



Automated fabric inspection system development aided with convolutional autoencoder-based defect detection

Evrişimli otokodlayıcı tabanlı hata tespiti ile desteklenen otomatik kumaş inceleme sistemi geliştirilmesi

Muharrem Mercimek^{1,*} , Muhammed Ali Nur Öz² , Özgür Turay Kaymakçı³ 

¹ Department of Control and Automation Engineering, Faculty of Electrical and Electronics Engineering, Yıldız Technical University, İstanbul, Türkiye

² Hendek Vocational School, Sakarya University of Applied Sciences, Sakarya, Türkiye

³ Department of Electrical and Electronics Engineering, Faculty of Engineering, Canakkale Onsekiz Mart University, Canakkale, Türkiye

Abstract

Industrial automatic fabric inspection system, a critical technology in the industry, enhances both total production quantity and quality compared to conventional inspection techniques. This study aims to create a reliable and effective real-time automated visual inspection system for fabrics, focusing on defect detection. The goals of the study can be stated as; installing a system with advanced technology for capturing and processing images swiftly, the development and deployment of a system capable of autonomously learning and scanning fabrics in use, and the creation of a smart framework for accurate fabric defect detection and classification. We focus on the development of unsupervised fabric defect detection using a convolutional autoencoder model, and defect classification using a convolutional neural network model, which takes input as the feature vector generated by the convolutional autoencoder. The experimental outcomes have displayed significant success rates in both detecting defects and classifying them, confirming the effectiveness of the framework in real-time visual inspection systems.

Keywords: Fabric defect detection, Fabric inspection system, Convolutional autoencoder.

1 Introduction

The process of fabric inspection involves checking and assessing textile lots to make sure they meet certain standards, specifications, or requirements, as well as measuring the fabrics to verify that they meet the necessary criteria. Color, appearance, fabric construction, printing defects, and cross-shading are usually checked during a fabric inspection. By controlling fabrics, it is possible to determine where any defects may be and also to get more yields from production, thus guaranteeing that the materials to be manufactured in the future are having the highest quality. In the traditional visual inspection system, a single individual is responsible for all the inspection tasks and report writing. This method has the potential to introduce

Öz

Endüstriyel otomatik kumaş inceleme sistemi, endüstride klasik inceleme tekniklerine göre hem toplam üretim miktarını hem de kaliteyi artıran kritik bir teknolojidir. Bu çalışma, kumaşlar için güvenilir ve etkili bir gerçek zamanlı otomatik görsel inceleme sistemi oluşturmayı amaçlamaktadır ve odak noktası olarak hata tespiti üzerinde yoğunlaşmaktadır. Çalışmanın hedefleri; hızlı bir şekilde görüntüleri yakalama ve işleme yeteneğine sahip gelişmiş teknolojiye sahip bir sistem kurmak, kullandığı kumaşları otomatik olarak öğrenme ve tarayabilme yeteneğine sahip bir sistem geliştirme ve, doğru kumaş hata tespiti ve sınıflandırması için akıllı bir yaklaşım oluşturmak şeklinde ifade edilebilir. Çalışmada, evrişimli otokodlayıcı modeli kullanarak denetimsiz bir kumaş hata tespiti ve evrişimli sinir ağı modeli kullanarak hata sınıflandırma geliştirme işlemleri üzerinde durulmaktadır. Sınıflandırmada kullanılan evrişimli sinir ağının girişine evrişimli otokodlayıcı tarafından üretilen özellik vektörü sunulmaktadır. Deney sonuçları analiz edildiğinde, hem hataları tespitinde hem de bunları sınıflandırmada önemli başarı sergilenmiştir ve yaklaşımın gerçek zamanlı görsel inceleme sistemlerindeki etkinliğini gösterilmiştir.

Anahtar kelimeler: Kumaş hatası tespiti, Kumaş inceleme sistemi, Evrişimli otokodlayıcı.

human errors and slow down textile production. Commonly the defects are detected and classified with the help of artificial intelligence (AI) supported systems. The basic requirement of AI training is presenting the related datasets. The manufacturer piles up thousands of defect-free and defective image samples for a certain operation period from the initial setup. Afterward, trained models are obtained for testing or indeed for visual inspection and evaluation tasks. These systems permit manufacturers to store data regarding both fabric defects and repairs and generate relevant reports.

The formation of fabric is accomplished by joining fabric fibers in different ways. At the end of the manufacturing process, fabric surfaces may display errors known as defects, and it is estimated that this decreases fabric sales prices by

* Sorumlu yazar / Corresponding author, e-posta / e-mail: mercimek@yildiz.edu.tr (M. Mercimek)

Geliş / Recieved: 10.05.2024 Kabul / Accepted: 08.07.2024 Yayınlanma / Published: xx.xx.20xx

doi: 10.28948/ngumuh.1481769

45% to 60% [1]. The goal of quality control is to detect and locate any imperfections in the fabric. Due to fatigue and carelessness, the control process done by people can cause human errors. Thus, automatic optical fabric inspection systems have been designed to produce better-quality fabrics in greater quantities. Automatic fabric defect inspection systems, with all the required components, can detect mistakes at a rate of 90%, whereas traditional methods only manage to do so at a rate of 60% [2].

Despite the potential of automated inspection systems to be free of human mistakes, the complexity of more than 60 types of fabric defects and the speed of the inspection table still make these systems error-prone [3]. Various fabric defect inspection techniques with high detection rates have been studied, yet they are not suitable for real-time operation due to their high processing load or they can only detect fabric defects of a few fabric types. Despite the solutions that have been implemented in the market, there is still a demand to develop systems for various fabrics and defects. The manufacturer must invest in updating the inspection system for any fabric they plan to manufacture. This is commonly observed when the utilized approaches for the analysis involve efforts other than machine learning.

This study is mainly focused on efforts towards producing a real-time smart inspection system that utilizes deep learning-based inspection to reduce the need for feature selection and also processes large amounts of data more effectively than traditional methods. This differs from the majority of studies found in the literature, which have been developed for ideal laboratory conditions with non-moving textile lots and minimal noise. A system is investigated that is capable of detecting over 90% of significant fabric defects on plain fabric samples. Efforts have been made to address identified deficiencies through the development of an autonomous system. Deep learning, which does not necessitate feature selection and has a lower processing load, is utilized. An offline system is developed to quickly adjust the parameters of the devised methods for the fabrics. An AI framework is studied based on the convolutional autoencoders (CAE), model both for defect detection and classification. This framework is designed to perform defect detection, thereby identifying defects that do not conform to the fabric weave pattern. The classification of identified fabric defects is established using a convolutional neural network (CNN) model, which accepts input as the feature vector generated by CAE. This classification process allows for the determination of fabric quality based on designated categories. To facilitate the mapping of detected and classified defects, a defect localization system is also implemented, and the data related to identified defects and their locations can be stored in the database.

In Section 2, we will first discuss conventional approaches found in existing literature dealing with fabric defects, addressing the necessity of AI-based defect detection and classification. Following that, we will discuss AI-based methods, with the main focus on the convolutional autoencoder used in this study. In Section 3, the visual inspection system configuration is introduced with its hardware components, and the dataset acquired with the

inspection system is explained. The experimental results and a comprehensive analysis of the outcomes of the AI framework for fabric defect detection and classification are presented in Section 4. Finally, in Section 5, the study concludes by highlighting the accomplishments and discussing the potential future work.

2 Related Work

The utilization of computer-based quality control systems is commonly needed when the manufacturing process cost is very high in the medium and long term. The most important component for such a system is to have an image acquisition setup suitable for detecting errors with the least amount of effort to be spent in image processing. To address these issues, automated fabric inspection systems have been developed. It has been stated that 70% of the defects on fabric surfaces can be detected in generic visual inspection tasks. A group of commercial products by Agteks as quality bar and finishing bar are declared to be used as online fabric defect control systems on woven, non-woven, and knitted fabrics [4]. The scanning speed has been stated as 50m/min. Defect location on the fabric, defect image, defective region properties, and assigned defect labels are recorded as the common output of such systems. In general, the detection of fabric defects has relied on manual inspection conducted by an expert. This process involves the expert visually identifying defects in fabric rolls as they pass through an illuminated inspection platform at a speed ranging from 8 to 10 meters per minute. In this regard, there is a trade-off between generating an accurate solution and the system cost & inspection time.

We can outline the primary blocks of defect detection systems as; image transformation, defect detection, and classification. In this section, we discuss the conventional solutions for handling fabrics mentioned in the existing literature, to raise a particular focus on the importance of AI-based defect detection and classification. There is a tremendous number of studies on AI-based fabric defect detection and classification, the Discussion of AI-involved literature will be covered next.

