the requirement for powerful hardware resources. The basic layers of CNN based models can be listed as in Table 5.1.

Table 5.1 Basic layers of CNN

Layer	**Description**
Convolution	A new image is created in convolution layer by extracting more specific features in the image. Filters have smaller dimensions than input data.
Activation Function	The activation functions of convolutional neural networks are crucial. The most often used activation function is Relu. Negative values in the input data are set to 0 in this layer, allowing the network to learn faster.
Pooling	Pooling is a process used to reduce sizes in deep learning models to avoid overfitting and reduce time complexity.
Fully Connected Layer	A fully connected layer is a one-dimensional matrix that is connected to all of the neurons in the layer before it. This layer is utilized to optimize class scores and is usually found near the end of the CNN architecture.

Table 5.1 Continues

Layer	Description
Classification	The final layer of CNN models is this one. This layer is responsible for classification process. In this layer, the softmax classifier is often employed, which gives probabilistic values between 0 and 1 for each class based on the architectures. As a result, the class predicted by the model is determined by the highest probability value.

In our study, we tested the performances of four CNN based models such as ResNet18, ResNet50, GoogLeNet, and AlexNet (Figure 5.2).

5.1.1.1 ResNet

ResNet is a pre-trained CNN algorithm (He et al., 2016). It has been trained on more than one million images. ResNet is the abbreviation form of Residual Network. It has different versions such as ResNet18, ResNet50, ResNet101, and ResNet152. They include 18, 50, 101, and 152 layers, respectively. As the number of layers increases, the accuracy rate increases and the execution time also increases. The sizes of input images must be 224-by-224 for ResNet architecture. ResNet18 and ResNet50 have been used in this study.

5.1.1.2 GoogLeNet

GoogleNet (also known as Inception v1) is a pre-trained CNN algorithm like ResNet18 (Szegedy et al., 2015). The sizes of input images must be 224-by-224 for GoogLeNet. It has an architecture consisting of deeper 22 layers with fewer parameters compared to other networks (Tellawi, 2019). Therefore, it gives higher success results in less time.

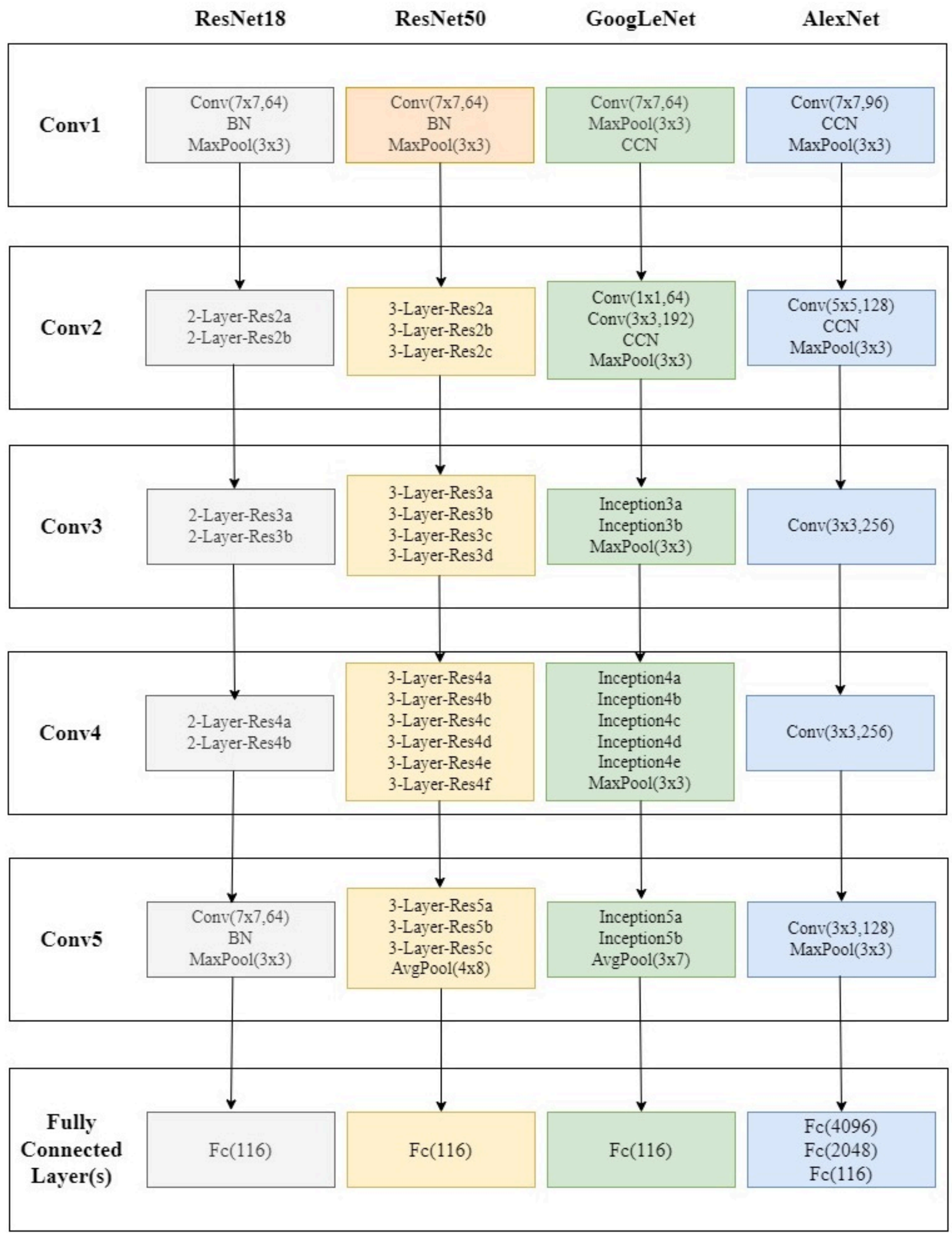


Figure 5.2 CNN models used for our study

5.1.1.3 AlexNet

The AlexNet architecture, which won the ImageNet 2012 competition, is a type of convolutional neural network designed by Krizhevsky (2012). Information about AlexNet analysis is as in Figure 5.3. AlexNet is very similar to the LeNet network. However, it differs from LeNet in that it has more layers. Input dimensions should be 227×227×3 for AlexNet. Local Response Normalization (LRN) and dropout are new additions to this network (Alom et al., 2018).

	Name	Type	Activations	Learnables
1	data 227x227x3 images with 'zerocenter' normalization	Image Input	227×227×3	-
2	conv1 96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]	Convolution	55×55×96	Weights 11×11×3×96 Bias 1×1×96
3	relu1 ReLU	ReLU	55×55×96	-
4	norm1 cross channel normalization with 5 channels per element	Cross Channel Nor...	55×55×96	-
5	pool1 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	27×27×96	-
6	conv2 2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]	Grouped Convolution	27×27×256	Weigh... 5×5×48×128... Bias 1×1×128×2
7	relu2 ReLU	ReLU	27×27×256	-
8	norm2 cross channel normalization with 5 channels per element	Cross Channel Nor...	27×27×256	-
9	pool2 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	13×13×256	-
10	conv3 384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	13×13×384	Weights 3×3×256×384 Bias 1×1×384
11	relu3 ReLU	ReLU	13×13×384	-
12	conv4 2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Grouped Convolution	13×13×384	Weigh... 3×3×192×192... Bias 1×1×192×2
13	relu4 ReLU	ReLU	13×13×384	-
14	conv5 2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Grouped Convolution	13×13×256	Weigh... 3×3×192×128... Bias 1×1×128×2
15	relu5 ReLU	ReLU	13×13×256	-
16	pool5 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	6×6×256	-
17	fc6	Fully Connected	1×1×4096	Weights 4096×9216
18	relu6 ReLU	ReLU	1×1×4096	-
19	drop6 50% dropout	Dropout	1×1×4096	-
20	fc7 4096 fully connected layer	Fully Connected	1×1×4096	Weights 4096×4096 Bias 4096×1
21	relu7 ReLU	ReLU	1×1×4096	-
22	drop7 50% dropout	Dropout	1×1×4096	-
23	fc8 1000 fully connected layer	Fully Connected	1×1×1000	Weights 1000×4096 Bias 1000×1
24	prob softmax	Softmax	1×1×1000	-
25	output crossentropyex with 'tench' and 999 other classes	Classification Output	-	-

Figure 5.3 AlexNet analsis result

5.2 The Proposed Feature Extraction System: Multi-Feature Fusion

Multi-feature fusion technique is proposed in this thesis. The features extracted using Method1 is vector X, and the features extracted using Method2 is vector Y. The feature fusion is obtained by adding the Y vector after the last element of the X vector (5.1). The visualized version of the proposed system is in the Figure 5.4.

$$F = (X, Y) = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) \tag{5.1}$$

Figure 5.5 is the flow chart of the system. Six combinations have been used fort he step of feature fusion. Then, the fused features have been classified using a classifier. Three algorihms such as ensemble learning, k-nearest neighbor, and support vector machine have been tested in the classification step.

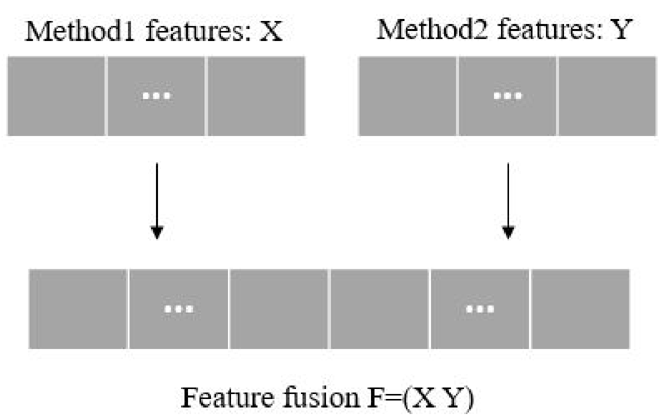


Figure 5.4 The procedure of the method proposed

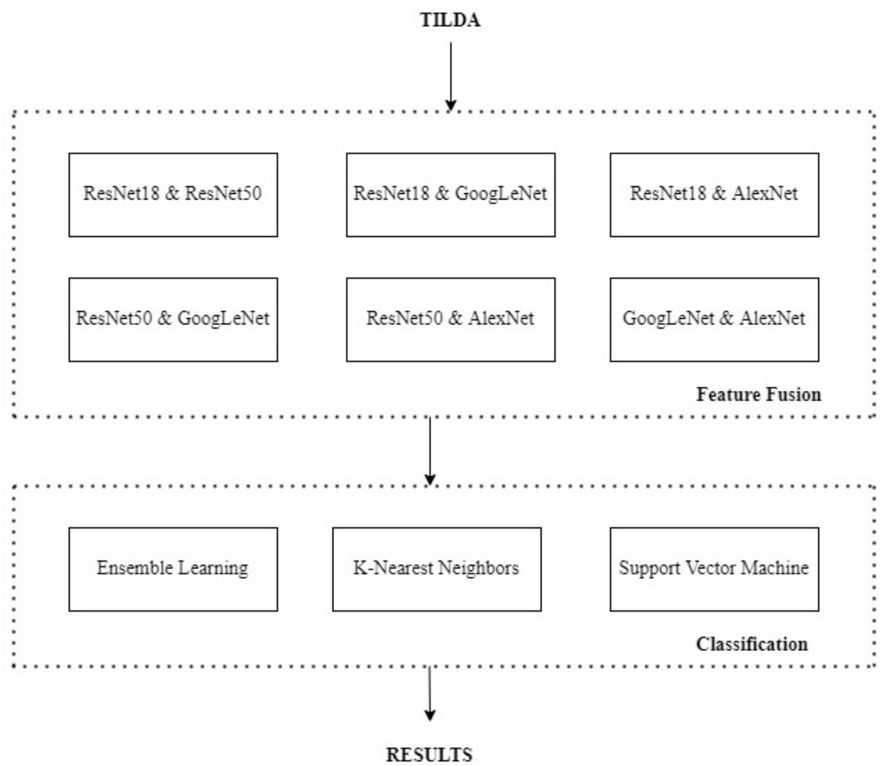


Figure 5.5 Proposed system

CHAPTER SIX

BACKGROUND FOR CLASSIFICATION

6.1 Ensemble Learning

Ensemble Learning (EL) is a method for performing classification based on predictions and decisions from multiple classifiers (Baran, 2020). It uses more than one classifier's information at the same time to apply each classifier's conclusion by consensus. This method outperforms a single classifier in most cases. The fact that the classifiers' mistakes differ from one another improves ensemble classification performance. Different subsets of the training dataset are used to achieve differences in the classifiers' predictions. To produce and train subsets of the training dataset, the bootstrap approach is utilized.

The predictions from the trained networks must be combined to arrive at a final outcome. The choice of the right combining approach for the predictions has an impact on classification performance in the ensemble learning method. The selection of the proper approach for the classifiers should be considered when choosing the combination technique. According to the combining procedures, sample selection for the training data set, and processing processes, there are many ensemble learning approaches. Bagging, boosting, and voting are the examples of these approaches.

6.1.1 Bagging

The bagging approach is the oldest, simplest, and an effective ensemble-based approach, in which learners are connected in parallel. It is based on bootstrap sampling method. By changing the examples each time, the bootstrap sampling approach creates various subsets from the training data set. A classifier is used to train each sub-training set. All classifiers categorize distinct sub-training sets at the same time. To aggregate the classifier estimates, the bagging approach employs the majority vote technique.

6.1.2 Boosting

The output of the preceding learning algorithm is used as input by algorithms. Unlike bagging approach, each classifier is affected by the performance of the previous algorithm in this method, where the algorithms are connected in sequence. The major goal of this approach is to help learner algorithms with low success rates achieve the target success rate.

Among the ensemble learning algorithms based on the Boosting approach, the Adaptive Boosting (Adaboost) algorithm, which was also used in this study, is one of the most powerful and widely used ensemble methods (Freund and Schapire, 1996).

6.1.3 Voting

Multiple classifiers of the same type train separate subsets of the dataset in boosting and bagging based ensemble approaches. The same set is classified by several types of classifiers in the voting technique. Because multiple types of classifiers increase the variability of the predictions, it improves performance by lowering the ensemble prediction error.

6.2 K Nearest Neighbor

One of the most popular and straightforward classification techniques is the K Nearest Neighbor (KNN) algorithm. A K value is determined for the operation of the algorithm. This K value means the number of elements to look at for classification. The distances between the sample element and the other elements in the dataset are calculated using different distance metrics (Chomboon et al., 2015). Euclidean, Manhattan, and Minkowski are among the most known distance metrics. In Table 6.1, the equation of them are given for m feature. In these equations, s is sample, and n is neighbor. To estimate which class the sample element in the dataset belongs to, the nearest K-neighbors of the element are used. The element is assigned to the class which neighbors belong to most.

Table 6.1 Distance metrics

Euclidean	$\sqrt{\sum_{i=1}^{m}(s_i - n_i)^2}$
Manhattan	$\sum_{i=1}^{m}\lvert s_i - n_i\rvert$
Minkowski	$\left(\sum_{i=1}^{m}(s_i - n_i)^q\right)^q$

6.3 Support Vector Machine

SVM is a machine learning technique proposed for classification problems in datasets where the patterns between variables are not known. It is a non-parametric classifier. This technique is also used for regression analysis like DTs.

Dataset is separated into two classes; training set, and test set. In this technique, the optimal hyperplane separating the classes is found using a labelled training set (Cortes and Vapnik, 1995). There may be more than one plane separating the two classes. An optimal hyperplane is the farthest plane to the nearest data points of the classes.

If the problem is two-dimensional, the hyperplane is a line (Fig. 6.1). As the number of dimensions increases, it becomes difficult to find the optimal hyperplane.

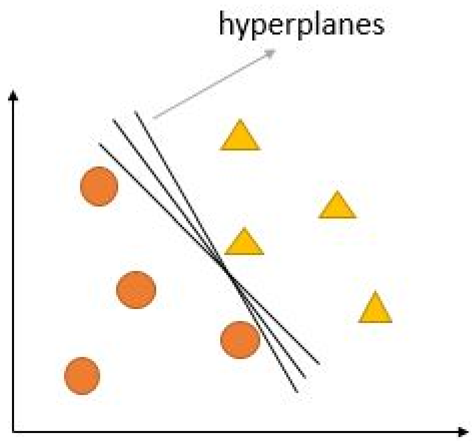


Figure 6.1 SVM

CHAPTER SEVEN

EXPERIMENTS

7.1 Evaluation Metrics

In determining how well machine learning algorithms rank, the selection of performance assessment criteria is crucial. The metrics used have an impact on how algorithms are assessed and comparisons are made.

In fabric defect classification systems, it is aimed to prevent recurrence after any defect is found. The occurrence of each defect depends on some reasons such as corruption of machine settings. For example, the machine and its elements should be checked for the defects caused by the machine. Fixing these problems prevents defects to occur. Studies in this field aim to achieve the highest performance in the shortest time.

In this study, the methods used in case 1 and case 3 have been compared in terms of accuracy, sensitivity, specificity, and F-score. These indicators are calculated using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) (Table 7.1). Accuracy seeks an answer to the question of "How many of all fabric samples have we labeled correctly?" (7.1). It is investigated how many of all defected fabrics are predicted correctly through the sensitivity metric (7.2). The specificity metric (7.3) investigates the inverse: How many un-defected fabrics have been discovered as un-defected? Precision investigates how many fabrics labeled as defected are actually defected. The harmonic mean of precision and recall gives the F-Score (7.5).

The majority of earlier studies merely compared the accuracy of learning algorithms. However, Huang et al. showed the success of the AUC value in their studies in 2003 and 2005. The methods used in case 2 and case 4 have been compared using the values of accuracy and AUC. The reason for using metrics different from case 1 and case 3 in case

2 and case 4 is to give the diagonal values of the confusion matrix obtained. Thus, it is possible to examine the classes classified with the highest and lowest success.

The area under the ROC (Receiver Operating Characteristic) curve is expressed by the AUC (Area Under Curve) value, which demonstrates how well the classification model distinguishes between the classes. The ROC curve is an increasing function between (0, 0) and (1, 1). The higher the AUC value, the higher the classification success of the model. AUC can take the largest value of "1" and the smallest value of "0.5".

Table 7.1 TP, TN, FP, and FN

		Actual	
		Positive	**Negative**
Predicted	**Positive**	TP	FP
	Negative	FN	TN

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{7.1}$$

$$Sensitivity\ (or\ Recall) = \frac{TP}{TP+FN} \tag{7.2}$$

$$Specificty = \frac{TN}{TN+FP} \tag{7.3}$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7.4}$$

The models' classification performance has been evaluated using 10-fold cross validation. Thus, the data set is split into ten equal parts as D*1*, D*2*, ..., D*10* (Han et al., 2011). Training and testing processes are repeated 10 times. Partition D*i* for iteration i is used for testing while the remaining partitions are utilized for training (Figure 7.1). The arithmetic average of the results obtained from each partition gives the success of the method.

Figure 7.1 10-fold cross validation

The suggested method makes use of fabric images from the Tilda database that are bo th defected and un-defected. Our experimental results consist of four cases. Classification of the images as defected or un-defected is performed in the first case. The successes of the methods in the Tilda dataset have been investigated from three different aspects (un-patterned dataset, patterned dataset, mix). In the second case, the groups of Tilda dataset have been tried to classify according to the eight classes (e0, e1, e2, e3, e4, e5, e6, e7). Classification of defects using feature fusion has been tested in case 3 and case 4.

