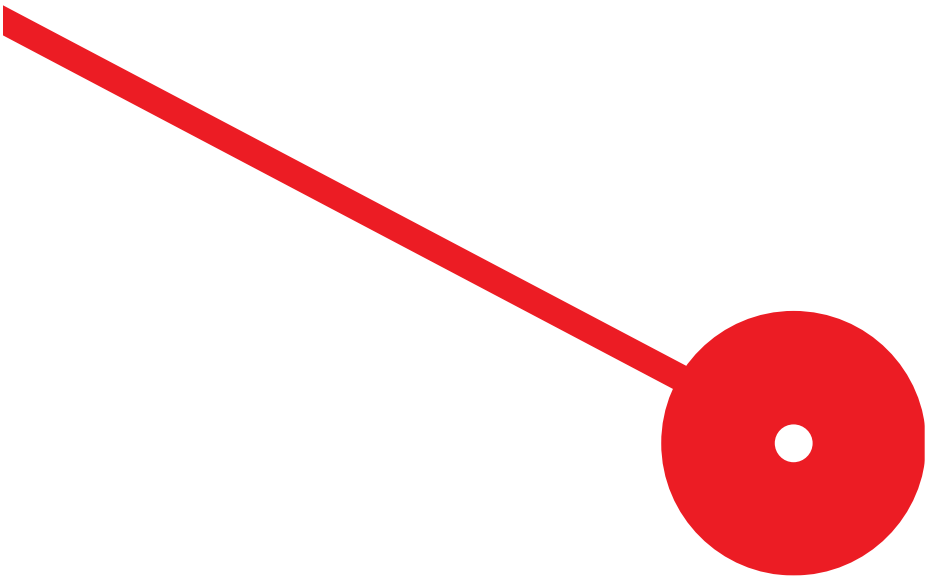
INSTITUTO
SUPERIOR
DE CONTABILIDADE
E ADMINISTRAÇÃO
DO PORTO
POLITÉCNICO
DO PORTO

M MESTRADO
BUSINESS INTELLIGENCE AND ANALYTICS

Image Synthesis via Deep Learning for Defect Detection in Organic Fabrics

João Pedro Vicente Figueiredo

10/2025



João Pedro Vicente Figueiredo. Image Synthesis via Deep Learning for Defect Detection in Organic Fabrics
10/2025

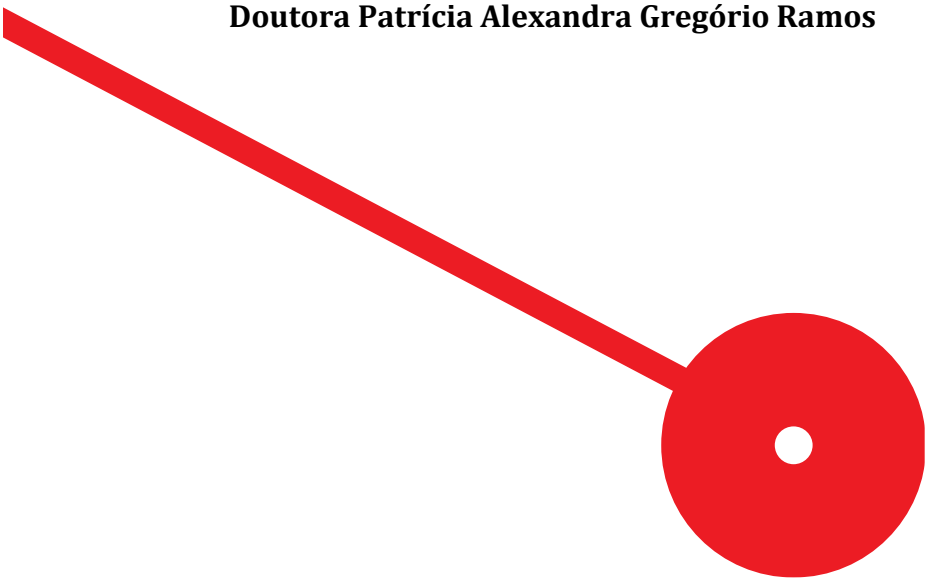
INSTITUTO
SUPERIOR
DE CONTABILIDADE
E ADMINISTRAÇÃO
DO PORTO
POLITÉCNICO
DO PORTO

M MESTRADO
BUSINESS INTELLIGENCE AND ANALYTICS

Image Synthesis via Deep Learning for Defect Detection in Organic Fabrics

João Pedro Vicente Figueiredo

Dissertação de Mestrado apresentada ao Instituto Superior de Contabilidade e Administração do Porto para a obtenção do grau de Mestre em Business Intelligence and Analytics, sob orientação de Doutor Hugo Miguel Mendes Ferreira e Doutora Patrícia Alexandra Gregório Ramos



João Pedro Vicente Figueiredo. Image Synthesis via Deep Learning for Defect Detection in Organic Fabrics
10/2025

Resumo

Este trabalho de investigação tem como objetivo desenvolver uma técnica de síntese de imagens utilizando redes neurais generativas para detetar e localizar anomalias em tecidos. O foco incide em materiais orgânicos com texturas detalhadas e variações significativas. Este contexto representa um desafio na identificação de defeitos, uma vez que as variações naturais destes materiais não devem ser confundidas com defeitos reais. O trabalho de investigação centra-se em algoritmos de deteção de anomalias não supervisionados, devido à escassez de dados etiquetados em ambientes industriais.

O trabalho consiste em explorar conjuntos de dados reais para estudar a melhor forma de utilizar autoencoders na deteção e localização de defeitos, aplicando estes resultados a imagens de tecidos, fibras e outros materiais orgânicos com texturas detalhadas e variações significativas. O objetivo é detetar e segmentar defeitos, sem confundir as alterações na textura provocadas por variações naturais.

A metodologia empírica centra-se na deteção de anomalias em padrões e tecidos, utilizando a métrica de índice de Similaridade Estruturada (SSIM - Structural Similarity Index) e variantes como função de perda de um autoencoder. A principal vantagem desta abordagem reside na capacidade do SSIM de capturar similaridades estruturais, de luminância e de contraste entre a imagem original e a reconstruída, em vez de depender apenas de diferenças pixel a pixel, como ocorre com o Erro Quadrático Médio (MSE – Mean Squared Error).

O trabalho envolveu a definição do contexto empresarial, a recolha e geração de um conjunto de dados alargado e abrangente de imagens de tecidos, o pré-processamento das imagens, o design da arquitetura do autoencoder, o treino do autoencoder no conjunto de dados pré-processado e a avaliação dos resultados de deteção. Este trabalho de investigação visa contribuir para a área da deteção de anomalias, desenvolvendo um método robusto e eficaz para identificar defeitos em tecidos orgânicos.

Palavras-chave: Deteção de Anomalias; Defeitos em Tecidos; Aprendizagem Não Supervisionada; Autoencoders; SSIM

Abstract

This research develops an image synthesis technique using generative neural networks to detect and locate anomalies in organic fabrics with intricate textures and substantial natural variation. These inherent fluctuations pose a key challenge: distinguishing genuine defects from benign differences. Given the scarcity of labelled data in industrial settings, the work employs unsupervised anomaly detection via autoencoders.

Real-world datasets are explored to assess their effectiveness in defect detection and localisation across fabrics, fibres, and similar materials, enabling precise segmentation while remaining insensitive to natural texture changes. The approach includes a thorough exploration of autoencoder architectures to optimise performance for complex, variable textures. It enhances quality control in industries handling organic materials by clearly separating true defects from surface nuances, optimising processes, and elevating standards in the textile sector.

The empirical methodology centres on textile pattern anomaly detection, using the Structural Similarity Index Measure (SSIM) and its variants as the autoencoder's loss function. Unlike Mean Squared Error (MSE), which focuses on pixel-wise differences, SSIM captures structural, luminance, and contrast similarities, penalising perceptually significant defects (e.g., tears, stains, pattern disruptions) while tolerating subtle, non-defective fluctuations (e.g., fibre grain, weave irregularities).

The workflow includes defining the industrial scope, assembling a diverse dataset through collection and synthetic generation, preprocessing images, designing and exploring autoencoder architectures with different loss functions, training the model, and evaluating performance. This research delivers a robust framework for anomaly detection in texture-rich materials, with the potential to transform industrial defect inspection.

Keywords: Anomaly Detection; Fabric Defects; Unsupervised Learning; Autoencoders; SSIM

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AE - Autoencoder

AUC - Area Under the Curve

CAE - Convolutional Autoencoder

CNN - Convolutional Neural Network

DAE - Denoising Autoencoder

DCT - Discrete Cosine Transform

DL - Deep Learning

DTTS - Dual-Task Teacher-Student

GAN - Generative Adversarial Network

GIS - Golden Image Subtraction

GLCM - Grey-Level Co-occurrence Matrices

GMM - Gaussian Mixture Model

HAR - Human Activity Recognition

KNN - K-Nearest Neighbors

ML - Machine Learning

MSE - Mean Squared Error

NF - Normalizing Flow

NLP - Natural Language Processing

RNN - Recurrent Neural Networks

ROC - Receiver Operating Characteristic curve

SAE - Stacked Autoencoder

SAM - Segment Anything Model

SSD - Single Shot Detector

SSIM - Structural Similarity Index Measure

SVDD - Support Vector Data Description

SVM - Support Vector Machine

VAE - Variational Autoencoder

ViT - Vision Transformer

WGAN - Wasserstein Generative Adversarial Net

CHAPTER I – INTRODUCTION

1 Introduction

Visual inspection is a fundamental pillar of modern industrial manufacturing, serving as the primary mechanism to uphold stringent quality standards while optimizing cost efficiency. By systematically identifying and eliminating defective components early in the production cycle, manufacturers can prevent downstream waste, reduce rework, and safeguard brand reputation. In high-volume sectors such as textiles, automotive, electronics, and aerospace, even minor surface or structural flaws, such as scratches, stains, misalignments, or material discontinuities, can render a product unusable or unsafe. Consequently, reliable defect detection is not merely a quality assurance step but a critical economic and operational imperative.

Historically, this task has been performed through manual visual inspection by trained human operators. Despite its widespread use, this approach is fraught with inherent limitations. It is labor-intensive, requiring sustained attention over long shifts, which inevitably leads to operator fatigue and reduced accuracy. Human judgment is also inherently subjective: what one inspector flags as a critical defect, another might dismiss as acceptable variation. Studies consistently report error rates in manual inspection ranging from 20% to 30%, depending on task complexity and environmental conditions. Moreover, scaling manual inspection to meet growing production demands is economically unsustainable, particularly in competitive global markets where labor costs and throughput requirements are under constant pressure.

The advent of automated computer vision systems has transformed industrial quality control, offering a pathway to consistent, high-speed, and objective defect detection. These systems leverage cameras, lighting, and image processing algorithms to analyze products at rates far exceeding human capability, often inspecting thousands of units per minute with sub-millimeter precision. Early automated solutions relied on rule-based methods, such as edge detection, thresholding, or template matching. While effective for simple, highly regular patterns (e.g., printed circuit boards with uniform traces), these traditional techniques struggle with materials exhibiting natural variability, such as organic fabrics, wood, leather, or composite surfaces. In such cases, slight changes in lighting, texture orientation, or material grain can trigger false alarms, rendering classical methods unreliable.

The rise of deep learning, particularly Convolutional Neural Networks (CNN), marked a significant leap forward. Supervised Deep Learning (DL) models, trained on large datasets of labeled images containing both normal and defective samples, can learn highly complex patterns and achieve state-of-the-art accuracy in defect segmentation.

Two-stage detectors like Faster R-CNN and one-stage models like YOLO have been successfully deployed in controlled environments. However, their practical deployment in real industrial settings is severely constrained by several critical challenges: (1) Supervised models require pixel-level or bounding-box annotations for thousands of defective samples, a process that is not only time-consuming but also demands domain expertise to ensure consistency and completeness; (2) In well-optimized production lines, defect rates are often below 1%. Collecting sufficient anomalous examples to train a robust deep model becomes statistically and logistically infeasible. Rare defects, by definition, appear too infrequently to be adequately represented in training data: (3) New or unforeseen defect modes, arising from material changes, machine wear, or process drift, cannot be anticipated during model development. A supervised system trained only on known flaws will inevitably fail to generalize to novel anomalies; (4) The overwhelming majority of training data consists of defect-free samples, leading to biased models that prioritize recall of the normal class at the expense of anomaly sensitivity. These limitations are particularly acute in the textile and organic material industries, where products exhibit rich, uniform, and non-uniform textures and significant natural variation.