2.1 Defect detection and classification

Raw images obtained through cameras contain a lot of data such as patterns and colors, which is difficult to process and detect defects in. To ensure the system works properly and reduces the processing load, it is important to identify and examine the parts that break the monotony on the fabric. Li et al. categorized fabric defect detection algorithms into two main groups: traditional algorithms and learning-based algorithms [5]. A comprehensive literature review was conducted on both categories. Traditional algorithms, as outlined by the authors, rely on feature extraction methods, encompassing statistical, structural, spectral, and model-based approaches. On the other hand, learning-based algorithms were further subdivided into classical machine learning algorithms and deep learning algorithms. Talu et al. stated that three main groups of methods were proposed in the literature for defect detection; statistical, spectral, and learning-based methods [6].

Statistical methods aim to reveal fabric defects by taking advantage of the fact that the defective parts on the fabric have different statistical properties than the defect-free parts. The biggest advantage of this approach, which is widely used in real-time systems, is its low processing load. Yet, it is very sensitive to noise and it fails to identify different types of defects. Autocorrelation measure and its variations can be used to identify defects in an image by utilizing a metric that takes into account the similarity between pixels [7]. An alternate approach, the co-occurrence matrix, is utilized to measure the repetition of the gray-level pixel distribution values. This method is commonly used for defect detection and can compute low-level statistical parameters quickly and with minimal processing effort. It is possible to determine the location of the defective parts by calculating the first-order statistical parameters of the gray level. Examining the pixels in groups is more effective than examining them individually when carrying out this calculation. One of the techniques that can be used in this context is the fractal dimension method, which measures the geometric structure of the objects [8]. This method, which takes advantage of its geometric structure to separate different objects in the image, has also been used to reveal defects. Although it is a method with a very low processing load, it is still insufficient to determine the location of the defects.

Spectral methods are utilized since standard statistical techniques are not enough to uncover errors concealed by intensity changes in the spatial domain. Forming fabrics through the repetition of simple patterns is a crucial requirement for discovering defects with spectral methods, as they are composed of periodic structures. In contrast, random-textured images are made up of unpredictable gray-level distributions, with no repetition or periodicity. As a result, spectral approaches are not effective in detecting imperfections in materials with a random texture [9]. Fourier transform is a method used to decompose signals to express them as a sum of many signals of different frequencies. It allows extraction of the meaningful information necessary to detect defects by revealing gray-level distributions repeating over the image. The biggest drawback of this popularly used method is that it cannot provide information about where in the image are the distributions of frequencies while providing correct results about the frequency distribution. To overcome this limitation, Wavelet transform can be used [10]. This transform can calculate the frequency spectrum wherein the image is the spectrum component for a temporal interval. However, it is difficult to express patterns in the image using this method. Gabor filters with angular, axial frequency bandwidths and center frequency are more suitable for examining flaws in the image than Fourier and Wavelet transform [11]. The researchers in [6], utilized six different groups of features. Histogram of Oriented Gradient (HoG), Co-occurrence HOG (CoHoG), and Statistical features (energy, contrast, correlation, homogeneity) were the spatial features. The other three included Fast Fourier Transform (FFT), Wavelet, and Shearlet. In addition to these feature extraction methods, they employed several models; Inception V3, MobileNet V2, a Self-Supervised neural

network, and a CNN model trained using the proposed Fourier transform-based defective patch capture algorithm.

Learning-based algorithms have been utilized in many studies for fabric defect detection and classification. A powerful structure to accommodate machine learning strategies, artificial neural networks can create complex decision boundaries and they have been widely used due to their low processing load and ease of training [12]. With the development of advanced data processing capabilities such as cloud computing, more complex neural networks can be created and trained. CNN as the introductory model incorporating these learning strategies can autonomously acquire hierarchical feature representations. In this process, lower layers specialize in getting fundamental features, while higher layers focus on understanding complicated ones. Additionally, CNNs are built to identify patterns within spatial relationships, interpreting them highly efficiently for tasks such as image recognition, where an understanding of the spatial arrangement of pixels is important. Also, pre-trained CNN models on large datasets (like ImageNet) can be fine-tuned for specific tasks, conveying the knowledge gained during the initial training. The study in [13] made use of pre-trained versions of extensively used VGG16, Densenet, Mobilenet, Inception V3, Resnet50, and Xception models.

U-Net (U-shaped Encoder-Decoder Network), a type of CNN different from traditional convolutional deep learning models, performs well with small datasets [14]. Since data sets available for fabric defects are mostly small, U-Net has been commonly utilized for defect detection. The researchers in [15] introduced an innovative model for identifying and categorizing fabric defects. They trained and assessed the model using the AITEX dataset [16]. In this approach, fabric images went through initial processing with U-Net to ascertain the presence of defects. Subsequently, VGG16 (Visual Geometry Group-16) and random forest algorithms were employed for the classification of specific defects within the fabrics. In the paper by Jing et al. [17], researchers employed an enriched architecture of AlexNet for fabric defect classification. They optimized network layers and convolution filters specifically to identify yarn dye defects. In the paper by Guo et al. [18] the atrous spatial pyramid pooling (ASPP) module was used in the YOLOv5 network. They mentioned that this integration facilitated the extraction of multiscale feature information from feature maps encompassing diverse receptive fields. This allowed for the detection of defects of varying sizes without modifying the resolution of the input image. Wang et al. constructed a dual convolutional network (ConvNet) architecture for fabric defect detection [19]. In this setup, the initial ConvNet was dedicated to fabric category classification, while the second ConvNet, specific to each category, was employed for detecting defective regions. In the study by Zhou et al. the study was on the modification of the Faster region-based convolutional neural network (Faster R-CNN) model, creating a deep learning model named FabricNet. Reportedly, the classification success for fabric defects was increased through the use of the deformable convolution block structure [20]. In [21] researchers incorporated a visual

long-short-term memory (VLSTM) module within a CNN framework; they aimed at mimicking human visual perception.

Autoencoders are a type of artificial neural network used in unsupervised learning [22]. Since our ablation study involves the real-time smart inspection system aided by convolutional autoencoders, the discussion of AI-involved literature will continue with autoencoders and will be covered next.

2.2 Autoencoders

Although autoencoders have been known for a long time, their practical applications emerged in the 1980s. What sets autoencoders apart from other types of neural networks is their ability to compress data primarily in hidden layers and then attempt to reconstruct the input data using the compressed data [23]. The fundamental goal of this process is to demonstrate that the features present within the data can be extracted, allowing for the reconstruction of the data. The phase where features are extracted takes place in the hidden layers, which contain the lowest-dimensional data.

This feature is used in studies that aim to have autoencoders discover feature vectors within the data. It is an undeniable fact that finding the features of data is crucial when training neural networks. Using the concept, data to be used for training a network can be generated by another network. When autoencoders reconstruct data, it is expected that they cannot reproduce noisy signals or any pattern-less components present in the input data [24]. Due to this characteristic, they can also be considered for separating data from noise or defects, serving as a key point for our study. Namely, the defects can be highlighted and located within the fabric images. Bergmann et al. improved the autoencoder's performance in defect detection by applying a perceptual loss function [25]. They observed that current techniques resulted in significant residual errors in edge areas with minor localization inaccuracies. To address this issue, they introduced a novel approach employing a perceptual loss function grounded in structural similarity for defect detection. This loss function gauges luminance, contrast, and structural disparities among image patches.

Autoencoders have two different structures within their artificial neural networks, known as the encoder and the decoder. To explain their roles, the neural network in the encoder part is responsible for reducing the dimension of the input data in a way that allows the extraction of features sufficient for the decoder to reconstruct it. On the other hand, the neural network in the decoder part is responsible for enlarging the dimensions of the features extracted by the encoder to reconstruct the input data. The features lying between these two structures find applications in various fields. We can express autoencoders mathematically as;

$$\begin{aligned} \varphi : X &\rightarrow F \\ \gamma : F &\rightarrow X' \end{aligned} \quad (1)$$

$$\varphi, \gamma = \arg \min_{\varphi, \gamma} M(X - X') \quad (2)$$

where φ represents an encoder neural network, γ represents a decoder neural network, X represents the data given as input to the autoencoder and intended to be

reconstructed, F represents the feature data generated by the encoder, and M represents a cost function, X' is the output of the autoencoder. The general autoencoder structure is depicted in Figure 1.

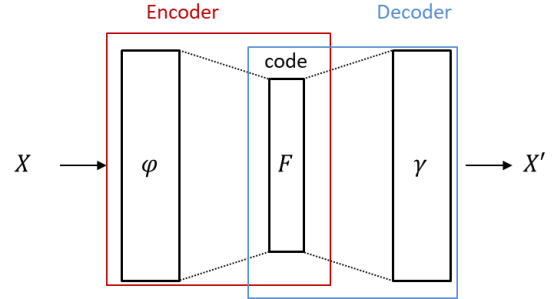


Figure 1. The autoencoder structure [26].

CNN can be used in the encoding and decoding parts of an autoencoder, making it a convolutional autoencoder (CAE). There are various types of autoencoders. From one point of view, they can be categorized into two main headings: regularized autoencoders and variational autoencoders. In another aspect with the development of the autoencoders, one can group methods as single-layer autoencoders and stacked autoencoders.