7.2 Preliminaries to experiments

If CNN based methods are used only for feature extraction, there is no need to split the dataset into train and test. So, entire database may be used for training in such type of studies. However, 70% of the database is reserved for training and 30% for testing to make comparisons in this study.

Classification of features drawn from previous layers of CNN is generally less successful. Therefore, the features taken from the last layers were used in this study (Table 7.2). Layers of ‘pool5’, ‘avg_pool’, ‘pool5-drop_7x7_s1’, and ‘pool5’ have been used to extract the features for the methods of ResNet18, ResNet50, GoogLeNet, and AlexNet, respectively.

Table 7.2 Feature extracted layers of CNN methods

METHOD	LAYER
ResNet18	pool5
ResNet50	avg_pool
GoogLeNet	pool5-drop_7x7_s1
AlexNet	pool5

7.3 Experimental Results

7.3.1 Case study 1 - Classification of fabrics as defected/undefected

In the first case, the Tilda dataset has been handled in three different ways (Table 7.3). The first set consists of defected fabric images and un-defected fabric images in the c1 folder of Tilda. There are 1400 defected images and 200 un-defected images. The second set consists of defected fabric images and un-defected fabric images in the c2 folder of Tilda. There are 1400 defected images and 200 un-defected images. The third set consists of defected fabric images and un-defected fabric images in both the c1 and c2 folders. There are 2800 defected images and 400 un-defected images. Its goal is to see how well patterned, un-patterned, and both patterned and un-patterned fabrics in the sets can be classified. Fabrics are classified as defected/un-defected in all three ways (Figure 7.2).

Table 7.3 Details for case 1

Dataset	Number of Elements
Un-patterned Fabrics of Tilda	1600 (200 un-defected, 1400 defected)
Patterned Fabrics of Tilda	1600 (200 un-defected, 1400 defected)
Mix of Tilda	3200 (400 un-defected, 2800 defected)

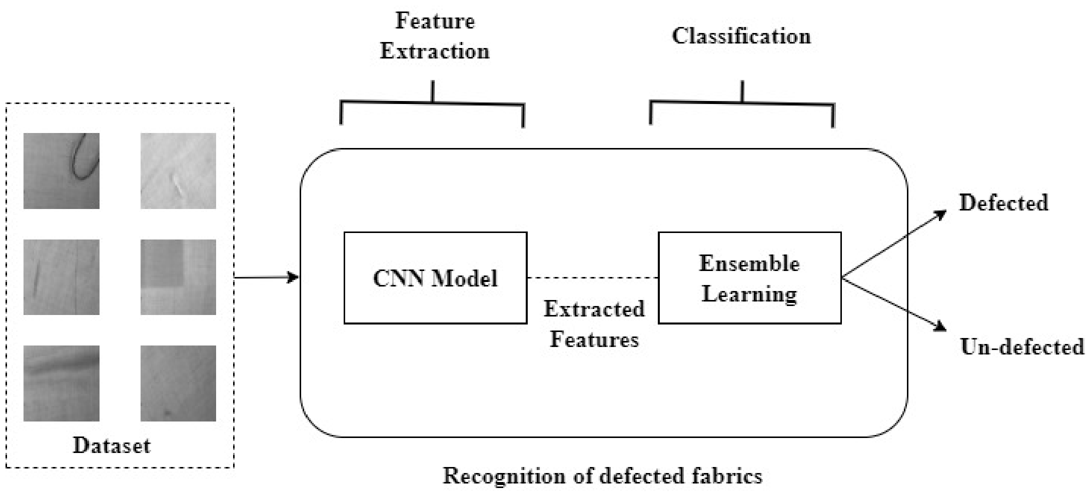


Figure 7.2 Sample Visualisation of Case 1

Table 7.4 shows the results of the methods. It can be said that the dataset including un-patterned fabrics has a greater classification success rate than the others (in general). The low specificity rates in various datasets and methodologies are due to the fact that the number of un-defected fabric images is smaller than the number of defected fabric images.

When the accuracy rates of the methods are compared, it is seen that highest accuracy rate is obtained when using GoogLeNet & EL for unpatterned fabrics (93.67%). The method that performs the best for patterned fabrics is ResNet50 & SVM (89.68%). It has also been the most successful method for mixed fabrics (89.72%).

Table 7.4 Results for case 1 (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix
ResNet18 & EL	**Sensitivity**	89.00	87.84	87.93
	Specificity	65.22	40.00	69.23
	F-Measure	93.67	93.15	93.45
	Accuracy	**88.31**	**87.24**	**87.78**
ResNet50 & EL	**Sensitivity**	91.58	90.05	88.74
	Specificity	72.12	77.05	75.00
	F-Measure	94.65	94.31	93.78
	Accuracy	**90.31**	**89.56**	**88.47**
GoogLeNet & EL	**Sensitivity**	90.31	88.06	87.94
	Specificity	59.14	48.00	78.26
	F-Measure	93.89	93.24	93.51
	Accuracy	**93.67**	**87.43**	**87.87**
AlexNet & EL	**Sensitivity**	89.01	88.67	87.70
	Specificity	68.18	78.57	81.82
	F-Measure	93.74	82.74	93.42
	Accuracy	**88.44**	**88.50**	**87.68**
ResNet18 &KNN	**Sensitivity**	94.96	92.28	93.17
	Specificity	39.54	39.75	38.28
	F-Measure	89.66	90.97	90.09
	Accuracy	**82.88**	**84.43**	**83.21**
ResNet50 & KNN	**Sensitivity**	96.40	93.47	94.14
	Specificity	41.87	42.32	42.25
	F-Measure	90.02	91.18	90.91
	Accuracy	**83.63**	**84.93**	**84.62**
GoogLeNet & KNN	**Sensitivity**	94.93	91.91	92.83
	Specificity	40.65	36.26	36.85
	F-Measure	90.08	90.21	89.80
	Accuracy	**83.50**	**83.18**	**82.71**
AlexNet & KNN	**Sensitivity**	94.11	92.22	93.00
	Specificity	23.75	29.26	26.45
	F-Measure	78.55	86.92	83.68
	Accuracy	**67.81**	**78.36**	**74.05**
ResNet18 & SVM	**Sensitivity**	87.48	87.64	87.72
	Specificity	33.33	60.00	66.67
	F-Measure	93.26	93.35	93.38
	Accuracy	**87.38**	**87.55**	**87.62**
ResNet50 & SVM	**Sensitivity**	93.06	89.70	90.15
	Specificity	84.48	88.89	79.03
	F-Measure	95.80	94.41	94.40
	Accuracy	**92.44**	**89.68**	**89.72**
GoogLeNet & SVM	**Sensitivity**	87.97	87.99	88.13
	Specificity	83.33	58.82	66.67
	F-Measure	93.54	93.39	93.49
	Accuracy	**87.94**	**87.68**	**87.87**
AlexNet & SVM	**Sensitivity**	88.38	88.60	88.18
	Specificity	72.00	72.41	49.18
	F-Measure	93.61	93.70	93.23
	Accuracy	**88.13**	**88.30**	**87.43**

Classifier based comparison is made in Table 7.5. Considering the average accuracy rates of the classifiers, it is seen that the best classifier is EL (88.70%). SVM is just as successful as EL (88.48%). KNN is the classifier with the lowest average with 81.11% accuracy. The specificity of KNN is extremely low (36.44%). The obtained results show that KNN should not be used in defect detection systems. When the mean of Table 10 is calculated, sensitivity, specificity, f-measure and accuracy values are 90.42%, 57.35%, 91.93%, 86.12%, respectively. The extremely low specificity values of KNN are the reason why the specificity average is so low.

Table 7.5 Classifier based comparison (%)

EL BASED AVERAGES	
Sensitivity	88.90
Specificity	67.72
F-Measure	92.80
Accuracy	88.77

KNN BASED AVERAGES	
Sensitivity	93.62
Specificity	36.44
F-Measure	88.51
Accuracy	81.11

SVM BASED AVERAGES	
Sensitivity	88.75
Specificity	67.90
F-Measure	93.80
Accuracy	88.48

AVERAGES	
Sensitivity	90.42
Specificity	57.35
F-Measure	91.93
Accuracy	86.12

When the dataset-based averages of the results given in Table 7.6 are examined, it is seen that the average accuracy rates of un-patterned and patterned fabrics are close to each other (86.20% and 86.40%, respectively). The average success of the mix dataset, which includes both patterned and unpatterned fabric samples, is not much lower than the average rates of the first two datasets.

Table 7.6 Dataset based comparison (%)

	UN-PATTERNED	PATTERNED	MIX
Sensitivity	91.43	89.87	89.97
Specificity	57.00	55.94	59.14
F Measure	91.71	91.46	91.93
Accuracy	86.20	86.40	85.75

Summary graphic for case 1 is in Figure 7.3.

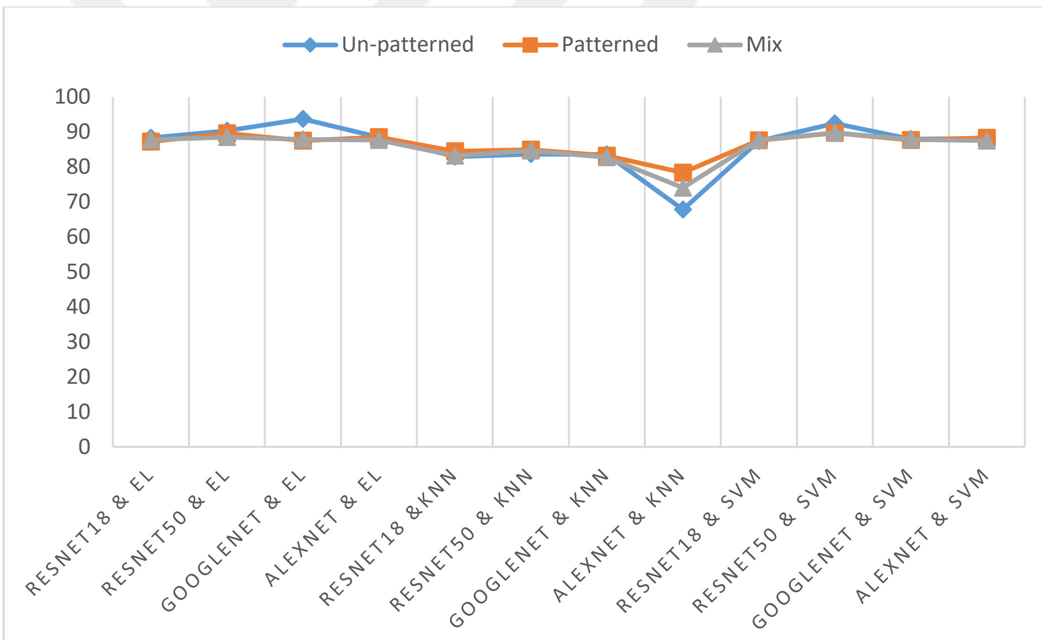


Figure 7.3 Summary for case 1

The execution times of the methods to classify different datasets are given in Table 7.7. Since there are twice as many elements in the Mix dataset as in other datasets, the execution time of classification methods for this dataset is longer (329.12 seconds). The average execution times for un-patterned and patterned fabrics are 140.16 and 171.10 seconds, respectively. In the table, the average execution time of the AlexNet&EL method is very high. The average execution time is 493.75 seconds when EL is used as the classifier. On the other hand, when KNN is used as the classifier, the execution time is 90.93 seconds, and when SVM is employed, it is 55.70 seconds.

Table 7.7 Time comparison (seconds)

METHOD	Un-patterned Fabrics	Patterned Fabrics	Mix	AVERAGE
ResNet18 & EL	116.19	189.40	249.47	185.02
ResNet50 & EL	295.90	361.90	614.99	424.26
GoogLeNet & EL	186.50	206.51	413.68	268.90
AlexNet & EL	720.73	912.26	1657.40	1096.80
ResNet18 &KNN	11.67	15.16	32.40	19.74
ResNet50 & KNN	28.32	38.39	86.39	51.03
GoogLeNet & KNN	23.16	21.61	74.10	39.62
AlexNet & KNN	152.88	152.82	454.28	253.33
ResNet18 & SVM	4.86	5.45	12.42	7.58
ResNet50 & SVM	16.85	18.40	39.43	24.89
GoogLeNet & SVM	9.03	9.46	26.54	15.01
AlexNet & SVM	115.81	121.79	288.32	175.31
AVERAGE	**140.16**	**171.10**	**329.12**	

7.3.2 Case Study 2 – Classification of Defects According to Their Types

In the second case, the Tilda dataset's groups (Table 7.8) have been classified using the eight classes (e0, e1, e2, e3, e4, e5, e6, e7) as in Figure 7.4. In each group, there are 50 images of each class.

ResNet50&SVM has the best accuracy rate (77.5%) for the c1r1 of Tilda database (Table 7.9). E0 has un-defected fabric samples. Considering the class-based accuracy rates, the class which is classified with the highest success is e0 (81.83%). E2 and e3 are the classes classified with the lowest rates (the accuracy rate of both is 56.17%).

Table 7.8 Details for case 2

Folder	Dataset	Number of elements
C1	c1r1 group of Tilda	400
	c1r3 group of Tilda	400
	c2r2 group of Tilda	400
	c2r3 group of Tilda	400
C2	c3r1 group of Tilda	400
	c3r3 group of Tilda	400
	c4r1 group of Tilda	400
	c4r3 group of Tilda	400

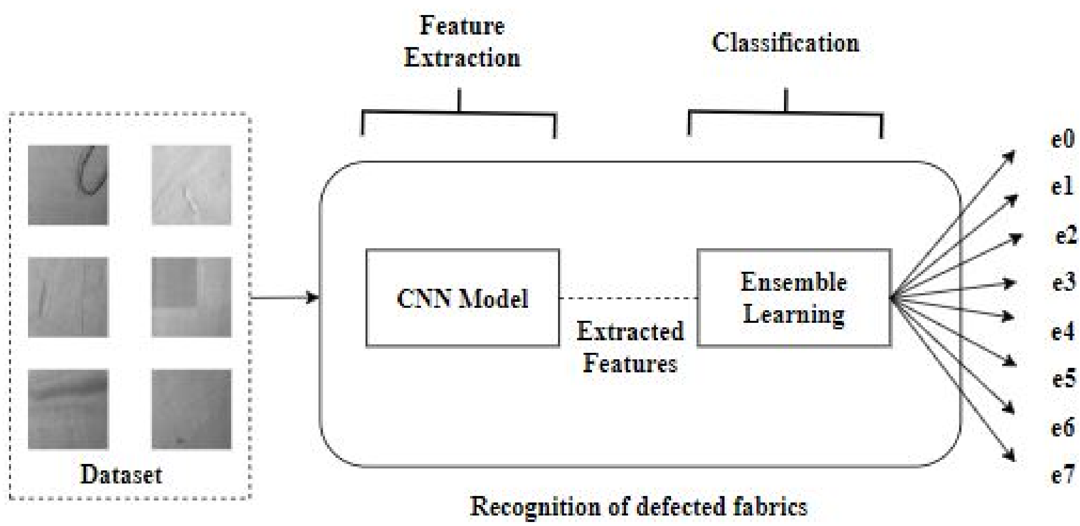


Figure 7.4 Sample Visualisation of Case 2

Table 7.9 Results for c1r1

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	Acc (%)	63.7	56.00	60.00	62.00	52.00	70.00	72.00	76.00	62.00
	AUC	0.86								
ResNet18&KNN	Acc (%)	67.8	72.00	62.00	44.00	58.00	76.00	78,00	90.00	62.00
	AUC	0.80								
ResNet18&SVM	Acc (%)	75.3	92.00	68.00	70.00	66.00	64.00	80,00	78.00	84.00
	AUC	0.96								
ResNet50&EL	Acc (%)	71.8	82.00	66.00	70.00	70.00	66.00	84,00	74.00	62.00
	AUC	0.95								
ResNet50&KNN	Acc (%)	66.8	90.00	46.00	60.00	44.00	64.00	78.00	86.00	66.00
	AUC	0.85								
ResNet50&SVM	Acc (%)	**77.5**	96.00	56.00	84.00	82.00	66.00	76,00	86.00	74.00
	AUC	0.99								
GoogLeNet&EL	Acc (%)	62.5	72.00	64.00	50.00	48.00	50.00	76.00	80.00	60.00
	AUC	0.90								

Table 7.9 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
GoogLeNet&KNN	Acc (%)	70.3	84.00	70.00	58.00	56.00	54,00	78.00	90.00	72.00
	AUC	0.86								
GoogLeNet&SVM	Acc (%)	72.8	92.00	60.00	58.00	76.00	54.00	78.00	88.00	76.00
	AUC	0.97								
AlexNet&EL	Acc (%)	56.80	70.00	46.00	50.00	36.00	44.00	74.00	72.00	62.00
	AUC	0.92								
AlexNet&KNN	Acc (%)	42.3	84.00	28.00	20.00	26.00	24.00	44.00	56.00	56.00
	AUC	0.74								
AlexNet&SVM	Acc (%)	60.8	92.00	20.00	48.00	60.00	50.00	56.00	82.00	78.00
	AUC	0.96								
AVERAGE ACCURACY (%)		**65.70**	**81.83**	**53.83**	**56.17**	**56.17**	**56.83**	**72.83**	**79.83**	**67.83**

ResNet18&SVM is the most successful method (69.50%) for classifying the c1r3 dataset (Table 7.10). Unlike Table 15, class e5 has the highest classification performance (71.67%) in this table. On the other hand, it is clear that e2 is the class classified with the lowest performance (39.50%).

Table 7.10 Results for c1r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	Acc (%)	52.50	54.00	56.00	32.00	36.00	50.00	84.00	42.00	66.00
	AUC	0.81								
ResNet18&KNN	Acc (%)	59.50	62.00	44.00	40.00	56.00	66.00	84.00	62.00	62.00
	AUC	0.73								
ResNet18&SVM	Acc (%)	**69.50**	84.00	66.00	58.00	58.00	68.00	88.00	54.00	80.00
	AUC	0.95								
ResNet50&EL	Acc (%)	59.80	68.00	64.00	42.00	52.00	64.00	82.00	46.00	60.00
	AUC	0.89								
ResNet50&KNN	Acc (%)	54.30	60.00	44.00	48.00	40.00	78.00	64.00	50.00	50.00
	AUC	0.71								
ResNet50&SVM	Acc (%)	67.00	78.00	54.00	56.00	64.00	74.00	80.00	58.00	72.00
	AUC	0.92								
GoogLeNet&EL	Acc (%)	51.50	58.00	36.00	42.00	38.00	56.00	74.00	56.00	52.00
	AUC	0.82								
GoogLeNet&KNN	Acc (%)	52.00	52.00	30.00	42.00	36.00	70.00	64.00	62.00	60.00
	AUC	0.68								
GoogLeNet&SVM	Acc (%)	56.80	72.00	42.00	38.00	48.00	50.00	84.00	52.00	68.00
	AUC	0.89								
AlexNet&EL	Acc (%)	45.30	58.00	40.00	20.00	18.00	44.00	70.00	66.00	46.00
	AUC	0.86								
AlexNet&KNN	Acc (%)	36.80	78.00	14.00	22.00	12.00	46.00	24.00	40.00	58.00
	AUC	0.73								
AlexNet&SVM	Acc (%)	51.20	90.00	22.00	34.00	44.00	44.00	62.00	54.00	60.00
	AUC	0.96								
AVERAGE ACCURACY (%)		**54.68**	**67.83**	**42.67**	**39.50**	**41.83**	**59.17**	**71.67**	**53.50**	**61.17**

According to Table 7.11, ResNet18&SVM is the best method for the c2r2 dataset of Tilda. The classification performances of e0 and e6 are very close to each other (73.83% and 73.33%, respectively). This dataset's most challenging class, e3, has a classification success rate of just 30.33 percent.

