# Deep Learning Models for Defect and Anomaly Detection on Industrial Surfaces

by

Alireza Saberironaghi

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

**Master of Applied Science in Electrical and Computer Engineering**

Department of Electrical, Computer, and Software Engineering (ECSE)

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

December 2023

**Thesis Examination Information**

Submitted by: **Alireza Saberironaghi**

**Master of Applied Science** in **Electrical and Computer Engineering**

| Thesis title: |
| --- |
| Deep Learning Models for Defect and Anomaly Detection on Industrial Surfaces |

An oral defense of this thesis took place on November 30, 2023 in front of the following examining committee:

**Examining Committee:**

| | |
| --- | --- |
| Chair of Examining Committee | Dr. Akramul Azim |
| Research Supervisor | Dr. Jing Ren |
| Examining Committee Member | Dr. Hossam Gaber |
| Thesis Examiner | Dr. Masoud Makrehchi |

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

**Abstract**

Automated quality control is essential across various industries to reduce manual inspection and improve operational efficiency. While there are advances in computer vision and machine learning for defect detection, challenges persist, such as defect variability and the computational burden. This thesis presents specialized deep learning architectures addressing defect classification, segmentation, and detection in textiles, civil engineering, and manufacturing. For textiles, a novel system merges capsule networks with convolutional neural networks and a spatial attention module, achieving a 99.42% accuracy on the TILDA dataset. In civil engineering, the DepthCrackNet model, optimized for pavement crack detection, attains mIoU scores of 77.0% and 83.9% on the Crack500 and DeepCrack datasets. In manufacturing, the E-UNet3+ model for steel defect detection showcases a mIoU score of 86.19% on the SD-saliency-900 dataset. The research's core contribution lies in pioneering deep learning architectures that precisely detect defects across sectors.

**Keywords:** Defect detection; defect segmentation; defect classification; anomaly detection; deep learning

**Author's Declaration**

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

_____ Alireza Saberironaghi

## Statement of Contributions

The results of this thesis have been published in the following papers:

- A. Saberironaghi and J. Ren, "E-Unet3+: Enhanced UNet3+ model with Multiscale Feature Learning and Attention Mechanisms for Steel Surface Defect Detection," (Submitted to Applied Soft Computing Journal, 22 pages)

- A. Saberironaghi and J. Ren, "DepthCrackNet: Integrating 3D Spatial Features and Multi-Head Attention Mechanism for Automatic Pavement Crack Detection.," (Submitted to IEEE Canadian Journal of Electrical and Computer Engineering, 22 pages)

- A. Saberironaghi and J. Ren, "A Capsule-based Neural Network for Texture Defect Classification," (Submitted to International Journal of Machine Learning and Cybernetics, 19 pages)

- A. Saberironaghi, J. Ren, and M. El-Gindy, "Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review," Algorithms, vol. 16, no. 2, p. 95, Feb. 2023, doi: 10.3390/a16020095.

- J. Ren, H. A. Gabbar, X. Huang, and A. Saberironaghi, "Defect Detection for Printed Circuit Board Assembly Using Deep Learning," in 2022 8th International Conference on Control Science and Systems Engineering (ICCSSE), Jul. 2022, pp. 85–89. doi: 10.1109/ICCSSE55346.2022.10079777.

**Acknowledgments**

First and foremost, I extend my deepest gratitude to God for granting me the strength and resilience needed to complete my research. Additionally, heartfelt thanks are in order for my supervisor, Prof. Jing Ren. Not only did she provide an invaluable opportunity to conduct my research, but she also guided me meticulously through each phase. Her mentorship enabled me to engage in comprehensive research and explore numerous new aspects of the subject. I am deeply grateful for her constant support and professional advice.

I would also like to convey my genuine appreciation to the members of my examination committee. Your willingness to critically read my thesis and provide insightful feedback and comments has been instrumental in elevating the quality of my work. Your perspectives have proven invaluable in refining my scholarship to a higher standard. Additionally, I must extend a heartfelt thank you to Dr. Zaferanieh. Your guidance and insights throughout my academic journey have been profoundly influential and deeply valued. Last but certainly not least, my family - specifically my parents and my brother Mohammadreza - deserves a special mention for their unconditional support throughout my academic journey. The sacrifices they've made have been indispensable in enabling me to pursue higher education and reach my aspirations. This thesis is lovingly dedicated to them. They have consistently provided love and guidance through every challenge I've faced. In conclusion, I want to thank everyone for being a part of this amazing journey. Your contributions have greatly influenced this project and, consequently, my academic path.

**Table of Contents**

## List of Tables

### Chapter 2



### Chapter 3



### Chapter 4

**Chapter 5**

## List of Figures

### Chapter 1



### Chapter 3

**Chapter 4**



**Chapter 5**

**List of Abbreviations and Symbols**

| | |
|---|---|
| $\otimes$ | Element-wise multiplication |
| $\sigma$ | Sigmoid activation function |
| $F$ | Original feature map |
| $F'$ | Feature map after channel attention |
| $F''$ | Feature map after spatial attention |
| $Mc$ | Channel attention mechanism |
| $Ms$ | Spatial attention mechanism |
| $FCavg$ | Squeezed feature map after average pooling |
| $FCmax$ | Squeezed feature map after max pooling |
| $Avgpool$ | Average pooling operation |
| $Maxpool$ | Max pooling operation |
| $f7\times7$ | Convolution operation with a 7x7 filter |
| $H$ | Height of the feature map |
| $W$ | Width of the feature map |
| $FSavg$ | Feature map obtained after average pooling |
| $FSmax$ | Feature map obtained after max pooling |
| IOT | Internet of Things |
| 3D | Three-dimensional |
| 2D | Two-dimensional |
| U.S. | United States |
| TIFF | Tagged Image File Format |
| DFG’s | Deutsche Forschungsgemeinschaft's |
| GPU | Graphics Processing Unit |
| RAM | Random Access Memory |
| GLCM | Gray Level Co-occurrence Matrix |

| | |
|---|---|
| LBP | Local Binary Patterns |
| GMRF | Gaussian Markov Random Field |
| MRF | Markov Random Field |
| HOG | Histogram of Oriented Gradients |
| CRF | Conditional Random Field |
| ESP | Edge-preserving Smoothing Process |
| ANN | Artificial Neural Networks |
| SVM | Support Vector Machines |
| GANs | Generative Adversarial Networks |
| CNN | Convolutional Neural Network |
| KNN | K-Nearest Neighbors |
| NN | Nearest Neighbor |
| CapsNet | Capsule Network |
| CBAM | Convolutional Block Attention Module |
| SE | Squeeze-and-Excitation |
| MLP | Multi-Layer Perceptron |
| MSFLM | Multiscale Feature Learning Module |
| IDS | Improved Down-Sampling Module |
| TP | True Positive |
| FP | False Positive |
| FN | False Negative |
| TN | True Negative |
| mIoU | Mean Intersection over Union |
| DCE | Double Convolutional Encoder |
| TMSA | TriInput Multi-Head Spatial Attention |
| SDE | Spatial Depth Enhancer |
| BN | Batch Normalization |
| CTD | Convolution Transpose Decoder |

| | |
|---|---|
| Conv2D | 2D Convolutional Layer |
| MPS | Minimal Path Selection |
| FFA | Free-Form Anisotropy |

# Chapter 1. Introduction

## 1.1 Overview

In the era of Industry 4.0, advanced technologies like the Internet of Things (IoT), robotics, and machine learning are revolutionizing the landscape of manufacturing processes. These advancements significantly enhancing productivity, efficiency, and quality in various industrial settings. Machine learning and computer vision technologies are essential in manufacturing for tasks like object tracking [1], object recognition [2], visual servoing [3], pattern recognition [4], and defect detection [5], as depicted in Figure 1.1. Deep learning techniques enhance the efficacy and accuracy of these applications [6].