Essentially autoencoders aim to obtain the input data as the output, they always strive to create a copy mechanism that moves data from input to output. Studies aimed at preventing autoencoders from simply copying data are referred to as regularized autoencoders. Examples of these regularized autoencoders include denoising autoencoders (DAEs) and no-peaking autoencoders. Different approaches have been tried in these models to prevent copying, resulting in successful outcomes. Mei et al. employed the multi-scale convolutional denoising autoencoder (MSCDAE) design, harnessing different image scales produced by the Gaussian pyramid and incorporating a salt-and-pepper noise model into the input image for the training of denoising autoencoders of varying dimensions [27]. As a final example of regularized autoencoders, split-brain autoencoders can be mentioned. The second type is variational autoencoder (VAE), for which generative adversarial networks (GANs) can be given as an example. In GAN, there are at least two competing neural networks [28]. Following its introduction by Goodfellow, the GAN framework has garnered significant interest among researchers [29]. In the paper [30] the researchers proposed a fabric defect detection method using a hybrid of CNN and VAE. The convolutional layers were employed to extract pattern features from fabric images, while the variational autoencoder was utilized to model latent characteristics and infer a reconstruction.

While a single-layer autoencoder proves effective for basic tasks, the extraction of complex features from many real-world datasets remains challenging. Addressing this limitation, a stacked autoencoder (SAE) moderates the issue by employing cascaded multiple layers of autoencoders. These layers progressively learn complex features, establishing a hierarchical representation of the data. The integration of SAE with CNN resulted in the convolutional

autoencoder stack, subsequently evolving into stacked convolutional autoencoders (SCAEs). SCAEs represent multi-layer neural networks specifically designed to capture complex representations. It is stated in [31], that SCAEs can retain more spatial locality information and extract more representative image features while minimizing redundancy, in contrast to conventional CNNs and AEs. Therefore, SCAE is rather suitable for extracting related visual perception information. Han and Yu proposed a defect detection system

using synthetic defect data based on a stacked convolutional autoencoder (SCAE) structure. Their straightforward adjustment involved utilizing autoencoders in the shape of CAE, where conventional feedforward autoencoders employ dense layers, whereas convolutional autoencoders incorporate convolutional and transposed convolutional layers [32]. Some of the recent deep learning-based studies for fabric defect detection and classification are listed in Table 1.

Table 1. Deep learning-based studies for fabric defect detection and classification.

Author	Proposed or Tested Model	AE-based [Y/N]	Dataset	Evaluation	Drawbacks
Li et al. (2023) [5]	N/A	N/A	N/A	Literature Review	Lacks specific model experimentation.
Talu et al. (2022) [6]	Fourier transform-based DPC algorithm and CNN	N	Images from their defect detection system	Classification Accuracy, Run time for one image	Utilizes traditional feature extraction but lacks unsupervised learning approaches.
Dewan et al. (2023) [13]	VGG16, Densenet, Mobilenet, Inception V3, Resnet50, Xception	N	Wang dataset	Classification Accuracy	Limited dataset size affects model performance, no custom model training is implemented.
Mohammed and Clarke (2022) [15]	U-Net, VGG16, Random Forest	N	AITEX dataset	Pixel and Classification Accuracy	Lower accuracy for certain defect classifications.
Jing et al. (2017)[17]	Modified CNN (based on a variant of AlexNet)	N	N/A	Classification Accuracy	Lacks an unsupervised learning approach.
Guo et al. (2023)[18]	AC-YOLOv5	N	Self-built fabric defect dataset	Classification Accuracy	Limited information on specific aspects like scalability and computational load.
Wang et al. (2017)[19]	Deep CNN	N	Competition dataset from the DAGM 2007 symposium	Classification Accuracy, Run time for one image	May require a larger and more diverse dataset due to complexity.
Zhou et al. (2019)[20]	VLSTM-CNN	Y	Fabric defect dataset of aliyun Tianchi competition	Classification Accuracy, Precision, Recall, F1-measure, G-mean, AUC	Relatively intricate architecture with specialized modules.
Zhao et al. (2020)[21]	FabricNet	N	Competition dataset from the DAGM 2007 symposium	Distance IoU (DIoU)	Dependency on Region Proposal Network (RPN).
Tian and Li. (2019)[24]	Autoencoder-based Method	Y	TILDA dataset	Classification Accuracy, Precision, Recall,	Iterative update approach for defect detection.
Bergmann et al. (2018)[25]	Perceptual Loss-based Model	Y	Self-built Woven fabric datasets	AUC	Lacks a comprehensive exploration of nuanced structural differences between input and reconstruction
Mei et al. (2018)[28]	Convolutional Denoising Autoencoder	Y	KTH-TIPS, Kylberg Texture, and Self-built dataset	Pixel classification Accuracy, Precision, Recall,	Computational complexity and potential sensitivity to parameter tuning.
Rippel et al. (2020) [29]	Image-to-Image Translation	N	Self-built fabric defect dataset	AUROC	Computational complexity involved in training the models
Fan et al. (2021)[30]	CNN-VAE Hybrid	Y	Patterned fabric datasets dataset from the Uni. of Hong Kong	Image Level detection rate, false alarm rate, and accuracy	Computational complexity.
Han and Yu (2020)[32]	Stacked Convolutional Autoencoders	Y	Self-built fabric defect dataset	Pixel level Precision, F1-score, Recall	Potential challenges in scalability and robustness to diverse defect types.

Throughout the training process, AEs, CAEs, and many other models for defect detection employ identical parameters such as the loss function, optimizer, batch size, and number of epochs. Autoencoders, generally structured with an Encoder-Bottleneck-Decoder architecture, serve the purpose of reconstructing the base data. Consequently, if a particular sample exhibits a significantly larger error in its output reconstruction, it may be indicative of an outlier or defect. The reconstruction error serves as a valuable metric for identifying unusual data points within a dataset. This proves particularly advantageous in unsupervised learning scenarios. In this paper, the ablation study has been conducted on CAE models which have been extensively worked on, well-received, and widely accepted in defect detection.

3 Visual inspection system configuration and the dataset

3.1 Visual inspection system

The performance of the imaging system is dependent on a combination of different elements, including the lighting, fiber material type of the fabric to be inspected for errors, the versatility of the imaging setup, and the specifications of the cameras and lenses. The layout of the developed system for visual inspection is depicted in Figure 2.

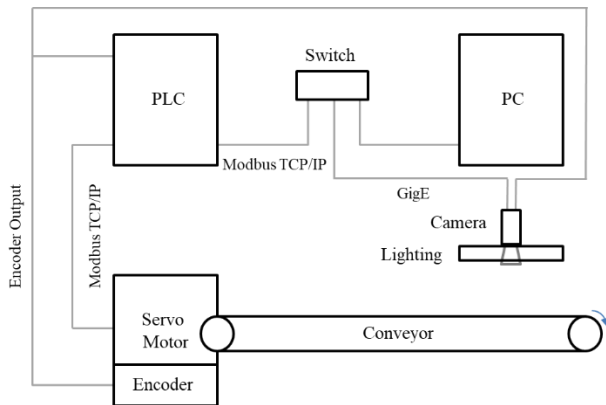


Figure 2. The layout of the developed visual inspection system.

In the developed system we used area scan cameras as the main image acquisition devices, in two distinct phases; the offline training phase and the online testing phase. Area scan cameras have been selected in visual inspection systems commonly, considering the advantages such as ease of use and affordability. On the other hand, several important limitations of area scan cameras impede the efficiency of visual inspection. For example, if the object under inspection is large, and constantly moving, or the system requires high-resolution and blur-free images, then area scan cameras are inadequate. For such problems, the use of an advanced group of cameras, namely, line scan cameras can be chosen. In cases of insufficient or blurred imaging of an object, the use of a line scan camera yields better results than area scan cameras. In this study we focus on image acquisition with

low/mid speeds of the conveyor belt and area scan cameras are utilized with ease.

To prevent under-sampling or over-sampling of the fabrics during image acquisition, a programmable encoder connected to the system is employed to synchronize the camera with the motion system and lighting. The most important factor affecting the performance of the image acquisition process for defect detection is the lighting of the environment [33]. The image acquisition process indeed is dependent on the lighting component that is dominant enough to eliminate any unwanted light reflected from the objects or environment, while also properly distributing the target areas in the image acquisition field.