Table 7.11 Results for c2r2

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	Acc (%)	49.00	70.00	38.00	22.00	36.00	56.00	52.00	72.00	50.00
	AUC	0.85								
ResNet18&KNN	Acc (%)	58.80	82.00	44.00	34.00	30.00	56.00	72.00	86.00	66.00
	AUC	0.84								
ResNet18&SVM	Acc (%)	**65.80**	90.00	60.00	60.00	46.00	52.00	72.00	76.00	70.00
	AUC	0.96								
ResNet50&EL	Acc (%)	61.80	68.00	72.00	62.00	22.00	50.00	78.00	78.00	64.00
	AUC	0.92								
ResNet50&KNN	Acc (%)	58.30	74.00	48.00	52.00	28.00	52.00	76.00	80.00	56.00
	AUC	0.80								
ResNet50&SVM	Acc (%)	61.30	68.00	50.00	42.00	52.00	54.00	76.00	66.00	82.00
	AUC	0.94								
GoogLeNet&EL	Acc (%)	48.30	72.00	50.00	36.00	26.00	56.00	54.00	66.00	26.00
	AUC	0.88								
GoogLeNet&KNN	Acc (%)	51.70	70.00	36.00	40.00	28.00	60.00	60.00	74.00	46.00
	AUC	0.76								
GoogLeNet&SVM	Acc (%)	55.30	82.00	38.00	52.00	38.00	50.00	58.00	72.00	52.00
	AUC	0.93								
AlexNet&EL	Acc (%)	40.50	72.00	22.00	18.00	16.00	34.00	24.00	80.00	58.00
	AUC	0.84								
AlexNet&KNN	Acc (%)	37.80	62.00	12.00	16.00	14.00	40.00	14.00	74.00	70.00
	AUC	0.69								
AlexNet&SVM	Acc (%)	49.00	76.00	46.00	28.00	28.00	36.00	50.00	56.00	72.00
	AUC	0.92								
AVERAGE ACCURACY (%)		**53.13**	**73.83**	**43.00**	**38.50**	**30.33**	**49.67**	**57.17**	**73.33**	**59.33**

When compared to other methods, ResNet50&SVM has the highest accuracy rate for the Tilda c2r3 dataset (Table 7.12). The class classified with the best accuracy rate in this dataset is e3, which has the lowest rate in Table 17.

Table 7.12 Results for c2r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	Acc (%)	55.00	53.00	62.00	45.00	62.00	58.00	50.00	60.00	50.00
	AUC	0.85								
ResNet18&KNN	Acc (%)	52.50	63.00	52.00	31.00	60.00	60.00	46.00	56.00	52.00
	AUC	0.74								
ResNet18&SVM	Acc (%)	64.00	80.00	56.00	55.00	72.00	66.00	66.00	56.00	60.00
	AUC	0.91								
ResNet50&EL	Acc (%)	61.5	75.00	62.00	31.00	80.00	64.00	68.00	60.00	52.00
	AUC	0.90								

Table 7.12 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet50&KNN	**Acc (%)**	58.50	71.00	66.00	45.00	62.00	60.00	60.00	58.00	46.00
	AUC	0.78								
ResNet50&SVM	**Acc (%)**	**67.00**	88.00	56.00	61.00	88.00	66.00	74.00	48.00	54.00
	AUC	0.93								
GoogLeNet&EL	**Acc (%)**	53.50	45.00	60.00	37.00	78.00	62.00	54.00	50.00	42.00
	AUC	0.77								
GoogLeNet&KNN	**Acc (%)**	59.50	59.00	68.00	49.00	68.00	58.00	54.00	66.00	54.00
	AUC	0.71								
GoogLeNet&SVM	**Acc (%)**	65.00	71.00	56.00	59.00	86.00	62.00	64.00	56.00	66.00
	AUC	0.89								
AlexNet&EL	**Acc (%)**	40.80	39.00	38.00	12.00	56.00	46.00	44.00	40.00	50.00
	AUC	0.75								
AlexNet&KNN	**Acc (%)**	35.50	45.00	26.00	39.00	30.00	44.00	20.00	54.00	26.00
	AUC	0.61								
AlexNet&SVM	**Acc (%)**	48.00	75.00	50.00	35.00	54.00	42.00	46.00	36.00	46.00
	AUC	0.84								
AVERAGE ACCURACY (%)		**55.07**	**63.67**	**54.33**	**41.58**	**66.33**	**57.33**	**53.83**	**53.33**	**49.83**

In Table 7.13, ResNet50&SVM has the highest accuracy rate (71.30%), while AlexNet&KNN has the lowest accuracy rate (48.30%). In this dataset, e4 is the class most successfully classified (70.67%) among the eight classes, while e1 is the class with the lowest success (39.33%).

Table 7.13 Results for c3r1

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	**Acc (%)**	54.5	40.00	20.00	42.00	58.00	74.00	60.00	66.00	76.00
	AUC	0.76								
ResNet18&KNN	**Acc (%)**	58.5	60.00	44.00	48.00	58.00	62.00	54.00	66.00	76.00
	AUC	0.76								
ResNet18&SVM	**Acc (%)**	63.20	64.00	40.00	46.00	54.00	90.00	72.00	64.00	76.00
	AUC	0.95								
ResNet50&EL	**Acc (%)**	61.00	74.00	30.00	48.00	46.00	74.00	78.00	68.00	70.00
	AUC	0.94								
ResNet50&KNN	**Acc (%)**	56.3	64.00	34.00	42.00	66.00	70.00	54.00	52.00	68.00
	AUC	0.78								
ResNet50&SVM	**Acc (%)**	**71.30**	72.00	52.00	68.00	82.00	80.00	66.00	82.00	68.00
	AUC	0.94								
GoogLeNet&EL	**Acc (%)**	51.5	38.00	34.00	36.00	44.00	68.00	66.00	80.00	46.00
	AUC	0.79								
GoogLeNet&KNN	**Acc (%)**	58.8	72.00	48.00	42.00	68.00	52.00	54.00	68.00	66.00
	AUC	0.82								
GoogLeNet&SVM	**Acc (%)**	62.7	66.00	52.00	58.00	52.00	82.00	64.00	66.00	62.00
	AUC	0.95								

Table 7.13 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
AlexNet&EL	**Acc (%)**	57.8	70.00	32.00	36.00	50.00	66.00	60.00	66.00	82.00
	AUC	0.91								
AlexNet&KNN	**Acc (%)**	48.3	74.00	38.00	36.00	56.00	48.00	16.00	54.00	64.00
	AUC	0.79								
AlexNet&SVM	**Acc (%)**	62.5	78.00	48.00	56.00	54.00	82.00	60.00	66.00	56.00
	AUC	0.98								
AVERAGE ACCURACY (%)		**58.87**	**64.33**	**39.33**	**46.50**	**57.33**	**70.67**	**58.67**	**66.50**	**67.50**

When compared to other methods, ResNet50&SVM has the highest accuracy rate for the Tilda c3r3 dataset (Table 7.14). The class classified with the best accuracy rate in this dataset is e5 (65.33%), while the class classified with the lowest accuracy rate is e2 (31.33%).

Table 7.14 Results for c3r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	**Acc (%)**	39.80	24.00	28.00	14.00	30.00	60.00	68.00	56.00	38.00
	AUC	0.75								
ResNet18&KNN	**Acc (%)**	45.50	48.00	24.00	30.00	30.00	78.00	68.00	46.00	48.00
	AUC	0.65								
ResNet18&SVM	**Acc (%)**	52.30	46.00	54.00	40.00	32.00	66.00	72.00	70.00	38.00
	AUC	0.84								
ResNet50&EL	**Acc (%)**	56.30	66.00	62.00	24.00	46.00	72.00	80.00	56.00	44.00
	AUC	0.89								
ResNet50&KNN	**Acc (%)**	50.50	50.00	22.00	44.00	26.00	68.00	80.00	66.00	48.00
	AUC	0.67								
ResNet50&SVM	**Acc (%)**	**56.50**	64.00	54.00	40.00	50.00	68.00	74.00	62.00	40.00
	AUC	0.88								
GoogLeNet&EL	**Acc (%)**	47.50	28.00	28.00	34.00	44.00	64.00	74.00	70.00	38.00
	AUC	0.69								
GoogLeNet&KNN	**Acc (%)**	45.00	40.00	14.00	44.00	32.00	54.00	66.00	58.00	52.00
	AUC	0.63								
GoogLeNet&SVM	**Acc (%)**	51.50	42.00	48.00	40.00	46.00	60.00	72.00	62.00	42.00
	AUC	0.80								
AlexNet&EL	**Acc (%)**	42.00	46.00	28.00	12.00	32.00	68.00	44.00	58.00	48.00
	AUC	0.74								
AlexNet&KNN	**Acc (%)**	36.80	54.00	24.00	20.00	26.00	40.00	32.00	52.00	46.00
	AUC	0.64								
AlexNet&SVM	**Acc (%)**	41.80	26.00	36.00	34.00	34.00	56.00	54.00	53.00	42.00
	AUC	0.72								
AVERAGE ACCURACY (%)		**47.13**	**44.50**	**35.17**	**31.33**	**35.67**	**62.83**	**65.33**	**59.08**	**43.67**

ResNet50&SVM has the highest accuracy rate (57.50%) for c4r1 compared to other methods (Table 7.15). What is interesting about this table is that e0 is among the classes classified with the lowest accuracy rate (26.33%). E3 has the highest average accuracy rate (76.33%).

Table 7.15 Results for c4r1

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	**Acc (%)**	43.00	12.00	36.00	56.00	80.00	28.00	56.00	34.00	42.00
	AUC	0.62								
ResNet18&KNN	**Acc (%)**	52.00	26.00	58.00	68.00	78.00	38.00	44.00	48.00	56.00
	AUC	0.57								
ResNet18&SVM	**Acc (%)**	55.30	38.00	58.00	68.00	92.00	42.00	58.00	40.00	46.00
	AUC	0.73								
ResNet50&EL	**Acc (%)**	48.80	24.00	66.00	66.00	82.00	14.00	48.00	32.00	58.00
	AUC	0.66								
ResNet50&KNN	**Acc (%)**	53.50	34.00	56.00	82.00	88.00	36.00	32.00	42.00	58.00
	AUC	0.63								
ResNet50&SVM	**Acc (%)**	**57.50**	36.00	70.00	74.00	96.00	38.00	58.00	46.00	42.00
	AUC	0.73								
GoogLeNet&EL	**Acc (%)**	39.50	20.00	48.00	58.00	78.00	26.00	22.00	30.00	34.00
	AUC	0.69								
GoogLeNet&KNN	**Acc (%)**	37.80	18.00	46.00	66.00	72.00	12.00	30.00	26.00	32.00
	AUC	0.52								
GoogLeNet&SVM	**Acc (%)**	45.30	26.00	56.00	60.00	88.00	28.00	34.00	30.00	40.00
	AUC	0.63								
AlexNet&EL	**Acc (%)**	28.50	14.00	20.00	46.00	74.00	8.00	22.00	14.00	30.00
	AUC	0.55								
AlexNet&KNN	**Acc (%)**	20.30	38.00	4.00	58.00	8.00	10.00	8.00	16.00	20.00
	AUC	0.50								
AlexNet&SVM	**Acc (%)**	37.30	30.00	54.00	44.00	80.00	16.00	42.00	10.00	22.00
	AUC	0.58								
AVERAGE ACCURACY (%)		**43.23**	**26.33**	**47.67**	**62.17**	**76.33**	**24.67**	**37.83**	**30.67**	**40.00**

ResNet50&SVM is the most successful method in c4r3 dataset (59.10%) in Table 7.16. The method that fails the most is AlexNet&EL (24.80%). When class-based performances are examined, it is seen that e0 is classified with the highest success (56.50%), and e2 is classified with the lowest success (28.00%).

Table 7.16 Results for c4r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&EL	**Acc (%)**	41.40	36.00	50.00	26.00	70.00	16.00	67.00	20.00	46.00
	AUC	0.78								
ResNet18&KNN	**Acc (%)**	52.40	70.00	68.00	24.00	62.00	42.00	33.00	64.00	56.00
	AUC	0.80								
ResNet18&SVM	**Acc (%)**	55.10	64.00	52.00	28.00	74.00	42.00	71.00	64.00	46.00
	AUC	0.91								
ResNet50&EL	**Acc (%)**	46.40	72.00	52.00	32.00	30.00	48.00	59.00	38.00	40.00
	AUC	0.91								
ResNet50&KNN	**Acc (%)**	52.40	80.00	60.00	36.00	58.00	48.00	33.00	54.00	50.00
	AUC	0.85								
ResNet50&SVM	**Acc (%)**	**59.10**	82.00	68.00	44.00	68.00	44.00	65.00	68.00	34.00
	AUC	0.97								

Table 7.16 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	**e1**	**e2**	**e3**	**e4**	**e5**	**e6**	**e7**
GoogLeNet&EL	**Acc (%)**	35.60	38.00	34.00	14.00	30.00	46.00	59.00	30.00	34.00
	AUC	0.73								
GoogLeNet&KNN	**Acc (%)**	45.10	56.00	52.00	32.00	68.00	42.00	18.00	46.00	46.00
	AUC	0.74								
GoogLeNet&SVM	**Acc (%)**	47.10	50.00	56.00	36.00	58.00	36.00	63.00	40.00	38.00
	AUC	0.86								
AlexNet&EL	**Acc (%)**	24.80	46.00	18.00	16.00	24.00	16.00	10.00	32.00	36.00
	AUC	0.70								
AlexNet&KNN	**Acc (%)**	37.80	46.00	20.00	24.00	48.00	56.00	18.00	46.00	44.00
	AUC	0.68								
AlexNet&SVM	**Acc (%)**	33.60	38.00	36.00	24.00	36.00	28.00	35.00	38.00	34.00
	AUC	0.73								
AVERAGE (%)		**44.23**	**56.50**	**47.17**	**28.00**	**52.17**	**38.67**	**44.25**	**45.00**	**42.00**

Considering the average accuracy rates of all methods for all sets of Tilda, it is seen that the success of EL-based classification is 49.77%, the success of KNN-based classification is 51.00%, and the success of SVM-based classification is 58.49% (Table 7.17). ResNet50&SVM is the most successful method with an average accuracy of 64.65%, while AlexNet&KNN is the least successful method with a rate of 36.95%.

Table 7.17 Average accuracy rates of the methods (%)

ResNet18&EL:	49.86
ResNet18&KNN:	55.88
ResNet18&SVM:	62.56
ResNet50&EL:	58.43
ResNet50&KNN:	56.33
ResNet50&SVM:	64.65
GoogLeNet&EL:	48.74
GoogLeNet&KNN:	52.53
GoogLeNet&SVM:	57.06
AlexNet&EL:	42.06
AlexNet&KNN:	36.95
AlexNet&SVM:	48.03

Average for EL-based classification:	49.77
Average for KNN-based classification:	50.42
Average for SVM-based classification:	58.08

Summary graphic for case 2 is in Figure 7.5.

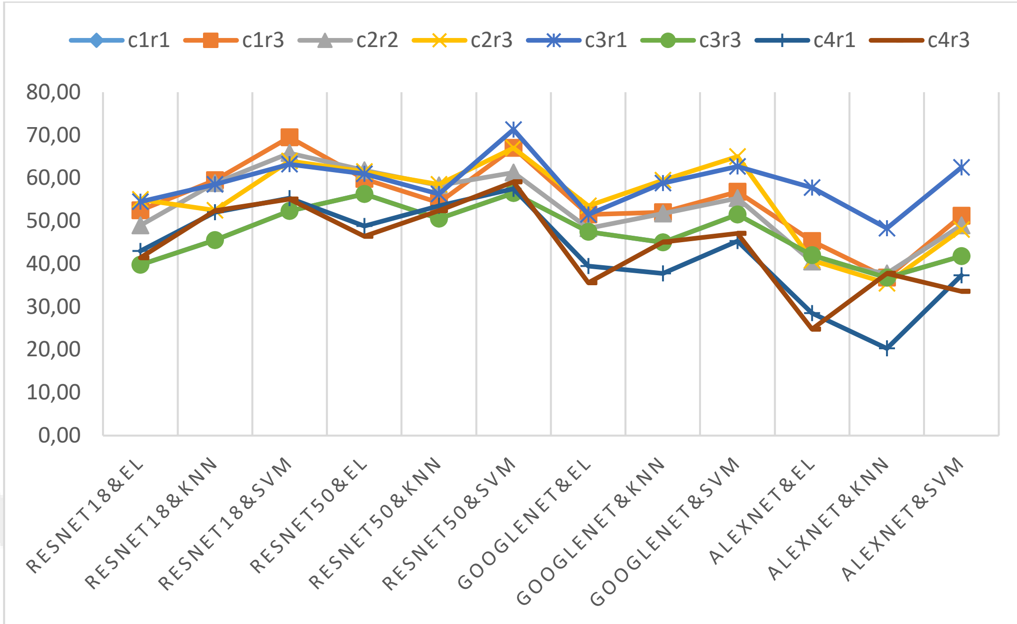


Figure 7.5 Summary for case 2

Average times to classify the features are given in Table 7.18. The EL-based classification time is 228.55 seconds on average. The KNN-based classification time is 20.59 seconds on average. The SVM based classification time is 34.61 seconds on average. It is seen that the classification time of EL is approximately 11 times longer than KNN and approximately 7 times longer than SVM.

Table 7.18 Average classification times (seconds)

ResNet18&EL	83.68
ResNet18&KNN	4.58
ResNet18&SVM	7.11
ResNet50&EL	220.10
ResNet50&KNN	9.48
ResNet50&SVM	17.41
GoogLeNet&EL	123.63
GoogLeNet&KNN	5.49
GoogLeNet&SVM	9.62
AlexNet&EL	486.79
AlexNet&KNN	62.80
AlexNet&SVM	104.31

Average time for EL-based classification:	228.55
Average time for KNN-based classification:	20.59
Average time for SVM-based classification:	34.61

7.3.3 Case Study 3 – Classification of fabrics as defected/undefected using feature fusion

In the third case, the features obtained using different feature extraction methods are brought together and classified with the help of a classifier (Figure 7.6).