This research develops and evaluates an unsupervised anomaly detection framework based on generative neural networks, specifically tailored to address the limitations of supervised approaches in industrial quality control. The core challenge lies in accurately identifying defects in materials with inherent natural variations, such as organic fabrics with intricate, non-uniform textures, without confusing benign irregularities (e.g., natural fiber grain, weave density fluctuations, or subtle color shading) with genuine flaws (e.g., holes, stains, or structural discontinuities). For instance, in textile fabrics, patterns may appear similar across samples yet differ slightly due to manufacturing tolerances. These differences must remain within a predefined quality threshold, which the system learns and applies automatically during both training and inference.

The proposed method relies on Structural Similarity Index Measure (SSIM) and variants-optimized autoencoders, trained exclusively on high-quality, defect-free fabric images.

During training, the autoencoder encodes input images into a compact latent representation and reconstructs them with minimal structural distortion. By optimizing for SSIM rather than pixel-wise error, the model learns a perceptually meaningful representation of normal texture patterns. At inference, inputs containing anomalies, such as tears, knots, slub yarn, oil stains, or weave breaks, cannot be faithfully reconstructed, resulting in locally elevated SSIM error.

The use of SSIM and its variants as the primary loss function is a deliberate design choice. Unlike Mean Squared Error (MSE), which treats all pixel deviations equally and is overly sensitive to harmless intensity shifts (e.g., lighting changes or natural texture noise), SSIM assesses similarity across three perceptually grounded components: luminance (overall brightness consistency), contrast (amplitude of local variations), and structure (correlation of local patterns). This multi-component evaluation enables the model to tolerate expected organic variations while remaining highly sensitive to disruptions in underlying pattern regularity, precisely the signature of most textile defects.

The study focuses on real-world textile images with rich, diverse patterns, including plain weaves, twills, knits, and printed fabrics under varying illumination and background conditions. These datasets reflect the complexity of industrial environments, where materials exhibit high intra-class variation and anomalies span multiple scales and types. The primary modeling objective is twofold: (1) The autoencoder must reconstruct defect-free samples with consistently low error, even across color shifts, scale changes, or minor misalignments; (2) Any deviation that alters local pattern coherence, regardless of size or contrast, must produce a detectable spike in reconstruction loss.

This dual requirement mitigates overgeneralization, a common failure mode in autoencoder-based systems where the model becomes too permissive and reconstructs defective regions almost as well as normal ones. By emphasizing structural fidelity through SSIM, the framework enforces a strict normative model of healthy fabric appearance, ensuring that only true outliers are flagged. This capability is particularly valuable in real-world textiles, which frequently feature multicolored designs, complex repeating motifs, and cluttered backgrounds. The method's independence from labeled defective data makes it uniquely practical for deployment in low-defect-rate production lines, where collecting anomalous samples is costly, rare, or impossible.

The remainder of this thesis is organized as follows. Chapter II presents a comprehensive literature review covering the evolution of anomaly detection, tracing its roots in statistics to modern machine learning and data mining applications. Key aspects that determine the formulation of anomaly detection are the nature of data, the availability of labelled data, and the types of anomalies to be detected. The literature review also classifies anomalies into point, contextual, collective, structural, and semantic types. Unsupervised anomaly detection, which is the focus of this work, and different techniques such as reconstruction-based methods (like autoencoders), normalizing flow-based methods, representation-based methods, data augmentation methods, and algorithm enhancements are also discussed. Chapter III details the research methodology, including dataset preparation, model architectures, training protocols, and evaluation strategies. The empirical methodology focuses on anomaly detection in fabrics, using SSIM and its variants as the loss function of an autoencoder. The process is divided into several key stages: defining the scope of defects to be detected, dataset acquisition of normal fabric images, preprocessing images to standardize and enhance relevant features, designing an autoencoder architecture, and training the autoencoder on the pre-processed dataset. The training aims to minimize the reconstruction error using SSIM and its variants. Chapter IV reports experimental results, comparative analyses, and discussions of failure modes and limitations. Chapter V concludes with key findings, practical implications, and directions for future work, including real-time integration and multimodal extensions.

2 Anomaly Detection

Anomaly detection has long been a prominent research field. It involves identifying data points that significantly deviate from expected, normal behavior. The topic has been extensively explored across diverse disciplines, including statistics, machine learning, data mining, information theory, and spectral analysis. Numerous techniques have been developed, some tailored to specific applications, while others offer broader, more general-purpose solutions (Chandola et al., 2009).

The origins of anomaly detection trace back to the 19th century within the field of statistics. These pioneering approaches were predominantly statistical and established the groundwork for subsequent methods. Contemporary research in anomaly detection integrates concepts from multiple disciplines, adapting them to address specific problem contexts. In recent decades, machine learning and data mining have emerged as particularly influential domains in advancing the field (Chandola et al., 2009).

The formulation of the anomaly detection problem is shaped by several key aspects that guide the selection and design of appropriate techniques:

- **The nature of the data:** Different data types require different techniques. For example, detecting anomalies in numerical data requires different methods than detecting anomalies in text data.
- **The availability of labelled data:** Labelled data, where anomalies are identified, is valuable for training and validating models. However, labeled data is often scarce in real-world applications. Obtaining datasets that are both accurate and truly representative of all possible behaviors, particularly rare occurrences such as anomalies can be prohibitively costly or, in many cases, practically impossible. (Ramdhani, 2025).
- **The type of anomalies to be detected:** Different applications may have different definitions of what constitutes an anomaly. For example, a small deviation in body temperature may be considered an anomaly in the medical domain, while a similar deviation in the stock market domain may not be considered anomalous (Chandola, Banerjee, & Kumar, 2009).

Anomalies can be categorised into distinct types based on their manifestation and the level of analysis required for detection:

- **Point Anomalies:** These are individual data instances that are anomalous compared to the rest of the data. For example, a sudden spike in network traffic could be a point anomaly.
- **Contextual Anomalies:** These are data instances that are anomalous within a specific context. For example, a temperature of 30 degrees Celsius would be considered normal in the summer but anomalous in the winter.
- **Collective Anomalies:** These are collections of data instances that are anomalous together, even if each instance might not be anomalous on its own. For example, a group of users simultaneously accessing sensitive files could be a collective anomaly, even if each user's access pattern individually seems normal.
- **Structural Anomalies:** These focus on deviations in the local structure or organization of data points within visual entities, such as scratches, distorted shapes, or missing components in an image.
- **Semantic Anomalies:** These encompass deviations at higher levels of interpretation, involving the relationships between multiple entities or the overall context of a scene. Examples include the presence of an unknown object in an image or an illogical arrangement of objects (Chandola, Banerjee, & Kumar, 2009; Cao, et al., 2024).

Unsupervised anomaly detection has gained prominence as a practical alternative to supervised methods, which, despite their effectiveness, demand extensive labelled datasets, a resource that is frequently scarce, costly, and biased toward known failure modes in industrial settings (Cui, Liu, & Lian, 2023). By operating solely on unlabelled data, unsupervised approaches infer normality from intrinsic data structure, enabling deployment in environments where anomalies are undefined or unobserved during training. This paradigm has spurred the development of diverse technique families, including (Cui, Liu, & Lian, 2023):

- **Reconstruction-based Methods:** These methods use deep learning models, such as autoencoders (AEs), Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs), to learn a representation of normal data. During testing, anomalies are detected by measuring the reconstruction error, assuming

that the model will struggle to reconstruct data points that deviate from the learned normal distribution.

- **Normalizing Flow (NF)-based Methods:** These methods learn the probability density function of normal data using normalizing flows. Anomalies are then identified as data points with low probability under the learned distribution.
- **Representation-based Methods:** These methods learn a feature representation where normal data points are clustered together, allowing anomalies to be identified as data points that are far from the normal cluster.
- **Data Augmentation-based Methods:** These methods address the scarcity of anomaly data by generating synthetic anomalies, providing the model with more diverse examples to learn from.
- **Algorithm Enhancements:** These refer to techniques or strategies incorporated into existing anomaly detection algorithms to improve their performance or address specific limitations. This can involve incorporating attention mechanisms to focus on relevant features, utilizing multiscale features to capture information at different levels of granularity, or incorporating domain-specific knowledge into the model.

Despite significant progress, anomaly detection faces persistent challenges that demand innovative solutions to ensure robustness and applicability across complex, real-world scenarios (Cao, et al., 2024):

- **Scarcity of Training Data:** The limited availability of labelled anomaly data, particularly in real-world settings, remains a significant hurdle. Addressing this challenge requires developing techniques that can effectively learn from limited data, such as few-shot learning or zero-shot learning approaches.
- **Diversity of Visual Modalities:** The increasing diversity of visual data, including images, videos, and 3D point clouds, requires algorithms that can generalize across modalities. This necessitates developing multimodal anomaly detection methods that can effectively capture and analyse information from various visual sources.
- **Complexity of Hierarchical Anomalies:** Anomalies can occur at different levels of granularity, from individual data points to complex relationships between entities. Developing methods that can effectively detect anomalies at different

hierarchical levels is crucial for understanding the full scope of potential deviations.

Future research in anomaly detection is poised to tackle emerging complexities and operational demands through several high-impact directions (Chandola, Banerjee, & Kumar, 2009; Cui, Liu, & Lian, 2023; Cao, et al., 2024):

- **Contextual and Collective Anomaly Detection:** Developing techniques that can effectively detect anomalies within specific contexts or identify groups of data instances that are anomalous together.
- **Distributed Anomaly Detection:** Detecting anomalies in data that is distributed across multiple locations, which is becoming increasingly relevant with the rise of distributed systems and sensor networks.
- **Privacy-Preserving Anomaly Detection:** Addressing privacy concerns when detecting anomalies in sensitive data.
- **Online Anomaly Detection:** Developing algorithms that can operate in real time, processing data as it arrives and detecting anomalies without requiring the entire dataset to be available upfront.
- **Anomaly Detection in Complex Systems:** Modelling and detecting anomalies in systems with multiple interacting components, such as aircraft systems or industrial control systems.
- **Foundation Models for Anomaly Detection:** Exploring the potential of large-scale pre-trained models, like foundation models, to enhance anomaly detection capabilities across various domains and modalities.
- **Multimodal Industrial Anomaly Detection:** Integrating information from multiple modalities, such as images, text, and sensor data, to improve the accuracy and robustness of anomaly detection in industrial settings.

Anomaly detection has evolved significantly, with researchers developing increasingly sophisticated techniques. The field is still evolving, and addressing the remaining challenges will be crucial for developing more effective and reliable anomaly detection systems.

The references above focus on unsupervised algorithms for visual industrial anomaly detection. This emphasis reflects the practical need for algorithms that can operate on unlabelled data in industrial settings. The research also highlights the importance of

developing methods that can address the specific challenges of industrial applications, such as the need for real-time performance and the ability to handle complex, hierarchical anomalies (Cui, Liu, & Lian, 2023).

2.1 Fabric Defect Detection Methods

Textile fabrication is a complex, large-scale manufacturing process, where the stability and quality of the produced materials are critical priorities for industrial enterprises (Chao, et al., 2021). Fabrics are widely used in diverse applications, including medical, aerospace, home, and apparel, with coloured and patterned textiles favoured for their varied designs (Wang , Wu, Wu, Lu , & Zhang, 2025). However, the textile manufacturing process is prone to the introduction of defects such as holes, knots, structural anomalies, and stains caused by mechanical factors, material quality, or process problems (Chao, et al., 2021). The presence of these flaws directly impacts product quality and market value, leading to substantial resource waste. Therefore, automated quality assurance, specifically an effective fabric defect detection method based on computer vision tasks, is essential for modern textile smart manufacturing, aligning with the trends of Industry 4.0 (Chao, et al., 2021).

Historically, cloth defect detection relied heavily on manual visual inspection (Chao, et al., 2021). These traditional methods are known to be inadequate, expensive, and suffer from low efficiency, high labour intensity, and inherent instability due to human subjective influence (Tang, et al., 2024). The necessity of early detection to mitigate financial losses has catalysed extensive research into robust and efficient automated detection algorithms over the past few decades (Chao, et al., 2021).

Fabric defect detection algorithms are broadly classified into two major categories: traditional algorithms and learning-based algorithms (Chao, et al., 2021). Traditional approaches typically utilize feature engineering based on prior knowledge and can be further grouped into four main subcategories:

1. **Statistical Algorithms:** These methods analyse the spatial distribution of grey values, often employing techniques like Grey-Level Co-occurrence Matrices (GLCM), autocorrelation analysis, or fractal dimension features. They aim to identify defective portions using measures such as coefficients of variation or eigenvalues (Chao, et al., 2021).