The imaging process has to be done in real-time or near real-time, in industrial computer-aided visual inspection systems. Hence, a certain size of fabric regions within certain time intervals must be processed and classified. In this context, a basic requirement arises for ensuring the quick passing of the fabrics through the system for defect detection in one cycle. To meet such a requirement, it is necessary to synchronize the encoder, AC servo motor, lighting, and computer vision codes. The mechanical setup consists of solid parts mounted on the belt which can be moved to adjust the height of the camera and the lighting device. An OMRON MX2 Inverter is used for speed control of the AC servo motor. This inverter is chosen for its compliance with industrial standards and its ability to meet the required speed conditions. To obtain visual data for use in fault detection algorithms, a conveyor belt system integrated with a camera module is developed. When capturing the images, the lighting device provides a uniform ambient light. The necessary lighting equipment has been procured, and its installation has been ensured in a way that does not obstruct the flow of fabric on the conveyor belt. Cognex CAM-CIC-1300-60-G area scan camera (1.3MP, 60fps 1/1.8" CMOS sensor, C-mount, GigE) is used for offline training at the low speed of the conveyor belts. The camera is equipped with a Fujinon DF6HA-1B Lens featuring an aperture of f/1.2. A lens with such an aperture allows more light in, allowing for a faster shutter speed. To be able to describe the defects properly, a minimum defect size of 5 pixels is required. The cameras are equipped with proper lenses so that the field of view enables us to express one mm length of fabric section corresponding to a minimum of five pixels in the images. The square-shaped lighting device and the camera at the center are mounted onto the aluminum extrusion frames installed on the belt system. The triggers from the encoder on the belt are transmitted to the camera via the I/O cable thus the camera can be synchronized. The images of the system are given in Figure 3. The PC can communicate with the PLC through its Ethernet port and the speed of the band can be adjusted. Images of the fabric surface are acquired through trigger signals from the encoder. Power Over Ethernet (POE) adapters have two Ethernet connections and one power connection. The electrical energy from the power connection is transferred to the output Ethernet connection, allowing power to be

supplied to the camera through the Ethernet cable. The input Ethernet port connects to the computer, creating a local network connection between the computer and the camera in this way. The acquired images are transferred to the PC via Ethernet and presented to the defect detection algorithm to run various stages of autonomous visual inspection tasks. A desktop application capable of communicating with the camera module is created also to store the acquired images in a database.

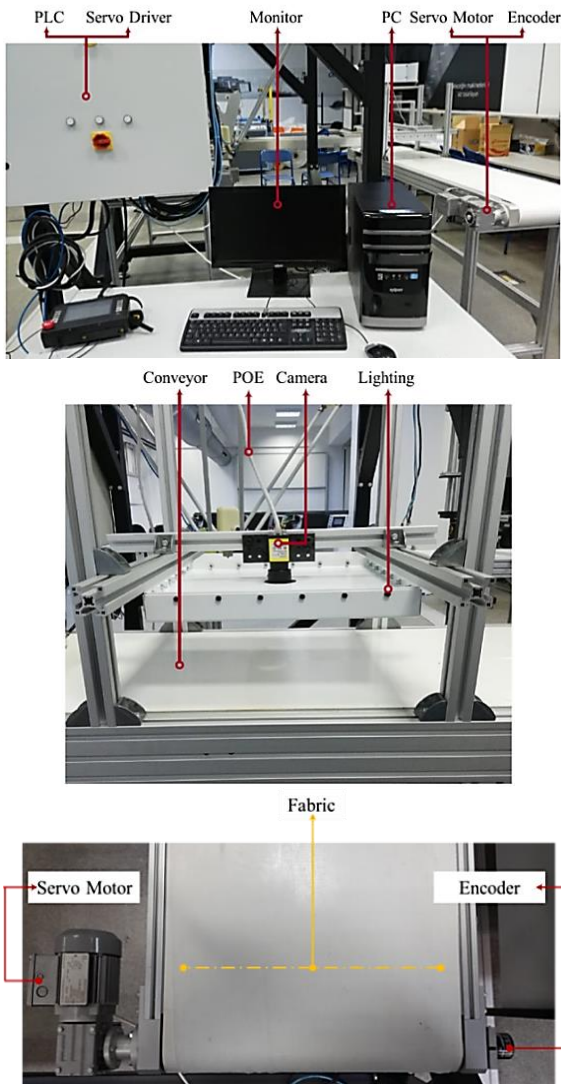


Figure 3. The complete system for visual inspection.

To ensure the recording of images during active processing and their storage in the necessary databases, ready-made interfaces do not meet the required needs. Therefore, a custom interface has been developed for the project using the SDK and developer libraries provided by Pylon Viewer. The developed interface shown in Figure 4 has been prepared using the C# language, and communication with the camera has been established using the Pylon viewer libraries. The original images have dimensions of 1280×1024, which are later cropped to 1280×704.

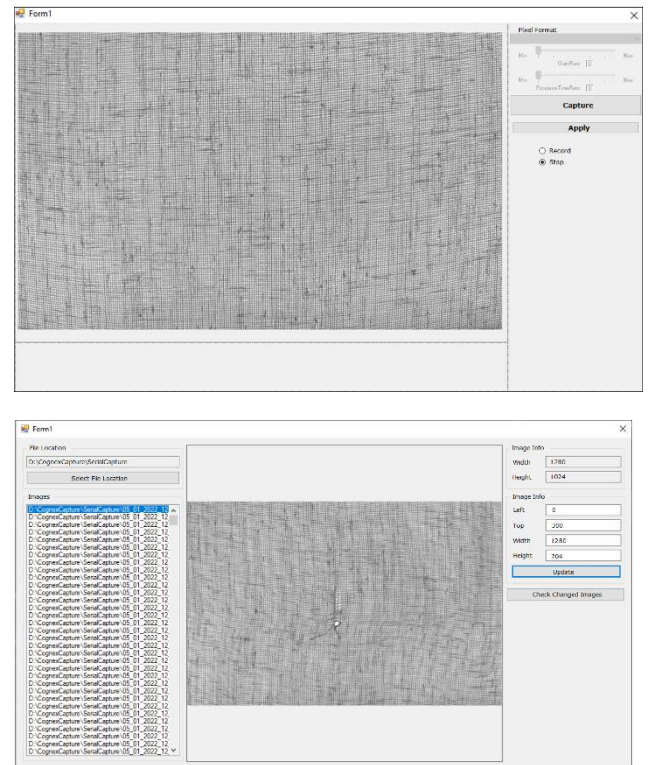


Figure 4. The interfaces for image capturing and reshaping.

3.2 The dataset

We applied the proposed system to examine tulle fabric samples containing various defects. The system captures images at a variable speed of 25.23×10^{-3} seconds per image, which equals 40 images per second. Each pixel represents a length of 3.94×10^{-3} inches, resulting in an approximate spatial resolution of 253 pixels per inch or 100 pixels per centimeter. We configured the raw image height to be 1024 pixels, meaning each image represents a 10.24 cm-length section of fabric. The fabric roll under study is 20.8 meters in length, thus, we obtained 203 non-overlapping images to cover the entire roll. The duration for shifting 1024 pixels during image acquisition is 4.35 seconds. The entire roll can be checked in 883.6 seconds (which is slow) and normally 35343 images are created. However, many of them contain excessively repeated instances of the same defects. Consequently, a set of 203 non-overlapping images is decided to be sufficient. Additionally, there are still defect-free images in this group. For defect detection and classification purposes in this study, we focused on the 58 non-overlapping images that specifically exhibit defects. Many of the fabric sample defects were intentionally produced using tools like knives, needles, etc., while others occurred during the fabric manufacturing process. The strategy for dividing large 1280×704 images into smaller 64×64 patches is depicted in Figure 5. There are 220 patches per image, totaling 12760 patches in 58 images. In the provided image in Figure 4, 17 patches exhibit defects, while 203 patches are defect-free. The patches colored in red indicate the presence of defects.

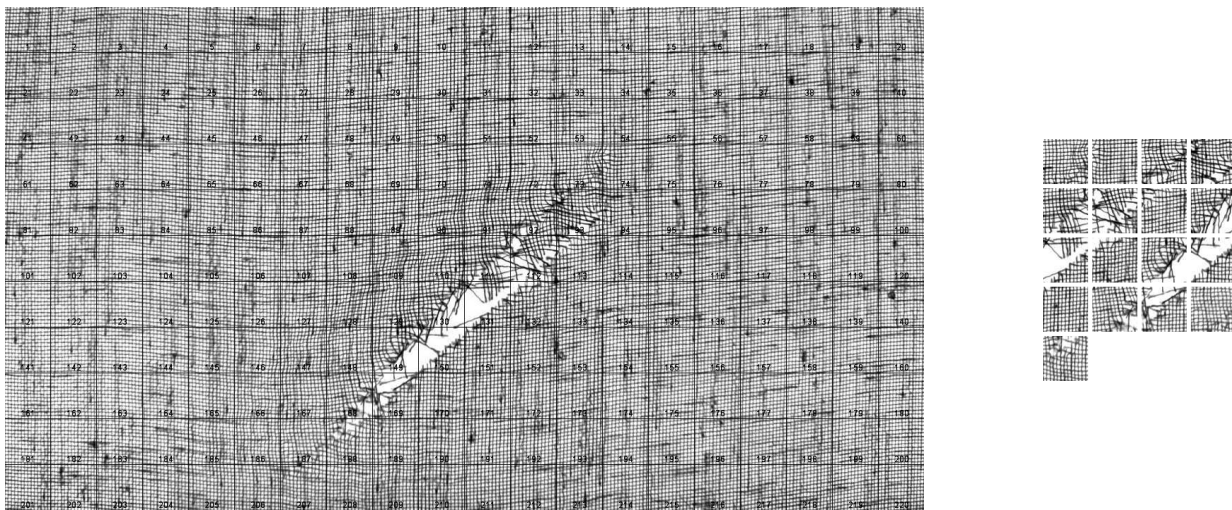


Figure 5. The strategy for dividing large 1280×704 images into smaller 64×64 patches. The numbers are overlaid on patch images and a pixel-wide boundary is added for this illustration. (The images are down-scaled to 35%)

Notably, the defect-free regions are significantly larger than the regions with defects. Out of the 12760 patches, 556 patches contain defects. 660 defect-free patches are randomly picked up and used during the training phase of the CAE model to balance the number of patches with and without defects. A defect type {hole, cut, thread wear, stain, missing thread, knots, starting mark} is assigned by an expert from the textile industry to each patch image containing a defect. The defect types adopted and used in this study are presented and exemplified in Figure 6. The patches may have different types of defects at the same time. As much as we could, labels have been assigned for the defects in all patch samples. One cannot claim that the labeling process is a definitive success because, in certain samples, multiple defect types may be present. The number of assigned labels for the defects is listed in Table 2. If such a dataset is utilized for training because three classes have a notably higher number of samples, the defect will probably be assigned to one of these classes during the defect detection.