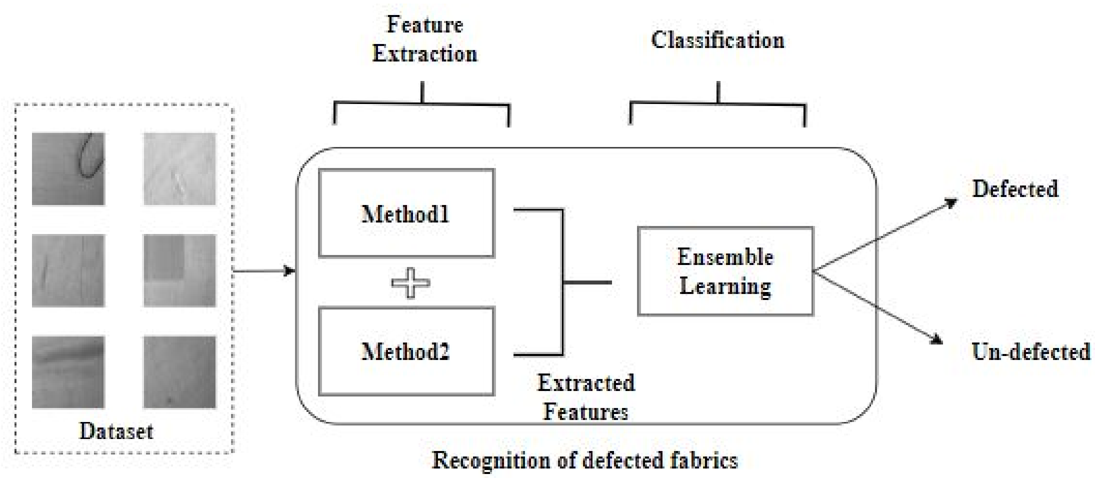


Figure 7.6 Sample visualisation of case 3

In Table 7.19, it is seen that the features obtained using ResNet18&ResNet50 are classified with the highest accuracy by SVM (90.52%). Additionally, EL's accuracy rate is close to that of SVM (89.40%). KNN performs the classification with the accuracy of 84.72%. Besides, un-patterned fabrics is a dataset that can be classified with the highest average accuracy rate (92.38%).

Table 7.20 shows that the features obtained using ResNet18&GoogLeNet are classified with the highest accuracy by SVM (88.29%). Additionally, EL's accuracy rate is close to that of SVM (88.27%). KNN performs the classification with the accuracy of 83.65%. On the other hand, un-patterned fabrics is a dataset that can be classified with the highest average accuracy rate (87.38%).

Table 7.19 Performance of ResNet18 & ResNet50 based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
ResNet18 & ResNet50 & EL	**Sensitivity**	92.05	89.66	88.93	90.21
	Specificity	71.55	67.21	84.38	74.38
	F-Measure	94.76	93.91	93.98	94.22
	Accuracy	**90.56**	**88.80**	**88.84**	**89.40**
ResNet18 & ResNet50 & KNN	**Sensitivity**	96.14	93.03	93.99	94.39
	Specificity	42.86	42.40	42.56	42.61
	F-Measure	90.46	91.34	91.04	90.95
	Accuracy	**84.25**	**85.12**	**84.80**	**84.72**
ResNet18 & ResNet50 & SVM	**Sensitivity**	92.77	89.42	90.20	90.80
	Specificity	86.92	87.50	80.00	84.81
	F-Measure	95.78	94.25	94.45	94.83
	Accuracy	**92.38**	**89.37**	**89.81**	**90.52**
AVERAGE ACCURACY (%)		**89.06**	**86.35**	**87.82**	

Table 7.20 Performance of ResNet18 & GoogLeNet based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
ResNet18 & GoogLeNet & EL	**Sensitivity**	91.18	88.41	87.97	89.19
	Specificity	66.35	51.35	62.50	60.07
	F-Measure	94.23	88.27	93.41	91.97
	Accuracy	**89.56**	**87.55**	**87.71**	**88.27**
ResNet18 & GoogLeNet & KNN	**Sensitivity**	96.2	92.08	93.55	93.94
	Specificity	42.19	37.50	39.23	39.64
	F-Measure	90.21	90.47	90.21	90.30
	Accuracy	**83.88**	**83.61**	**83.46**	**83.65**
ResNet18 & GoogLeNet & SVM	**Sensitivity**	88.7	88.44	88.24	88.46
	Specificity	88.46	75.00	66.67	76.71
	F-Measure	93.91	93.68	93.51	93.70
	Accuracy	**88.69**	**88.24**	**87.93**	**88.29**
AVERAGE ACCURACY (%)		**87.38**	**86.47**	**86.37**	

In Table 7.21, it is seen that the features obtained using ResNet18&AlexNet are classified with the highest accuracy by SVM (88.08%). In addition, EL's accuracy rate is close to that of SVM (88.00%). KNN performs the classification with the accuracy of 74.04%.

Table 7.21 Performance of ResNet18 & AlexNet based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
ResNet18 & AlexNet & EL	**Sensitivity**	89.34	88.27	87.86	88.49
	Specificity	59.68	68.18	78.95	68.94
	F-Measure	93.56	93.55	93.48	93.53
	Accuracy	**88.19**	**88.00**	**87.80**	**88.00**
ResNet18 & AlexNet & KNN	**Sensitivity**	94.77	92.27	93.53	93.52
	Specificity	24.00	31.76	27.72	27.83
	F-Measure	77.84	88.21	84.11	83.39
	Accuracy	**67.13**	**80.24**	**74.74**	**74.04**
ResNet18 & AlexNet &SVM	**Sensitivity**	88.57	88.54	87.99	88.37
	Specificity	84.00	71.43	58.49	71.31
	F-Measure	93.81	93.67	93.41	93.63
	Accuracy	**88.50**	**88.24**	**87.50**	**88.08**
AVERAGE ACCURACY (%)		**81.27**	**85.49**	**83.35**	

Table 7.22 shows that the ResNet50&GoogLeNet-obtained features are classified by SVM with the best degree of accuracy (91.01%). The average accuracy rates of EL and KNN are 89.65% and 84.28%, respectively.

Table 7.22 Performance of ResNet50 & GoogLeNet based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
ResNet50 & GoogLeNet & EL	**Sensitivity**	92.32	89.48	89.05	90.28
	Specificity	75.65	75.51	76.62	75.93
	F-Measure	95.08	94.07	93.92	94.36
	Accuracy	**91.13**	**89.06**	**88.75**	**89.65**
ResNet50 & GoogLeNet & KNN	**Sensitivity**	95.77	93.34	94.02	94.38
	Specificity	40.05	44.36	41.39	41.93
	F-Measure	89.53	91.71	90.69	90.64
	Accuracy	**82.81**	**85.74**	**84.28**	**84.28**
ResNet50 & GoogLeNet & SVM	**Sensitivity**	93.45	89.83	90.56	91.28
	Specificity	86.67	91.30	82.35	86.77
	F-Measure	96.08	94.51	94.66	95.08
	Accuracy	**92.94**	**89.87**	**90.22**	**91.01**
AVERAGE ACCURACY (%)		**88.96**	**88.22**	**87.75**	

While SVM is always the classifier with the highest accuracy rate in tables 7.19-7.23, it is seen that the situations have changed in Table 7.23. This table demonstrates that EL provides the most accurate classification of the ResNet50&AlexNet-obtained features (89.57%). Both KNN and SVM have average accuracy rates of 75.93% and 88.73%, respectively.

Table 7.23 Performance of ResNet50 & AlexNet based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
ResNet50 & AlexNet & EL	**Sensitivity**	92.13	89.85	88.90	90.29
	Specificity	73.68	70.97	82.81	75.82
	F-Measure	94.90	94.07	93.95	94.31
	Accuracy	**90.81**	**89.12**	**88.78**	**89.57**
ResNet50 & AlexNet & KNN	**Sensitivity**	95.54	92.18	93.86	93.86
	Specificity	26.44	33.33	29.62	29.80
	F-Measure	80.12	88.98	88.25	85.78
	Accuracy	**70.06**	**81.36**	**76.37**	**75.93**
ResNet50 & AlexNet &SVM	**Sensitivity**	89.43	88.80	88.53	88.92
	Specificity	92.30	85.71	61.76	79.92
	F-Measure	94.32	93.94	93.51	93.92
	Accuracy	**89.50**	**88.74**	**87.96**	**88.73**
AVERAGE ACCURACY (%)		**83.46**	**86.41**	**84.37**	

Table 7.24 demonstrates that EL provides the most accurate classification of the ResNet50&AlexNet-obtained features (88.49%). Both KNN and SVM have average accuracy rates of 74.88% and 88.41%, respectively. In this table, we would like to point out that EL has the highest accuracy.

When the classifier-based averages of the results given in Table 7.25 are examined, it is seen that classification can be performed with an average accuracy rate of about 85.88%. While the classification successes of EL and SVM are close to each other (88.90% and 89.17%, respectively), the classification success of KNN is lower than EL and SVM (79.58%). The table shows that the specificity of KNN is extremely low. This shows that the success of KNN in correctly recognizing un-defected fabrics is low.

Table 7.24 Performance of GoogLeNet & AlexNet based feature extraction (%)

Method	Metric	Un-patterned Fabrics	Patterned Fabrics	Mix	Average
GoogLeNet & AlexNet & EL	**Sensitivity**	90.53	88.70	88.15	89.13
	Specificity	64.44	71.88	76.47	70.93
	F-Measure	93.98	93.73	93.58	93.76
	Accuracy	**89.06**	**88.37**	**88.03**	**88.49**
GoogLeNet & AlexNet & KNN	**Sensitivity**	95.17	92.49	93.35	93.67
	Specificity	25.21	33.44	27.62	28.76
	F-Measure	79.01	88.77	84.25	84.01
	Accuracy	**68.63**	**81.11**	**74.90**	**74.88**
GoogLeNet & AlexNet &SVM	**Sensitivity**	88.82	88.62	88.30	88.58
	Specificity	96.15	80.77	58.93	78.62
	F-Measure	94.05	93.81	93.42	93.76
	Accuracy	**88.94**	**88.50**	**87.78**	**88.41**
AVERAGE ACCURACY (%)		**82.21**	**85.99**	**83.57**	

Table 7.25 Performance comparison (%)

EL AVERAGE	
Sensitivity	89.60
Specificity	71.01
F Measure	93.69
Accuracy	88.90

KNN AVERAGE	
Sensitivity	93.96
Specificity	35.10
F Measure	87.51
Accuracy	79.58

SVM AVERAGE	
Sensitivity	89.40
Specificity	79.69
F Measure	94.15
Accuracy	89.17

AVERAGE	
Sensitivity	90.99
Specificity	61.93
F Measure	91.79
Accuracy	85.88

Table 7.26 shows the dataset-based averages of the results. It is seen that the average accuracy rate is highest for patterned fabrics (86.72%). The success of finding unpatterned fabric defects in fusion features created with AlexNet is lower than others. In all fusions where AlexNet is not used, the success of finding unpatterned fabric defects is higher than the others (88.47%).

Table 7.26 Dataset based performance comparison

	UN-PATTERNED	PATTERNED	MIX
Sensitivity	92.38	90.19	90.39
Specificity	67.00	62.20	59.89
F Measure	91.20	92.27	91.88
Accuracy	85.39	86.72	85.54

In order to summarize the findings obtained, the graph of case 3 is in Figure 7.7.

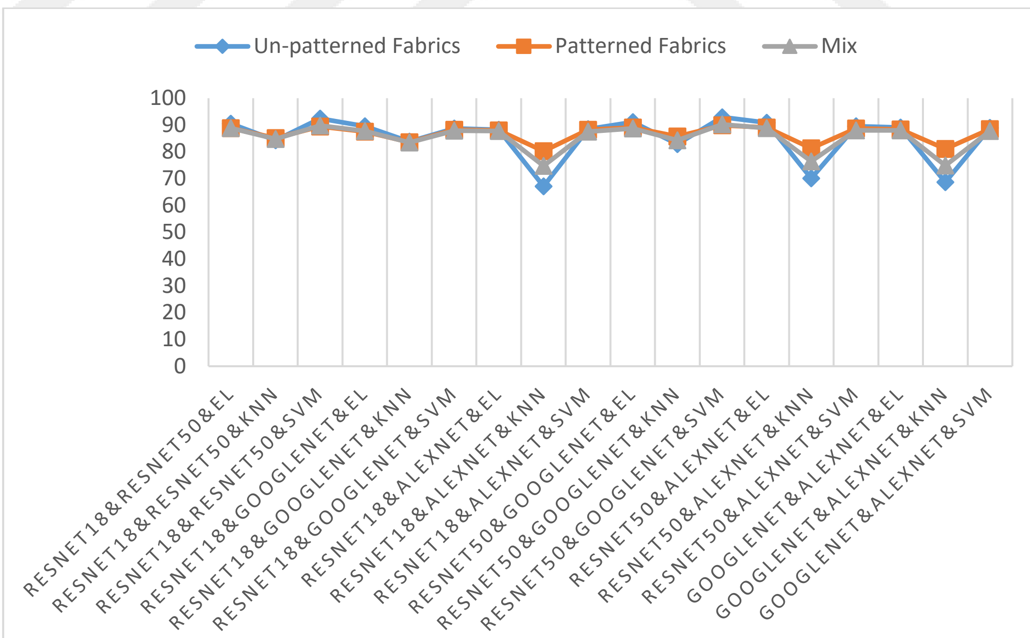


Figure 7.7 Summary for case 3

Table 7.27 provides the average classification times of the features. The average time for EL-based classification is 997.43 seconds. The average classification time for KNN is

228.44 seconds. The average time for SVM-based classification is 120.36 seconds. It can be shown that EL takes considerably longer to classify the data than KNN and SVM-roughly 4 and 8 times longer, respectively.

Table 7.27 Classification times (seconds)

	DATASET			
METHOD	**Un-patterned Fabrics**	**Patterned Fabrics**	**Mix**	**AVERAGE**
ResNet18&Resnet50&EL	382.54	495.02	923.20	600.22
ResNet18&Resnet50&KNN	47.48	47.98	156.59	84.02
ResNet18&Resnet50&SVM	21.94	23.44	55.61	33.66
ResNet18&GoogLeNet&EL	305.30	347.88	630.50	427.89
ResNet18&GoogLeNet&KNN	27.93	26.90	89.46	48.10
ResNet18&GoogLeNet&SVM	12.46	13.87	33.98	20.10
ResNet18&AlexNet&EL	853.13	1061.40	1869.10	1261.21
ResNet18&AlexNet&KNN	254.71	202.32	606.10	354.38
ResNet18&AlexNet&SVM	161.04	143.02	294.94	199.67
ResNet50&GoogLeNet&EL	471.55	576.10	1007.60	685.08
ResNet50&GoogLeNet&KNN	55.50	55.03	117.65	76.06
ResNet50&GoogLeNet&SVM	26.50	29.47	66.66	40.88
ResNet50&AlexNet&EL	1014.20	1273.20	2333.10	1540.17
ResNet50&AlexNet&KNN	249.78	254.24	757.74	420.59
ResNet50&AlexNet&SVM	159.20	169.13	356.27	228.20
GoogLeNet&AlexNet&EL	999.97	1377.50	2032.50	1469.99
GoogLeNet&AlexNet&KNN	217.55	221.85	723.11	387.50
GoogLeNet&AlexNet&SVM	136.48	172.10	290.43	199.67

7.3.4 Case Study 4 – Classification of Defects According to Their Types Using Feature Fusion

Visualisation of case 4 is as in Figure 7.8. The feature fusion created by combining the features of two different CNN models is tried to be classified. The classification is not binary as in case 1 and case 3, but 8-class as in case 2.

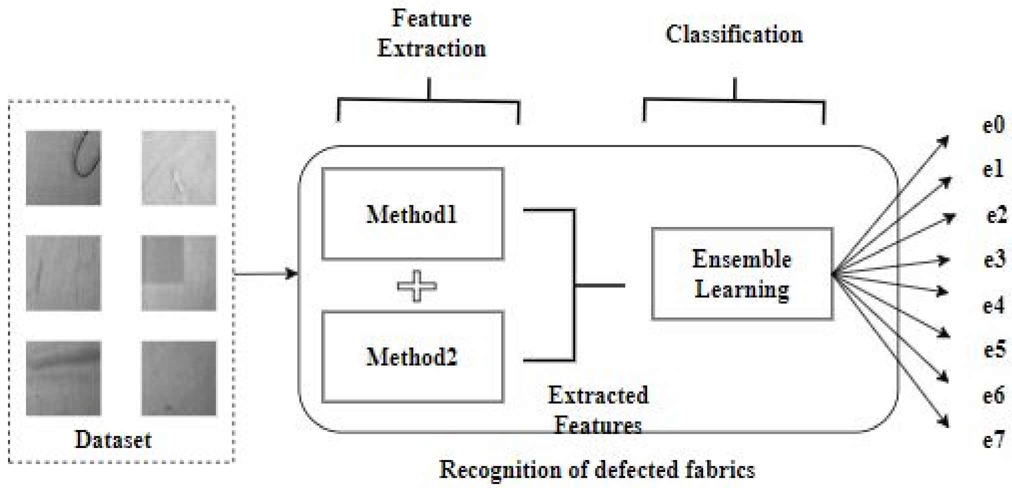


Figure 7.8 Sample visualisation of case 4

In Table 7.28, ResNet50&GoogLeNet&SVM has the highest accuracy rate (80.00%), while ResNet18&AlexNet&KNN has the lowest accuracy rate (46.30%). In this dataset, e0 is the class most successfully classified (88.22%) among the eight classes, while e2 is the class with the lowest success (53.00%).