Figure 1.1: Industrial computer vision applications.

The arrival of Industry 4.0 marks a new era where automation is essential element in manufacturing technologies. This revolution is not only transforming traditional manufacturing processes but also emphasizing the role of quality assurance in the entire

production pipeline. Defect detection and classification have thus become critical aspects that ensure quality control in industrial manufacturing processes [7]. Automated systems that can promptly and accurately identify and categorize defects in materials or products can significantly contribute to cost-efficiency and product reliability. The significance of this cannot be overstated in various industries such as automotive manufacturing, aerospace, and consumer electronics, where even minor defects can have critical safety or financial implications.

Traditionally, defect detection has been performed through visual inspection by human operators [8]. However, this approach has numerous limitations, including inefficiency and inconsistency. With advancements in computer vision and deep learning technologies, there has been a significant shift toward automated defect detection systems. These systems leverage the power of algorithms to scan, identify, and categorize defects in materials or products with high accuracy and efficiency [9].

## 1.2 Problem Statement and Research Questions

In the realm of manufacturing and infrastructure maintenance, the automatic detection and classification of defects play a crucial role in ensuring quality and safety. However, this task is filled with challenges, mainly because of the complexity and differences in the defects that must be detected.

Firstly, in the textile industry, the quality control of fabrics is a significant concern. The diversity in textures, coupled with the irregular shapes, varying sizes, and complex backgrounds of fabric defects, poses a substantial challenge for automated systems.

Traditional approaches often fall short in accurately identifying these defects due to their limited capability in handling complex, multiscale features inherent in fabric images. Similarly, in the field of civil engineering, specifically in road maintenance, the automatic detection of pavement cracks is vital for road safety. The variability of cracks, differences in pavement materials, and the presence of various anomalies on the pavement surface add layers of complexity to the automated detection process. Existing systems struggle with effectively capturing and processing the 3D spatial relationships and contextual information necessary for accurate crack detection. Moreover, in steel manufacturing, the detection of surface defects is critical for maintaining product quality. Automated defect detection in this domain faces hurdles due to the varied types of defects, their subtle contrasts against complex backgrounds, and diverse sizes. The conventional defect detection methods often fail to achieve the high precision required due to limitations in feature extraction and the inability to focus on relevant features amidst diverse backgrounds. Addressing these challenges, this thesis proposes three novel deep learning models tailored for defect classification and segmentation in different contexts: textile fabrics, pavement surfaces, and steel surfaces. Each model introduces solutions to overcome the specific challenges encountered in its respective application domain. By leveraging advanced neural network architectures, attention mechanisms, and feature learning strategies, these models aim to set new standards in automated defect detection, ensuring both efficiency and accuracy in quality control across various industries. The overarching goal of this research is to push the boundaries of what's achievable with automated defect detection systems, demonstrating that with the right combination of deep learning techniques, these systems can not only match but surpass the capabilities of

traditional methods. The models proposed in this thesis aim to contribute significantly to their respective fields by providing more reliable, efficient, and accurate defect detection solutions, thus enhancing quality control and safety in various industrial and infrastructural sectors.

### 1.2.1 Research Questions

This thesis seeks to advance the field of defect classification and segmentation using deep learning models in various industrial contexts. The research is specifically focused on developing models that can handle the complexities and variability of defects in textile manufacturing, road maintenance, and steel production. Based on these objectives, the research questions are as follows:

- How can the proposed deep learning models be optimized to achieve superior performance in defect classification and segmentation compared to traditional methods in their respective industrial applications? This question involves a comparative analysis of the proposed models against existing methods, seeking to understand the advancements these models offer in terms of accuracy, efficiency, and reliability.
- How can deep learning, specifically a capsule-based neural network, be enhanced to improve the accuracy of texture defect classification in fabrics? This question aims to explore the limitations of traditional capsule networks and investigate how integrating CNNs and spatial attention modules can enhance their capability to accurately classify complex fabric defects.
- What deep learning architecture, particularly in the context of DepthCrackNet, is most effective for detecting and segmenting pavement cracks, considering their

variable nature and the presence of anomalies? This question seeks to assess the effectiveness of the DepthCrackNet model, focusing on its unique combination of 3D spatial features and Multi-Head Attention mechanism, in accurately detecting pavement cracks under various public datasets.

- In what ways can the UNet3+ model be modified, particularly through the integration of a MultiScale Feature Learning Module and CBAM, to enhance the detection of surface defects on steel? The aim here is to evaluate the effectiveness of the E-Unet3+ model in detecting diverse types of defects on steel surfaces, focusing on how its enhanced feature learning capabilities and attention mechanisms contribute to its performance.

By addressing these questions, this thesis aims to provide contributions to the field of automated defect detection, offering solutions that enhance the accuracy, efficiency, and adaptability of defect classification and segmentation processes in various industrial settings.

## 1.3 Different Approaches in Defect Analysis

The field of defect analysis has multiple facets, and the problem can be subdivided into various approaches, each with its own set of challenges and requirements [10]. Below are the key tasks commonly encountered in defect analysis:

a) **Defect Classification**: Classification of defects relies on identifying the specific type of defect presented in an image, usually by labeling it as one of several predefined classes. Machine learning models, particularly neural

networks, have shown great promise in performing this task. However, the majority of these models lack the capability to focus on localized, intricate patterns in the images, which is often crucial for identifying subtle defects. This is crucial for quality control processes that require an understanding of defect types.

b) **Defect Detection**: Aims to not only classify the objects (or defects) in images but also to locate them by drawing bounding boxes around each object. This is particularly useful in scenarios where the spatial location of defects matters.

c) **Defect Segmentation**: Segmentation tasks go beyond classification and object detection by identifying the precise pixel locations of the defect in the image. This enables a more detailed analysis, which is essential for applications where the extent and shape of the defect are crucial for assessing the severity and potential impact.

d) **Anomaly Detection**: Anomaly detection stands apart as it does not rely on prior labeling of defect types. Instead, it learns the 'normal' state of an object or scene and identifies anomalies or deviations from this learned norm. This is particularly useful for identifying new or rare types of defects that have not been previously cataloged.