Table 2. The number of patches having certain types of defects

Defect Types	Number of patches
Hole	38
Cut	144
Thread wear	178
Stain	4
Missing thread	20
Knots	23
Starting mark	149

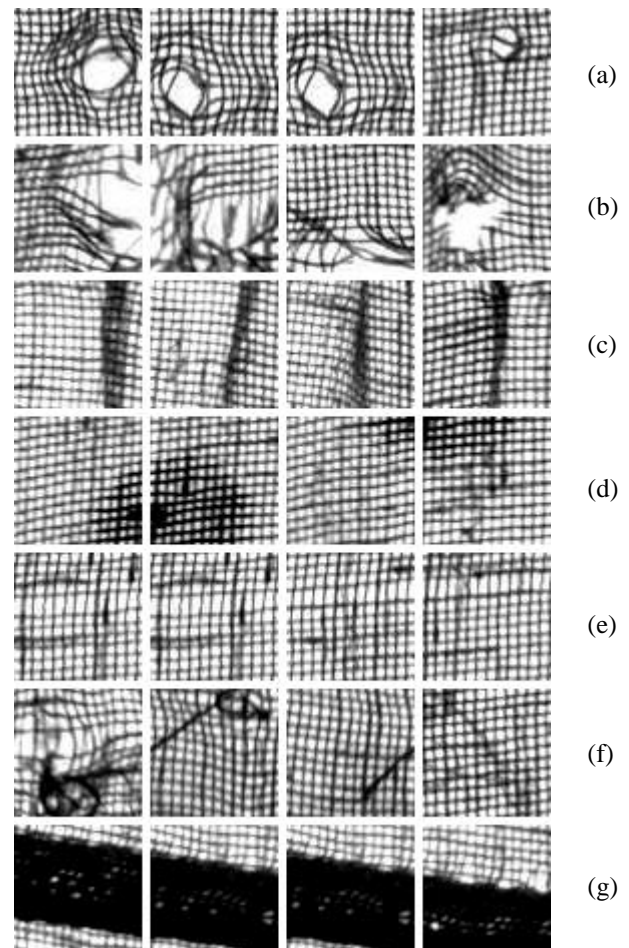


Figure 6. The defect type samples visible on the fabric used in this study: a) hole, b) cut, c) thread wear, d) stain, e) missing thread, f) knots, and g) starting mark.

4 Experimental study

It is essential to detect and categorize defects within the images. This involves a series of sequential tasks, beginning with preprocessing the fabric images, followed by

unsupervised fabric defect detection using CAE, and concluding with defect classification using a CNN model, which receives input from the feature vector generated by CAE.

The CAE model that we adopted to use both in defect detection and defect classification is given in Figure 7 and Figure 8. The structural form of the model is given as well. As can be seen from the depiction when autoencoders perform data reconstruction, it is anticipated that they will be unable to replicate noisy signals or any random elements found in the input data. This makes them suitable for distinguishing between data and noise or flaws. Specifically, this enables the identification and localization of defects within the fabric images. In this study, L₂ Loss Function is used for the training of the model to minimize the error which is the sum of all the squared differences between the original patches and the reconstructed patches. The training phase employs a standard patch-splitting strategy to reduce the number of manually labeled patches. Conversely, during the testing phase, a different but widely used approach is adopted. This involves sliding a window of the same size as the training patch images across the larger image and passing each section through the convolutional autoencoder. This enables the comprehensive testing of the larger image for defect detection, piece by piece. An alternative advanced

approach is the utilization of multi-scale convolutional autoencoders, which can handle images at various resolutions. This method entails training distinct autoencoders for different image scales. However, due to considerations of ease of implementation, multi-scale convolutional autoencoders are not currently favored in the study's current state.

Yet the error maps as the difference between the original image data and the reconstructed versions are not readily usable, various post-processing needs to be applied such as thresholding, morphological operators, and blob detection steps depicted in Figure 8. The post-processing of the reconstructed patch images holds significance in various aspects. The application of consistent post-processing techniques plays a crucial role in establishing uniformity across a set of images, thereby enhancing the reliability of comparisons and analyses. Particularly in the training of machine learning models, the preprocessing of images proves to be pivotal, exerting a substantial influence on the overall performance of the model. Keras/Tensorflow libraries are used for building, training, and testing neural network models, and the OpenCV library is used for image-based functions.

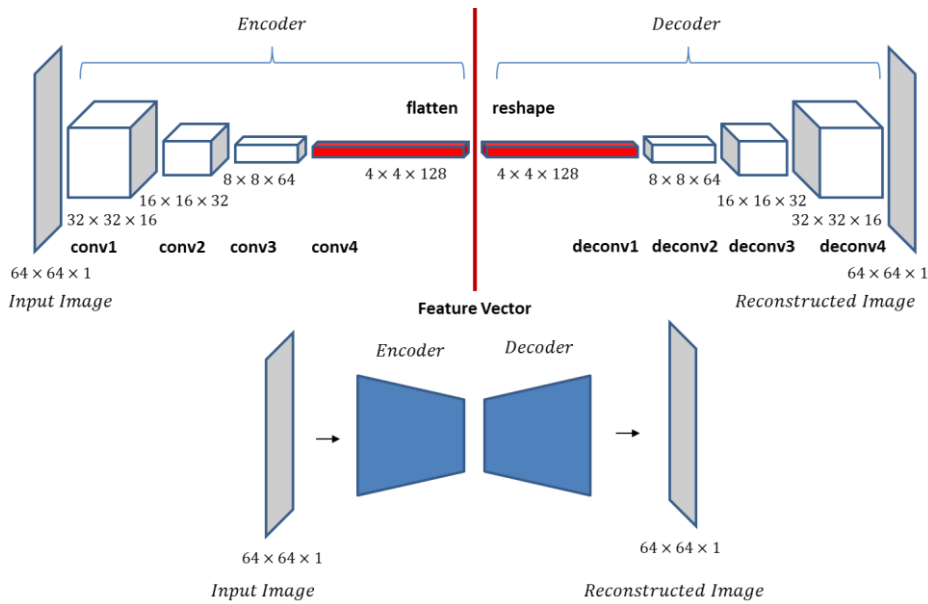


Figure 7. Defect detection: CAE model trained for defect detection using the small patch images.

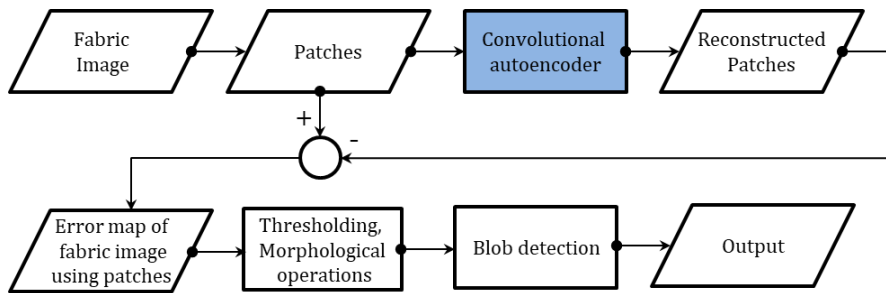
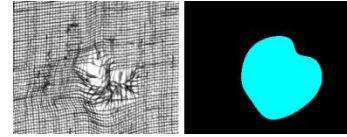
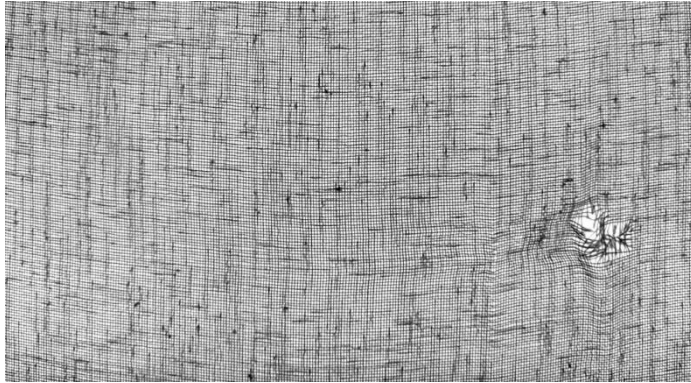


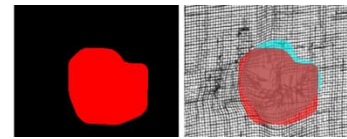
Figure 8. Computation steps of fabric defect detection – testing phase.