Table 7.28 Results for c1r1

METHOD	GENERAL PERFORMANCE		CLASS-BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	Acc (%)	74.50	92.00	76.00	70.00	64.00	68.00	72.00	78.00	76.00
	AUC	0.96								
ResNet18&Resnet50&KNN	Acc (%)	66.80	88.00	48.00	6.00	50.00	62.00	84.00	88.00	54.00
	AUC	0.86								
ResNet18&Resnet50&SVM	Acc (%)	79.00	96.00	54.00	86.00	82.00	68.00	82.00	84.00	80.00
	AUC	0.99								
ResNet18&GoogLeNet&EL	Acc (%)	65.30	62.00	64.00	52.00	56.00	62.00	76.00	90.00	60.00
	AUC	0.90								
ResNet18&GoogLeNet&KNN	Acc (%)	71.00	76.00	64.00	54.00	66.00	62.00	82.00	94.00	70.00
	AUC	0.82								
ResNet18&GoogLeNet&SVM	Acc (%)	76.80	96.00	62.00	74.00	72.00	64.00	78.00	84.00	84.00
	AUC	0.98								
ResNet18&AlexNet&EL	Acc (%)	66.80	86.0	70.00	54.00	40.00	70.00	80.00	66.00	68.00
	AUC	0.95								
ResNet18&AlexNet&KNN	Acc (%)	46.30	86.00	32.00	18.00	18.00	36.00	52.00	62.00	66.00
	AUC	0.77								
ResNet18&AlexNet&SVM	Acc (%)	62.70	96.00	26.00	54.00	62.00	52.00	62.00	74.00	76.00
	AUC	0.97								
ResNet50&GoogLeNet&EL	Acc (%)	74.30	82.00	66.00	78.00	66.00	66.00	86.00	90.00	60.00
	AUC	0.94								

Table 7.28 Continues

METHOD	GENERAL PERFORMANCE		CLASS-BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet50&GoogLeNet&KNN	**Acc (%)**	70.30	96.00	56.00	60.00	48.00	52.00	84.00	92.00	74.00
	AUC	0.91								
ResNet50&GoogLeNet&SVM	**Acc (%)**	**80.00**	96.00	60.00	78.00	86.00	66.00	82.00	92.00	80.00
	AUC	0.98								
ResNet50&AlexNet&EL	**Acc (%)**	74.80	94.00	86.00	72.00	58.00	72.00	88.00	68.00	60.00
	AUC	0.98								
ResNet50&AlexNet&KNN	**Acc (%)**	46.50	86.00	30.00	22.00	28.00	32.00	54.00	64.00	56.00
	AUC	0.77								
ResNet50&AlexNet&SVM	**Acc (%)**	67.30	98.00	38.00	54.00	72.00	54.00	62.00	80.00	80.00
	AUC	0.98								
GoogLeNet&AlexNet&EL	**Acc (%)**	66.8	80.00	68.00	56.00	44.00	60.00	76.00	88.00	62.00
	AUC	0.93								
GoogLeNet&AlexNet&KNN	**Acc (%)**	48.30	86.00	34.00	18.00	30.00	36.00	48.00	66.00	68.00
	AUC	0.78								
GoogLeNet&AlexNet&SVM	**Acc (%)**	67.30	92.00	38.00	48.00	70.00	52.00	70.00	82.00	86.00
	AUC	0.97								
AVERAGE ACCURACY (%)		**66.93**	**88.22**	**54.00**	**53.00**	**56.22**	**57.44**	**73.22**	**80.11**	**70.00**

According to Table 7.29, ResNet18&GoogLeNet&SVM is the best method (70.00%) for the c1r3 dataset of Tilda. Compared to other classes, e0's classification performance has the greatest rate (75.67%). This dataset's most challenging class, e2, has a classification success rate of just 38.22 percent.

Table 7.29 Results for c1r3

METHOD	GENERAL PERFORMANCE		CLASS-BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	61.50	56.00	62.00	40.00	62.00	70.00	82.00	58.00	62.00
	AUC	0.89								
ResNet18&Resnet50&KNN	**Acc (%)**	59.50	70.00	34.00	50.00	50.00	76.00	82.00	48.00	66.00
	AUC	0.76								
ResNet18&Resnet50&SVM	**Acc (%)**	68.50	84.00	54.00	56.00	66.00	82.00	82.00	50.00	74.00
	AUC	0.94								
ResNet18&GoogLeNet&EL	**Acc (%)**	56.00	60.00	50.00	40.00	48.00	54.00	86.00	54.00	56.00
	AUC	0.83								
ResNet18&GoogLeNet&KNN	**Acc (%)**	60.30	70.00	36.00	38.00	48.00	68.00	76.00	74.00	72.00
	AUC	0.78								
ResNet18&GoogLeNet&SVM	**Acc (%)**	**70.00**	86.00	66.00	54.00	58.00	74.00	84.00	56.00	82.00
	AUC	0.96								
ResNet18&AlexNet&EL	**Acc (%)**	56.30	92.00	38.00	30.00	56.00	54.00	70.00	42.00	66.00
	AUC	0.96								
ResNet18&AlexNet&KNN	**Acc (%)**	38.50	80.00	12.00	20.00	16.00	44.00	30.00	42.00	64.00
	AUC	0.75								
ResNet18&AlexNet&SVM	**Acc (%)**	56.00	92.00	38.00	30.00	56.00	54.00	70.00	42.00	66.00
	AUC	0.96								
ResNet50&GoogLeNet&EL	**Acc (%)**	61.00	66.00	58.00	52.00	56.00	64.00	78.00	52.00	62.00
	AUC	0.89								
ResNet50&GoogLeNet&KNN	**Acc (%)**	56.00	66.00	40.00	40.00	42.00	78.00	66.00	60.00	56.00
	AUC	0.74								
ResNet50&GoogLeNet&SVM	**Acc (%)**	65.50	74.00	58.00	54.00	54.00	72.00	84.00	54.00	74.00
	AUC	0.92								

Table 7.29 Continues

METHOD	GENERAL PERFORMANCE		CLASS-BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet50&AlexNet&EL	**Acc (%)**	62.50	66.00	64.00	42.00	56.00	76.00	86.00	54.00	56.00
	AUC	0.90								
ResNet50&AlexNet&KNN	**Acc (%)**	38.00	80.00	14.00	18.00	16.00	40.00	34.00	44.00	58.00
	AUC	0.74								
ResNet50&AlexNet&SVM	**Acc (%)**	54.00	92.00	34.00	36.00	54.00	44.00	62.00	48.00	62.00
	AUC	0.95								
GoogLeNet&AlexNet&EL	**Acc (%)**	53.50	62.00	26.00	44.00	42.00	64.00	80.00	58.00	52.00
	AUC	0.85								
GoogLeNet&AlexNet&KNN	**Acc (%)**	38.30	76.00	12.00	20.00	12.00	42.00	32.00	46.00	66.00
	AUC	0.73								
GoogLeNet&AlexNet&SVM	**Acc (%)**	53.50	90.00	32.00	24.00	44.00	56.00	70.00	48.00	64.00
	AUC	0.96								
AVERAGE ACCURACY (%)		**56.05**	**75.67**	**40.44**	**38.22**	**46.44**	**61.78**	**69.67**	**51.67**	**64.33**

The methods of ResNet18&GoogLeNet&SVM and ResNet50&GoogLeNet&SVM have highest rates (both have a classification success of 64.50%) in Table 7.30. E6 is the class with the highest success (77.89%) while e3 is the class with the lowest success (31.22%).

Table 7.30 Results for c2r2

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	60.50	80.00	58.00	58.00	28.00	50.00	76.00	78.00	56.00
	AUC	0.95								
ResNet18&Resnet50&KNN	**Acc (%)**	56.80	74.00	38.00	48.00	24.00	50.00	74.00	86.00	60.00
	AUC	0.79								
ResNet18&Resnet50&SVM	**Acc (%)**	65.30	82.00	62.00	44.00	58.00	50.00	74.00	74.00	78.00
	AUC	0.97								
ResNet18&GoogLeNet&EL	**Acc (%)**	53.50	82.00	50.00	38.00	22.00	48.00	56.00	76.00	56.00
	AUC	0.93								
ResNet18&GoogLeNet&KNN	**Acc (%)**	56.80	74.00	38.00	44.00	34.00	54.00	58.00	82.00	70.00
	AUC	0.79								
ResNet18&GoogLeNet&SVM	**Acc (%)**	**64.50**	84.00	46.00	66.00	52.00	50.00	68.00	74.00	76.00
	AUC	0.96								
ResNet18&AlexNet&EL	**Acc (%)**	50.20	74.00	38.00	28.00	16.00	54.00	54.00	86.00	52.00
	AUC	0.89								
ResNet18&AlexNet&KNN	**Acc (%)**	40.50	72.00	10.00	16.00	14.00	46.00	16.00	84.00	66.00
	AUC	0.73								
ResNet18&AlexNet&SVM	**Acc (%)**	53.50	76.00	50.00	28.00	32.00	42.00	64.00	70.00	66.00
	AUC	0.93								
ResNet50&GoogLeNet&EL	**Acc (%)**	62.00	64.00	66.00	58.00	30.00	52.00	82.00	78.00	66.00
	AUC	0.91								
ResNet50&GoogLeNet&KNN	**Acc (%)**	57.50	72.00	38.00	56.00	34.00	52.00	64.00	88.00	56.00
	AUC	0.78								
ResNet50&GoogLeNet&SVM	**Acc (%)**	**64.50**	80.00	62.00	50.00	50.00	56.00	68.00	72.00	78.00
	AUC	0.95								
ResNet50&AlexNet&EL	**Acc (%)**	60.80	72.00	58.00	54.00	34.00	50.00	68.00	88.00	62.00
	AUC	0.94								
ResNet50&AlexNet&KNN	**Acc (%)**	40.50	66.00	16.00	18.00	22.00	42.00	22.00	76.00	62.00
	AUC	0.71								

Table 7.30 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet50&AlexNet&SVM	**Acc (%)**	59.80	76.00	62.00	38.00	36.00	48.00	76.00	68.00	74.00
	AUC	0.94								
GoogLeNet&AlexNet&EL	**Acc (%)**	49.50	80.00	54.00	26.00	14.00	52.00	50.00	76.00	44.00
	AUC	0.93								
GoogLeNet&AlexNet&KNN	**Acc (%)**	37.80	64.00	12.00	8.00	18.00	36.00	22.00	78.00	64.00
	AUC	0.69								
GoogLeNet&AlexNet&SVM	**Acc (%)**	57.50	82.00	60.00	24.00	44.00	42.00	64.00	68.00	76.00
	AUC	0.94								
AVERAGE ACCURACY (%)		**53.96**	**75.22**	**45.44**	**39.00**	**31.22**	**48.56**	**58.67**	**77.89**	**64.56**

According to Table 7.31, ResNet18&GoogLeNet&SVM is the best method (71.30%) for the c3r3 dataset of Tilda. E0 is the class classified with the highest accuracy rate (71.00%). E2 is the only class with a classification success of less than 50% (48.22%).

Table 7.31 Results for c2r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	63.70	75.00	64.00	47.00	80.00	52.00	72.00	64.00	56.00
	AUC	0.91								
ResNet18&Resnet50&KNN	**Acc (%)**	59.80	69.00	64.00	49.00	66.00	62.00	64.00	52.00	52.00
	AUC	0.77								
ResNet18&Resnet50&SVM	**Acc (%)**	69.80	94.00	60.00	71.00	82.00	70.00	76.00	50.00	54.00
	AUC	0.96								
ResNet18&GoogLeNet&EL	**Acc (%)**	59.50	53.00	62.00	51.00	82.00	64.00	54.00	64.00	46.00
	AUC	0.84								
ResNet18&GoogLeNet&KNN	**Acc (%)**	64.30	71.00	72.00	53.00	72.00	66.00	54.00	60.00	66.00
	AUC	0.78								
ResNet18&GoogLeNet&SVM	**Acc (%)**	**71.30**	78.00	58.00	65.00	92.00	76.00	68.00	60.00	72.00
	AUC	0.93								
ResNet18&AlexNet&EL	**Acc (%)**	58.30	65.00	60.00	43.00	70.00	64.00	54.00	52.00	58.00
	AUC	0.87								
ResNet18&AlexNet&KNN	**Acc (%)**	34.30	47.00	24.00	35.00	24.00	42.00	26.00	44.00	32.00
	AUC	0.60								
ResNet18&AlexNet&SVM	**Acc (%)**	50.00	76.00	46.00	41.00	62.00	42.00	44.00	42.00	46.00
	AUC	0.84								
ResNet50&GoogLeNet&EL	**Acc (%)**	66.00	71.00	68.00	61.00	78.00	74.00	64.00	54.00	58.00
	AUC	0.92								
ResNet50&GoogLeNet&KNN	**Acc (%)**	63.20	84.00	72.00	41.00	72.00	58.00	60.00	60.00	58.00
	AUC	0.84								
ResNet50&GoogLeNet&SVM	**Acc (%)**	70.80	88.00	58.00	67.00	88.00	74.00	68.00	58.00	64.00
	AUC	0.95								
ResNet50&AlexNet&EL	**Acc (%)**	65.30	80.00	64.00	51.00	70.00	58.00	74.00	64.00	60.00
	AUC	0.95								
ResNet50&AlexNet&KNN	**Acc (%)**	38.50	51.00	26.00	35.00	40.00	42.00	34.00	48.00	32.00
	AUC	0.63								
ResNet50&AlexNet&SVM	**Acc (%)**	56.30	88.00	48.00	45.00	70.00	48.00	54.00	48.00	48.00
	AUC	0.91								
GoogLeNet&AlexNet&EL	**Acc (%)**	56.80	41.00	66.00	35.00	80.00	68.00	60.00	54.00	50.00
	AUC	0.81								
GoogLeNet&AlexNet&KNN	**Acc (%)**	37.00	67.00	22.00	33.00	34.00	32.00	24.00	54.00	30.00
	AUC	0.70								

Table 7.31 Continues

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
GoogLeNet&AlexNet&SVM	**Acc (%)**	52.80	80.00	42.00	45.00	72.00	46.00	44.00	44.00	48.00
	AUC	0.87								
AVERAGE ACCURACY (%)		**57.65**	**71.00**	**54.22**	**48.22**	**68.56**	**57.67**	**55.22**	**54.00**	**51.67**

ResNet18&Resnet50&SVM is the method with the highest accuracy rate (73.30%) in Table 7.32. While e0 is the class classified with the highest accuracy rate (72.22%) in this dataset, e1 is the class classified with the lowest rate (44.78%).

Table 7.32 Results for c3r1

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	61.80	70.00	28.00	52.00	62.00	72.00	72.00	68.00	70.00
	AUC	0.94								
ResNet18&Resnet50&KNN	**Acc (%)**	62.50	58.00	40.00	58.00	72.00	68.00	66.00	66.00	72.00
	AUC	0.77								
ResNet18&Resnet50&SVM	**Acc (%)**	**73.30**	72.00	60.00	56.00	80.00	88.00	74.00	82.00	74.00
	AUC	0.95								
ResNet18&GoogLeNet&EL	**Acc (%)**	56.30	50.00	34.00	36.00	42.00	64.00	62.00	86.00	76.00
	AUC	0.83								
ResNet18&GoogLeNet&KNN	**Acc (%)**	60.00	66.00	54.00	44.00	68.00	46.00	56.00	72.00	74.00
	AUC	0.79								
ResNet18&GoogLeNet&SVM	**Acc (%)**	72.30	80.00	74.00	66.00	70.00	82.00	70.00	64.00	72.00
	AUC	0.96								
ResNet18&AlexNet&EL	**Acc (%)**	58.00	68.00	34.00	32.00	56.00	64.00	60.00	78.00	72.00
	AUC	0.90								
ResNet18&AlexNet&KNN	**Acc (%)**	53.8	82.00	38.00	40.00	58.00	60.00	24.00	62.00	66.00
	AUC	0.84								
ResNet18&AlexNet&SVM	**Acc (%)**	64.3	78.00	50.00	54.00	58.00	82.00	58.00	72.00	62.00
	AUC	0.97								
ResNet50&GoogLeNet&EL	**Acc (%)**	61.30	76.00	28.00	42.00	54.00	76.00	70.00	74.00	70.00
	AUC	0.95								
ResNet50&GoogLeNet&KNN	**Acc (%)**	61.00	66.00	50.00	48.00	76.00	56.00	58.00	62.00	72.00
	AUC	0.80								
ResNet50&GoogLeNet&SVM	**Acc (%)**	71.00	76.00	58.00	68.00	74.00	74.00	72.00	74.00	72.00
	AUC	0.95								
ResNet50&AlexNet&EL	**Acc (%)**	65.30	82.00	32.00	34.00	62.00	76.00	78.00	76.00	82.00
	AUC	0.98								
ResNet50&AlexNet&KNN	**Acc (%)**	53.30	70.00	30.00	50.00	64.00	58.00	32.00	58.00	64.00
	AUC	0.79								
ResNet50&AlexNet&SVM	**Acc (%)**	68.80	78.00	62.00	54.00	62.00	86.00	64.00	76.00	68.00
	AUC	0.97								
GoogLeNet&AlexNet&EL	**Acc (%)**	60.00	70.00	26.00	36.00	58.00	70.00	64.00	76.00	80.00
	AUC	0.92								
GoogLeNet&AlexNet&KNN	**Acc (%)**	54.00	76.00	46.00	48.00	62.00	46.00	30.00	54.00	70.00
	AUC	0.81								
GoogLeNet&AlexNet&SVM	**Acc (%)**	68.5	82.00	62.00	58.00	60.00	74.00	70.00	70.00	72.00
	AUC	0.98								
AVERAGE ACCURACY (%)		**62.53**	**72.22**	**44.78**	**48.67**	**63.22**	**69.00**	**60.00**	**70.56**	**71.56**

In Table 7.33, ResNet50&GoogLeNet&SVM has the highest accuracy rate (60.30%), while GoogLeNet &AlexNet&KNN has the lowest accuracy rate (37.50%). In this dataset, e5 is the class most successfully classified (67.33%) among the eight classes, while e2 is the class with the lowest success (35.11%).

Table 7.33 Results for c3r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	57.50	68.00	62.00	24.00	52.00	72.00	84.00	58.00	40.00
	AUC	0.89								
ResNet18&Resnet50&KNN	**Acc (%)**	51.70	58.00	26.00	42.00	30.00	68.00	76.00	60.00	54.00
	AUC	0.70								
ResNet18&Resnet50&SVM	**Acc (%)**	59.30	68.00	48.00	42.00	56.00	70.00	74.00	68.00	48.00
	AUC	0.90								
ResNet18&GoogLeNet&EL	**Acc (%)**	48.30	22.00	26.00	36.00	36.00	76.00	76.00	76.00	38.00
	AUC	0.71								
ResNet18&GoogLeNet&KNN	**Acc (%)**	47.50	40.00	22.00	44.00	32.00	56.00	72.00	60.00	54.00
	AUC	0.61								
ResNet18&GoogLeNet&SVM	**Acc (%)**	56.80	52.00	54.00	48.00	42.00	68.00	72.00	68.00	50.00
	AUC	0.84								
ResNet18&AlexNet&EL	**Acc (%)**	49.80	50.00	38.00	28.00	34.00	62.00	72.00	74.00	40.00
	AUC	0.78								
ResNet18&AlexNet&KNN	**Acc (%)**	38.00	58.00	26.00	30.00	20.00	44.00	32.00	54.00	40.00
	AUC	0.66								
ResNet18&AlexNet&SVM	**Acc (%)**	44.80	32.00	32.00	34.00	32.00	70.00	60.00	62.00	36.00
	AUC	0.75								
ResNet50&GoogLeNet&EL	**Acc (%)**	58.30	68.00	64.00	24.00	52.00	60.00	86.00	66.00	46.00
	AUC	0.90								
ResNet50&GoogLeNet&KNN	**Acc (%)**	46.30	42.00	24.00	38.00	28.00	54.00	74.00	62.00	48.00
	AUC	0.62								
ResNet50&GoogLeNet&SVM	**Acc (%)**	**60.30**	68.00	50.00	52.00	56.00	64.00	74.00	66.00	52.00
	AUC	0.89								
ResNet50&AlexNet&EL	**Acc (%)**	59.50	76.00	68.00	26.00	48.00	68.00	86.00	54.00	50.00
	AUC	0.91								
ResNet50&AlexNet&KNN	**Acc (%)**	38.50	48.00	24.00	26.00	26.00	42.00	40.00	54.00	48.00
	AUC	0.62								
ResNet50&AlexNet&SVM	**Acc (%)**	52.80	52.00	50.00	40.00	38.00	70.00	62.00	64.00	46.00
	AUC	0.85								
GoogLeNet&AlexNet&EL	**Acc (%)**	53.50	40.00	46.00	38.00	36.00	70.00	74.00	74.00	50.00
	AUC	0.79								
GoogLeNet&AlexNet&KNN	**Acc (%)**	37.50	58.00	22.00	26.00	26.00	40.00	36.00	52.00	40.00
	AUC	0.66								
GoogLeNet&AlexNet&SVM	**Acc (%)**	47.00	32.00	36.00	34.00	36.00	72.00	62.00	66.00	38.00
	AUC	0.77								
AVERAGE ACCURACY (%)		**50.41**	**51.78**	**39.89**	**35.11**	**37.78**	**62.56**	**67.33**	**63.22**	**45.44**

According to Table 7.34, ResNet18& ResNet50&SVM is the best method (60.50%) for the c4r1 dataset of Tilda. The classification performance of e3 is greatest (72.89%) compared to other classes. This dataset's most challenging classes are e0 and e4 with the accuracy rates of 27.89% and 23.11%, respectively.