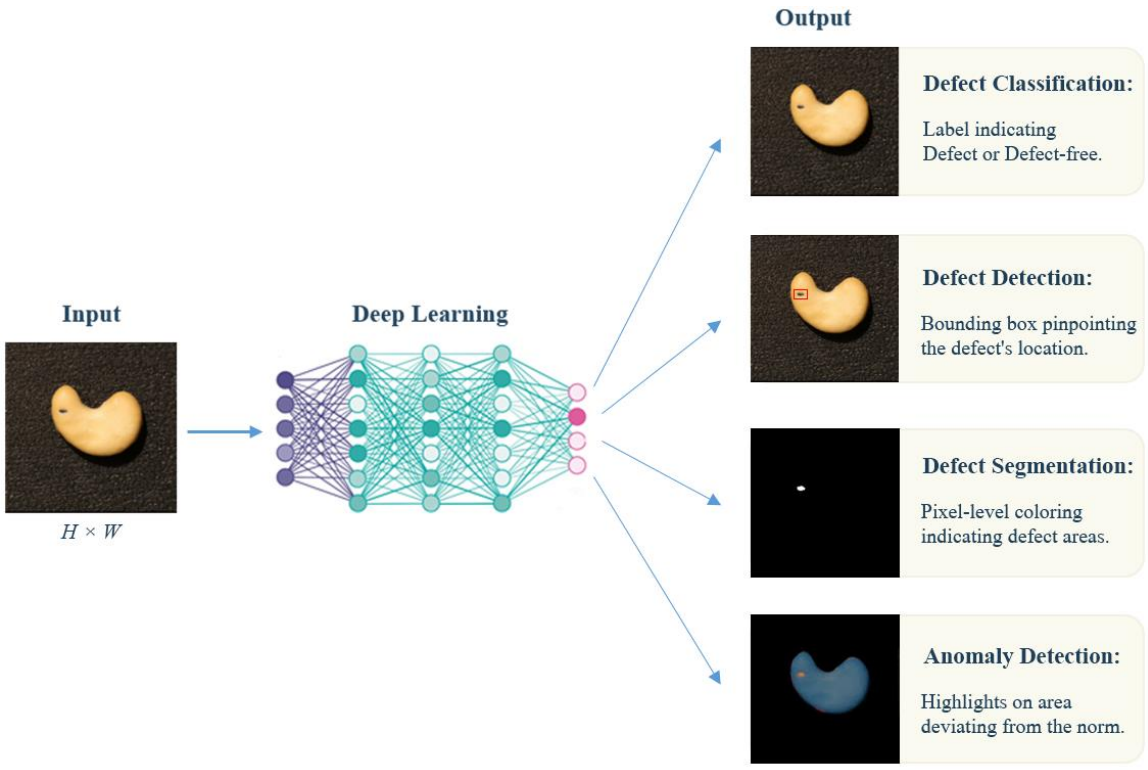


Figure 1.2: An illustration of defect classification, localization, and segmentation.

These tasks require sophisticated models that can understand the intricate patterns and variations in image data to make accurate predictions. The advances in machine learning algorithms and computational hardware have paved the way for more complex and efficient models capable of performing these tasks with high accuracy. However, with the rise of complex manufacturing processes that employ a wide range of materials and techniques, there is a growing need for robust and versatile defect detection systems.

This thesis aims to address these challenges by proposing three distinct but interconnected machine learning models. Each model is designed to solve a specific sub-problem in this domain - ranging from a texture defect classification model using a capsule-based neural network, pavement crack segmentation using 3D spatial features, steel surface defect segmentation using enhanced UNet3+.

## 1.4 Scope and Limitations

### 1.4.1 Scope

**a) Which Defect Analysis Approach Covers (Technological Scope)**

This research integrates machine learning and computer vision for industrial quality control applications. My focus is on identifying defects, with a special emphasis on their classification and segmentation.

**b) Where It Can Be Used (Industrial Scope)**

The models I have created are particularly fine tuned for industrial applications such as texture defects, pavement cracks, and steel surface defects. Additionally, these models and models have potential utility in other sectors requiring accurate defect detection, like medical imaging in healthcare or infrastructure inspection in civil engineering.

**c) Range of Data Used (Dataset Scope)**

The research will use multiple domain-specific datasets to train and validate and fine-tune the proposed models. Each dataset is selected to represent the type of defects that the corresponding model aims to detect, classify, or segment. These datasets include images with varying degrees of complexity, captured under different environmental conditions, to ensure that the models are robust and adaptable.

### 1.4.2 Limitations

**a) Computational Complexity**

While aiming for high accuracy and precision, some of the proposed models involve complex architectures that may require substantial computational resources for training and inference. This could restrict their immediate use in settings with limited resources.

**b) Generalizability Across Materials**

The thesis primarily focuses on specific types of defects on certain materials (e.g., fabric surface defects, steel surface defects). While the principles may apply broadly, the models might require fine-tuning or adaptation to work effectively on other types of materials or defects.

**c) Lack of Real-world Validation**

Although the models are rigorously tested using available datasets, the validation will primarily be simulation-based. Actual industrial settings might present additional challenges, such as varying lighting conditions, that are not fully accounted for in the datasets used for training and validation.

This thesis aiming to improve automated systems for detecting defects by introducing new, deep learning models. However, the research acknowledges its limitations in computational demands, generalizability, and real-world applicability. Understanding these limitations is essential for interpreting the results of this study and for figuring out where to focus future research.

## 1.5 Research Contributions

This thesis introduces improvements in the field of defect classification and segmentation through deep learning models, specifically targeting applications in textile manufacturing, road maintenance, and steel production. In summary, this thesis makes the following contributions:

1. Advancements in Fabric Defect Classification with Capsule Networks and CNN Integration:
    - The thesis integrates Capsule Networks with traditional CNNs, utilizing pre-trained models like DenseNet201 and InceptionV3. This integration enhances the model's capability in capturing complex hierarchical feature relationships, crucial for accurate defect classification in fabric textures.
    - The research establishes a new benchmark on the TILDA texture datasets, demonstrating defect classification performance and effectively handling real-world fabric defect scenarios.
2. Development of DepthCrackNet for Enhanced Pavement Crack Detection:
    - The DepthCrackNet model is another contribution of this thesis, employing the Double Convolutional Encoder (DCE) structure for optimized feature extraction and parameter efficiency.
    - A novel TriInput Multi-Head Spatial Attention (TMSA) mechanism is introduced, processing three input feature maps simultaneously. This mechanism uses multi-head attention to extract richer contextual information, thereby enhancing segmentation precision.

- The Spatial Depth Enhancer (SDE) module is another innovative addition, skillfully extending two-dimensional feature maps into a three-dimensional context to amplify depth perception and spatial representation.

3.Creation of the Enhanced UNet3+ Model for Steel Surface Defect Detection:

- The E-UNet3+ model is introduced with a novel encoder structure that incorporates varied dilation rates and DropBlock regularization. This structure is crucial for capturing multi-scale features, enhancing the model's adaptability and robustness.
- An innovative adaptation involves augmenting traditional max-pooling in UNet3+ with strided Conv2D layers, followed by concatenation and a 1x1 convolution. This approach helps retain critical feature information and elevates the model's predictive power.
- Experimental evaluations demonstrate that the E-UNet3+ model exhibits enhanced performance compared to previous studies and state-of-the-art models. The architecture effectively identifies surface defects, including various types, background similarities, and diverse sizes, marking a substantial advancement in steel surface defect detection.

These contributions collectively represent advancement in the realm of automated defect detection using deep learning. The findings and innovations presented here can contribute valuable insights and tools for enhancing quality control and safety in textile manufacturing, road maintenance, and steel production through advanced deep learning applications.

## 1.6 Thesis Outline

This thesis is divided into several chapters, each with a specific role that adds to the overall story of improving automated defect detection. Here's a summary of what each chapter covers:

- **Chapter 1** presents the context for the research, highlighting the significance of quality control and defect detection in various industries.
- **Chapter 2** offers a comprehensive overview of the existing literature in the field of machine learning models for defect detection and classification. It also identifies the gaps in the current state of the art, providing the academic backdrop against which this thesis is situated.
- In **chapter 3** the development and evaluation of a capsule-based neural network for texture defect classification are discussed. The chapter begins with the rationale behind choosing a capsule-based architecture, followed by a detailed explanation of the methodology, experiments, and results.
- **Chapter 4** focuses on a specialized model for pavement crack segmentation. The chapter outlines the model architecture, incorporating 3D spatial features and a multi-head attention mechanism, and presents experimental results validating its effectiveness.
- **Chapter 5** discusses a model developed for steel surface defect segmentation. It describes the enhanced UNet3+ architecture employed and how it incorporates multiscale feature learning and attention mechanisms for better segmentation performance. The model's efficacy is demonstrated through a series of experiments.

- **Chapter 6** summarizes the findings from the research chapters and outlines avenues for future work. It also discusses the broader implications of the research and recommends subsequent studies to be conducted.