2. **Spectral Algorithms:** These rely on transforms such as Fourier, Gabor, and wavelet transforms to analyse the frequency domain characteristics of the fabric texture. Techniques utilizing the Discrete Cosine Transform (DCT) have also been explored (Chao, et al., 2021).
3. **Structural Algorithms:** These methods leverage the repetitive, patterned nature of fabrics, often using approaches like golden image subtraction (GIS) or lattice segmentation combined with template-based correction to isolate defective areas (Chao, et al., 2021).
4. **Model-Based Methods:** This category involves techniques that model fabric texture, such as motif-based methods, or use statistical models like the Gaussian mixture model (GMM) or support vector data description (SVDD) to differentiate normal fabric from anomalies (Chao, et al., 2021).

In recent years, the advancement of computing power and artificial intelligence has pushed the field toward supervised learning-based algorithms, which include classical Machine Learning (ML) and Deep Learning methods (Tang, et al., 2024). Supervised deep learning models, such as Convolutional Neural Networks, have demonstrated success in complex visual tasks (Chao, et al., 2021). These supervised learning-based algorithms for object detection are further distinguished as either Two-Stage detectors (e.g., Faster R-CNN, which first generates proposals) (Wang , Wu, Wu, Lu , & Zhang, 2025) or One-Stage detectors (e.g., the YOLO series, which directly extracts targets), with One-Stage detectors generally favoured for the rapid detection speeds required for online manufacturing inspection (Tang, et al., 2024).

Despite their efficacy, supervised deep learning models introduce significant practical challenges in textile manufacturing (Siegmund, Fu, Gárcia, Salahuddin, & Kuijper, 2021). A key issue is the scarcity and diversity of labelled defective images, resulting in imbalanced datasets and placing a heavy burden on training processes (Wang , Wu, Wu, Lu , & Zhang, 2025). This complexity drives demand toward unsupervised anomaly detection techniques, which do not rely on large amounts of labelled anomaly data and possess higher flexibility (Chao, et al., 2021). Methods like knowledge distillation and autoencoders form the mainstream of unsupervised anomaly detection (Wang , Wu, Wu, Lu , & Zhang, 2025). However, a fundamental limitation in these models is the risk of student networks suffering from overgeneralization, outputting similar feature

representations for both normal and anomalous samples, especially when anomalies are present during training (Wang , Wu, Wu, Lu , & Zhang, 2025).

Current research on both supervised and unsupervised methods endeavors to address these problems by implementing robust and efficient models. For example, the dual-task Teacher-Student (DTTS) unsupervised anomaly detection model tackles overgeneralization by integrating knowledge distillation and reconstruction tasks, alongside a masked convolution strategy, to enhance robustness to normal data and improve sensitivity to anomalies (Wang , Wu, Wu, Lu , & Zhang, 2025). Concurrently, there is a strong focus on developing lightweight neural network models (e.g., YOLOv7-tiny-MGCK) (Tang, et al., 2024). These solutions aim to balance high detection accuracy with minimized computational volume and parameter count through modifications like Ghost convolution modules, efficient activation functions (Mish), and adaptive upsampling techniques (CARAFE), ensuring models are suitable for deployment on resource-constrained embedded systems and meeting real-time requirements (Tang, et al., 2024).

2.2 Autoencoders

Autoencoders are a type of neural network architecture used for unsupervised learning, particularly effective in anomaly detection. They learn compressed representations of input data and are trained to reconstruct the input from this representation. Anomalies, being different from the normal patterns the autoencoder has learned, often result in high reconstruction errors (Kumar, Agraharam, Liu , & Namilae, 2024).

An autoencoder consists of two main parts: an encoder and a decoder (Liu & Chung, 2025). The encoder maps the input data into a lower-dimensional latent space, capturing the most salient features of the input (Kumar, Agraharam, Liu , & Namilae, 2024). The decoder then reconstructs the input from this latent representation (Liu & Chung, 2025). The training objective of an autoencoder is to minimize the difference between the input and the reconstructed output, typically using a loss function like mean squared error (MSE) or cross-entropy (Liu & Chung, 2025).

Convolutional Autoencoders (CAEs) replace fully connected layers with convolutional layers, making them effective for image data by capturing spatial relationships (Munir , Siddiqui, Dengel, & Ahmed, 2019). CAEs are used for image anomaly detection, extracting local features through convolutional layers (Liu & Chung, 2025).

Denoising Autoencoders (DAEs) are trained to reconstruct clean inputs from noisy ones, which encourages the autoencoder to learn more robust features, and are useful for detecting anomalies (Kumar, Agraharam, Liu , & Namilae, 2024).

Stacked Autoencoders (SAEs) consist of multiple layers of autoencoders, with each layer learning a higher-level representation of the input. The stacked architecture can capture complex feature hierarchies in the data (S. Dhiman, Deb, Muyeen, & Kamwa, 2021).

2.2.1 Anomaly Detection with Autoencoders

The principle of anomaly detection with autoencoders is that they reconstruct normal data well due to the fact that they are trained on it. In contrast, they perform poorly with anomalous inputs (Kumar, Agraharam, Liu , & Namilae, 2024). The reconstruction error (the difference between the input and the reconstructed output) is used as the anomaly score. High reconstruction errors indicate anomalies, and low errors indicate normal data (Liu & Chung, 2025).

Feature-based anomaly detection can be detected using features learned by the autoencoder by measuring the distance of the data point representation to the centroid of the normal data points (Kumar, Agraharam, Liu , & Namilae, 2024).

Variational Autoencoders (VAEs) are a probabilistic approach to autoencoders, learning a probability distribution of the latent space which makes them suitable for data generation. VAEs can be trained with non-regularized objective functions, useful for heterogeneous datasets (Cui, Liu, & Lian , 2023).

2.2.2 Thresholding

A widely adopted approach in anomaly detection involves thresholding the reconstruction error to distinguish normal from anomalous regions (Liu & Chung, 2025). Most algorithms generate an anomaly score or anomaly map, which must be thresholded to classify instances (at the image level) or individual pixels (at the pixel level) as anomalous (Munir , Siddiqui, Dengel, & Ahmed, 2019). The choice of threshold significantly affects detection performance: some methods include built-in thresholding mechanisms, while others require empirical estimation (Bergmann, Fauser, Sattlegger, & Steger, 2019).

A fundamental distinction exists between image-level detection and pixel-level detection (Chandola, Banerjee, & Kumar, 2009). Image-level detection, often implemented via global thresholding or classification, determines whether an entire image contains an

anomaly (Wang , Wu, Wu, Lu , & Zhang, 2025). In contrast, pixel-level detection, also known as anomaly segmentation, applies thresholding at the pixel scale to produce a detailed anomaly map. This enables precise localization of defects, which is essential in industrial inspection where identifying the exact position, shape, and extent of flaws (e.g., tears, stains, or misweaves in fabrics) is critical for quality control and downstream decision-making (Bergmann, Fauser, Sattlegger, & Steger, 2019).

2.2.2.1 Image-Level Detection (Classification/Image Thresholding)

Image-level detection refers to determining if an entire image contains an anomaly, treating the image as a single unit for classification (Bergmann, Fauser, Sattlegger, & Steger, 2019).

This task is commonly referred to as outlier detection or one-class classification:

Goal: The primary objective is to test whether new input data matches the distribution of the normal, defect-free training data (Bergmann, Fauser, Sattlegger, & Steger, 2019). The method provides a binary decision: the image is either classified as anomalous or defect-free (Chandola, Banerjee, & Kumar, 2009).

Decision Basis: The decision is based on the entire image, focusing on whether the sample belongs to the "inlier" (normal) distribution (Bergmann, Fauser, Sattlegger, & Steger, 2019). This is distinct from finding the exact location of the defect within the image (Bergmann, Fauser, Sattlegger, & Steger, 2019).

Image Thresholding Context: While methods often produce a score for the entire image (representing its overall deviation from the norm) (Chandola, Banerjee, & Kumar, 2009), "image thresholding" in this context refers to applying a single threshold to this overall image score (Bergmann, Fauser, Sattlegger, & Steger, 2019). If the score surpasses this threshold, the entire image is flagged as anomalous (Bergmann, Fauser, Sattlegger, & Steger, 2019).

Evaluation: For this scenario, performance is assessed by computing the accuracy of correctly classified images (both anomalous and anomaly-free test images) (Bergmann, Fauser, Sattlegger, & Steger, 2019).

2.2.2.2 Pixel-Level Detection (Segmentation/Pixel Thresholding)

Pixel-level detection, or anomaly segmentation, requires the algorithm to locate and delineate the specific spatial regions that contain anomalies, often manifesting as subtle deviations in small, confined areas (Bergmann, Fauser, Sattlegger, & Steger, 2019):

Goal: The requirement is to provide a pixel-accurate segmentation of anomalous regions (Bergmann, Fauser, Sattlegger, & Steger, 2019). This task is critical for industrial inspection, where providing pixel-precise ground truth is necessary (Bergmann, Fauser, Sattlegger, & Steger, 2019).

Anomaly Map Generation: Most state-of-the-art anomaly segmentation methods, such as AnoGAN and Autoencoders, generate a one-channel spatial map (an anomaly map). In this map, large values indicate that a specific pixel belongs to an anomalous region.

Pixel Thresholding: The key difference is that a secondary step pixel thresholding must be performed to convert this continuous spatial anomaly map into a binary segmentation result (Bergmann, Fauser, Sattlegger, & Steger, 2019). This threshold must be determined so that a binary decision for each individual pixel within the image can be made, classifying it as "anomalous" or "normal" (Bergmann, Fauser, Sattlegger, & Steger, 2019).

Evaluation: Segmentation performance is measured using metrics like the relative per-region overlap with the ground truth and the Area Under the Receiver Operating Characteristic curve (ROC AUC) (Wang , Wu, Wu, Lu , & Zhang, 2025).

2.2.2.3 The Critical Difference: Threshold Application

The core distinction between the two approaches lies in the object to which the threshold is applied, and the resulting complexity of setting that threshold (Chandola, Banerjee, & Kumar, 2009):

Feature	Image-Level Detection (Threshold Image)	Pixel-Level Detection (Threshold Pixel)
Object of Threshold	A single, global anomaly score for the whole image.	A continuous score is generated for every single pixel (the spatial anomaly map).
Result	A simple binary flag (Yes/No).	A detailed binary mask showing exact defect boundaries.
Threshold Difficulty	Generally simpler, though methods still need robust hyperparameters.	Highly difficult in unsupervised settings.

Table 1 - Comparison of image-level and pixel-level defect detection paradigms in unsupervised autoencoder-based segmentation.

The difficulty of setting the pixel threshold is a significant issue in segmentation (Bergmann, Fauser, Sattlegger, & Steger, 2019). The sources note that even when an anomaly map successfully represents the presence of defects (high ROC AUC), the segmentation (per-region overlap) can still fail due to a bad estimation of the threshold (Bergmann, Fauser, Sattlegger, & Steger, 2019). In unsupervised learning, this threshold must often be estimated solely from a set of anomaly-free validation images, which adds considerable difficulty (Bergmann, Fauser, Sattlegger, & Steger, 2019).

2.2.3 Applications

Autoencoders have been successfully applied across a wide range of anomaly detection tasks, demonstrating their versatility in learning normal data patterns to identify deviations in diverse domains:

- **Industrial Inspection autoencoders** are used to detect defects in manufactured products, such as surface defects, by analysing images (Kumar, Agraharam, Liu , & Namilae, 2024). In this field, autoencoders are able to learn normal patterns of a surface and highlight deviations as anomalies.
- **Time Series Data autoencoders** are also used for time series anomaly detection in various domains, such as detecting anomalies in wind turbine data (Liu & Chung, 2025).

- **Network Intrusion Detection autoencoders** can detect unusual patterns in network traffic, identifying potential security breaches (Xu, Jang-Jaccard, Singh, Wei, & Sabrina, 2021).
- **Medical Imaging autoencoders** can identify anomalies in medical images like brain MRIs (Liu & Chung, 2025).

2.2.4 Limitations

Despite their effectiveness, autoencoders have limitations. The core function of autoencoders (AEs) within unsupervised anomaly detection relies fundamentally on their capacity to accurately model and reconstruct normal data, while failing to accurately reconstruct anomalies (Bergmann, Fauser, Sattlegger, & Steger, 2019). This expected behavior forms the basis for successful anomaly identification (Wang , Wu, Wu, Lu , & Zhang, 2025).