Table 7.34 Results for c4r1

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	50.50	18.00	52.00	62.00	88.00	32.00	64.00	34.00	54.00
	AUC	0.64								
ResNet18&Resnet50&KNN	**Acc (%)**	54.30	28.00	58.00	78.00	84.00	32.00	40.00	42.00	72.00
	AUC	0.58								
ResNet18&Resnet50&SVM	**Acc (%)**	**60.50**	42.00	68.00	74.00	96.00	40.00	62.00	48.00	54.00
	AUC	0.73								
ResNet18&GoogLeNet&EL	**Acc (%)**	42.30	22.00	38.00	44.00	78.00	26.00	58.00	26.00	46.00
	AUC	0.63								
ResNet18&GoogLeNet&KNN	**Acc (%)**	45.30	16.00	56.00	70.00	66.00	18.00	46.00	42.00	48.00
	AUC	0.53								
ResNet18&GoogLeNet&SVM	**Acc (%)**	53.00	34.00	64.00	56.00	92.00	34.00	54.00	44.00	46.00
	AUC	0.69								
ResNet18&AlexNet&EL	**Acc (%)**	41.3	4.00	46.00	62.00	78.00	28.00	46.00	24.00	42.00
	AUC	0.57								
ResNet18&AlexNet&KNN	**Acc (%)**	23.30	26.00	12.00	76.00	14.00	10.00	8.00	18.00	22.00
	AUC	0.50								
ResNet18&AlexNet&SVM	**Acc (%)**	37.50	24.00	50.00	42.00	78.00	14.00	48.00	22.00	22.00
	AUC	0.60								
ResNet50&GoogLeNet&EL	**Acc (%)**	51.00	32.00	52.00	64.00	78.00	34.00	50.00	40.00	58.00
	AUC	0.70								
ResNet50&GoogLeNet&KNN	**Acc (%)**	50.50	30.00	54.00	88.00	84.00	18.00	38.00	40.00	52.00
	AUC	0.60								
ResNet50&GoogLeNet&SVM	**Acc (%)**	58.30	42.00	72.00	72.00	96.00	38.00	54.00	42.00	50.00
	AUC	0.73								
ResNet50&AlexNet&EL	**Acc (%)**	48.80	20.00	62.00	62.00	84.00	16.00	54.00	36.00	56.00
	AUC	0.62								
ResNet50&AlexNet&KNN	**Acc (%)**	32.80	30.00	36.00	64.00	42.00	8.00	16.00	32.00	34.00
	AUC	0.58								
ResNet50&AlexNet&SVM	**Acc (%)**	47.50	36.00	58.00	54.00	84.00	18.00	58.00	38.00	34.00
	AUC	0.62								
GoogLeNet&AlexNet&EL	**Acc (%)**	38.30	24.00	46.00	60.00	66.00	24.00	18.00	28.00	40.00
	AUC	0.64								
GoogLeNet&AlexNet&KNN	**Acc (%)**	25.30	32.00	8.00	68.00	22.00	10.00	14.00	26.00	22.00
	AUC	0.53								
GoogLeNet&AlexNet&SVM	**Acc (%)**	42.00	42.00	48.00	54.00	82.00	16.00	44.00	22.00	28.00
	AUC	0.63								
AVERAGE ACCURACY (%)		**44.58**	**27.89**	**48.89**	**63.89**	**72.89**	**23.11**	**42.89**	**33.56**	**43.33**

ResNet18&Resnet50&SVM has the highest accuracy rate (61.70%) while GoogLeNet&AlexNet&EL has the lowest accuracy rate (34.30%) in Table 7.35. On the other hand, e0 is the class with the highest success (61.00%), while e2 is the class with the lowest success (29.56%).

Table 7.35 Results for c4r3

METHOD	GENERAL PERFORMANCE		CLASS BASED PERFORMANCE (%)							
			e0	e1	e2	e3	e4	e5	e6	e7
ResNet18&Resnet50&EL	**Acc (%)**	50.10	68.00	68.00	32.00	42.00	40.00	63.00	46.00	42.00
	AUC	0.89								
ResNet18&Resnet50&KNN	**Acc (%)**	52.60	76.00	58.00	38.00	60.00	52.00	35.00	52.00	50.00
	AUC	0.82								
ResNet18&Resnet50&SVM	**Acc (%)**	**61.70**	82.00	68.00	44.00	78.00	48.00	67.00	64.00	42.00
	AUC	0.97								
ResNet18&GoogLeNet&EL	**Acc (%)**	45.40	50.00	54.00	20.00	56.00	30.00	71.00	38.00	44.00
	AUC	0.80								
ResNet18&GoogLeNet&KNN	**Acc (%)**	50.90	72.00	56.00	34.00	72.00	50.00	22.00	52.00	48.00
	AUC	0.81								
ResNet18&GoogLeNet&SVM	**Acc (%)**	54.90	70.00	52.00	30.00	82.00	52.00	65.00	50.00	38.00
	AUC	0.93								
ResNet18&AlexNet&EL	**Acc (%)**	40.10	44.00	46.00	24.00	60.00	20.00	71.00	20.00	36.00
	AUC	0.75								
ResNet18&AlexNet&KNN	**Acc (%)**	38.80	40.00	26.00	30.00	50.00	56.00	20.00	44.00	44.00
	AUC	0.64								
ResNet18&AlexNet&SVM	**Acc (%)**	38.30	50.00	38.00	30.00	50.00	28.00	43.00	38.00	30.00
	AUC	0.80								
ResNet50&GoogLeNet&EL	**Acc (%)**	45.90	62.00	50.00	36.00	36.00	32.00	61.00	40.00	50.00
	AUC	0.86								
ResNet50&GoogLeNet&KNN	**Acc (%)**	51.60	86.00	54.00	34.00	66.00	48.00	29.00	50.00	46.00
	AUC	0.87								
ResNet50&GoogLeNet&SVM	**Acc (%)**	59.60	78.00	70.00	38.00	76.00	46.00	71.00	56.00	42.00
	AUC	0.96								
ResNet50&AlexNet&EL	**Acc (%)**	42.90	70.00	64.00	32.00	42.00	40.00	59.00	32.00	44.00
	AUC	0.88								
ResNet50&AlexNet&KNN	**Acc (%)**	40.10	52.00	36.00	20.00	52.00	54.00	20.00	48.00	38.00
	AUC	0.69								
ResNet50&AlexNet&SVM	**Acc (%)**	49.10	72.00	48.00	34.00	60.00	48.00	53.00	50.00	28.00
	AUC	0.91								
GoogLeNet&AlexNet&EL	**Acc (%)**	34.30	42.00	36.00	16.00	38.00	26.00	51.00	30.00	36.00
	AUC	0.75								
GoogLeNet&AlexNet&KNN	**Acc (%)**	36.80	40.00	26.00	22.00	50.00	52.00	16.00	50.00	38.00
	AUC	0.64								
GoogLeNet&AlexNet&SVM	**Acc (%)**	37.10	44.00	44.00	18.00	46.00	30.00	35.00	50.00	30.00
	AUC	0.79								
AVERAGE ACCURACY (%)		**46.12**	**61.00**	**49.67**	**29.56**	**56.44**	**41.78**	**47.33**	**45.00**	**40.33**

In terms of accuracy rates, ResNet18&Resnet50&SVM performs the best (67.18%) when compared to the performances of other methods (Table 7.36). On the other hand, it is seen that ResNet18&AlexNet&KNN has the lowest performance (39.19%). If a classifier-based comparison is made, the highest average is obtained in SVM-based classification (58.36%). 48.61% and 55.75% of classification accuracy rates have been obtained for the classifiers of KNN and EL, respectively.

Table 7.36 Average accuracy rates

AVERAGE ACCURACY RATES (%)					
ResNet18&Resnet50&EL	60.01	**ResNet18&AlexNet&EL**	52.60	**ResNet50&AlexNet&EL**	59.99
ResNet18&Resnet50&KNN	58.00	**ResNet18&AlexNet&KNN**	39.19	**ResNet50&AlexNet&KNN**	41.03
ResNet18&Resnet50&SVM	**67.18**	**ResNet18&AlexNet&SVM**	50.89	**ResNet50&AlexNet&SVM**	56.95
ResNet18&GoogLeNet&EL	53.33	**ResNet50&GoogLeNet&EL**	59.98	**GoogLeNet&AlexNet&EL**	51.59
ResNet18&GoogLeNet&KNN	57.01	**ResNet50&GoogLeNet&KNN**	57.05	**GoogLeNet&AlexNet&KNN**	39.38
ResNet18&GoogLeNet&SVM	64.95	**ResNet50&GoogLeNet&SVM**	66.25	**GoogLeNet&AlexNet&SVM**	53.21

Summary graphic for case 4 is in Figure 7.9. While c1r1 is the dataset with the highest classification success (66.93%), c4r1 is the dataset with the lowest classification success (44.6%).

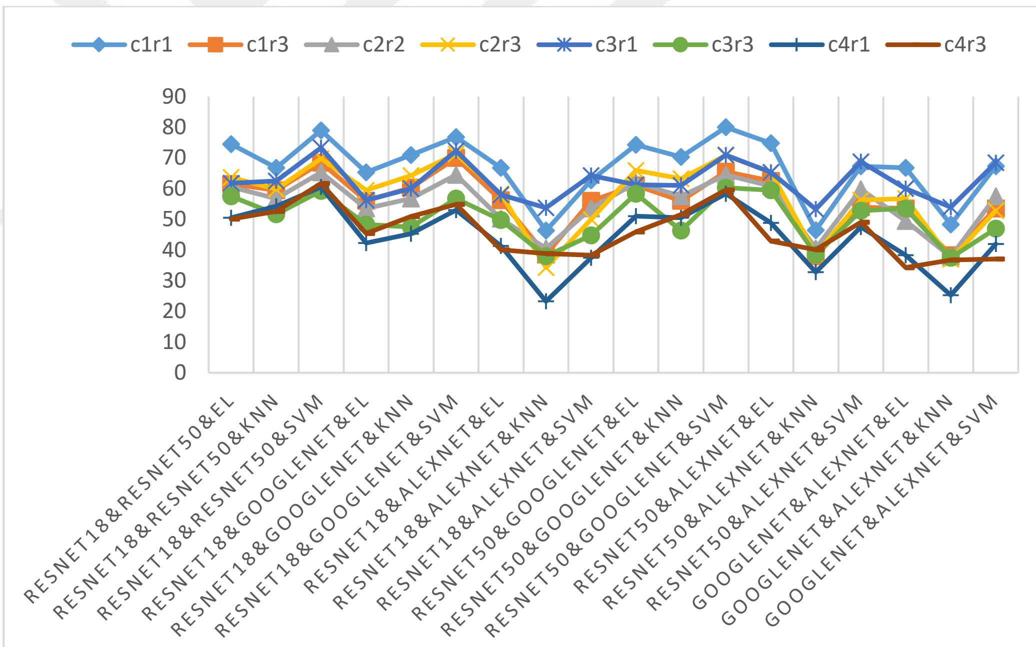


Figure 7.9 Summary for case 4

Average times to classify the features are given in Table 7.37. The EL-based classification time is 449.83 seconds on average. The KNN-based classification time is 44.44 seconds on average. The SVM based classification time is 74.12 seconds on average. It is seen that the classification time of EL is approximately 10 times longer than KNN and approximately 6 times longer than SVM.

Table 7.37 Classification times (seconds)

	DATASET							
METHOD	**c1r1**	**c1r3**	**c2r2**	**c2r3**	**c3r1**	**c3r3**	**c4r1**	**c4r3**
ResNet18&Resnet50&EL	231.10	242.58	224.33	261.33	252.85	233.20	302.36	446.85
ResNet18&Resnet50&KNN	12.38	12.89	9.28	9.28	9.75	13.08	9.24	14.13
ResNet18&Resnet50&SVM	23.78	22.20	19.88	21.87	21.83	21.98	22.32	23.05
ResNet18&GoogLeNet&EL	189.53	178.76	197.30	180.78	196.56	205.25	211.15	243.09
ResNet18&GoogLeNet&KNN	8.20	4.80	4.74	5.24	7.85	4.73	11.29	7.87
ResNet18&GoogLeNet&SVM	13.16	12.26	12.37	12.34	14.86	14.81	15.43	14.17
ResNet18&AlexNet&EL	423.62	118.32	506.69	475.42	431.72	468.47	589.67	805.64
ResNet18&AlexNet&KNN	70.71	68.63	73.93	65.48	71.55	70.55	70.65	62.62
ResNet18&AlexNet&SVM	111.60	118.32	113.37	109.12	114.25	112.27	108.06	105.34
ResNet50&GoogLeNet&EL	277.84	287.12	260.18	277.47	313.64	256.21	334.60	513.38
ResNet50&GoogLeNet&KNN	15.26	14.93	14.91	13.38	14.93	14.74	14.38	12.28
ResNet50&GoogLeNet&SVM	24.77	25.20	26.83	28.62	27.74	27.75	26.28	25.83
ResNet50&AlexNet&EL	490.67	519.97	578.70	629.04	569.36	540.96	687.69	1141.30
ResNet50&AlexNet&KNN	85.75	78.08	88.71	97.33	92.75	81.55	83.71	83.74
ResNet50&AlexNet&SVM	130.16	133.84	140.65	153.35	149.33	122.74	133.65	127.95
GoogLeNet&AlexNet&EL	473.67	450.88	561.71	544.76	442.38	485.71	640.37	958.20
GoogLeNet&AlexNet&KNN	73.49	68.52	82.18	80.26	73.04	76.67	77.04	78.02
GoogLeNet&AlexNet&SVM	110.59	117.23	124.10	131.76	112.21	127.73	130.34	119.74

7.4 Analysis

7.4.1 Analysis for Binary Classification (for Case 1 and Case 3)

In this section, the results of the tested combinations have been also compared with the results of the CNN models. In the models, epoch value was taken 5, and learning rate was taken 0.001. In order for the results to be comparable, 10-fold cross validation was used in these models as well. The confusion matrices of ResNet18 have been given in Figure 7.10. The results of the CNN models have been also given for 10-fold cross validation. The target class is shown on the horizontal axis, while the output class is shown on the vertical axis.

Figure 7.10 Confusion matrix for ResNet18 (binary classification)

As shown in Table 7.38, the accuracy rates of ResNet50 and GoogleNet are close to each other (87.87% and 88.07%, respectively). AlexNet's success is the lowest compared to the other three models. The average accuracy rate for the classification of unpatterned fabrics is higher than in patterned and mixed fabrics. The graph showing the success of the models is in Figure 7.11.

According to the classification times (Table 7.39), ResNet50 is the model with the longest time to result (8774.67 seconds on average). The model that results in the shortest time is AlexNet (849 seconds on average).

Table 7.38 Accuracy rates of CNN models for binary classification (%)

METHOD	DATASET			AVERAGE
	Un-patterned Fabrics	Patterned Fabrics	Mix	
ResNet18	85.25	88.30	85.31	**86.29**
Resnet50	91.10	87.50	85.00	**87.87**
GoogLeNet	88.90	87.60	87.70	**88.07**
AlexNet	86.80	83.10	86.80	**85.57**
AVERAGE	**88.01**	**86.63**	**86.20**	

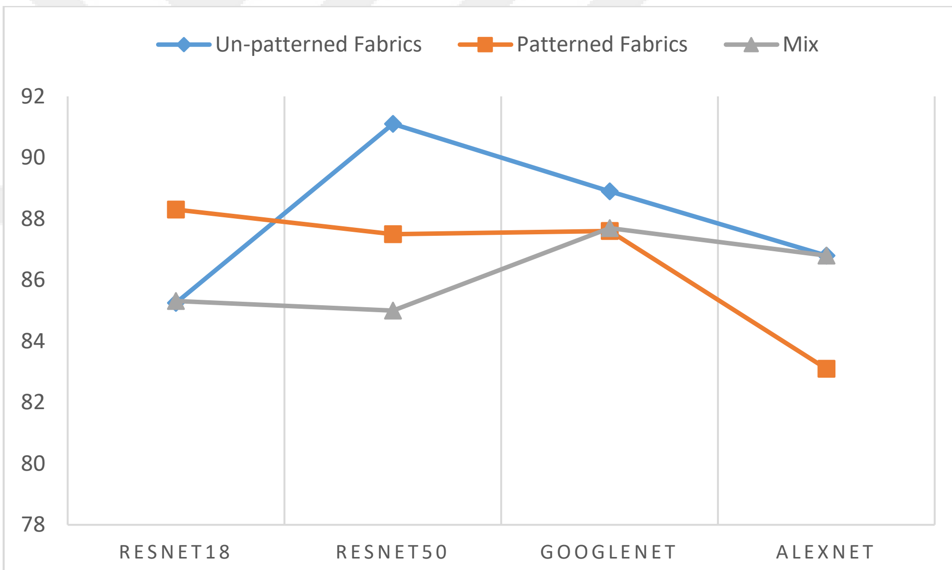


Figure 7.11 Summary of CNN models for 2-class classification

Table 7.39 Classification times of CNN models for binary classification (seconds)

METHOD	DATASET			AVERAGE
	Un-patterned Fabrics	Patterned Fabrics	Mix	
ResNet18	1530	1452	3033	**2005.00**
Resnet50	8408	9824	8092	**8774.67**
GoogLeNet	2635	2617	5019	**3423.67**
AlexNet	641	635	1271	**849.00**

The comparison of case 1 and case 3 with CNN models is given in Table 7.40. It can be observed that the success of case 3 in patterned fabrics (86.72%) is slightly higher than the success of case 1 (86.40%) and CNN models (86.63%) for patterned fabrics. On the other hand, the success of CNN is higher than the successes of case 1 and case 3 in other datasets (patterned and mix).

Since a feature fusion is created by combining the features of the two models in case 3, the classification time in case 3 is approximately twice the classification time of case 1.