# Chapter 2. Literature Review

## 2.1 Introduction

Advances in technology and research are making more industries, like textiles and road safety, interested in using machines to detect defects. Manual inspections, although considered the traditional method, are not only labor-intensive but are also prone to human errors. With the increasing demand for high-quality products and the need for timely maintenance in infrastructures, the requirement for accurate, efficient, and swift automated defect detection systems has been more pressing than ever. The field of detecting and classifying defects has greatly evolved, with many methods developed to address the various challenges in different industries. In the textile industry, the main focus is on making sure the fabric is of good quality by correctly identifying fabric defects, which come in different patterns, shapes, and sizes and often blend into complicated backgrounds. Similarly, in road safety, effective identification of pavement cracks can substantially reduce accidents and maintain roads for longer. Meanwhile, the manufacturing sector, especially steel production, emphasizes the timely detection of surface defects to guarantee product quality. All these challenges converge on a common point: the necessity of automated defect detection systems that can outperform manual inspections in terms of accuracy and efficiency. Recent advances in deep learning and computer vision have led to new, innovative methods specifically designed to meet these challenges. Notably, the applications of advanced neural network architectures, such as capsule networks, U-Net shaped models, and transformers, have demonstrated considerable promise in enhancing defect and anomaly detection capabilities. Furthermore, these models often incorporate

mechanisms like spatial attention, multiscale feature learning, and novel encoding techniques to elevate their performance metrics.

This chapter delves into the existing literature on defect and anomaly detection, aiming to provide a comprehensive understanding of current methodologies, their underlying principles, and their practical applications. It also examines recent key studies to clearly explain the latest techniques used in specific areas like textile defect classification, pavement crack segmentation, steel surface defect detection, and general anomaly detection and localization. The following sections will go into detail about each area, providing a thorough review of significant research contributions and offering insights into their strengths, weaknesses, and possible future developments.

## 2.2 Texture Defect Classification

In recent years, the textile industry has gained attention for its major impact on the world's economies. With increasing demand, ensuring the quality of fabrics has become a paramount concern for manufacturers. As fabric defect detection stands as a pivotal aspect of quality assurance, the advancements in computer vision and machine learning offer immense promise. By integrating these technologies, researchers hope to redefine the boundaries of fabric inspection, paving the way for both accuracy and efficiency. This literature review delves into the different methods used in detecting and classifying fabric defects, uncovering the challenges and successes of each approach.

Fabrics, with their intricate patterns and varying textures, pose unique challenges for defect detection. Imperfections can range from minuscule thread anomalies to significant texture variations. Traditional manual inspections, although detailed, are time-consuming

and can be prone to human errors. Hence, the quest for automated solutions has led to an influx of research in this realm. Numerous studies have embarked on the journey to decipher fabric anomalies using advanced algorithms [11]. When broadly categorizing the research efforts, one can identify four dominant paradigms: statistical, spectral, model-based, and learning-based techniques. While each category has its distinct methodologies and principles, their unified goal remains consistent: to address and overcome the intricacies inherent in textile defect analysis [12].

- **Statistical Methods**:

A major fraction of fabric defect detection methodologies relies on statistical analysis, rooted deeply in fundamental image processing procedures. The essence of these methods lies in their capability to decipher variations in patterns, drawing upon the inherent statistical properties of fabrics. These techniques operate on a core assumption that fabrics, in their pristine state, exhibit a uniform statistical behavior. Thus, any deviation from this uniformity potentially signals a defect. The art of these methods lies in discerning these deviations from the norm. Widely recognized strategies in this domain include morphological operations [13] which focus on altering the structure of objects within an image, thresholding [14] that bifurcates an image into segments based on pixel intensity, and auto-correlation functions [15] which assess the similarity between sequences of pixels in an image. These techniques are often employed in tandem to enhance defect detection rates. A particularly noteworthy method integrates the Gray Level Co-occurrence Matrix (GLCM) with Local Binary Patterns (LBP) [16]. Here, LBP delves into capturing the minute nuances and variations in fabric images, thereby identifying local anomalies. On the other hand, GLCM zooms out to observe the broader picture, gathering data on the

global texture statistics of the fabric. This duality, combining both micro and macro perspectives, enriches the data pool, setting the stage for a more refined classification process. Another method couples Coordinated Clusters Representation with LBP, harnessing the power of the Rough Set Theory for a nuanced fabric classification [17]. In light of these developments, it's evident that statistical methods, with their foundational principles and innovative adaptations, continue to be a cornerstone in the world of fabric defect detection. As research progresses, it is anticipated that these methods will undergo further refinements, elevating their precision and applicability.

- **Spectral Methods**:

At the intersection of mathematics and image processing, spectral methods have emerged as a dominant force in the fabric defect detection arena. They center on manipulating images across various domains, providing a unique lens to discern potential anomalies. These methods transition image data between the spatial and frequency domains, thereby highlighting aspects of the image that might not be evident in its native form. When applied to fabrics, these transitions can reveal subtle irregularities in texture and pattern, which might be indicative of defects. Three techniques often steal the spotlight in spectral analyses: the Wavelet transform, the Fourier transform, and the Gabor transform [18]. While the Wavelet transform is adept at capturing localized nuances across multiple orientations, the Fourier technique in converting signals from the time domain to the frequency domain, offering a broader overview [19]. The Gabor transform, with its versatility, straddles both domains, offering insights that are both detailed and comprehensive. The Gabor filter, for instance, has seen numerous implementations and refinements to optimize its defect detection capabilities [20], [21]. One standout approach

involved an adaptive wavelet-based feature extraction coupled with a Euclidean distance detector, yielding good results, especially in the context of plain and twill fabrics [22].

- **Model-Based Methods**:

As the name suggests, model-based approaches hinge on the use of a predetermined model, usually representing defect-free fabric, to compare and evaluate inspection images. These methods revolve around the principle of using a standard or reference. Any deviation from this reference, which represents the ideal fabric, potentially points to a defect. Among the forerunners in this domain are the Autoregressive models and the Markov Random Field methods [23]. The Autoregressive model, in particular, offers a computational edge, characterizing randomness based on the time domain and often yielding solutions derived from linear equations [24]. A noteworthy exploration in this category saw the use of the Gaussian Markov Random Field (GMRF) to emulate defect-free textures on fabric images [25]. This research framed defect detection as a statistical hypothesis testing challenge, and though the results were promising, it also highlighted the need for more expansive testing and evaluation. With the rise of machine learning, it was only a matter of time before its ability was harnessed for fabric defect detection. Learning-based methods utilize labeled data sets to train models, which then extrapolate from this learning to identify defects in new, unseen fabric images. At the heart of these methods lies the principle of learning from data. By feeding these models a plethora of examples, they "learn" to recognize patterns, which they subsequently use to identify anomalies in fabric samples. The techniques under this umbrella are varied, encompassing Artificial Neural Networks (ANN), Support Vector Machines (SVM), Regression models, Nearest Neighbor (NN) techniques, and more elaborate methodologies. An intriguing proposal came from [26], who introduced the

MSCDAE network model for fabric defect detection, using defect-free images for training and incorporating data augmentation techniques. Another significant contribution emerged from [27], which melded Artificial Neural Networks with CoHog features, achieving real-time detection efficiencies in an industrial context. A further innovation saw the introduction of a CNN-based approach [28], segmenting fabric images into patches and utilizing distance-matching functions for enhanced defect detection.