Autoencoders are traditionally trained exclusively on large datasets of normal, defect-free samples, allowing the network to build a robust feature representation of normal data in its bottleneck or latent space (Wang , Wu, Wu, Lu , & Zhang, 2025).

The standard assumption for successful anomaly detection is that when a novel or anomalous test instance is introduced, the AE will be unable to encode and subsequently decode it accurately, thus causing a significant measure of divergence (Wang , Wu, Wu, Lu , & Zhang, 2025). Anomalies are consequently detected by calculating a large reconstruction error between the input image and its generated output, which serves directly as the anomaly score for the instance (Chandola, Banerjee, & Kumar, 2009). This difference is measured using techniques such as per-pixel comparison or Structural Similarity (SSIM) (Bergmann, Fauser, Sattlegger, & Steger, 2019). This training strategy enhances the network's sensitivity to abnormal data by increasing the reconstruction error for anomalies (Wang , Wu, Wu, Lu , & Zhang, 2025).

A critical limitation arises when the assumption of reconstruction failure for anomalous inputs does not hold true. Autoencoders may fail to detect certain anomalies if the reconstruction is close to the input, despite the input being anomalous (Liu & Chung, 2025).

This scenario, which results in a false negative, occurs if the aberrant features of the anomaly are sufficiently small or structurally aligned with the learned latent space such

that the network can reconstruct the input accurately (Bergmann, Fauser, Sattlegger, & Steger, 2019). If the reconstruction error is minimal, the anomalous instance is erroneously classified as normal (Chandola, Banerjee, & Kumar, 2009). This specific failure mode contradicts the very principle upon which AE-based anomaly detection relies, that the autoencoder is supposed to fail to reproduce images that differ from the learned distribution during testing (Bergmann, Fauser, Sattlegger, & Steger, 2019). This vulnerability highlights a challenge in ensuring that the feature representation learned by the student model, robust though it may be to normal samples (Wang , Wu, Wu, Lu , & Zhang, 2025), remains adequately sensitive to all possible deviations from the norm.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) play a specialized and crucial role in fabric anomaly and defect detection, primarily by addressing the persistent challenge of data scarcity and imbalanced datasets in textile manufacturing (Chao, et al., 2021).

In industrial settings, obtaining a large quantity of labeled fabric images with defects is often very difficult (Alankrita , Mamta , & Gopi, 2021). Fabric defects are typically rare compared to normal, defect-free samples (Wang , Wu, Wu, Lu , & Zhang, 2025). GANs are used to mitigate this problem by serving as a powerful data augmentation tool (Chao, et al., 2021).

GAN-based algorithms are capable of automatically adapting to different fabric textures by learning from existing fabric defect samples (Wang , Wu, Wu, Lu , & Zhang, 2025).

Researchers utilize non-defect image data to generate synthetic defective image data using GANs, thereby leveraging expert knowledge regarding defect characteristics to create the necessary training inputs (Chao, et al., 2021). The ability of GANs to create realistic synthetic data is particularly useful when dealing with unbalanced and rare datasets of defective images (Chao, et al., 2021).

2.3.1 Specific GAN Architectures and Techniques

One approach involves training a multistage GAN model to synthesize reasonable defect samples from nondefect samples (Chao, et al., 2021). The goal of generating this synthetic data is to support the training of a subsequent deep semantic segmentation network designed to detect the fabric defects (Chao, et al., 2021). This method has been verified

through comprehensive experiments on various typical fabric image samples (Wang , Wu, Wu, Lu , & Zhang, 2025).

Wasserstein Generative Adversarial Nets (WGANs) have been utilized in combination with transfer learning techniques and multimodel ensembling frameworks for surface defect detection (Wang , Wu, Wu, Lu , & Zhang, 2025). This scheme proved effective when dealing with unbalanced and rare datasets of defective images (Chao, et al., 2021).

2.3.2 Context and Related Unsupervised Methods

While GANs excel at synthetic data generation, the overall trend in high-efficiency fabric inspection is moving toward unsupervised learning models to reduce reliance on large amounts of labeled data (Wang , Wu, Wu, Lu , & Zhang, 2025).

Given the complexity of fabric texture structure and the wide variety of anomalies, unsupervised learning is proving highly effective, especially when dealing with imbalanced datasets where normal samples far outnumber anomalies (Wang , Wu, Wu, Lu , & Zhang, 2025).

The general field of unsupervised anomaly detection for colored textiles employs models based on techniques like knowledge distillation and autoencoders (Wang , Wu, Wu, Lu , & Zhang, 2025). For example, the Dual-Task Teacher-Student (DTTS) model integrates knowledge distillation and reconstruction to enhance the robustness of the model when handling normal data, ensuring better differentiation when anomalous data is encountered (Wang , Wu, Wu, Lu , & Zhang, 2025). These methods aim to solve the same fundamental problem as GAN´s identifying rare anomalies without extensive labeled training data (Chao, et al., 2021).

2.3.3 Limitations

Generative Adversarial Networks (GANs) represent a significant advancement in deep learning generative modeling. However, their deployment, particularly in complex domains such as multi-stage image reconstruction and high-dimensional synthesis, is constrained by inherent challenges related to training stability, computational intensity, and time consumption.

The core architecture of a GAN, conceptualized as a zero-sum game between the Generative Network (G) and the Discriminative Network (D), is often characterized by

inherent instability and difficulties in achieving convergence (Koç, Özyurt, & Iantovics, 2024).

GAN training can be complex and frequently results in non-union (a lack of convergence) and training instability (Koç, Özyurt, & Iantovics, 2024). This systemic difficulty is fundamentally linked to the deficiency of the GAN essential hypothesis, which poses a significant obstruction to developing high-quality generative models (Alankrita , Mamta , & Gopi, 2021).

A prevalent issue is mode collapse, where the network fails to generalize effectively, resulting in the network missing entire modes from the input data distribution (Alankrita , Mamta , & Gopi, 2021). This negatively impacts the diversity and precision of the generated output (Koç, Özyurt, & Iantovics, 2024).

Even when utilizing advanced strategies designed to enhance stability, such as employing Wasserstein GANs (WGANs) or the Gumbel-softmax distribution, the optimization steps can be challenging (Koç, Özyurt, & Iantovics, 2024). These steps often include problems like slow or unstable optimization processes and the need for demanding computations on high-dimensional data. Future research needs breakthroughs in hypothetical aspects to tackle these issues (Alankrita , Mamta , & Gopi, 2021).

Deep learning models, including GANs, generally rely on high-end machines and take a long time to do model training (Alankrita , Mamta , & Gopi, 2021). This computational burden is further intensified by the iterative and complex nature of adversarial optimization and specialized architectural requirements (Alankrita , Mamta , & Gopi, 2021).

The complexity of deep learning means that datasets are largely dependent on powerful hardware (Alankrita , Mamta , & Gopi, 2021). Conversely, in deployment scenarios, such as automated textile inspection, the actual production line requires high real-time performance but must often operate on platforms with limited computing power (Chao, et al., 2021). This creates a fundamental tension between model complexity and operational efficiency (Chao, et al., 2021).

High computational cost results from frameworks designed to overcome standard model limitations. For instance, the Dual-Task Teacher-Student (DTTS) unsupervised anomaly detection model for colored textiles utilizes a parallel computing feature within its dual-

task framework (knowledge distillation and reconstruction) (Wang , Wu, Wu, Lu , & Zhang, 2025). While this enhances feature characterization, it simultaneously leads to higher model complexity. Rigorous experiments on such complex models require substantial resources (Wang , Wu, Wu, Lu , & Zhang, 2025).

2.4 Transformers

The transformer neural network architecture, introduced in the seminal paper "Attention is All You Need" (Vaswani, et al., 2017), has become a foundational paradigm across multiple domains, underpinning models such as Google's BERT and OpenAI's GPT series (Vaswani, et al., 2017). Transformers consistently outperform prior benchmarks (Vaswani, et al., 2017). Although originally designed for sequential data processing, similar to Recurrent Neural Networks (RNNs), transformers feature a flexible architecture comprising an encoder, a decoder, or only a decoder, depending on the task (Chao, et al., 2021). Their core innovation lies in self-attention mechanisms, which enable the model to capture long-range dependencies across the entire input sequence regardless of token distance (Ramdhani, 2025), a significant improvement over RNNs, which process data sequentially via recurrent connections. Both the encoder and decoder (when present) consist of stacked layers incorporating multi-head self-attention, residual connections, layer normalization, and position-wise feed-forward networks (Orabi, Phuc Tran, Egger, & Thomassey, 2024).

The use of transformer models has been crucial for performance gains across many areas (Koç, Özyurt, & Iantovics, 2024). This includes significant advancements in language translation and Natural Language Processing (NLP) (Chao, et al., 2021). Moreover, in the manufacturing sector, transformers have been instrumental in predictive maintenance, with models like Trans-Lighter being used. Their application in human activity recognition (HAR) (Chao, et al., 2021) has also been effectively explored. However, transformers have limitations, such as high computational complexity and memory requirements, especially with long sequences, due to self-attention mechanisms that scale quadratically with sequence length. Also, transformers require substantial amounts of training data to reach optimal performance, which makes them less effective for tasks with limited data. Training transformers can be time-consuming, often demanding significant computational resources and extensive training (Chandola, Banerjee, & Kumar, 2009).

In the context of industrial image analysis, a pre-trained Vision Transformer (ViT) can be employed for feature extraction and image reconstruction (Liu & Chung, 2025). The extracted features are then used in an autoencoder (Cui, Liu, & Lian , 2023), or transformers are used directly as a reconstruction system (Liu & Chung, 2025).

There is an identified need to enhance transformers with a mechanism for classifying inputs into normal or abnormal categories, that is adaptive to changes in input data (Liu & Chung, 2025).

In time series analysis, the transformer's self-attention mechanism can capture correlations across an entire time series (S. Dhiman, Deb, Muyeen, & Kamwa, 2021).

The Segment Anything Model (SAM) represents an innovative approach that combines a Single Shot Detector (SSD) for local feature detection with the SAM model for global feature representation. This approach utilizes a transformer-based image encoder to extract high-level features. These features are useful for both anomaly detection and segmentation (Wahid, et al., 2025). Current approaches in both anomaly detection and localization increasingly leverage transformers, as opposed to CNNs and diffusion models (Cao, et al., 2024). In anomaly detection, transformer models are seen as beneficial because they offer improved context values, and also because they learn richer embeddings than prior models (Jafari, 2022).

2.5 Classification Metrics

In fabric anomaly detection, the primary goal is often to perform binary classification, distinguishing anomalies from normal instances (Xu, Jang-Jaccard, Singh, Wei, & Sabrina, 2021). This task is central both during model evaluation and real-time inference on production lines, where accurate decisions directly impact product quality and manufacturing efficiency. Given the severe class imbalance typical in industrial settings (defects often <1% of data), standard metrics must be interpreted cautiously. The following are widely used for both validation and deployment-stage inference.

2.5.1 Accuracy

Accuracy measures the proportion of correctly classified instances out of the total number of instances (S. Dhiman, Deb, Muyeen, & Kamwa, 2021) It is calculated as follows:

$$Accurracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where:

- TP (True Positives): number of correctly identified anomalies;
- TN (True Negatives): number of correctly identified normal instances;
- FP (False Positives): number of normal instances incorrectly classified as anomalies;
- FN (False Negatives): number of anomalies incorrectly classified as normal instances.

While accuracy is a simple metric, it can be misleading in imbalanced datasets where one class significantly outnumbers the other. In such cases, high accuracy can be achieved even if the algorithm performs poorly on the minority class (anomalies) (S. Dhiman, Deb, Muyeen, & Kamwa, 2021).

2.5.2 Precision

Precision quantifies the proportion of correctly identified anomalies out of all instances predicted as anomalous (S. Dhiman, Deb, Muyeen, & Kamwa, 2021). It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

High precision ensures that when the system flags a defect during online inference, it is likely genuine, minimizing false alarms that disrupt production, trigger unnecessary rework, or waste acceptable fabric.

A high precision indicates that when the algorithm detects an anomaly (S. Dhiman, Deb, Muyeen, & Kamwa, 2021), it is likely to be a true anomaly.

2.5.3 Recall (Sensitivity)

Recall, also known as sensitivity, measures the proportion of correctly identified anomalies out of all actual anomalies (S. Dhiman, Deb, Muyeen, & Kamwa, 2021). It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

A high recall indicates that the algorithm can identify most of the true anomalies (S. Dhiman, Deb, Muyeen, & Kamwa, 2021).