Table 7.40 Comparison for binary classification

Comparison	**Methods**	**Results**
Average accuracy rates for unpatterned fabrics	CNN	88.01%
	Case 1	86.20%
	Case 3	85.39%
Average accuracy rates for patterned fabrics	CNN	86.63%
	Case 1	86.40%
	Case 3	86.72%
Average accuracy rates for mix fabrics	CNN	86.20%
	Case 1	85.75%
	Case 3	85.54%
Dataset classified with highest success	CNN	unpatterned
	Case 1	patterned
	Case 3	patterned
Dataset classified with lowest success	CNN	mix
	Case 1	mix
	Case 3	unpatterned
Average time	CNN	3763.08 sec
	Case 1	213.46 sec
	Case 3	448.74 sec

7.4.2 Analysis for 8-Class Classification (for Case 2 and Case 4)

The outcomes of the tested combinations have also been compared with the outcomes of the CNN models in this section (Like section 7.4.5.1). Epoch value was set to 5 in the models, and learning rate was set at 0.001. Additionally, 10-fold cross validation was applied to these models to ensure that the outcomes could be compared. Figure 7.12 contains the confusion matrices of ResNet18. The vertical axis indicates the output class, and the horizontal axis indicates the target class. Table 7.41 displays the results of CNN models. ResNet18 has the greatest accuracy rate (69.89%) compared to other models (ResNet50, GoogLeNet, AlexNet). The second most successful method is ResNet50 (65.75%). The averages of GoogLeNet and AlexNet are quite low (48.86%, and 41.89%, respectively). On the other hand, when dataset-based comparison is made, it is seen that c1r1 is the dataset with the highest average accuracy rate (Figure 7.13). When this dataset is classified using ResNet18, approximately 80% success is achieved.

The average classification accuracy rate of the first four datasets with unpatterned samples (c1r1, c1r3, c2r2, c2r3) is 61.74%, while the average classification success of the last four datasets with patterned samples (c3r1, c3r3, c4r1, c4r3) is 51.46%.

The classification times of CNN models are shown in Table 7.42. AlexNet is the model with the lowest classification time (350 seconds), while ResNet50 has the longest classification time (2203.75 seconds).

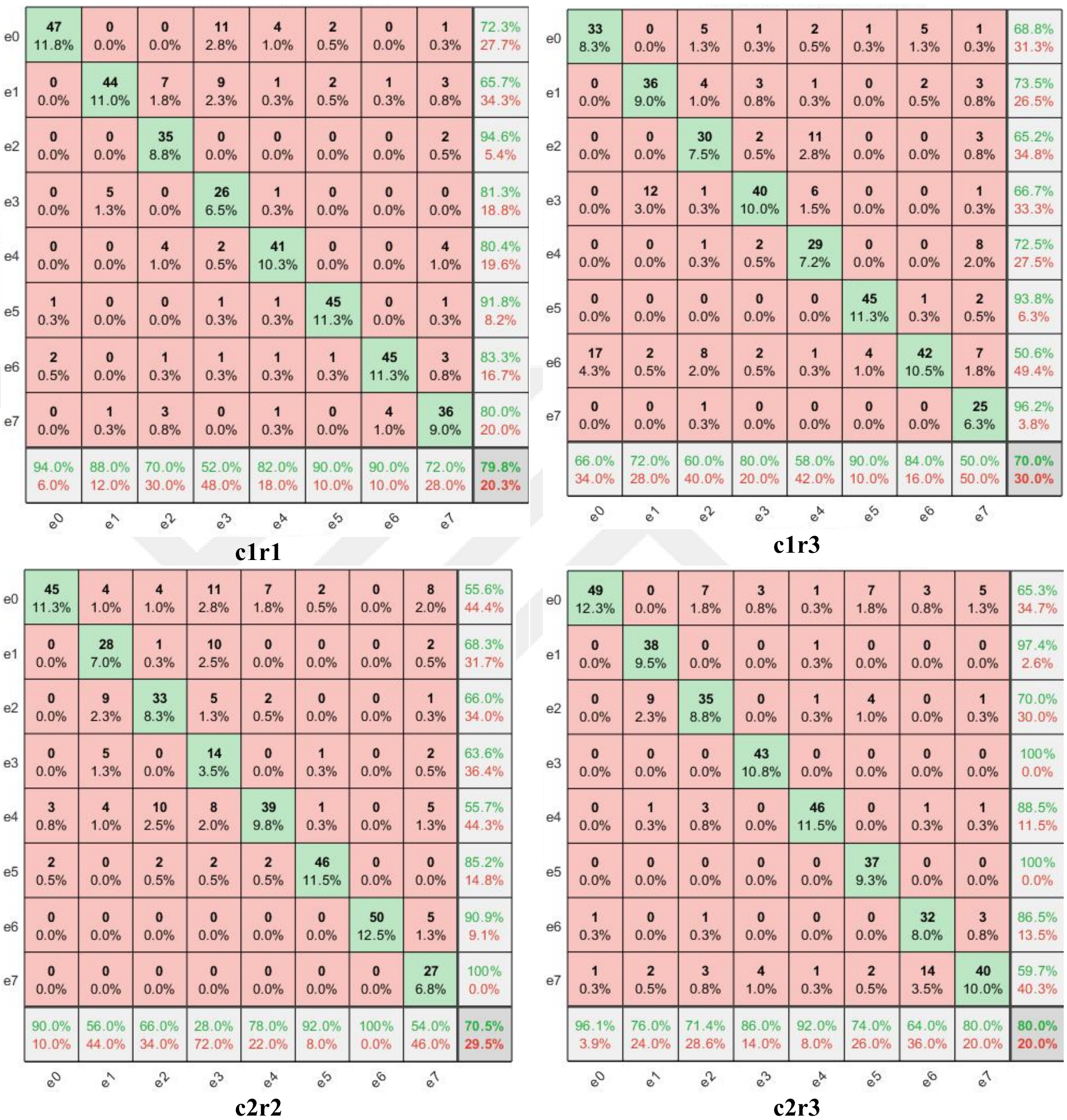
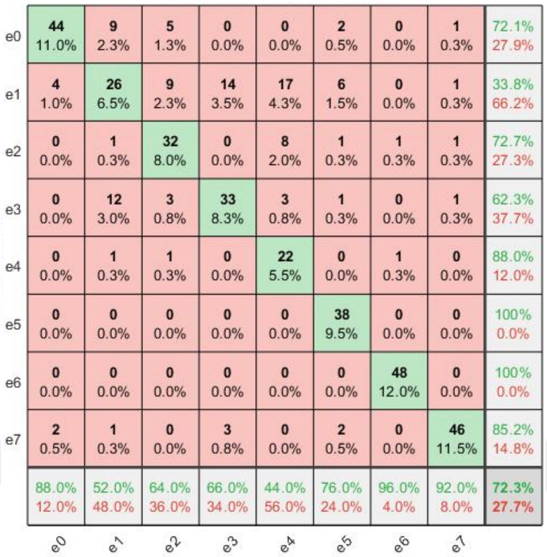
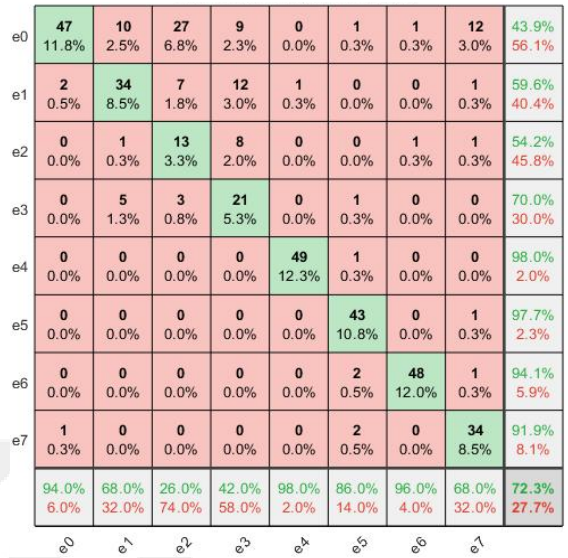


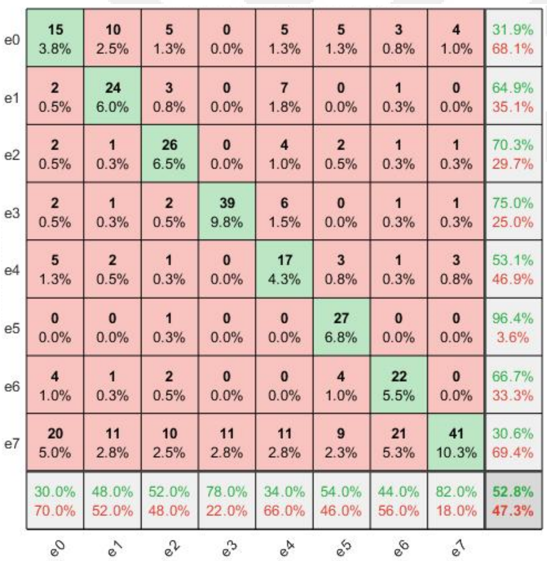
Figure 7.12 Confusion matrix for ResNet18 (8-class classification)



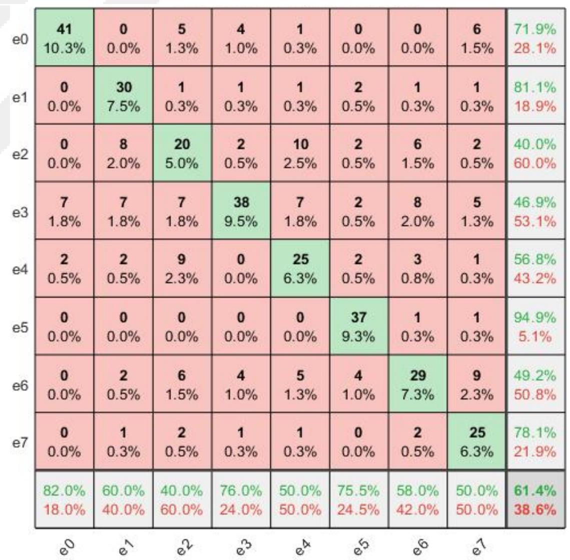
c3r1



c3r3



c4r1



c4r3

Figure 7.12 Continues

Table 7.41 Accuracy rates of CNN models for 8-class classification (%)

	DATASETS								
METHOD	**c1r1**	**c1r3**	**c2r2**	**c2r3**	**c3r1**	**c3r3**	**c4r1**	**c4r3**	**AVERAGE**
ResNet18	79.80	70.00	70.50	80.00	72.30	72.30	52.80	61.40	**69.89**
Resnet50	74.30	63.50	70.50	71.00	72.50	64.30	54.30	55.60	**65.75**
GoogLeNet	63.25	54.00	58.30	47.50	51.50	42.50	40.50	33.33	**48.86**
AlexNet	48.80	44.25	46.00	46.00	51.20	42.75	31.00	25.10	**41.89**
AVERAGE	**66.54**	**57.94**	**61.33**	**61.13**	**61.88**	**55.46**	**44.65**	**43.86**	

Figure 7.13 Summary of CNN models for 8-class classification

Table 7.42 Classification times of CNN models for 8-class classification (seconds)

	DATASETS								
METHOD	**c1r1**	**c1r3**	**c2r2**	**c2r3**	**c3r1**	**c3r3**	**c4r1**	**c4r3**	**AVERAGE**
ResNet18	1580	1570	1630	1640	1620	1630	1670	1550	**1611.25**
ResNet50	2270	2090	2210	2250	2280	2250	2180	2100	**2203.75**
GoogLeNet	1350	1330	1300	1310	1300	1320	1280	1400	**1323.75**
AlexNet	390	330	350	350	360	340	350	330	**350.00**

There can be some intriguing conclusions made. The average accuracy rates for the first four datasets (c1r1, c1r3, c2r2, and c2r3) are 58.65% for case 2 and 57.15% for case 4, while the average accuracy rates for the next four datasets (c3r1, c3r3, c4r1, and c4r3) are 50.91% for case 2 and 48.37% for case 4. Similar results were obtained when examining the results of CNN models. In the light of this information, it can be concluded that the defects in the unpatterned fabric samples are more easily classified than the defects in the patterned fabrics.

C1r1 is the dataset classified with the highest success on average for case4 and CNN models. When the maximum performances obtained for the c1r1 data set are examined, classification is achieved with 80% success in case 4, 77.5% in case 2 and 79.8% in CNN models.

Now that the results of the CNN models have been examined, they can now be compared with the results of this study. Comparison of them are given in Table 7.43.

Table 7.43 Comparison for 8-class classification

Comparison	**Methods**	**Results**
Average accuracy rates for unpatterned fabrics	CNN	61.74%
	Case 2	58.65%
	Case 4	57.15%
Average accuracy rates for patterned fabrics	CNN	51.46%
	Case 2	50.91%
	Case 4	48.37%
Dataset classified with highest success	CNN	c1r1 (66.54%)
	Case 2	c3r1 (58.87%)
	Case 4	c1r1 (66.93%)
Dataset classified with lowest success	CNN	c4r3 (43.86%)
	Case 2	c4r1 (43.23%)
	Case 4	c4r1 (44.58%)
Average time	CNN	1372 sec
	Case 2	95 sec
	Case 4	189 sec

CHAPTER EIGHT

CONCLUSION AND FUTURE WORK

8.1 Conclusion

The decision on the average quality of the fabric rolls is based on the number of defects detected per unit fabric area. Traditionally, fabric defect control is based on humanpower. On the basis of the training he/she received and the experience he/she has accrued, an experienced and specifically trained personel may identify the visible defects in a fabric (Ala & İkiz, 2014). He or she can then rectify any defects discovered or mark them for future correction. Although the process has highly cost, it is a control type that does not achieve high success, as was mentioned in this study. For this reason, the idea of automating this process has emerged and studies have been started in this area. Given the recent advancements in technology, automated fabric defect detection systems have garnered a lot of attention for a number of reasons, including improved product quality.

Especially in recent years, the use of CNN-based models has been very popular not only in this field but also in all fields. It is obvious that CNN has advantages as well as disadvantages. However, the disadvantages of CNN are not enough to stain the popularity of CNN. In the literature, it is seen that CNN models developed by considering the weak points of CNN models are presented. This thesis aims to present a system that will give results in less time and is at least as successful as CNN. For these purposes, many cases have been created and a comprehensive analysis has been carried out for these cases.

The lack of a database containing fabric defects and the difficulties encountered in creating a new database led us to use the Tilda database, which is the only database open to access in this field. The study consists of four cases in which the data sets in Tilda are handled differently. For case 1 and case 3, subdirectories other than e0 have been collected in one directory, since fabric samples have been tried to be classified as defected or non-defected. As a consequence, for un-patterned fabrics, we have 200 non-defected images and 1400 defected images. On the other hand, the number of images we have for patterned fabrics is the same as the number of images for un-patterned fabrics. In addition to all of these, the study looked into the accuracy of classification in a dataset that included both patterned and un-patterned fabrics. This dataset is called 'mix' and it consists of 400 un-defected images and 2800 defected images from the preceding two datasets. Images with and without patterns, as well as various textures, have all been identified as defected or un-defected in this way. For case 2 and case 4, sets of Tilda database have been used as they are. The performances of eight sets (c1r1, c1r3, c2r2, c2r3, c3r1, c3r3, c4r1, c4r3) have been analyzed and compared separately. One feature extraction method and one classifier are used in case 1 and case 3, while fusion features are obtained and classified in case 2 and case 4.

Different approaches using four CNN-based models such as ResNet18, ResNet50, GoogLeNet, and AlexNet have been used in feature extraction step, while EL, KNN, and SVM have been used in classification step. In this study, unlike other studies in the field of fabric defect classification, feature fusion has been used for feature extraction. In feature fusion approach, binary combinations of ResNet18, ResNet50, GoogLeNet, and AlexNet (ResNet18&ResNet50, ResNet18&GoogLeNet, ResNet18&AlexNet, ResNet50&GoogLeNet, ResNet50&AlexNet, GoogLeNet&AlexNet) have been preferred.

The methods used in case 1 and case 2 are as follows:

- ResNet18&EL
- ResNet18&KNN
- ResNet18&SVM
- ResNet50&EL
- ResNet50&KNN
- ResNet50&SVM

- GoogLeNet&EL
- GoogLeNet&KNN
- GoogLeNet&SVM
- AlexNet&EL
- AlexNet&KNN
- AlexNet&SVM

The methods used in case 3 and case 4 are as follows:

- ResNet18&Resnet50&EL
- ResNet18&Resnet50&KNN
- ResNet18&Resnet50&SVM
- ResNet18&GoogLeNet&EL
- ResNet18&GoogLeNet&KNN
- ResNet18&GoogLeNet&SVM
- ResNet18&AlexNet&EL
- ResNet18&AlexNet&KNN
- ResNet18&AlexNet&SVM
- ResNet50&GoogLeNet&EL
- ResNet50&GoogLeNet&KNN
- ResNet50&GoogLeNet&SVM
- ResNet50&AlexNet&EL
- ResNet50&AlexNet&KNN
- ResNet50&AlexNet&SVM
- GoogLeNet&AlexNet&EL
- GoogLeNet&AlexNet&KNN
- GoogLeNet&AlexNet&SVM

If we come to the conclusions of the cases mentioned above:

- ResNet50 & SVM has the highest performance in the first case. Considering their average performances, EL and SVM have close values while KNN's value is low. Additionally, it is seen that the average specificity value of KNN is about half of the specificity values of other classifiers in case 1, and less than half of the specificity values of other classifiers in case 3. KNN is a classifier that is affected by the number of samples, sensitive to variables, and therefore not robust.

- In the second case, the methods are not as successful as in case 1. ResNet50&SVM has been the highest performing method in case 2 as well as case 1. SVM is the most efficient classifier.

- In the third case, ResNet50&GoogLeNet&SVM achieved over 90% on average. SVM is the highest performing classifier. It is seen that the average specificity value of KNN is less than half of the specificity values of other classifiers (as in case 1). EL and SVM are less impacted by the quantity of samples than KNN, indicating that they are more robust classifiers.

- In the next case, ResNet18&Resnet50&SVM has the highest performance compared to others. ResNet50 GoogLeNet&SVM is the second most successful method. The most effective classifier is SVM.

To summarize, it was concluded that the methods tested in all cases gave results close to CNN. Time comparisons reveal that fusion methods (used in case 3 and case 4) produce results more slowly than CNN&machine learning combination methods used in cases 1 and 2, but that this delay is incredibly minimal when compared to the completion times of CNN methods. In this case, machine learning-based classification is more preferable after CNN-based feature extraction instead of CNN-based classification, considering the time advantage. Additionally, classification success of un-patterned fabrics is remarkably higher than that of patterned fabrics. Therefore, it would be appropriate to make improvements to increase success in detecting defects on patterned fabrics. Moreover, when the success rates in case 2 and case 4 are considered, the low rates are striking. Higher rates may result from doing classification once the images have undergone the proper preprocessing.

8.2 Future Work

The following issues can be solved in the future:

- It is aimed to increase the performance by increasing the number of samples by using augmentation techniques.

- It is aimed to shorten the time by applying feature selection methods to the features obtained from CNN-based models.

- Converting the system into a product as a real-time defect checking machine will close the gap in this area and will be economical as it is a domestic solution.

- It is obvious that there is a dataset problem in this area. In the future, it is aimed to create a large database containing a large number of fabric defect samples.

REFERENCES

Abdellah, H., Ahmed, R., & Slimane, O. (2014). Defect detection and identification in textile fabric by SVM method. *IOSR Journal of Engineering, 4*(12), 69-77.

Ala, D. M., & İkiz, Y. (2014). A statistical investigation into the determining woven fabric defects that occur on raw terry fabrics during weaving production. *Pamukkale University Journal of Engineering Sciences, 20*(7), 244-252.