Figure 9-12 depict examples of defect results on various fabric images. These images showcase the following: the complete fabric images with defects, the defect detection results using CAE for the entire image, the ground truth for the defects, and an overlay that displays the defects,

detection results, and ground truth in these images. 58 images with dimensions of 1280×704 have been manually segmented using tools of ImageJ program to obtain ground truth regions for the defects used in this study.

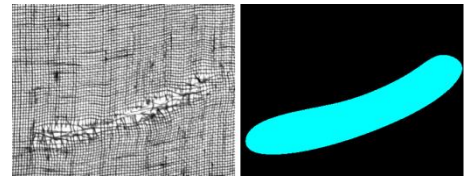
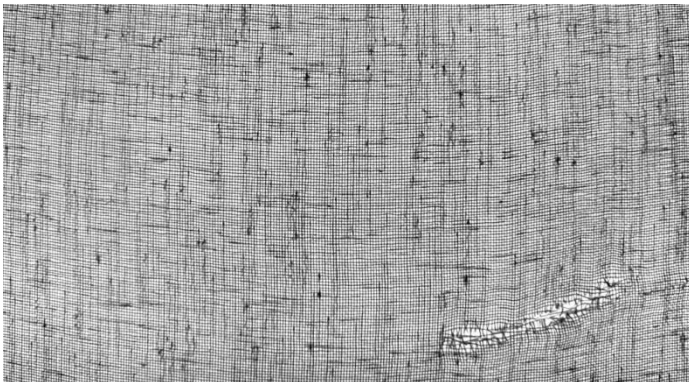


(a)

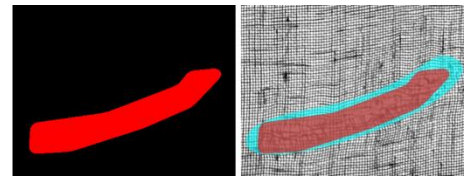


(d,e)

Figure 9. Defect detection result for a fabric image, a) fabric image as whole b) a patch with defect, c) defect detection result of the whole image with CAE d) the ground truth for the defect e) overlay of the defect, detection result, and the ground truth. (The images down-scaled to 30%)

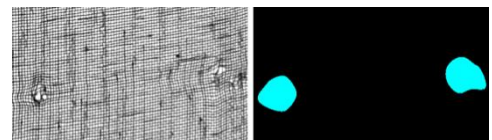
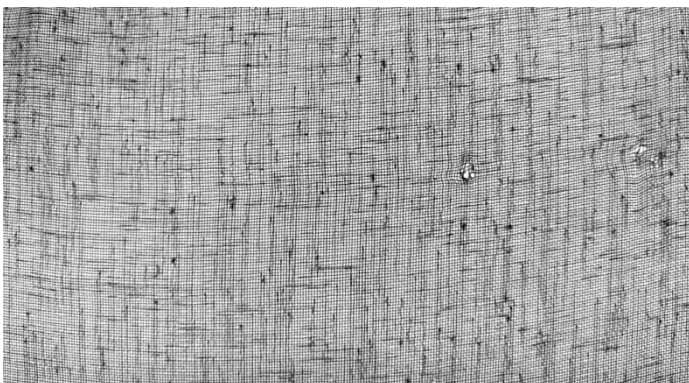


(a)

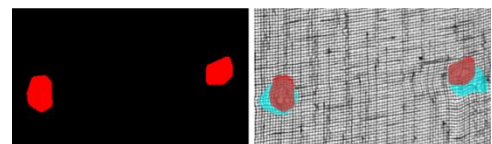


(d,e)

Figure 10. Defect detection result for a fabric image, a) fabric image as whole b) a patch with defect, c) defect detection result of the whole image with CAE d) the ground truth for the defect e) overlay of the defect, detection result, and the ground truth. (The images down-scaled to 30%)



(a)



(d,e)

Figure 11. Defect detection result for a fabric image, a) fabric image as whole b) a patch with defect, c) defect detection result of the whole image with CAE d) the ground truth for the defect e) overlay of the defect, detection result, and the ground truth. (The images are down-scaled to 30%)

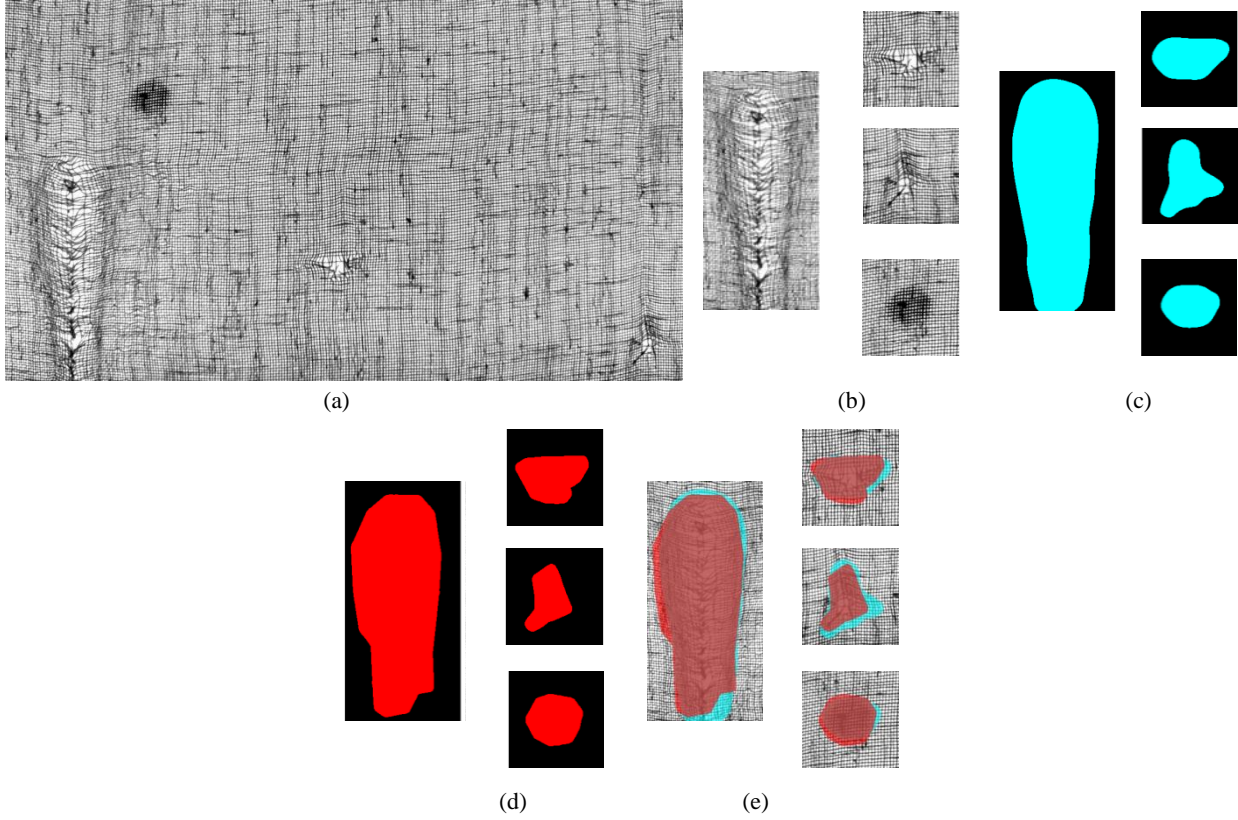


Figure 12. Defect detection result for a fabric image for multiple defects, Results for four defects are depicted a) fabric image as whole b) patches with defects, c) defect detection result of the whole image with CAE d) the ground truth for the defects e) overlay of the defects, detection results, and the ground truth.. (The images are down-scaled to 30%)

This paper places a greater emphasis on defect detection rather than defect classification. This decision is based on the recognition that, as mentioned earlier, we cannot definitively assert the success of the labeling process for the specific fabric type employed in this study. The established framework is designed to generate defect detection scores for each test image. Since all 58 images are tested collectively, overall performance scores can then be computed. These scores are derived by comparing the ground truth regions for defects with those highlighted through CAE-based defect detection. It's important to note that determining an appropriate threshold value is necessary for this process, and post-processing steps applied to candidate defective regions can inadvertently affect the scores. Following thresholding, individual pixels may be classified as either defective or defect-free. To evaluate the success of segmentation four fundamental values must be calculated. These fundamental values as true positive pixels (TP_p), false positive pixels (FP_p), false negative pixels (FN_p) and true negative pixels (TN_p) are illustrated in Figure 13.