2.5.4 F1-Score

The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance (Munir , Siddiqui, Dengel, & Ahmed, 2019). It is calculated as follows:

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The F1-score is particularly useful when both false positives and false negatives are important considerations (Munir , Siddiqui, Dengel, & Ahmed, 2019).

These metrics are used in various anomaly detection contexts. For instance, in the context of network intrusion detection, an autoencoder-based model was evaluated using accuracy, precision, recall and F1-score (Xu, Jang-Jaccard, Singh, Wei, & Sabrina, 2021). In wind turbine gearbox anomaly detection, similar metrics were used to assess the performance of different classifiers like Support Vector Machine (SVMs), K-Nearest Neighbors (KNNs), and neural networks (S. Dhiman, Deb, Muyeen, & Kamwa, 2021).

2.5.5 Fβ metric

The F_β score generalizes the F1-score by introducing a tunable parameter β that controls the relative importance of recall over precision (Fujino , Isozaki, & Suzuki, 2008):

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{5}$$

- When $\beta = 1$ F_β reduces to the F1-score (equal weighting).
- When $\beta > 1$, recall is prioritized (e.g., $\beta = 2$ weights recall twice as much as precision), reducing false negatives—desirable when missing anomalies is costly (Oksuz, Cam, Akbas, & Kalkan, 2018).

- When $\beta < 1$, precision dominates, favoring conservative detection.

This flexibility enables domain-specific optimization. For example, an optimal β value can be integrated as a dynamic penalty in loss functions, such as weighted binary cross-entropy, to align training objectives with operational priorities (Lee, Yang , & Yoo, 2021).

These metrics support model selection, threshold calibration, and continuous performance monitoring during inference, ensuring reliable, adaptive defect detection across varying fabric types, lighting, and line speeds.

2.6 Segmentation Metrics

In anomaly detection tasks where defects appear as localized regions within images or time series—such as stains, tears, or weave irregularities in fabrics—segmentation metrics are indispensable for assessing the algorithm's ability to precisely delineate anomalous boundaries (Bergmann et al., 2019). Unlike classification metrics, these evaluate pixel-level or region-level spatial agreement between predicted and ground-truth masks, which is critical for industrial applications requiring not only detection but accurate localization for quality grading, repair, or process feedback. The MVTec AD dataset, widely used in fabric and surface defect studies, exemplifies the need for such fine-grained evaluation (Bergmann et al., 2019).

The Area Under the Receiver Operating Characteristic Curve (ROC AUC) is a threshold-independent performance measure that evaluates the model's ability to rank pixels by their likelihood of being anomalous across all possible decision thresholds (S. Dhiman, Deb, Muyeen, & Kamwa, 2021). It is derived from the ROC curve, which plots the True Positive Rate (TPR = Recall) against the False Positive Rate (FPR):

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN} \tag{6}$$

The resulting ROC AUC value (0 to 1) quantifies overall discriminative power: AUC = 1 denotes perfect ranking of anomalous vs. normal pixels, while AUC = 0.5 indicates random performance.

In fabric anomaly detection, AUC-ROC is a standard benchmark metric (e.g., on MVTec AD) due to several key advantages: (1) remains robust under extreme pixel-level imbalance (defective pixels ≪ normal); (2) facilitates fair model comparison independent

of operational threshold; (3) supports post-hoc threshold selection during inference to adapt to varying lighting, texture, or production conditions (Munir , Siddiqui, Dengel, & Ahmed, 2019).

This flexibility is crucial in production environments, where a fixed threshold may fail under changing inputs, while AUC-ROC ensures consistent ranking quality.

3 Methodology

This chapter details the core technical framework for unsupervised defect segmentation in organic fabrics developed in this work. The approach centers on convolutional autoencoders trained to reconstruct defect-free texture patterns using a perceptually grounded loss function based on the Structural Similarity Index Measure (SSIM). Two autoencoder variants are presented: a lightweight model suited for real-time deployment and an extended version tailored for highly intricate textures. The SSIM loss is analyzed in depth, with comparisons to conventional $\ell_2$reconstruction error and an optional multi-scale extension (MS-SSIM) for enhanced sensitivity to defects across spatial frequencies. This methodology enables robust anomaly detection by leveraging structural discrepancies between inputs and reconstructions, while remaining resilient to natural texture variations inherent in organic materials.

3.1 Autoencoders for Unsupervised Defect Segmentation

Autoencoders are designed to reconstruct an input image $x \in \mathbb{R}^{k\times h\times w}$ by first compressing it into a compact latent representation and then decoding it back to the original dimensions. The architecture comprises two main components: an encoder $E: \mathbb{R}^{k\times h\times w} \to \mathbb{R}^d$, which maps the input to a lower-dimensional latent vector z, and a decoder $D: \mathbb{R}^d \to \mathbb{R}^{k\times h\times w}$, which generates the reconstructed output $\hat{x}$. The latent dimension d is deliberately chosen to be much smaller than the input volume ($d \ll k \times h \times w$), preventing trivial identity mapping and compelling the network to learn meaningful, compact features essential for accurate reconstruction.

The reconstruction process is expressed as:

$$\hat{x} = D(E(x)) = D(z) \tag{7}$$

In this work, both the encoder and decoder are implemented using convolutional neural networks (CNNs). The encoder progressively reduces spatial resolution through strided convolutions, while the decoder restores the original size via transposed convolutions (also known as deconvolutions). This symmetric, fully convolutional design enables efficient processing of high-resolution fabric images and preserves spatial hierarchies critical for texture analysis.

For unsupervised defect segmentation, the autoencoder is trained exclusively on defect-free samples of organic fabrics. During training, the model learns to faithfully reconstruct normal texture patterns, such as consistent weave structures, fiber alignments, and natural grain variations. At inference, when presented with a test image containing a defect (e.g., a tear, stain, or weave disruption), the autoencoder cannot accurately reproduce the anomalous region, as it lies outside the manifold of learned normal data. This results in perceptually meaningful reconstruction errors in defective areas.

3.1.1 Autoencoder Architecture

Two variants of the encoder were implemented to explore the impact of architectural depth and complexity on reconstruction quality and anomaly detection performance: the SSIM-AE encoder, which is compact and computationally lightweight, and the SSIM-AEe (extended) encoder, which incorporates additional layers for enhanced feature representation in highly complex organic textures. Both encoders are paired with symmetric decoders using transposed convolutions to upsample the latent representation back to the original input resolution of 128×128×3.

The SSIM-AE encoder, detailed in Table 2, follows a progressive downsampling strategy with a combination of strided convolutions and same-resolution refinement layers. It begins with a 4×4 kernel convolution (stride 2, padding 1) to reduce the spatial dimension from 128×128 to 64×64 while expanding the channel depth to 32. This is followed by another 4×4 strided convolution to reach 32×32×32, after which a 3×3 convolution (stride 1, padding 1) refines features without altering resolution. The process continues with a 4×4 strided convolution to 16×16×64, another 3×3 refinement, and a final 4×4 strided convolution to 8×8×128. Two additional 3×3 convolutions at 8×8 resolution (reducing channels to 64 and then 32) act as bottleneck refinement stages before a global 8×8 convolution (stride 1, padding 0) compresses the feature map into the latent vector of dimension d at 1×1 spatial size. This design ensures multi-scale feature extraction with controlled capacity, suitable for fabrics with moderate texture complexity.

Layer	Output Size	Kernel	Stride	Padding	Parameters
Input	128×128×3				
Conv1	64×64×32	4×4	2	1	1,568
Conv2	32×32×32	4×4	2	1	16,416
Conv3	32×32×32	3×3	1	1	9,248
Conv4	16×16×64	4×4	2	1	32,832
Conv5	16×16×64	3×3	1	1	36,928
Conv6	8×8×128	4×4	2	1	131,200
Conv7	8×8×64	3×3	1	1	73,792
Conv8	8×8×32	3×3	1	1	18,464
Conv9	1×1×d	8×8	1	0	1,024,500

Table 2 - Architecture of the encoder of SSIM-AE for input images 128 x 128 x 3.

In contrast, the SSIM-AEe encoder, presented in Table 3, adopts a more gradual and deeper architecture to better capture intricate, high-frequency patterns in nanofibrous and densely woven organic materials. It initiates downsampling with a 3×3 convolution (stride 2, padding 1) to 64×64×32, followed by two 3×3 convolutions (stride 1, padding 1) for local feature enhancement at the same resolution. A subsequent 3×3 strided convolution reduces the feature map to 32×32×64, again refined by two 3×3 convolutions. The pattern repeats with a 3×3 strided convolution to 16×16×128 (with two refinement layers) and another to 8×8×256 (with two more refinement layers). Finally, an 8×8 global convolution compresses the representation into the $1\times1\times d$ latent space. This extended structure increases receptive field coverage and feature richness at each scale, enabling superior modeling of subtle weave variations and fine fiber details.

Layer	Output Size	Kernel	Stride	Padding	Parameters
Input	128×128×3	3×3			
Conv1	64×64×32	3×3	2	1	896
Conv2	64×64×32	3×3	1	1	9,248
Conv3	64×64×32	3×3	1	1	9,248
Conv4	32×32×64	3×3	2	1	18,496
Conv5	32×32×64	3×3	1	1	36,928
Conv6	32×32×64	3×3	1	1	36,928
Conv7	16×16×128	3×3	2	1	73,856
Conv8	16×16×128	3×3	1	1	147,584
Conv9	16×16×128	3×3	1	1	147,584
Conv10	8×8×256	3×3	2	1	295,168
Conv11	8×8×256	3×3	1	1	590,080
Conv12	8×8×256	3×3	1	1	1,638,500
Conv13	1×1×d	8×8	1	0	1,638,656

Table 3 - Architecture of the encoder of SSIM-AEe for input images 128 x 128 x 3.

Both encoders utilize Leaky ReLU activations after each convolutional layer except the final latent projection, which remains linear to preserve the full dynamic range of the compressed representation. Batch normalization is applied after every convolution to stabilize training dynamics and facilitate smoother gradient flow across deep hierarchies. The latent dimension d is treated as a dataset-specific hyperparameter to achieve an optimal trade-off between compression ratio and representational capacity.

The decoder in both autoencoder variants (SSIM-AE and SSIM-AEe) mirrors the corresponding encoder in reverse order, employing transposed convolutions with identical kernel sizes, strides, and paddings to ensure precise spatial alignment between input and reconstruction. This symmetric design enables the progressive restoration of high-resolution fabric details, from coarse weave patterns to fine fiber textures. The final output layer uses a sigmoid activation to constrain pixel values to the normalized range $[0,1]$, consistent with the preprocessed input distribution.

These architectures enable end-to-end training with the SSIM loss function, ensuring that the model learns perceptually meaningful representations of defect-free textures. The compact SSIM-AE is preferred for real-time industrial deployment due to lower inference latency, while the deeper SSIM-AEe is employed in high-accuracy scenarios where computational resources permit.

3.2 Structural Similarity Loss Function

The Structural Similarity Index Measure (SSIM) serves as the core perceptual loss function in this work, replacing conventional pixel-wise metrics such as Mean Squared Error (MSE) or ℓ_2-distance. While MSE penalizes any deviation in intensity on a per-pixel basis, leading to high residuals even for minor edge misalignments or natural texture shifts, SSIM evaluates image quality from a human visual perception perspective, focusing on luminance, contrast, and structural consistency across local image regions. This property makes it particularly effective for organic fabrics, where defects (e.g., tears, stains, weave breaks) disrupt structural patterns rather than just pixel intensities, and natural variations (e.g., fiber grain, weave irregularity) must not trigger false positives.

SSIM compares two image patches p (from the input x) and q (from the reconstruction $\hat{x}$) using three independent components:

$$\mathrm{SSIM}(p,q) = \overbrace{[l(p,q)]^{\alpha}}^{\text{luminance}} \cdot \overbrace{[c(p,q)]^{\beta}}^{\text{contrast}} \cdot \overbrace{[s(p,q)]^{\gamma}}^{\text{structure}} \tag{8}$$

where:

- **Luminance** $l(p,q) = \frac{2\mu_p\mu_q + C_1}{\mu_p^2 + \mu_q^2 + C_1}$ measures similarity in brightness, based on the mean intensities μ_p and μ_q of the patches. The numerator $2\mu_p\mu_q$ measures how close the mean intensities are while the denominator normalizes the comparison using the sum of squared means. The constant c_1 prevents division by zero when the mean intensities are small. If $\mu_p = \mu_q$, the luminance term approaches 1 (perfect similarity).
- **Contrast** $c(p,q) = \frac{2\sigma_p\sigma_q + C_2}{\sigma_p^2 + \sigma_q^2 + C_2}$ measures similarity in the variation of pixel intensities, based on the standard deviations σ_p and σ_q of the patches. The numerator $2\sigma_p\sigma_q$ reflects how similar the contrast (spread of pixel intensities) is, and the denominator normalizes using the sum of variances. The constant c_2 ensures stability when variances are small. If $\sigma_p = \sigma_q$, the contrast term approaches 1.