Ala, D.M., & İkiz, Y. (2015). A statistical investigation for determining fabric defects that occur during production. *Pamukkale University Journal of Engineering Sciences, 21*(7), 282-287.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:*1803.01164.

Ananthavaram, R. R., Rao, O. S., & Prasad, M. H. M. K. (2012). Automatic defect detection of patterned fabric by using RB method and independent component analysis. *International Journal of Computer Applications, 39*(18), 52-56.

Bağkur, S. (2013). *Fabric defect detection using image processing techniques* [MSc Thesis]. Dokuz Eylül University.

Baran, M. (2020). *Classification of multi-label data with machine learning methods* [MSc Thesis]. Sivas Cumhuriyet University.

Barış, B., Özek, H. Z. (2019). Dokuma kumaş hatalarının sistematik sınıflandırılması üzerine bir çalışma. *Tekstil ve Mühendis, 26*(114), 156-167.

Basibuyuk, K., Coban, K., & Ertuzun, A. (2008, March). *Model based defect detection problem: Particle filter approach.* In 2008 3rd International Symposium on Communications, Control and Signal Processing (pp. 348-351).

Basturk A., Ketencioglu H., Yugnak Z., Yuksel M.E. (2007). Inspection of defects in fabrics using gabor wavelets and principle component analysis. *In 2007 9th International Symposium on Signal Processing and Its Applications*, 1-4.

Baykal, A. (2006). Application fields of data mining. *Dicle University Journal of Ziya Gökalp Faculty of Education, 7*, 95-107.

Behera B.K, & Mani M.P. (2007). Characterization and classification of fabric defects using discrete cosine transformation and artificial neural network. *Indian Journal of Fibre & Textile Research*, *32*, 421-426.

Ben Salem, Y., & Nasri, S. (2011). Woven fabric defects detection based on texture classification algorithm. *8th International Multi-Conference on Systems, Signals & Devices*.

Bissi, L., Baruffa, G., Placidi, P., Ricci, E., Scorzoni, A., & Valigi, P. (2013). Automated defect detection in uniform and structured fabrics using Gabor filters and PCA. *Journal of Visual Communication and Image Representation, 24*(7), 838-845.

Büyükkabasakal, K. (2010). *Detection and classification of fabric defects* [MSc Thesis]. Ege University.

Campbell, J.G., Fraley, C., Stanford, D., Murtagh, F., & Raftery, A.E. (1999). Model-based methods for textile fault detection. *International Journal of Imaging Systems and Technology, 10*, 339-346.

Cao, J., Zhang, J., Wen, Z., Wang, N., & Liu, X. (2017). Fabric defect inspection using prior knowledge guided least squares regression. *Multimedia Tools and Applications, 76*(3), 4141-4157.

Chan, C., & Pang, G.K.H. (2000). Fabric defect detection by fourier analysis. *IEEE Transactions on Industry Applications, 36*(5).

Chang, X., Gu, C., Liang, J., & Xu, X. (2018). Fabric defect detection based on pattern template correction. *Mathematical Problems in Engineering, 2018*, 1-17.

Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015, March). An empirical study of distance metrics for k-nearest neighbor algorithm. *Proceedings of the 3rd international conference on industrial application engineering*, 280-285.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20*, 1-25.

Cuifang, Z., Yu, C., & Jiacheng, M. (2020). Fabric defect detection algorithm based on PHOG and SVM. *Indian Journal of Fibre & Textile Research (IJFTR), 45*(1), 123-126.

Çelik, H.İ., Dülger, L.C., & Topalbekiroğlu, M. (2014). Fabric defect detection using linear filtering and morphological operations. *Indian Journal of Fibre & Textile Research, 39*, 254-259.

Dongli, T., Zhitao, X., Fang, Z., Lei, G., & Jun, W. (2013). Cloth defect classification method based on SVM. *International Journal of Digital Content Technology and its Applications, 7*(3), 614-622.

Dorrity, L., Vachtsevanos, G., & Jasper, W. (1995). Real-time fabric defect detection and control in weaving processes. *National Textile Center*, G94-2, 143-152.

Dülgeroğlu Kısaoğlu, O. (2010). A statistical quality control system in a medium-scale weaving mill: I. control of fabric defects. *Pamukkale University Journal of Engineering Sciences, 16*(3), 291-301.

Faouzi, S. (2014). Classification of fabric defects using image analysis and fuzzy c-means method. *6th International Conference of Applied Research in Textile*, 12-15.

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, 96*, 148-156.

Ghosh, A., Guha, T., Bhar, R. B., & Das, S. (2011). Pattern classification of fabric defects using support vector machines. I*nternational Journal of Clothing Science and Technology*, *23*, 142-151.

Goldberg, J.B. (1950). *Fabric defects - Case histories of imperfections in woven cotton and rayon fabrics.* Mcgraw-Hill Book Company, Inc., New York.

Guan, M., Zhong, Z., Rui, Y., Zheng, H., & Wu, X. (2019, September). Defect detection and classification for plain woven fabric based on deep learning. *Seventh International Conference on Advanced Cloud and Big Data*, 297-302.

Habib, M. T., & Rokonuzzaman, M. (2013). An empirical approach to optimize design of backpropagation neural network classifier for textile defect inspection. *Journal of Advances in Mathematics and Computer Science*, 617-634.

Hamdi, A. A., Sayed, M. S., Fouad, M. M., & Hadhoud, M. M. (2016, May). Fully automated approach for patterned fabric defect detection. *Fourth International Japan-Egypt Conference on Electronics, Communications and Computers*, 48-51.

Hamdi, A.A., Sayed, M.S., Fouad, M.M., & Hadhoud, M.M. (2018). Unsupervised patterned fabric defect detection using texture filtering and K-Means clustering. *International Conference on Innovative Trends in Computer Engineering*.

Han, J., Pei, J., Kamber, M., & Pei, J (2011). *Data mining: concepts and techniques*. Elsevier.

Hanbay, K., Talu, M. F., & Özgüven, Ö. F. (2017). Real time fabric defect detection by using fourier transform. *Journal of the Faculty of Engineering and Architecture of Gazi University, 32*(1), 151-158.

Hanbay, K., & Talu, M. (2014). Online/offline detection systems of fabric defects and methods. *Sakarya University Journal of Science, 18*(1), 49-69.

Hanbay, K., Talu, M. F., Özgüven, Ö. F., & Öztürk, D. (2015, May). Fabric defect detection methods for circular knitting machines. *23nd Signal Processing and Communications Applications Conference (SIU)*, 735-738.

Hanbay, K., Talu, M. F., Özgüven, Ö. F., & Öztürk, D. (2019). Real-time detection of knitting fabric defects using shearlet transform. *Tekstil ve Konfeksiyon, 29*(1), 1-10.

He K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition,* 770-778.

Hoang, V. T., & Rebhi, A. (2018, July). On comparing color spaces for fabric defect classification based on local binary patterns. *IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, 297-300.

Hu, M. C., & Tsai, I. S. (2000). Fabric inspection based on best wavelet packet bases. T*extile Research Journal, 70*(8), 662-670.

Huang, C.C., & Chen I.C. (2001). Neural fuzzy classification for fabric defects. *Textile Res. J., 71*(3), 220-224.

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering, 17*(3), 299-310.

Huang, J., Lu, J., & Ling, C. X. (2003, November). Comparing naive bayes, decision trees, and SVM with AUC and accuracy. *Third IEEE International Conference on Data Mining*, 553-556.

Jackson, J. (2002). Data mining; a conceptual overview. *Communications of the Association for Information Systems, 8*(1), 19.

Jeyaraj, P. R., & Nadar, E. R. S. (2019). Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *International Journal of Clothing Science and Technology, 31*(4), 510-521.

Jianli, L., & Baoqi, Z. (2007). Identification of fabric defects based on discrete wavelet transform and back-propagation neural network. *Journal of the Textile Institute, 98*(4), 355-362.

Jing, J. F., Ma, H., & Zhang, H. H. (2019). Automatic fabric defect detection using a deep convolutional neural network. *Coloration Technology, 135*(3), 213-223.

Jing, J., Yang, P., & Li, P. (2015). Defect detection on patterned fabrics using distance matching function and regular band. *Journal of Engineered Fibers and Fabrics, 10*(2), 155892501501000210.

Jmali M., Zitouni B., & Sakli F. (2014). Fabrics defects detecting using image processing and neural networks. *Information and Communication Technologies Innovation and Application*. DOI: 10.1109/ICTIA.2014.7883765.

Kang, Z., Yuan, C., & Yang, Q. (2013, August). The fabric defect detection technology based on wavelet transform and neural network convergence. *IEEE International Conference on Information and Automation*, 597-601.

Karlekar, V. V., Biradar, M. S., & Bhangale, K. B. (2015, February). Fabric defect detection using wavelet filter. *International Conference on Computing Communication Control and Automation*, 712-715.

Kaynar, O., Işik, Y. E., Görmez, Y., & Demirkoparan, F. (2017, September). Fabric defect detection with LBP-GLMC. *International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1-5.

Kısaoğlu, Ö. (2002). *Orta büyüklükte bir dokuma işletmesinde istatistiksel kalite kontrol sisteminin kurulması* [MSc Thesis]. Dokuz Eylül Üniversity.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1-9.

Kumar A. (2003). Neural Network Based Detection of Local Textile Defects. *Pattern Recognition*, *36*, 1645-1659.

Kuo C.F.J., & Lee C.J. (2003). A back propagation neural network for recognizing fabric defects. *Textile Res. J., 73*(2), 147-151.

Kure, N., Biradar, M. S., & Bhangale, K. B. (2017, August). Local neighborhood analysis for fabric defect detection. *International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, 1-5.

Lambert, G., & Bock, F. (1997, October). Wavelet methods for texture defect detection. *Proceedings of International Conference on Image Processing, 3*, 201-204.

Latif-Amet, A., Ertüzün, A., & Erçil, A. (2000). An efficient method for texture defect detection: sub-band domain co-occurrence matrices. *Image and Vision computing, 18*(6-7), 543-553.

Li, C., Gao, G., Liu, Z., Huang, D., & Xi, J. (2019). Defect detection for patterned fabric images based on GHOG and low-rank decomposition. I*EEE Access, 7*, 83962-83973.

Li, W., Xue, W., & Cheng, L. (2012). Intelligent detection of defects of yarn-dyed fabrics by energy-based local binary patterns. T*extile Research Journal, 82*(19), 1960-1972.

Liu, L., Zhang, J., Fu, X., Liu, L., & Huang, Q. (2019). Unsupervised segmentation and elm for fabric defect image classification. *Multimedia Tools and Applications, 78*(9), 12421-12449.

Liu, Z., Liu, X., Li, C., Li, B., & Wang, B. (2018, April). *Fabric defect detection based on faster R-CNN*. Ninth International Conference on Graphic and Image Processing (ICGIP 2017), 10615, p. 106150A.

Lizarraga-Morales, R. A., Sanchez-Yanez, R. E., & Baeza-Serrato, R. (2017). Defect detection on patterned fabrics using texture periodicity and the coordinated clusters representation. *Textile Research Journal, 87*(15), 1869-1882.

Mak, K.L., Peng, P., & Yiu, K.F.C. (2009). Fabric Defect Detection Using Morphological Filters. *Image and Vision Computing, 27*, 1585–1592.

Makaremi, M., Razmjooy, N., & Ramezani, M. (2018). A new method for detecting texture defects based on modified local binary pattern. *Signal, Image and Video Processing, 12*(7), 1395-1401.

Martinez-Leon, M., Lizarraga-Morales, R. A., Rodriguez-Donate, C., Cabal-Yepez, E., & Mata-Chavez, R. I. (2016, May). Defect detection on patterned fabrics using entropy cues. *International Conference on Image and Signal Processing*, 71-78. Springer, Cham.

Mottalib M.M., Rokonuzzaman, M., Habib M.T., & Ahmed, F. (2015). *Fabric defect classification with geometric features using bayesian classifier.* International Conference on Advances in Electrical Engineering, 137-140.

Murino, V., Bicego, M., & Rossi, I.A. (2004). Statistical classification of raw textile defects. *Proceedings of the 17th International Conference on Pattern Recognition*. DOI: 10.1109/ICPR.2004.1333765

Ngan, H. Y., & Pang, G. K. (2008). Regularity analysis for patterned texture inspection. *IEEE Transactions on Automation Science and Engineering, 6*(1), 131-144.

Ngan, H. Y., Pang, G. K., Yung, S. P., & Ng, M. K. (2005). Wavelet based methods on patterned fabric defect detection. *Pattern Recognition, 38*(4), 559-576.

Oni, D. I., Ojo, J. A., Alabi, B. O., Adebayo, A. A., & Amoran, A. E. (2018). Patterned fabric defect detection and classification techniques: A review. *International Journal of Scientific & Engineering Research, 9*(2), 1156-1165.

Ozkaya U., Öztürk Ş., Tuna, K., Seyfi L., & Akdemir B. (2018). *Faults detection with image processing methods in textile sector*. 1st International Symposium on Innovative Approaches in Scientific Studies, 2, 405-409.

Qu, T., Zou, L., Zhang, Q., Chen, X., & Fan, C. (2016). Defect detection on the fabric with complex texture via dual-scale over-complete dictionary. *The Journal of the Textile Institute, 107*(6), 743-756.

Rasheed, A., Zafar, B., Rasheed, A., Ali, N., Sajid, M., Dar, S. H., & Mahmood, M. T. (2020). Fabric defect detection using computer vision techniques: A comprehensive review. *Mathematical Problems in Engineering, 2020*, 1-24.

Rebhi A., Benmhammed I., Abid S., & Fnaiech F. (2015). Fabric defect detection using local homogeneity analysis and neural network. *Journal of photonics*. http://dx.doi.org/10.1155/2015/376163.

Sabeenian, R. S., Paramasivam, M. E., & Dinesh, P. M. (2011). Detection and location of defects in handloom cottage silk fabrics using MRMRFM & MRCSF. *International Journal of Technology and Engineering System, 2*(2), 172-176.

Sakhare K., Kulkarni A., Kumbhakarn M., & Kare N. (2015, May). *Spectral and spatial domain approach for fabric defect detection and classification*. International conference on industrial instrumentation and control (ICIC), 640-644.

Salem, Y.B. & Nasri, S. (2011). *Woven fabric defects detection based on texture classification algorithm*. 8th International Multi-Conference on Systems, Signals & Devices, 43(24).

Şeker, A., Peker, K. A., Yüksek, A. G., & Delibaş, E. (2016, May). *Fabric defect detection using deep learning.* 24th Signal Processing and Communication Application Conference, 1437-1440.

Sezer, O. G., Erçil, A., & Ertuzun, A. (2007). Using perceptual relation of regularity and anisotropy in the texture with independent component model for defect detection. *Pattern Recognition, 40*(1), 121-133.

Silvestre-Blanes, J., Albero-Albero, T., Miralles, I., Pérez-Llorens, R., & Moreno, J. (2019). A public fabric database for defect detection methods and results. *Autex Research Journal, 19*(4), 363-374.

Simon, P., & Uma, V. (2020). Deep learning based feature extraction for texture classification. *Procedia Computer Science, 171*, 1680-1687.

Srinivasan, K., Dastoor, P. H., Radhakrishnaiah, P., & Jayaraman, S. (1992). FDAS: A knowledge-based framework for analysis of defects in woven textile structures. *Journal of the Textile Institute, 83(3)*, 431-448.

Su, J., & Zhang, H. (2006, July). A fast decision tree learning algorithm. *American Association for Artificial Intelligence, 6,*500-505.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.

Tajeripour, F., Kabir, E., & Sheikhi, A. (2007, December). *Defect detection in patterned fabrics using modified local binary patterns.* International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), 2, 261-267.

Tellawi, M. (2019). *Classification of Malaria Infected Cells Using Inception V1* [PhD Thesis]. Near East University.

Thorave, S., & Biradar, M.S. (2014). New technology for fabric defect detection based on K-means algorithm. *IOSR Journal of Engineering, 4*(3), 15-20.

Turkish Standards Institute (2005). Dokunmuş Kumaşlar Hata Tarifleri-Terimler. TS 471 ISO 8498.

Türkoğlu, M., Hanbay, K., Sivrikaya, I. S., & Hanbay, D. (2021). Classification of apricot siseases by using deep convolution neural network. *Bitlis Eren University Journal of Science, 9*(1), 334-345.

Uci Machine Learning Repository (2017, Nov 2). *Center for machine learning and intelligent systems.* https://archive.ics.uci.edu/ml/index.php

Uyanık, S., & Çelikel, D. C. (2019). Türk tekstil endüstrisi genel durumu. *Journal of Technical Sciences, 9*(1), 32-41.

Wang, T., Chen, Y., Qiao, M., & Snoussi, H. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology, 94*(9-12), 3465-3471.

Wei, B., Hao, K., Tang, X. S., & Ding, Y. (2019). A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes. *Textile Research Journal, 89*(17), 3539-3555.

Workgroup on Texture Analysis of DFG (Deutsche Forschungsgemeinschaft) (2020, May 5). *TILDA database for textile defect detection database.* http://lmb.informatik.uni-freiburg.de/research/dfgtexture/tilda.

Xin, B., Hu, J., Baciu, G., & Yu, X. (2009). Investigation on the classification of weave pattern based on an active grid model. *Textile Research Journal, 79*(12), 1123-1134.

Yang, X., Pang, G., & Yung, N. (2004). Discriminative training approaches to fabric defect classification based on wavelet transform. *Pattern Recognition, 37*(5), 889-899.

Yapi, D., Allili, M. S., & Baaziz, N. (2017). Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain. *IEEE Transactions on Automation Science and Engineering, 15*(3), 1014-1026.

Yıldız K., Buldu A., & Demetgul M. (2016). A thermal-based defect classification method in textile fabrics with K-nearest neighbor algorithm. *Journal of Industrial Textiles, 45*(5), 780-795.

Yu, X., Hu, J., & Baciu, G. (2005). Defect detection of jacquard fabrics using multiple color-channel analysis. *Research Journal of Textile and Apparel, 9*(1), 21.

Zaki, M. J. (1999). Parallel and distributed association mining: A survey. *IEEE Concurrency, 7*(4), 14-25.

Zhao, Y., Hao, K., He, H., Tang, X., & Wei, B. (2020). A visual long-short-term memory based integrated CNN model for fabric defect image classification. *Neurocomputing, 380*, 259-270.

Zhi, Y. X., Pang, G. K., & Yung, N. H. C. (2001, May). *Fabric defect detection using adaptive wavelet.* IEEE International Conference on Acoustics, Speech, and Signal Processing, 6, 3697-3700.

Zhu, D., Pan, R., Gao, W., & Zhang, J. (2015). Yarn-dyed fabric defect detection based on autocorrelation function and GLCM. *Autex Research Journal, 15*(3), 226-232.

Zhu, Z., Han, G., Jia, G., & Shu, L. (2020). Modified densenet for automatic fabric defect detection with edge computing for minimizing latency. I*EEE Internet of Things Journal, 7*(10), 9623-9636.

Zuo, H., Wang, Y., Yang, X., & Wang, X. (2012, October). *Fabric defect detection based on texture enhancement*. 5th International Congress on Image and Signal Processing, 876-880.