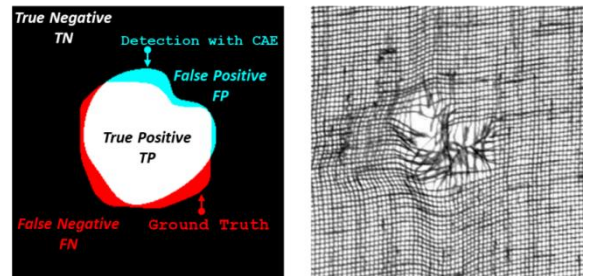


Figure 13. Categorization of image pixels. (The images are down-scaled to 75%)

A pixel labeled as defective is considered true positive. Similarly, a pixel labeled as defect-free is considered true negative. To evaluate the effectiveness of defect detection using CAE we utilize we use Recall, Precision, F1 score, Accuracy, and IoU metrics,

$$\begin{aligned}
 \text{Recall} &= TP_p / (TP_p + FN_p) \\
 \text{Precision} &= TP_p / (TP_p + FP_p) \\
 \text{F1 score} &= \\
 &= 2 \times \text{Recall} \times \text{Precision} / (\text{Recall} \\
 &\quad + \text{Precision}) \\
 \text{Accuracy} &= \\
 &= (TP_p + TN_p) / (TN_p + FN_p + TP_p + FP_p) \\
 &\quad \times 100\% \\
 \text{IoU} &= TP_p / (FP_p + TP_p + FN_p)
 \end{aligned} \tag{3}$$

Recall represents the percentage of actual defects correctly identified by the defect detection system. Precision, signifies the percentage of cases identified as defects by the detection system that are indeed actual defects. The accuracy metric quantifies the number of accurate predictions made by a model across the entire dataset. The F1 score assesses a model's performance by combining its Precision and Recall scores. IoU quantifies the degree of overlap between ground truth and result sets. The performance of the fabric defect detection using CAE for different evaluation metrics is given in Table 3.

Table 3. Performance of the fabric defect detection using CAE.

Recall	0.967
Precision	0.625
F1 score	0.760
Accuracy(%)	99.095
IoU	0.612

The Recall score primarily focuses on identifying or detecting defects, whereas the Precision score provides information about the exact location of the defects. In our experiments, the areas identified as defects by the CAE-based detection system often encompass a larger region, with the ground truth being a subset of this area, as shown in Figure 10. One possible explanation for this behavior may be the use of 64×64 patch images. When a defect is detected, it is enclosed within a larger region than necessary. As previously mentioned, several post-processing steps, such as thresholding, morphological operators, and blob detection, are required to eliminate weak candidates for defects and ensure that the defect detection framework accurately delineates the defect regions. The F1 score assesses performance using Precision and Recall scores. IoU score is proportional to the Precision score as both expressions have TP_p parameter in their numerators. If the primary concern is correctly identifying defects, Recall and Accuracy scores can be considered. Conversely, if precision in detecting defects is more important, Precision and IoU can be considered.

In defect classification, we utilize a total of 556 patch images, each has a size of 64×64 pixels. These patches exclusively contain defects. For the training dataset, we randomly select three-fourths of the patches from each defect group, reserving the remaining one-fourth for testing (Table 4). Notably, employing the patches as-is, without addressing the imbalance in sample numbers across defect classes, can lead to a classification task that inherently favors the classes with a larger sample size.

The defect classification task is employed regarding the model given in Figure 14. One can refer to the depiction of the CAE model trained for fabric defect detection using the small image patches given in Figure 6 to identify the blocks corresponding to the CAE encoder located on the right-hand side of the feature vector.

Table 4. The number of defective patch images used for the training and testing phases.

	Total	Training	Testing
Hole	38	29	9
Cut	144	108	32
Thread wear	178	120	58
Stain	4	3	1
Missing Thread	20	15	5
Knots	23	18	5
Starting mark	149	111	37
	556	404	147

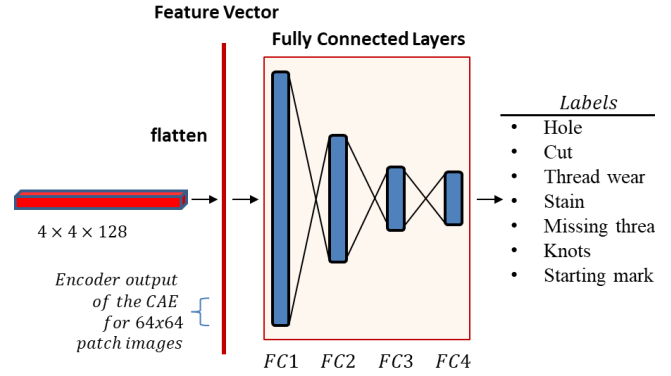


Figure 14. Defect classification: The feature vector from the encoder module already created in the patch reconstruction is to be introduced to the classifier module for the classification of the defects.

The confusion tables for the training and the test phases for the classification task are given in Figures 15 and 16 and classification rates for each class in given in Table 5.

Training \	Hole	Cut	Thread wear	Stain	Missing Thread	Knots	Starting mark
Hole	20	6	3	0	0	0	0
Cut	10	92	0	0	3	3	0
Thread wear	0	6	98	0	1	6	9
Stain	0	0	3	0	0	0	0
Missing thread	2	4	2	0	7	0	0
Knots	2	4	5	0	0	7	0
Starting mark	1	0	12	0	0	8	90

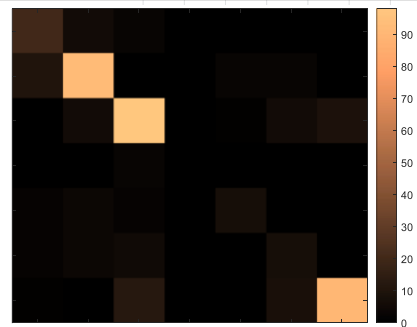


Figure 15. Confusion matrix for the training patch images.

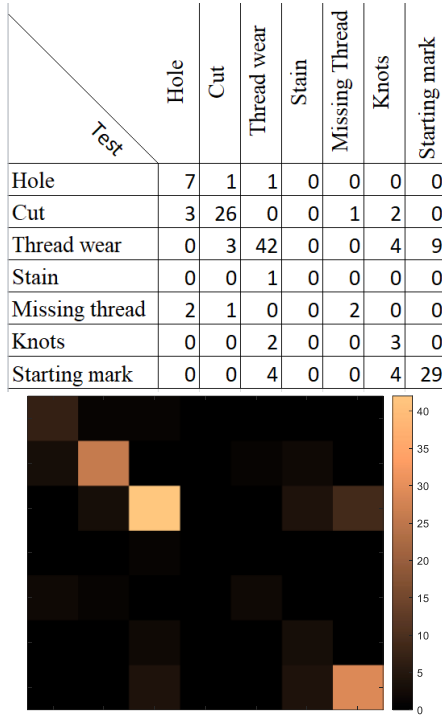


Figure 16. Confusion matrix for the test patch images.

As previously mentioned, the labeling process for the fabric types utilized in this study can pose a bottleneck for defect classification. Among the classes, there is one with an extremely limited number of samples (stain class: 3), while three other classes have only a small number of samples (missing thread: 20, knots: 23, hole: 38). In contrast, the remaining classes have a larger number of samples (cut: 144, starting mark: 149, thread wear: 178). It can be noticed that the stain class samples are misclassified mostly as thread wear (Table 5).

Table 5. Classification rates for defective patch images.

Training patch images		Test patch images	
Defect Type	Classification Rate (%)	Defect Type	Classification Rate(%)
Hole	68.97	Hole	77.78
Cut	85.18	Cut	81.25
Thread wear	81.67	Thread wear	72.41
Stain	0	Stain	0
Missing thread	46.67	Missing thread	40
Knots	38.89	Knots	60
Starting mark	81.08	Starting mark	78.38
	77.72		74.15

The class imbalance presents a challenge when training models, particularly in the context of defect detection. Neural networks may assign labels from the majority classes to the under-represented class samples, and this leads to poor generalization. Boosting algorithms are reported to be well-suited for addressing imbalanced datasets because they assign higher weights to the minority class for successive iterations [34]. Additionally, employing an appropriate evaluation metric that accounts for class imbalance can

provide a more accurate representation of a classifier's performance.

The scanning speed for the system was set to 1.412 m/min (or 0.02354 m/s), which is low for a commercial product. The main reason for this speed level is the lack of a fabric winding system associated with the inspection system. The winding process was managed mechanically at low scanning speeds. The conveyor belt can reach a speed of 8-10 m/s, yet the winding process will not support such a speed range.

The training duration of the neural network varies depending on the value desired in the loss function and the network size. In the studies conducted for fabric defect detection, the neural network was trained for an average of three hours. For defect detection, the testing duration of a test image with dimensions of 1280×704 was measured to be 5.156 seconds. For 203 non-overlapping images of 20.8 m fabric roll, overall testing was completed in 17.44 minutes. The defect classification was managed in a supervised manner utilizing the designated defective patches. The training phase of the classification took 18.50 minutes, and overall testing was completed in 7.843 seconds for 556 defective regions. All computations were evaluated on a computer with an Intel Core i7-6700HQ 2.59 GHz processor and 16.0 GB of RAM.