In essence, the advancements in spectral, model-based, and learning-based methods underscore the dynamic and evolving nature of fabric defect detection. As technology and research converge, it's anticipated that these methods will see further refinements, pushing the boundaries of what's possible in the realm of fabric quality assurance.

| Method Category | Description | Reference | Methodology | Advantages | Limitations |
|---|---|---|---|---|---|
| **Statistical** | Utilizes fundamental image processing and statistical analysis to detect variations in fabric patterns. | [13] | Morphological operations | Can clarify and emphasize key structural features; robust against noise in binary images | May alter or remove important features if not carefully applied; limited in handling color textures |
| | | [14] | Thresholding techniques | Easy to implement; efficient for real-time applications; well-suited for binary and simple textures | Poor performance under varying lighting conditions; not suitable for complex textures |
| | | [15] | Auto-correlation functions | Good for periodic pattern detection and textural feature analysis | Performance degrades with noise; computational complexity can be high for large images |
| | | [16] | GLCM with LBP | Offers a detailed textural analysis by combining two powerful descriptors | GLCM is sensitive to the choice of distance and angle parameters; LBP may miss larger patterns due to local analysis |
| | | [17] | Coordinated Clusters with LBP and Rough Set Theory | Integrates pattern recognition with uncertainty handling, which may offer robust performance in ambiguous scenarios | Method complexity can hinder real-time application; may require extensive training and parameter tuning |

| | | | | | |
|---|---|---|---|---|---|
| **Spectral** | Employs mathematical transformations to reveal fabric defects by highlighting irregularities in different domains. | [18] | Spectral methods (Wavelet, Fourier, Gabor transforms) | Effective in multi-scale and orientation-specific defect detection; useful for a wide range of textures | Choosing the right transform or a set of parameters can be non-trivial; may not be optimal for non-periodic patterns |
| | | [19] | Fourier transform | Excellent for identifying periodicity and global textural properties; relatively fast with FFT | Ineffective for localizing defects in the spatial domain; can be less intuitive to interpret |
| | | [20] | Gabor filter optimizations | Balances spatial and frequency information; tunable for specific orientations and scales | Optimization process can be complex; Gabor filters generate a large amount of data which can be cumbersome to process |
| | | [22] | Adaptive wavelet-based feature extraction | Particularly strong in capturing hierarchical and localized fabric defects | Requires careful tuning of wavelet functions; may be computationally expensive |
| **Model-Based** | Uses a pre-determined model representing defect-free fabric as a benchmark for detecting deviations indicative of defects. | [23] | Autoregressive models | Utilizes time-series analysis techniques that can be powerful for texture modeling | Assumptions of linearity and stationarity can be limiting for complex fabric patterns |
| | | [25] | Gaussian Markov Random Field (GMRF) | Capable of modeling complex stochastic textures with spatial dependencies | Computationally intensive, especially for large-scale problems; sensitive to model parameters |
| | | [26] | MSCDAE network model | Leverages the ability of deep networks to learn high-level features; effective even with a small number of defect samples when using augmentation | High computational resource demand during training; may not generalize well to unseen types of fabric or defects |
| | | [27] | ANN with CoHog features | Combines classic feature extraction with powerful neural networks for potentially high accuracy | Requires substantial data for training; ANN models can be opaque ('black box') and challenging to troubleshoot |
| | | [28] | CNN-based patch segmentation | CNNs are state-of-the-art in image recognition tasks; can handle raw image data effectively | High computational costs for training and inference; requires large and diverse datasets for optimal performance |

Table 2.1: Comparative summary of methodologies in texture defect detection.

## 2.3 Pavement Crack Segmentation

Crack detection stands as a cornerstone in the realm of structural maintenance and safety assessment. The detection, identification, and analysis of cracks in various materials, from civil infrastructure to aerospace components, can be the difference between structural integrity and catastrophic failure. Historically, engineers and technicians relied heavily on manual inspections - a laborious and often error - prone process. The quest to refine accuracy and efficiency in this domain led to numerous technological breakthroughs. Traditional methods emerged as an initial answer, leveraging image processing and analysis to automate detection. However, as with all innovations, these techniques came with their own set of challenges. Wavelet Analysis, one such method, presents a fascinating case of how traditional crack detection techniques evolved over time, trying to overcome inherent limitations and enhance performance.

### a) Traditional Crack Detection Techniques

Traditional crack detection, in essence, refers to the array of techniques developed during the earlier phases of automated structural health monitoring. These techniques emerged as a response to the pressing need for automated, consistent, and objective crack detection methods that could outperform or, at the very least, complement manual inspections. Over time, researchers and experts identified distinct methods, each with its unique mechanism and application domain. The overarching aim was always clear: to capture the minutest of cracks efficiently and accurately to ensure structural safety.

- **Wavelet Analysis:**

One of the pioneering techniques in traditional crack detection was the Wavelet Analysis. The foundational idea behind wavelet analysis is to use wavelet transforms - a mathematical tool designed to dissect information based on different frequency components. At its core, wavelet analysis breaks down an image into different frequency bands. This allows for the differentiation between possible crack patterns (typically of a higher frequency due to their intricate and sudden nature) and the regular, undamaged patterns of a structure (which usually reside in the lower frequency bands). Wavelet analysis found particular favor in detecting cracks in pavements. For instance, research [29] showcased its application in automating crack detection in pavement images. This was achieved by employing the continuous wavelet transform to identify potential crack regions against the backdrop of the pavement's texture. However, like all techniques, wavelet analysis wasn't without its challenges. A primary limitation surfaced when dealing with images that presented a diverse range of textures. As indicated in the same study [29], the wavelet-based approach faced difficulties in consistently performing on such images. This limitation underscored the challenge of distinguishing between genuine cracks and misleading patterns that could resemble cracks in varied textures.

This challenge led to more nuanced approaches. In another significant study [30], researchers attempted to enhance the robustness of wavelet-based crack detection. They processed pavement images through a wavelet transform, which allowed them to break the image into multiple frequency subbands. The objective was to isolate distress signals (indicative of cracks) from ambient noise by segregating high-amplitude wavelet coefficients (representing distresses) from low-amplitude coefficients (indicative of noise).

- **Image Thresholding:**

Image thresholding operates on a fundamental principle: by identifying and segmenting variations in pixel intensities within an image, one can potentially isolate features of interest, such as cracks. Image thresholding typically involves assigning a 'threshold' value of pixel intensity. Pixels with intensities above this value are classified one way (e.g., as potential cracks), while those below are classified another (e.g., as background or non-crack regions). This method found prominence in studies [31]–[33] where preprocessing algorithms were first used to counteract any inconsistencies in illumination. Once uniformity was achieved, thresholding identified potential crack zones. The initial detections, though a step forward, were not always perfect and thus required further refinement. Post-thresholding, techniques such as morphological operations were used to enhance the accuracy of crack detections. Morphological operations, involving processes like dilation and erosion, can help in reducing noise and bridging small gaps in detected cracks. Another intriguing method was presented in [34], where crack images were segmented using a combination of histogram analysis and Ostu's thresholding. Here, the image was partitioned into four equal sub-images, each undergoing individual analysis. The results were then integrated, providing a comprehensive overview of potential cracks, especially in images with a low signal-to-noise ratio.

- **Manual Feature Extraction and Classification:**

Moving away from generalized image operations, some methodologies focused on the extraction of specific, manually-defined features from images. In this technique, certain characteristics or 'features' within an image are manually identified, extracted, and then used as descriptors for potential cracks. Many current crack detection techniques rely on such handcrafted features. As exemplified in studies [35], [36], [37], features like the

Histogram of Oriented Gradients (HOG) [35] were extracted from segments of an image to describe cracks. Once these features were extracted, they were fed into classifiers, often machine learning models like Support Vector Machines (SVMs). These classifiers were trained to recognize and differentiate between crack and non-crack features, thus automating the detection process.

- **Boundary Detection Techniques:**

Boundary detection revolves around the idea that cracks often manifest as abrupt changes or 'edges' in an image. This technique emphasizes the identification and analysis of edges within an image. Since cracks create distinct boundaries, edge detection can be an effective way to spot them. Techniques such as the Sobel edge detection, as employed by [37], were used for crack identification after refining the image. This study further introduced a bidimensional empirical mode decomposition algorithm, aimed at reducing speckle noise which can interfere with edge detection. [38] took a slightly different route, incorporating morphological filters into their crack detection strategy. They leveraged a modified median filter to suppress noise, enhancing the clarity and precision of detected cracks.