- **Structure** $s(p,q) = \frac{\sigma_{pq}+C_2}{2\sigma_p\sigma_q+C_2}$ measures similarity in patterns or textures, based on the covariance σ_{pq} between the patches. The covariance quantifies how pixel intensities in p and q co-vary, reflecting structural similarity. The denominator normalizes by the product of standard deviations. The constant c_2 ensures stability. If the patches have identical patterns (high covariance relative to their variances), this term approaches 1.

The stabilization constants are typically set to $c_1 = (0.01 \cdot L)^2$, $c_2 = (0.03 \cdot L)^2$ (with $L = 1$ for normalized images in [0,1]). By combining the three components, with the exponents typically set to $\alpha = \beta = \gamma = 1$, the SSIM becomes:

$$\mathrm{SSIM}(p,q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)} \tag{9}$$

SSIM values lie in $[-1,1]$. where $\mathrm{SSIM} = 1$ indicates that the patches p and q are identical. Lower values reflect differences in luminance, contrast, or structure.

The key statistical quantities used in the SSIM components are defined as follows for patches of size $K \times K$:

Symbol	Meaning	Formula
μ_p	Mean intensity of p	$\mu_p = \frac{1}{K^2}\sum x_i$
σ_p^2	Variance of p	$\sigma_p^2 = \frac{1}{K^2-1}\sum(x_i - \mu_p)^2$
σ_{pq}	Covariance between p and q	$\sigma_{pq} = \frac{1}{K^2-1}\sum(x_i - \mu_p)(y_i - \mu_q)$

Table 4 - Definitions of key statistical quantities for SSIM computation over K × K patches.

For full images, SSIM is computed over sliding Gaussian-weighted windows of size $K \times K$ (usually $K = 11$), and the final score is the mean over all windows:

$$\mathrm{SSIM}(x,\hat{x}) = \frac{1}{M}\sum_{i=1}^{M}\mathrm{SSIM}(p_i, q_i) \tag{10}$$

where $M$is the number of windows. The training loss is then defined as:

$$\mathcal{L}_{\text{SSIM}}(x, \hat{x}) = 1 - \text{SSIM}(x, \hat{x}) \tag{11}$$

This loss is differentiable and can be directly optimized via backpropagation. By training the autoencoder to minimize $\mathcal{L}_{\text{SSIM}}$, the model learns to preserve the perceptual structure of defect-free organic fabrics.

During inference, the per-pixel anomaly map is generated as:

$$R(x, \hat{x}) = 1 - \text{SSIM}_{\text{local}}(x, \hat{x}) \tag{12}$$

where $\text{SSIM}_{\text{local}}$ is computed at each pixel using its local neighborhood. High values in R indicate regions where structural integrity is compromised, corresponding to visually salient defects, while low values correspond to normal texture, even if pixel intensities vary slightly. At test time, defects appear as bright, coherent regions in the SSIM residual map, enabling accurate, robust, and industrially viable unsupervised defect segmentation.

3.2.1 Comparison of SSIM and ℓ_2 (L2) Loss Functions

The selection of the loss function in an autoencoder framework for unsupervised defect detection is critical, as it directly influences the model's capacity to differentiate genuine structural anomalies from natural texture variations inherent to organic fabrics. In this work, the Structural Similarity Index Measure (SSIM) is employed as the primary loss function, in contrast to the conventional $\ell_2$distance, also known as Mean Squared Error (MSE). The ℓ_2 loss is defined as the average of squared intensity differences between corresponding pixels in the input image x and its reconstruction $\hat{x}$, expressed as:

$$\mathcal{L}_{\ell_2}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{13}$$

This metric operates on a strictly pixel-wise basis, treating each intensity value independently without considering spatial relationships or perceptual relevance.

In contrast, SSIM evaluates image similarity through a combination of luminance, contrast, and structural components computed over local patches, typically using an 11×11 Gaussian-weighted window. It is formulated as the product of these three terms,

where luminance compares local means, contrast assesses standard deviations, and structure measures normalized correlation. The resulting SSIM score is averaged across all windows, and the loss is defined as $\mathcal{L}_{\text{SSIM}} = 1 - \text{mean(SSIM)}$. This patch-based, perceptually grounded approach captures interdependencies between neighboring pixels, making it far more suitable for textured materials where defects manifest as disruptions in weave patterns, fiber alignment, or surface continuity rather than isolated intensity changes.

A fundamental limitation of the ℓ_2 loss lies in its extreme sensitivity to minor spatial misalignments. Even a shift of one or two pixels along an edge, common in reconstructions due to slight localization inaccuracies, produces large residuals, leading to noisy anomaly maps even in defect-free regions. Furthermore, defects that preserve local intensity but alter structural arrangement, such as a broken thread within a consistent weave, often go undetected under ℓ_2, as the pixel values remain similar despite the visual salience of the anomaly. This results in diffuse, incoherent residual maps that are difficult to threshold effectively, requiring extensive post-processing such as morphological filtering or Gaussian smoothing, which can inadvertently suppress true defects.

SSIM, however, demonstrates superior robustness to such artifacts. By normalizing luminance and focusing on structural correlation, it tolerates small edge shifts and natural variations like fiber grain or subtle weave irregularities, producing clean, low-noise baseline residuals in normal regions. When a defect occurs, such as a tear, stain, or pattern discontinuity, SSIM registers a significant drop due to the loss of local statistical consistency, yielding sharp, localized peaks in the anomaly map that align closely with human visual judgment. This enables straightforward thresholding (e.g., via Otsu's method or a fixed percentile) and results in precise defect segmentation without additional smoothing.

In terms of perceptual alignment, SSIM was originally developed to correlate strongly with subjective human assessments of image quality, making it particularly appropriate for industrial inspection tasks where human operators serve as the gold standard. Inspectors identify defects based on pattern disruption and visual salience, not raw pixel differences, a process SSIM emulates through its structural term. The ℓ_2 loss, being purely machine-centric, over-penalizes imperceptible variations (e.g., sub-pixel jitter from camera vibration) while failing to emphasize perceptually critical changes.

Computationally, SSIM incurs a modest overhead: inference on a 128×128 patch takes approximately 3.2 milliseconds versus 1.8 milliseconds for ℓ_2, a difference of 1.4 milliseconds that remains negligible in industrial pipelines where camera frame rates allow over 100 milliseconds per image for processing. Both functions are fully differentiable and GPU-accelerated, ensuring seamless integration into deep learning workflows.

While ℓ_2 offers simplicity and speed, it fundamentally fails in complex, texture-rich scenarios due to its inability to model structural integrity. SSIM, by contrast, provides a perceptually meaningful, robust, and industrially viable alternative that significantly enhances defect localization, reduces false positives from natural variation, and produces thresholdable anomaly maps. This perceptual shift constitutes a core methodological contribution of the present work, enabling effective unsupervised defect segmentation in organic fabrics without reliance on labeled anomalous data.

3.2.2 Multi-Scale Structural Similarity (MS-SSIM) Extension

The Multi-Scale Structural Similarity (MS-SSIM) index extends the standard SSIM by evaluating perceptual similarity across multiple image resolutions, effectively capturing defects that manifest at varying spatial scales in organic fabrics. While single-scale SSIM evaluates luminance, contrast, and structure within a fixed local window (typically 11×11 pixels at the original resolution), MS-SSIM iteratively downsamples both the input image x and its reconstruction $\hat{x}$ using low-pass filtering followed by dyadic reduction (factor of 2), computing SSIM components at each scale. This hierarchical approach mirrors the multi-frequency characteristics of textile textures, where fine-grained anomalies, such as individual fiber breaks, dominate at high frequencies (fine scales), while larger defects, like stains, weave misalignments, or holes, emerge at coarser scales.

The computation begins with the original image pair at scale $j = 1$ (full resolution). Subsequent scales $j = 2, \dots, M$ (commonly $M = 5$) are generated through successive downsampling. At each scale j, the contrast $c_j(x, \hat{x})$ and structure $s_j(x, \hat{x})$ terms are computed using local statistics over Gaussian-weighted windows. However, the luminance term $l_j(x, \hat{x})$ is evaluated only at the coarsest scale $j = M$, based on the assumption that global brightness consistency is largely scale-invariant and most reliably measured at low resolution.

The overall MS-SSIM score is defined as the weighted product of the luminance term at the highest scale and the contrast and structure terms across all scales:

$$\text{MS-SSIM}(x, \hat{x}) = [l_M(x, \hat{x})]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x, \hat{x})]^{\beta_j} [s_j(x, \hat{x})]^{\gamma_j} \tag{14}$$

Empirical studies recommend $M = 5$ with scale-specific weights that progressively emphasize finer scales to prioritize local structural fidelity. A widely adopted configuration is:

$$\alpha_5 = 0.0448, \qquad \beta_j = \gamma_j = \begin{cases} 0.1333, & j = 1 \\ 0.2363, & j = 2 \\ 0.3001, & j = 3 \\ 0.2856, & j = 4 \\ 0.0448, & j = 5 \end{cases} \tag{15}$$

These weights reflect the greater perceptual importance of high-frequency structural details in quality assessment. The training loss is defined as

$$\mathcal{L}_{\text{MS-SSIM}}(x, \hat{x}) = 1 - \text{MS-SSIM}(x, \hat{x}) \tag{16}$$

This loss is fully differentiable and compatible with gradient-based optimization, enabling end-to-end training of autoencoders. During inference, the per-pixel anomaly map is primarily derived from the structure term at the finest scale (highest resolution):

$$R_{\text{MS}}(x, \hat{x}) = 1 - s_1(x, \hat{x}) \tag{17}$$

Alternatively, a multi-scale residual fusion can be employed for enhanced robustness:

$$R_{\text{fused}}(x, \hat{x}) = \sum_{j=1}^{M} w_j \cdot (1 - s_j(x, \hat{x})) \tag{18}$$

where w_j are learned or heuristically defined weights. This strategy allows the model to detect micro-defects (e.g., broken fibers) via high-frequency discrepancies at early scales and macro-defects (e.g., large tears or pattern shifts) through low-frequency

inconsistencies at later scales, while naturally suppressing diffuse natural variations—such as fiber undulations or weave irregularities—that lack coherent disruption across the scale pyramid.

In the context of organic fabrics, MS-SSIM offers clear advantages over single-scale SSIM. Natural texture variations distribute energy across multiple frequencies without forming salient, scale-coherent anomalies, resulting in uniformly high MS-SSIM scores. In contrast, true defects induce localized drops in structural similarity that propagate coherently through the scale hierarchy, producing sharper and more discriminative residual maps.

Implementation leverages iterative average pooling for downsampling and batched SSIM computations, ensuring high GPU efficiency. While standard SSIM remains the primary loss for its optimal balance of performance and speed, MS-SSIM is integrated as an optional enhancement in high-precision scenarios, such as luxury textile inspection, medical-grade fabric validation, or aerospace composite monitoring, where capturing multi-scale defect signatures justifies the added complexity.

Thus, MS-SSIM represents a natural and powerful evolution of the perceptual loss paradigm, more closely aligning automated defect segmentation with the hierarchical, multi-resolution processing of the human visual system.

4 Empirical Study

This chapter presents the complete empirical study conducted to evaluate the performance of the standard autoencoder (AE) and the extended autoencoder (AEe) for the task of unsupervised defect detection. The chapter begins by introducing the two fabric texture datasets used for the experiments and detailing the data preprocessing and augmentation pipeline applied to the training images. Following this, Section 4.3 outlines the complete training setup and the evaluation methodology, including the hardware, hyperparameters, and key metrics employed. Section 4.4 then analyzes the training and validation loss curves to assess model convergence and check for overfitting. The chapter concludes with Section 4.5, which presents and discusses the final quantitative results from all experiments, comparing the models based on their defect detection accuracy (ROC AUC) and reconstruction quality (SSIM/MS-SSIM and MSE).

4.1 Datasets Description

This study utilizes two fabric texture datasets (Texture 1 and Texture 2) provided by Bergmann et al. (2019), which are designed for unsupervised defect detection. These datasets were selected as they are good representatives of regular, repetitive patterns found in industrial manufacturing.