5 Conclusions

The main thrust of this revolved around the development of an automatic visual inspection system that utilizes deep learning to reduce the need for feature selection. The system processes large amounts of data more effectively than traditional methods in this manner. Efforts have been made to address deficiencies through the development of an autonomous visual inspection system. We investigated unsupervised fabric defect detection using CAE, and defect classification using a CNN model, which takes input as the feature vector generated by the CAE. The model constructed and trained for defect detection is developed on our tulle fabric image dataset. Additionally, we implemented a defect localization system that records data of identified defects and their respective relative locations in a database. Experimental results have demonstrated high rates of success in both defect detection and classification, affirming the applicability of our approach to real-time visual inspection systems. Regarding the scanning speed of the inspection system for the data acquisition process, the conveyor belt can operate at speeds up to 8-10 m/s. However, the current winding process does not support this speed range. Additionally, the testing duration at the current project stage is moderate. Therefore, a light version of the model with fewer parameters should be considered to remain competitive.

In the future, we will put out efforts to improve the detection accuracy and experiment with different strategies to further enhance reconstruction detail. For defect detection multi-scale convolutional autoencoders that can handle images at various resolutions and boosting algorithms to overcome the problems of imbalanced dataset use will be investigated.

Acknowledgment

This research is funded by The Scientific and Technological Research Council of Turkey (TUBITAK) under project number 118E607.

Conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person/institution in the article.

Similarity rate (iThenticate): 9%

References

- [1] K. Srinivasan, P. H. Dastoor, and S. Jayaraman, FDAS: architecture and implementation. *Expert Systems*, 9, 115-124, 1992. <http://dx.doi.org/10.1111/j.1468-0394.1992.tb00392.x>.
- [2] C.H. Chan and G. K. H. Pang, Fabric defect detection by Fourier analysis. *IEEE Transactions on Industry Applications*, 36(5), 1267-1276, 2000. <http://dx.doi.org/10.1109/28.871274>.
- [3] Standard Test Methods for Visually Inspecting and Grading Fabrics. D5430-13, 2017.
- [4] Fabric inspection systems: Agteks. <https://www.agteks.com/fabric-inspection-systems> Accessed 25 April 2024.
- [5] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, Fabric defect detection in textile manufacturing: a survey of the state of the art. *Security and Communication Networks*, 1-13, 2023. <http://dx.doi.org/10.1155/2021/9948808>.
- [6] M. F. Talu, K. Hanbay, and M. H. Varjovi, CNN-based fabric defect detection system on loom fabric inspection. *Textile And Apparel*, 32(3), 208-219, 2022. <https://doi.org/10.32710/tekstilvekonfeksiyon.1032529>.
- [7] G. Gao C. Liu, Z. Liu, C. Li, and R. Yang, Fabric defect detection based on Gabor filter and tensor low-rank recovery. 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 2017, 73-78, 2017.
- [8] J. Chockalingam and S. Mondal, Fractal-based pattern extraction from time-Series NDVI data for feature identification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12), 5258-5264, December, 2017. <http://dx.doi.org/10.1109/JSTARS.2017.2748989>.
- [9] K. Sakhare, A. Kulkarni, M. Kumbhakarn, N. Kare, Spectral and spatial domain approach for fabric defect detection and classification. 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, 2015.
- [10] A. Bakhshipour, A. Jafari, S. Nassiri, and D. Zare, Weed segmentation using texture features extracted from wavelet sub-images. *Biosystems Engineering*, 157, 1-12, 2017. <https://doi.org/10.1016/J.BIOSYSTEMSENG.2017.02.002>.
- [11] J. Liang, C. Che, L. Jiuzhen, and H. Zhenjie, Fabric defect inspection based on lattice segmentation and Gabor filtering, *Neurocomputing*, 238, 84-102, 2017. <https://doi.org/10.1016/j.neucom.2017.01.039>.
- [12] Z. Ren, F. Fang, and Y. Y. Wu, State of the art in defect detection based on machine vision. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 9, 661-691 2022. <http://dx.doi.org/10.1007/s40684-021-00343-6>.
- [13] J. H. Dewan, R. Das, S. D. Thepade, H. Jadhav, N. Narsale, A. Mhasawade, and S. Nambiar, Image classification by transfer learning using pre-trained CNN models. 2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 1-6, 2023.
- [14] O. Ronneberger, P. Fischer, and T. Brox, U-net: convolutional networks for biomedical image segmentation. *MICCAI 2015, Lecture Notes in Computer Science*, 9351, 234-241. Springer, Cham, 2015. <https://doi.org/10.48550/arXiv.1505.04597>.
- [15] S.S. Mohammed and H.G. Clarke, A hybrid machine learning approach to fabric defect detection and classification, *ICECENG 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 436, 135-147, 2022. <http://dx.doi.org/10.1177/09544054231209782>.
- [16] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno, A public fabric database for defect detection methods and results. *Autex Res. J.* 19(4), 363-374, 2019. <http://dx.doi.org/10.2478/aut-2019-0035>.
- [17] J. Jing, A. Dong, P. Li, and K. Zhang, Yarn-dyed fabric defect classification based on convolutional neural network. *Optical Engineering*, 56(9), 93104, 2017. <https://doi.org/10.1117/1.OE.56.9.093104>.
- [18] Y. Guo, X. Kang, J. Li, and Y. Yang, Automatic fabric defect detection method using AC-YOLOv5, *Electronics*, 2023, 12, 2950, 2023. <http://dx.doi.org/10.3390/electronics12132950>.
- [19] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, A fast and robust convolutional neural network-based defect detection model in product quality control. *Int Journal of Advance Manufacturing Technology*, 94, 3465-3471, 2018. <https://doi.org/10.1007/s00170-017-0882-0>.
- [20] H. Zhou, B. Jang, Y. Chen, and D. Troendle, Exploring faster RCNN for fabric defect detection. *Third International Conference on Artificial Intelligence for Industries (AI4I)*, Irvine, CA, USA, 52-55, 2020.
- [21] Y. Zhao, K. Hao, H. He, X. Tang, and B. Wei, A visual long-short-term memory based integrated CNN model for fabric defect image classification, *Neurocomputing*, 380, 259-270, 2020. <http://dx.doi.org/10.1016/j.neucom.2019.10.067>.
- [22] B. Olimov, B. Subramanian, and J. Kim, Unsupervised deep learning-based end-to-end network for anomaly detection and localization. *Thirteenth International*

- Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 2022.
- [23] M. Sewak, S.K. Sahay, and H. Rathore, An overview of deep learning architecture of deep neural networks and autoencoders, *Journal of Computational and Theoretical Nanoscience*, 17(1), 182-188, January 2020. <http://dx.doi.org/10.1166/jctn.2020.8648>.
- [24] H. Tian and F. Li, Autoencoder-based fabric defect detection with drossp similarity. 16th International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 2019.
- [25] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, Improving unsupervised defect segmentation by applying structural similarity to autoencoders. Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAP'13), Madeira Portugal, 27-29 Jan 2018.
- [26] E. Solovyeva and A. Abdullah, Dual autoencoder network with separable convolutional layers for denoising and deblurring Images. *Journal of Imaging*, 8(9), 250, 2022. <https://doi.org/10.3390/jimaging8090250>.
- [27] S. Mei, Y. Wang, and G. Wen Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model, *Sensors*, 18, 1064, 2018. <https://doi.org/10.3390/s18041064>.
- [28] O. Rippel M. Müller, and D. Merhof, GAN-based defect dynthesis for anomaly setection in fabrics. 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 2020.
- [29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, *advances in neural information processing systems*. 3(6), 2014. <https://doi.org/10.48550/arXiv.1406.2661>.
- [30] J. Fan, W. K. Wong, J. Wen, C. Gao, D. Mo, and Z. Lai, Fabric defect detection using deep convolution neural network. *AATCC Journal of Research*, 8 (1_suppl),143-150, 2021. <https://doi.org/10.14504/ajr.8.S1.18>.
- [31] J. Masci, U. Meier, D. Cire, and J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction. *Proceedings of the International Conference on Artificial Neural Networks*, Springer, pp. 52–59, 2011.
- [32] Y. J. Han and H. J. Yu, Fabric defect detection system using stacked convolutional denoising auto-encoders trained with synthetic defect data. *Applied Sciences*, 10(7), 2511, 2020. <https://doi.org/10.3390/app10072511>.
- [33] C. C. Chen, C.H. We, and C.S. Lin, Fast detection of fabric defects based on neural networks. Sixth International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 2023.
- [34] A. Taherkhani, C. Georgina, T. McGinnity, AdaBoost-CNN: an adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing*, 404, 351-366, 2020. <https://doi.org/10.1016/j.neucom.2020.03.064>.