- **Shortest Path Techniques:**

Shortest Path Techniques focus on tracing the optimal path through an image, often capitalizing on the continuous nature of cracks. The idea is to identify paths or contours that resemble cracks in an image, even when some parts of the crack are not distinctly visible. A notable approach was introduced by [39], which employed an advanced minimal path method to detect contours, minimizing the need for prior knowledge about the crack's

topology or endpoints. In [40], the technique began by highlighting potential crack areas using a windowed intensity path-based approach. After this preliminary identification, crack segmentation was carried out using a model that relied on a multivariate statistical hypothesis test, aiming for a more precise crack delineation.

### b) Deep Learning-based Crack Detection

The world of crack detection underwent a paradigm shift with the introduction of deep learning. Deep learning, a subset of machine learning, employs neural networks with many layers (hence "deep") to analyze various forms of data. Within the realm of image analysis and computer vision, deep learning techniques, especially Convolutional Neural Networks (CNNs), have emerged as game changers. Deep learning models are trained on vast datasets to recognize patterns. In the context of crack detection, these networks learn to identify and differentiate crack-like features from the myriad of other possible patterns in an image. They do so by processing images through multiple layers, each extracting and refining features, culminating in the ability to recognize even subtle cracks. Recent years have seen tremendous success in the application of deep learning to crack detection. A significant driver behind this has been the performance of CNNs in computer vision tasks. These networks, designed specifically for image data, leverage spatial hierarchies and patterns to extract features and make predictions.

With advancements in semantic segmentation tasks, researchers have been tailoring these techniques for crack detection. For instance, [41] presented a technique inspired by SegNet [42], optimized for video frames from remote visual inspections. By compiling crack likelihood across overlapping frames, this method offers a dynamic way to detect cracks in videos. Another remarkable work is that of [43], which introduced an

architectural amalgamation, fusing feature pyramid and hierarchical boosting components. This sophisticated approach aimed to allocate importance based on the complexity of image samples, potentially improving crack recognition in challenging scenarios. Addressing the challenge of limited datasets, [44] tapped into Generative Adversarial Networks (GANs) to augment data for crack detection. By generating synthetic yet realistic images of cracks, this method enhanced the diversity of training data, leading to more robust models. The study by [45] brought forward a semi-supervised technique for crack segmentation. This method produces supervision signals for unlabeled images, which can help in training models even when comprehensive labeled data is scarce. Beyond general-purpose CNNs, researchers have begun tailoring networks specifically for crack detection challenges. For example, [46] designed a semantic segmentation model to not only detect cracks in infrastructure but also measure their width precisely, a critical metric for assessing structural health. Some researchers, like those in [47], blended traditional techniques with deep learning. By merging the capabilities of CNNs with multi-scale structured forests, they aimed to harness localized information amidst intricate backgrounds, presenting a potential solution to challenges faced by older edge detection methods. It's essential to mention studies like [48], which took the deep learning models from the lab to real-world scenarios. Utilizing the popular AlexNet network, they enhanced the reliability of detection mechanisms, proving the feasibility and effectiveness of deep learning in practical crack detection tasks.

Despite its remarkable successes, deep learning-based crack detection isn't free from challenges. Models require vast amounts of labeled data for training, and while techniques like data augmentation address this to some extent, obtaining high-quality labeled data

remains a hurdle. Additionally, while these models excel in controlled environments, their performance in varied real-world scenarios - with differing lighting, textures, and noise levels - remains a topic of ongoing research.

| Method Category | Reference | Methodology | Advantages | Limitations |
|---|---|---|---|---|
| Traditional | [29] | Wavelet Analysis | Targets high-frequency crack patterns; Automates pavement crack detection | Performance can degrade with complex textures; May confuse crack-like textures with actual cracks |
| | [30] | Enhanced Wavelet Analysis | Increased robustness through subband processing; Better isolation of distress signals | Potentially more computationally intensive; May require fine-tuning for different textures |
| | [31]–[33] | Image Thresholding with Preprocessing | Addresses illumination inconsistencies; Simplifies the detection process | Sensitive to threshold value selection; May miss fine cracks with uniform background |
| | [34] | Histogram Analysis and Otsu's Thresholding | Effectively segments images with varying intensities; Suitable for low contrast images | Requires separate processing for different image sections; Histogram-based methods can be fooled by noise |
| | [35] | Manual Feature Extraction and Classification | Can be highly accurate with well-defined features; Good for controlled environments | Time-consuming feature selection process; Classification accuracy depends on feature selection |
| | [38] | Boundary Detection Techniques | Directly identifies crack edges; Can be combined with noise reduction techniques | Edge detection can be ambiguous in noisy images; May require additional processing to confirm crack detection |
| | [39] | Shortest Path Techniques | Can detect non-visible parts of cracks; Less dependent on image quality | Complex implementation; May generate false positives if the path is incorrectly identified |
| Deep Learning | [41] | CNNs with SegNet for Videos | Allows for temporal analysis in video data; Can track crack propagation over time | Requires significant computational resources; May need fine-tuning for different video qualities |
| | [43] | Deep Learning with Architectural Amalgamation | Adapts to the complexity of samples; Potential for better generalization | Architectural complexity can lead to longer training times; May require extensive hyperparameter optimization |
| | [44] | GANs for Data Augmentation | Addresses data scarcity and diversity issues; Generates training data with novel crack patterns | GAN-generated data may diverge from real-world scenarios; Training GANs is resource-intensive and complex |
| | [45] | Semi-supervised Deep Learning | Utilizes unlabeled data, reducing annotation needs; Can work with partially labeled datasets | Semi-supervised results may vary widely; Still requires a sufficient amount of labeled data for best performance |

| | | | | |
|---|---|---|---|---|
| | [46] | Specialized Semantic Segmentation Models | Provides precise crack dimensions; High utility for structural health monitoring | Specialized models may not generalize well; Development and training can be resource-heavy |
| | [47] | CNNs with Multi-scale Structured Forests | Enhanced localized feature extraction; Integrates traditional and deep learning methods | Integration can be challenging and may lead to overfitting; Complex models require extensive validation |
| | [48] | AlexNet for Real-world Applications | Proven effectiveness in diverse conditions; Beneficial for large-scale deployment | May not handle extreme variability well; Real-world application requires continuous model updating |

Table 2.2: Comparative summary of methodologies in pavement crack detection.

## 2.4 Steel Surface Defect Segmentation

The journey of identifying surface defects has evolved remarkably over time. In the earliest phases of research, basic image processing techniques were the cornerstone for detecting these defects. Techniques such as thresholding, morphological operations, and Fourier transforms took center stage [49]. For instance, a study by [50] embraced rudimentary image processing techniques. They used tools like noise filtering, gradients, and morphological operations to zero in on the best thresholding values. Impressively, this method outperformed other established thresholding models, including OTSU. Meanwhile, another research [51] showcased a novel dynamic thresholding method that relied heavily on the distribution of pixel intensity. This technique demonstrated substantial potential, especially when applied to industrial steel images from a hot rolling apparatus. However, while powerful, these threshold-centric methods [50], [51] are not without vulnerabilities. They tend to be sensitive to noise, a consequence of depending heavily on pixel intensity distributions. On a tangent, another study [52] utilized the KNN classifier to pinpoint defects on standard surfaces. They refined their results by integrating fundamental image processing techniques, including edge detection and morphological processing. The