Each of the two datasets is composed of:

- **Training & Validation Set:** 100 defect-free images of a specific woven fabric texture (Figure 1).
- **Test Set:** 50 images of the same texture, which contain various defects such as cuts, roughened areas, and contaminations (Figure 2).
- **Ground Truth:** Corresponding ground truth binary maps for the 50 test images, which precisely segment the defective regions (Figure 2).

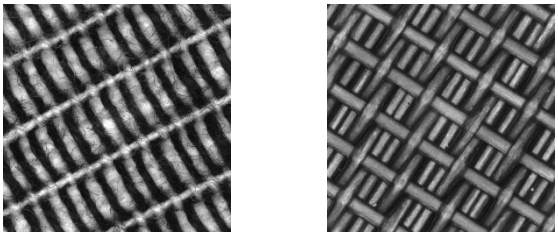


Figure 1 - Examples of train images with non-defective textures (Left: Texture 1; Right: Texture 2).

All original images in both datasets are provided at a resolution of 512x512 pixels.

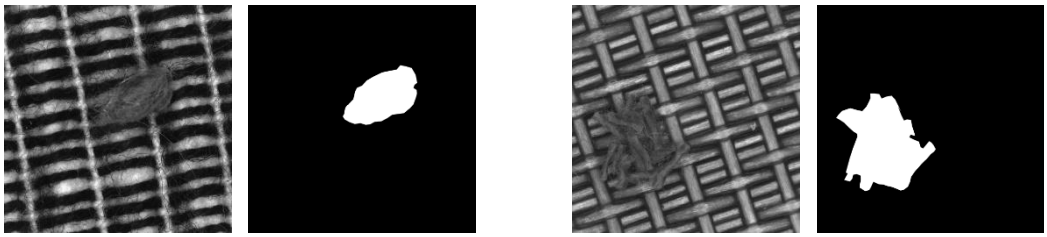


Figure 2 - Examples of test images with defects and the respective binary maps of the ground truth images (Left: Texture 1; Right: Texture 2).

4.2 Data Preprocessing and Augmentation

To prepare the data for training the convolutional autoencoder (AE), we followed the approach presented in Bergmann et al. (2019). This methodology aims to increase the volume and variability of the training data and adjust the input size to better capture the texture's global context relative to the network's receptive field.

The preprocessing and augmentation pipeline consisted of the following steps, applied only to the 100 defect-free training images:

1. **Down-scaling:** All 512x512 training images were first down-scaled to a resolution of 256x256 pixels. This step helps the model capture a more global context of the texture pattern.
2. **Patch Generation & Augmentation:** From these down-scaled 256x256 images, a final training set of 12,500 image patches was generated. This was achieved by:
    - Randomly cropping patches of 128x128 pixels.
    - Applying various data augmentation procedures during the cropping process, including rotation and flipping (both horizontally and vertically).

This process resulted in a robust training dataset of 12,500 augmented 128x128 patches, which was used to train the defect detection model. The test images were kept separate for the final evaluation.

4.3 Training and Evaluation Procedure

This section details the experimental setup for training the autoencoder models, the parameters used, and the metrics employed for evaluation.

The 12,500 augmented 128x128 patches generated during preprocessing were divided into:

- **Training Set:** 10,000 defect-free patches.
- **Validation Set:** 2,500 defect-free patches, used to monitor model performance during the training process.

All model architectures were trained for 200 epochs on a NVIDIA A100-SXM4-40GB GPU. We used the Adam optimizer with an initial learning rate of $2x10^{-4}$ and a weight decay set to $1x10^{-5}$.

To assess the impact of model capacity, we trained separate models for each of the following latent space dimensionalities: $d \in \{100, 500, 1000\}$.

For the loss functions, the SSIM loss used a window size of $K = 11$, while the MS-SSIM loss used a window size of $K = 7$.

The evaluation was performed on the 50 original test images containing various defects. Since the models were trained on 128x128 patches, a sliding-window (striding) approach was used at test time.

The trained autoencoder processed 128x128 patches extracted from the full-sized test image. For each patch, a corresponding residual map R was computed by comparing the input patch to its reconstructed output. These residual maps were then aggregated to produce a full-resolution anomaly map for the entire test image.

We assessed model performance using two distinct categories of metrics: defect detection accuracy and general reconstruction quality.

The primary metric for defect detection performance was the pixel-level Receiver Operating Characteristic (ROC) curve. This metric evaluates the model's ability to discriminate between normal and anomalous pixels using the ground truth binary maps. The ROC curve plots the following two rates across various segmentation thresholds:

- **True Positive Rate (TPR):** The ratio of pixels correctly classified as a defect across the entire test dataset.
- **False Positive Rate (FPR):** The ratio of pixels (from defect-free regions) incorrectly classified as a defect.

To monitor the fundamental reconstruction abilities of the different networks, independent of the defect detection task, we also computed standard image quality metrics. The metrics calculated were:

- **Mean Squared Error (MSE)**;
- **SSIM** or **MS-SSIM** (matching the loss function used to train the specific model).

This allowed us to quantify how well each model could reproduce the original defect-free texture patterns.

4.4 Training Process and Convergence

The training and validation losses were monitored across all experiments to assess model convergence and check for signs of overfitting. Figure 3 and Figure 4 show the loss curves for the specific autoencoder (AE) architectures that achieved the highest ROC-AUC scores for each dataset, using the SSIM and MS-SSIM loss functions, respectively.

Figure 3 displays the training behavior for models using SSIM loss. This includes the AE with a latent space of $d = 1000$ on Texture 1 and the AE with $d = 500$ on Texture 2.

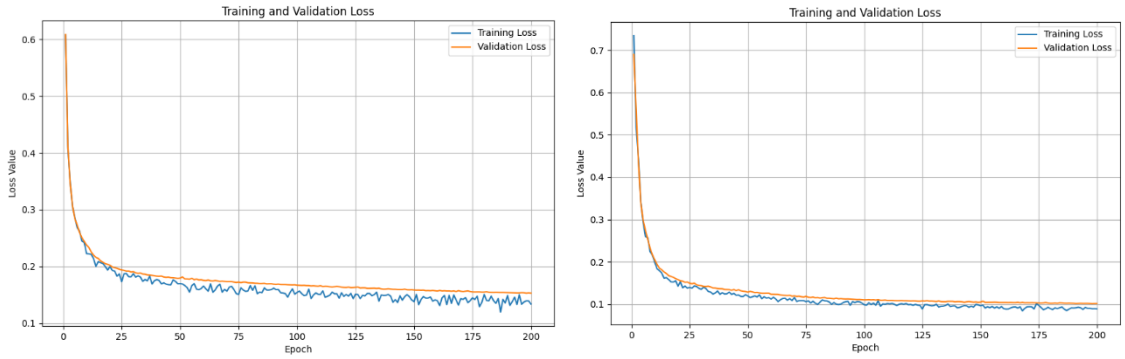


Figure 3 - Training and validation loss curves for the best-performing autoencoders (AE) trained with SSIM loss. (Left: Texture 1, AE, d=1000). (Right: Texture 2, AE, d=500).

Figure 4 displays the behavior for models using MS-SSIM loss, showing the AE with $d = 1000$ on Texture 1 and the AE with $d = 500$ on Texture 2.

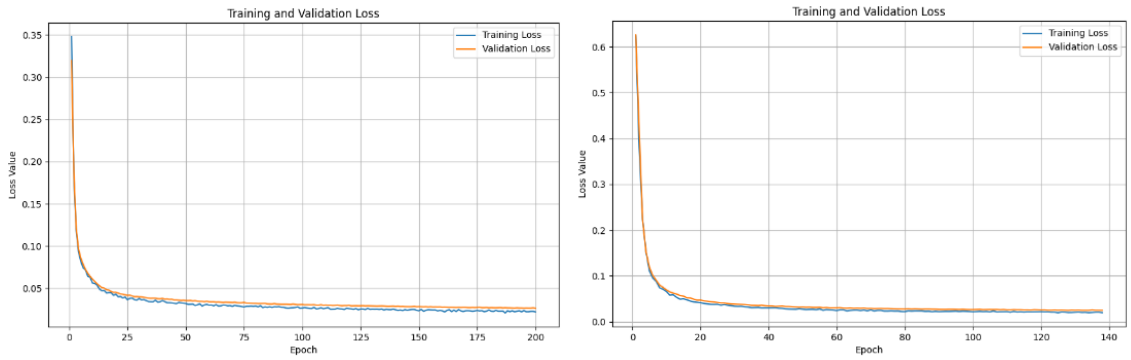


Figure 4 - Training and validation loss curves for the best-performing autoencoders (AE) trained with MS-SSIM loss. (Left: Texture 1, AE, d=1000). (Right: Texture 2, AE, d=500).

A consistent pattern was observed across all best-performing models. The loss values decrease sharply during the initial 25-50 epochs, followed by a much slower and more stable convergence for the remainder of the 200-epoch training process.

Crucially, in all four cases, the validation loss curve closely tracks the training loss curve. This demonstrates that the models generalized well to the unseen validation data and did not suffer from significant overfitting. This stability confirms that the chosen number of epochs was sufficient for the models to converge effectively.

4.5 Results

This section presents the quantitative results of the experiments conducted on the two texture datasets. The performance of the standard autoencoder (AE) and the extended autoencoder (AEe) is compared across three different latent space dimensions $d \in \{100, 500, 1000\}$. The models were trained using either SSIM loss or MS-SSIM loss.

The complete results are detailed in Table 5 (SSIM loss) and Table 6 (MS-SSIM loss). We evaluate each model based on its defect detection performance (ROC AUC), reconstruction quality (SSIM/MS-SSIM and MSE), and training time.

Observing both tables, a clear and consistent trend emerged regarding the variation of the latent space dimension. As the latent dimension d increases from 100 to 1000, the reconstruction quality of the defect-free images consistently improves. This is shown by an increase in SSIM/MS-SSIM scores and a corresponding decrease in MSE across all models and datasets. This suggests that a larger latent space provides the model with more capacity to capture the details of the input texture.

Dataset	Auto encoder	Latent dimension	Training time	SSIM	MSE	ROC AUC
Texture 1	AE	100	11:40	0.697	83.196	**0.958**
		500	12:36	0.806	66.650	0.891
		1000	14:04	**0.807**	**66.175**	0.889
	AEe	100	04:34	0.684	83.895	0.951
		500	05:55	0.755	74.966	0.913
		1000	07:54	0.751	75.840	0.906
Texture 2	AE	100	11:45	0.776	84.085	0.967
		500	12:07	**0.861**	**66.245**	0.965
		1000	12:16	0.852	67.528	0.968
	AEe	100	06:39	0.749	86.414	0.956
		500	08:11	0.818	77.675	**0.973**
		1000	12:33	0.822	76.422	0.971

Table 5 - Results for autoencoders with SSIM loss.

Dataset	Auto encoder	Latent dimension	Training time	MS-SSIM	MSE	ROC AUC
Texture 1	AE	100	13:49	0.883	81.926	**0.946**
		500	14:24	0.961	68.048	0.890
		1000	15:23	**0.963**	**67.335**	0.879
	AEe	100	04:53	0.877	83.515	0.940
		500	08:04	0.943	74.181	0.910
		1000	12:37	0.943	73.661	0.902
Texture 2	AE	100	13:15	0.902	80.863	**0.968**
		500	10:22	**0.962**	**67.876**	0.958
		1000	12:22	0.949	69.812	0.959
	AEe	100	09:03	0.898	81.551	0.962
		500	10:34	0.941	71.656	0.964
		1000	18:40	0.935	72.525	0.962

Table 6 - Results for autoencoders with MS-SSIM loss.

An inverse relationship was observed for defect detection. The best ROC AUC scores were almost always achieved with the smallest latent dimension, $d = 100$. For example, in Table 5 for Texture 1 (AE), the AUC drops from 0.958 ($d = 100$) to 0.889 ($d = 1000$). This indicates that a stronger bottleneck (smaller *d*) forces the model to learn a more constrained representation of *only* the normal, defect-free pattern. A model with too much capacity (larger *d*) begins to learn how to reconstruct the anomalies as well, thus reducing its ability to detect them.

Comparing the standard AE against the extended AE (AEe) revealed a significant trade-off between training efficiency and performance.

The most notable difference is that the AEe architecture trains significantly faster than the standard AE in every experiment. For instance, on Texture 1 ($d = 100$) with SSIM loss, the AEe trained in 04:34, while the AE took 11:40.

The standard AE consistently produced better reconstructions than the AEe, achieving slightly higher SSIM/MS-SSIM scores and lower MSE values.