literature also introduced innovative methods, such as the entity sparsity pursuit (ESP) approach proposed by [53]. This method harnessed the Local Binary Patterns (LBP) for feature extraction, which was then segmented into easy-to-perceive and dense super-pixels. The ESP algorithm then took the reins to detect defects. [42] adopted an alternative angle, presuming the surface to be even, and then honed in on defect zones that deviated from this pattern. However, it's essential to acknowledge that while these methods produced commendable results, they weren't free from challenges. The most glaring among these are the limitations tied to manual feature extraction techniques, which are not only time-intensive but also often lack broad applicability. Moreover, any deployment on varied surfaces often demands model recalibration. The realm of surface defect identification experienced a seismic shift with the advent of deep learning-based image segmentation techniques. Their unparalleled precision, computational efficacy, and user-friendliness set them apart. Unlike their predecessors, which demanded manual feature extraction followed by classification, deep learning models amalgamated these steps. They were adept at autonomously deriving the necessary features from training data, bypassing the need for expert-curated feature sets. Of these, encoder-decoder structures such as SegNet [42] and U-Net [54] have garnered immense attention, as have pyramid pooling structures like Deeplabv3+ [52] and PSPNet [55]. Capitalizing on these foundational architectures, several enhancements surfaced, bolstering model sturdiness and segmentation precision. These modifications fall broadly into three categories: attention-centric techniques, feature aggregation methods, and strategies rooted in Conditional Random Field (CRF). Delving deeper into feature aggregation, its essence lies in amplifying model accuracy in semantic segmentation undertakings. Presently, the literature chiefly recognizes two types of feature

aggregation techniques, multiscale and multi-level. The inception block [53] is a testament to multiscale feature aggregation. It amalgamates features gleaned using varied sizes of convolution kernels, facilitating a richer image representation. Conversely, multi-level feature aggregation is oriented towards fusing granular and broad features. This fusion, achieved by operations that upsample and downsample, ensures models harness both intricate details and the larger picture in images. An exemplar of this approach is [56], which devised a robust feature pyramid, yielding particularly good outcomes in remote sensing image segmentation. Feature aggregation, however, is not without its drawbacks. These techniques, while bolstering accuracy, increase the complexity of the models, which can be a double-edged sword. Recent times have seen the rise of attention-centric methods in deep learning, drawing inspiration from human vision mechanisms. Generally, these methods are categorized into channel attention, spatial attention, and non-local attention. For example, the SE module [57] emphasized the interplay between feature channels, while [58] underscored high-frequency features. CBAM [59] and subsequent improvements [60] balanced channel and spatial dimensions. However, while promising, these methods do pose computational challenges, particularly the non-local-based methods. Lastly, one notable shortcoming of traditional semantic segmentation models is their inability to capture adequate scene-level context, making object boundaries murky. To remedy this, researchers have turned to Conditional Random Fields (CRFs) for post-processing. An approach by [61] employed a Convolutional Neural Network (CNN) to construct superpixel potentials, which were then finessed using CRF parameters optimized by an SVM, yielding enhanced segmentation masks. In conclusion, the journey of surface defect identification has been transformative. From basic techniques to sophisticated deep

learning models, the evolution is palpable. Yet, as with any journey, there are milestones yet to be achieved, particularly in the realms of computational efficiency and model robustness.

| Reference | Methodology | Advantages | Limitations |
|---|---|---|---|
| [49] | Basic image processing | Pioneered surface defect detection with simple computational tools | Simple techniques may not be as effective with complex or noisy images |
| [50] | Rudimentary image processing | Surpassed established thresholding models; effective for specific applications | Sensitive to variable lighting conditions and noise |
| [51] | Dynamic thresholding based on pixel intensity | Flexible in dealing with different intensity distributions; beneficial for certain industrial applications | Struggles with high levels of image variation and noise |
| [52] | KNN classifier with image processing techniques | Combined traditional image processing with machine learning for improved accuracy | Limited by the need for feature engineering and potential overfitting |
| [53] | ESP approach with LBP | Enhanced defect detection by using texture-based features | Can be computationally intensive due to the complexity of the algorithm |
| [42] | Surface evenness assessment | Effective for planar surface defects; simple to apply to certain types of surfaces | Not universally applicable; struggles with complex surface geometries |
| [54] | U-Net architecture | High precision and efficacy in segmentation; capable of dealing with complex images | Can require substantial computational resources; may need large datasets |
| [55] | PSPNet architecture | Improved accuracy over other segmentation methods, especially in edge detail | Higher complexity can lead to slower inference times |
| [56] | Feature pyramid for remote sensing image segmentation | Capable of capturing multi-scale features for better segmentation | Potential increase in computational cost due to multi-scale processing |
| [57] | SE module for channel attention | Focused on relevant features, reducing the influence of irrelevant information | May not always capture spatial dependencies, leading to incomplete context |
| [58] | Spatial attention for high-frequency features | Highlighted textural and edge details for better segmentation | High-frequency emphasis might miss out on lower-level, yet critical, features |
| [59] | CBAM for channel and spatial dimensions | Aimed to capture both spatial and channel-wise feature dependencies | Balancing the two dimensions can be difficult and computationally intensive |
| [61] | CNN with CRF post-processing | Generated highly refined segmentation masks by effectively combining CNN outputs with CRF | Integration of CNN with CRF can be complex and requires careful parameter tuning |

Table 2.3: Comparative summary of methodologies in steel surface defect detection.

## 2.5 Summary

The field of defect detection and segmentation has experienced significant evolution across varied domains. This progression is evident across the three prominent areas examined in this thesis: texture defect classification, pavement crack segmentation, and steel surface defect segmentation.

- **Texture Defect Classification:**

Historically, the focus was primarily on traditional image processing techniques, with wavelet transforms and various statistical measures like entropy and variance being the mainstays. As research progressed, machine learning models emerged, providing a more robust solution by classifying defects based on these extracted features. However, the principal challenge remains in striking a balance between accuracy and computational efficiency.

- **Pavement Crack Segmentation:**

Pavement defect detection has seen an evolution from manual inspection to automated techniques. Earlier methods, such as thresholding and morphological operations, although effective, were susceptible to environmental factors and required constant calibration. The introduction of machine learning, particularly SVMs, delivered more accurate results, with the potential of reducing false positives. Deep learning models, particularly CNNs, then augmented this progress, proving highly competent in discerning intricate patterns and enhancing segmentation accuracy.

- **Steel Surface Defect Segmentation:**

Initial techniques were rooted in basic image processing, relying heavily on thresholding and pixel intensity distribution, despite their sensitivity to noise. The application of classifiers like KNN, alongside fundamental image processing, brought a degree of sophistication to this domain. Encoder-decoder structures, such as SegNet and U-Net, eliminated the need for manual feature extraction, thereby streamlining the process. Feature aggregation, while increasing accuracy, also added complexity to models. The recent rise of attention-centric methods in deep learning, inspired by human vision, and the application of Conditional Random Fields for post-processing, further fine-tuned the segmentation output.

Collectively, the journey from basic image processing to advanced deep learning models across these domains underscores the rapid advancements in defect detection and segmentation. While the strides have been impressive, challenges persist, especially in computational efficiency and model robustness, indicating ample opportunities for future research endeavors.