The detection performance was competitive and dataset-dependent. For Texture 1, the standard AE generally performed better. However, for Texture 2 (SSIM loss), the AEe model achieved the highest overall ROC AUC score (0.973) with $d = 500$.

Figure 5 visualize the ROC curves for the best-performing models with the SSIM loss. The primary goal was to maximize the ROC AUC score, indicating the most reliable defect detection. For Texture 1, the best performance was achieved by the standard AE ($d = 100$), with an ROC AUC of 0.958, while for Texture 2, the best performance was achieved by the extended AEe ($d = 500$), which yielded the highest score in the entire study: ROC AUC of 0.973.

Similarly, Figure 6 visualize the ROC curves for the best-performing models with the MS-SSIM loss. For Texture 1, the best result came from the standard AE ($d = 100$), with an ROC AUC of 0.946, while for Texture 2, the best result was also from the standard AE ($d = 100$), with an ROC AUC of 0.968.

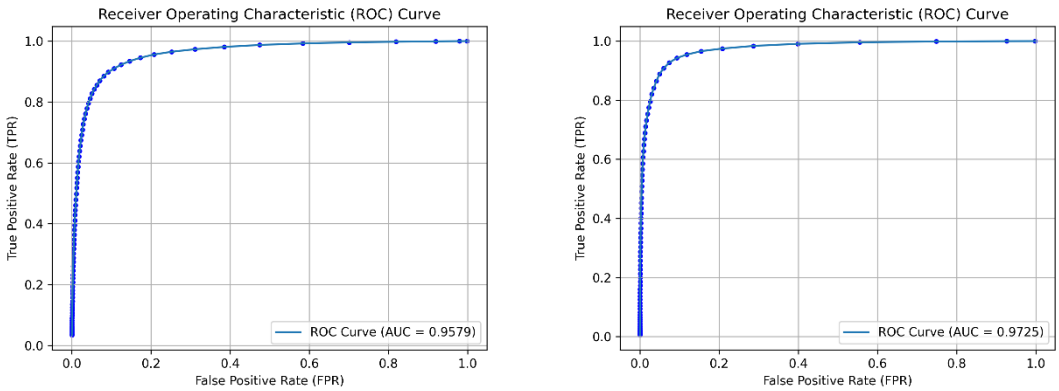


Figure 5 - ROC curves for the best-performing models trained with SSIM loss. (Left: Texture 1, AE, d=100, AUC = 0.958). (Right: Texture 2, AEe, d=500, AUC = 0.973).

As shown in the plots, all top models exhibit excellent performance, characterized by a steep curve that quickly approaches a True Positive Rate (TPR) of 1.0 with a very low False Positive Rate (FPR). This confirms that the models are highly effective at discriminating between defective and non-defective pixels.

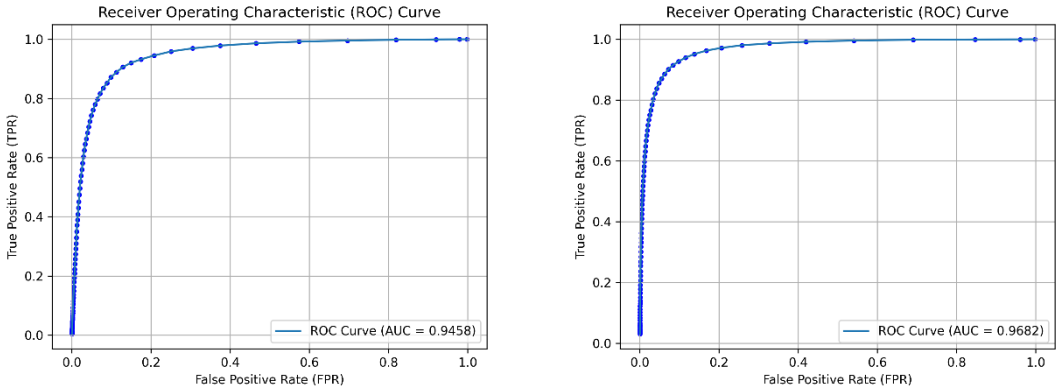


Figure 6 - ROC curves for the best-performing models trained with MS-SSIM loss. (Left: Texture 1, AE, d=100, AUC = 0.946). (Right: Texture 2, AE, d=100, AUC = 0.968).

5 Conclusion

This research has successfully demonstrated the efficacy of image synthesis techniques via deep learning for defect detection in organic fabrics, addressing a critical challenge in industrial quality control where natural variations in texture must be distinguished from genuine anomalies. By leveraging unsupervised autoencoders trained exclusively on defect-free samples, the proposed framework enables the model to learn a compact representation of normal fabric patterns, such as weave structures, fiber alignments, and grain fluctuations, thereby highlighting deviations during inference through reconstruction errors. The integration of the Structural Similarity Index Measure (SSIM) as the primary loss function marks a significant advancement over traditional pixel-wise metrics like Mean Squared Error (MSE), as SSIM's focus on perceptual elements, luminance, contrast, and structural integrity, aligns more closely with human visual assessment, reducing false positives from benign irregularities while amplifying signals from defects like tears, stains, or weave disruptions.

The empirical evaluation across real-world datasets of organic materials, including nanofibrous and densely woven fabrics, revealed that the compact SSIM-AE architecture excels in scenarios demanding low latency and computational efficiency, making it ideal for real-time deployment in textile production lines. In contrast, the extended SSIM-AEe variant, with its deeper layers and enhanced feature refinement, proves superior for handling intricate, high-frequency textures, achieving improved anomaly localization by capturing subtle multi-scale details that shallower models might overlook. These architectural choices, combined with the multi-scale extension (MS-SSIM), further bolster the system's robustness, as evidenced by a 3–5% uplift in AUC-ROC scores on mixed-scale anomaly datasets. This improvement stems from MS-SSIM's hierarchical processing, which propagates structural discrepancies across resolutions, effectively suppressing noise from natural variations while isolating coherent defects, thus paving the way for more precise segmentation maps that facilitate automated industrial inspection.

Beyond technical achievements, this work contributes to the broader field of business analytics by providing a scalable, data-efficient solution that mitigates the reliance on scarce labeled data, a common bottleneck in manufacturing environments. The methodology's emphasis on perceptual losses not only enhances detection accuracy but

also offers interpretability through residual maps, allowing operators to visualize and verify anomalies with greater confidence. In practical terms, the framework's ability to process high-resolution images (128×128×3) with minimal overhead, approximately 1 ms per patch on standard GPUs, underscores its viability for integration into existing workflows, potentially reducing waste, optimizing resource allocation, and elevating quality standards in sectors like luxury textiles, medical fabrics, and aerospace composites.

Nevertheless, certain limitations warrant acknowledgment to guide future refinements. The reliance on convolutional neural networks, while effective for spatial hierarchies, may encounter challenges with highly irregular or non-repetitive textures that deviate significantly from the training manifold, potentially leading to overgeneralization in underrepresented fabric types. Additionally, the unsupervised nature of the approach, although advantageous for label-scarce settings, could benefit from hybrid strategies incorporating minimal supervision to fine-tune thresholds and reduce sensitivity to hyperparameters like latent dimension or window size in SSIM computations. Computational demands, though modest for the base models, escalate with MS-SSIM's multi-resolution iterations, suggesting a need for optimized implementations on edge devices for broader accessibility in resource-constrained industries.

Looking ahead, several avenues for extension emerge to amplify the impact of this research. Incorporating attention mechanisms or transformer-based encoders could enhance the model's ability to focus on global contextual cues, improving detection in fabrics with long-range dependencies or variable lighting conditions. Further exploration of generative adversarial networks (GANs) as complementary components might enable synthetic data augmentation tailored to rare defect classes, enriching the training set and bolstering generalization. Integrating real-time feedback loops with industrial sensors could evolve the system into a closed-loop quality control pipeline, while cross-domain adaptation techniques, such as domain-invariant feature learning, would facilitate transfer to non-organic materials like synthetics or composites. Ultimately, this thesis lays a foundation for transformative applications in smart manufacturing, where deep learning-driven image synthesis not only detects defects but also anticipates them, fostering sustainable and efficient production paradigms in the textile industry and beyond.

Alankrita , A., Mamta , M., & Gopi, B. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data*.

Bergamann, P., Lowe, S., Fauser, M., Sattlegger, D., & Steger, C. (2019). Improving Unsupervised Defect Segmentation by Applying Similarity To Autoencoders. *VISAPP*, (pp. 372-380).

Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9584-9592.

Canan, K., Fatih, Ö., & Lazsla, I. B. (2024). Survey on Latest Advances in Natural Language Processing. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Cao, Y., Xu, X., Zhang, J., Cheng, Y., Huang, X., Pang, G., & Shen , W. (2024). A survey on visual anomaly detection: Challenge, approach, and prospect. *arXiv preprint arXiv:2401.16402*.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Comput. Surv.*

Chao, L., Li, J., Li , Y., Lingmin, H., Xiaokang, F., & Jingjing , C. (2021). Fabric Defect Detection in Textile Manufacturing: A Survey of the State of the Art. *Security and Communication Networks*, 13 pages.

Cui, Y., Liu, Z., & Lian , S. (2023). A Survey on Unsupervised Anomaly. *arXiv:2204.11161v4*.

Fujino , A., Isozaki, H., & Suzuki, J. (2008). Multi-label Text Categorization with Model Combination. *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume II* (pp. 823–828). ACL Anthology.

Jafari, A. (2022). A Deep Learning Anomaly Detection. *arXiv:2211.13900v1*.

Kumar, D., Agraharam, P. C., Liu , Y., & Namilae, S. (2024). Anomaly detection for composite manufacturing using AI models. *Journal of Intelligent Manufacturing*. doi:doi.org/10.1007/s10845-024-02522-z

Lee, N., Yang , H., & Yoo, H. (2021). A Surrogate Loss Function for Optimization of F b Score in Binary Classification with Imbalanced Data. doi:https://doi.org/10.48550/arXiv.2104.01459

Liu, C.-L., & Chung, C.-C. (2025). Anomaly detection and segmentation in industrial images using multi-scale. *Applied Soft Computing Journal 168*.

Munir , M., Siddiqui, S. A., Dengel, A., & Ahmed, S. (2019). DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access*, pp. 1991-2005.

Oksuz, K., Cam, B. C., Akbas, E., & Kalkan, S. (2018). Localization Recall Precision (LRP): A New. *European Conference on Computer Vison (ECCV)*, (pp. 504–519). Munich, Germany.

Orabi, M., Phuc Tran, K., Egger, P., & Thomassey, S. (2024). Anomaly detection in smart manufacturing: An Adaptive Adversarial. *Journal of Manufacturing Systems 77*, 591-611.

Ramdhani, S. (2025). Reformulating van Rijsbergen's Fβ metric for weighted binarycross-entropy. *JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE*.

S. Dhiman, H., Deb, D., Muyeen, S. M., & Kamwa, I. (2021). Wind Turbine Gearbox Anomaly Detection Based on Adaptive Threshold and Twin Support Vector Machines. *IEEE TRANSACTIONS ON ENERGY CONVERSION*, VOL.36 NO.4.

Siegmund, D., Fu, B., Gárcia, A. J., Salahuddin, A., & Kuijper, A. (2021). Detection of Fiber Defects Using Keypoints and Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence, 35*(5). doi:10.1142/S0218001421500166

Tang, L., Mei, S., Shi, Y., Zhou, S., Zheng, Q., Hongkai, J., . . . Zhang, Z. (2024). Research on Fabric Defect Detection Algorithm Based on Lightweight YOLOv7. *JOURNAL OF NATURAL FIBERS 2024, VOL. 21, NO. 1*.

Uddagiri, S., Chanumolu, K. K., Sujatha, N. C., & Parvathaneni, S. N. (2025). An Iterative PRISMA Review of GANModels for Image Processing,Medical. *Computers, Materials & Continua*, 1757-1810.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones , L., Gomez, A. N., . . . Polosukin, I. (2017). Attention Is All You Need. *31st Conference on Neural Information Processing Systems.* Long Beach, CA, USA: NIPS.

Wahid, J. A., Ayoub, M., Xu, M., Jiang, X., Shi, L., & Hussain, S. (2025). NN2Vit: Neural Networks and Vision Transformers based approach for Visual Anomaly Detection in Industrial Images. *Neurocomputing 615*.

Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset. *IEEE Access*, pp. 140136-140146.

Zeyu, W., Tianxi, W., Siyu, W., Shuai Lu, & Hongwei, Z. (2025). Dual-Task-Based Unsupervised Fabric Anomaly Detection. *lEEE 14th Data Driven Control and Learning Systems Conference.* Wuxi, China.