# Chapter 3. Texture Defect Classification Model Using a Capsule-Based Neural Network

## 3.1 Introduction and Problems

Textile materials, made from individual textile fibers, are crucial in our everyday lives. Their vast applications range from clothing to household items, making them an integral part of human life. However, the process of making textiles is filled with potential problems. During their production, textiles can develop a wide range of defects [62]. These defects can be caused by problems with equipment or poor-quality raw materials, resulting in lower quality fabric. Importantly, these defects can lead to serious financial losses. Imperfections can greatly decrease the retail value of these fabrics. , sometimes the value can drop by 45-65% [63]. This highlights the critical need for strict quality control in the textile industry. Maintaining top fabric quality is vital not only for protecting a brand's reputation but also for preventing possible economic losses [64]. Traditionally, the challenging task of finding these defects was mainly done by people using their hands and eyes. Skilled workers would carefully check rolls of fabric, looking for any irregularities. Once identified, these defects were then manually corrected. However, this method had its shortcomings. Some fabric defects are subtle, making them elusive to human detection [65], [5]. Relying on manual inspection also increases labor hours, leading to higher operational costs. Given the sheer number and variety of potential defects, there is an emerging need for an automated, more sophisticated inspection system. Such systems promise not just enhanced quality assurance but also significant reductions in labor costs [66]. Recent technological advancements have paved the way for a new era of textile

inspection. Contemporary inspection systems now leverage tools like machine learning, deep learning, and an array of machine vision technologies. These state-of-the-art methods have garnered substantial attention, both academically and industrially [67]. Their burgeoning appeal lies in their capability to revolutionize fabric inspection, heralding more efficient and effective quality control measures.

However, identifying fabric defects is still a complex task, filled with various challenges (refer to Figure 3.1):

a) **Defect Diversity**: Textiles can have a wide range of possible defects, appearing in various shapes, patterns, and visual characteristics. Some defects are obvious and easy to detect, but others are subtle, blending into the fabric's natural texture. This complex nature of defects makes it difficult for traditional inspection methods, which usually depend on fixed criteria or standards.

b) **Background Ambiguity**: Sometimes, defects can look so much like the surrounding fabric that they are almost impossible to tell apart. This similarity between the defect and its background makes detection hard. Traditional methods often struggle with this, leading to defects being missed or wrongly classified.

c) **Inconsistency in Defect Dimensions**: The size of defects on fabric can vary greatly. Some might be tiny spots, while others could cover large areas. These irregularities present a big challenge for automated systems. Creating a detection algorithm that can effectively handle such a wide range of defect sizes is a difficult task.

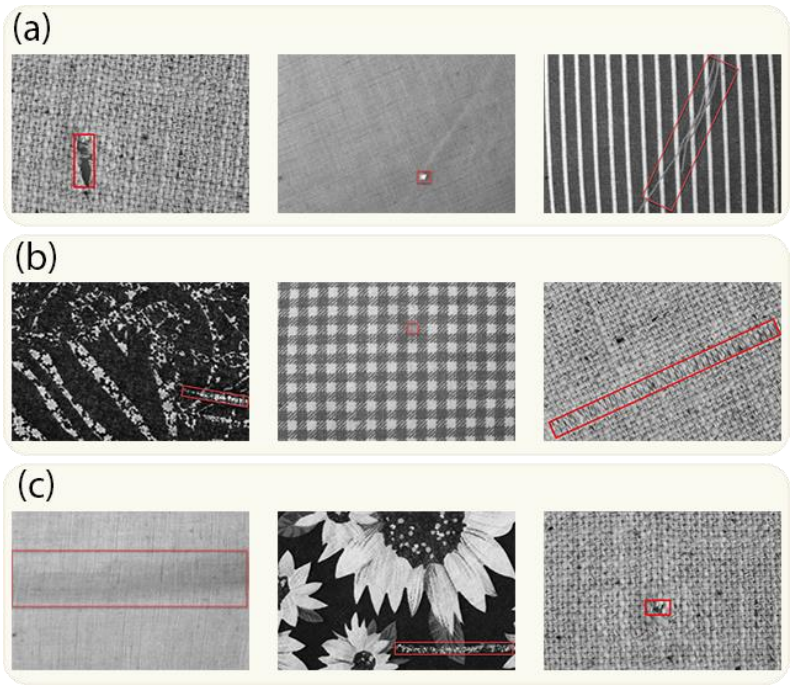


Figure 3.1: Illustrations of fabric flaws (highlighted with red indicators) and their locations, corresponding to the challenges in fabric defect categorization: (a) Variety in Defect Types, (b) Similar Backgrounds, and (c) Differences in Defect Dimensions.

In this section, I present a framework for texture defect classification, which leverages the capabilities of a capsule-based neural network. The main goal is to greatly improve how accurately flaws in textured items like fabrics are detected and categorized. Traditional Capsule Networks, though promising, often struggle because they use just one convolutional layer to extract features. This constraint becomes even more pronounced when identifying defects within intricate fabric textures, where the subtle intricacies of such defects can often go unnoticed. To overcome these obstacles, my model adopts a pioneering strategy: it merges modern Convolutional Neural Networks (CNNs) with a spatial attention module. This combination not only enhanced the weaknesses of the usual capsule networks but also takes advantage of the benefits of transfer learning, streamlining

and fine-tuning the feature extraction journey. The advantage of using transfer learning is its ability to use the strengths of previously trained models, which are good at identifying feature representations from large datasets. This combination significantly reduces the need for large training datasets and computational power, a common challenge in traditional methods. My model unfolds in two essential and crucial phases: preprocessing, and a process of feature extraction and classification. In preprocessing, each texture image is enhanced through data augmentation, which increases the variety and robustness of the dataset. These images are then resized to a uniform n × n dimension for consistency and normalized to create a standardized input data spectrum, ultimately enhancing and stabilizing the training journey. We extract deep features by combining a spatial attention module with pre-trained models like DenseNet201 and InceptionV3, all within a transfer learning approach. The strategic collaboration of CNN models and the spatial attention component within the capsule network proves to be a combination for feature extraction. The spatial attention component intelligently focusing on crucial image areas, highlighting defect-prone areas.

The main contributions of this model can be listed as:

- I utilized the Capsule Networks and conventional CNNs, addressing the shortcomings of CNNs in recognizing essential hierarchical feature relations, which are vital for precise defect detection in intricate fabric patterns.
- By utilizing pre-trained DenseNet201 and InceptionV3 architectures, the system is tailored to meet the specific demands of texture flaw identification, ensuring effective feature extraction.

- On the widely-referenced TILDA texture datasets, the model proved its defect classification capabilities and illustrating its utility in addressing real fabric defect challenges.

## 3.2 Background

In this section, my objective is to provide a comprehensive examination of the diverse methodologies that have been integrated into the proposed model. I commence the discourse by diving into the intricacies of Convolutional Neural Networks (CNNs), covered in Subsection 3.2.1 CNNs, as a fundamental component of deep learning, have transformed various applications, ranging from image recognition to natural language processing. This subsection aims to clarify the core principles, structure, and the important role of CNNs in the proposed model. Progressing further, Subsection 3.2.2 provides a comprehensive overview of Transfer Learning. This approach, frequently seen as a central component in contemporary machine learning, utilizes knowledge obtained from one domain to improve performance in another, albeit related, domain. By understanding the nuances of Transfer Learning, especially in scenarios where data might be limited or where pre-trained models offer a head start in the learning journey. Subsequently, in Subsection 3.2.3, the attention shifts to the Spatial Attention Module. This module enhances the model's capability by allowing it to selectively focus on specific spatial regions within an input. By doing so, the model can efficiently filter out irrelevant information and amplify crucial features, leading to potentially improved performance in tasks like image classification or object detection. Lastly, Subsection 3.2.4 introduces Capsule Networks, a type of neural network architecture. Unlike traditional neural networks, Capsule Networks

are designed to better understand and preserve the spatial relationships in data, which can enhance their performance in certain tasks. This section will explore the distinct structure of Capsule Networks, how they work, and the benefits they might offer to the model being discussed.

### 3.2.1 Convolutional Neural Network Models

In the area of Deep Learning, Convolutional Neural Networks (CNNs) are very important for tasks involving images. When looking closely at CNNs, we find they are made up of many layers, each with its own role. These layers work together to process and understand information from images in a complex but organized way.

The Convolutional Layer is a key part of CNNs (Convolutional Neural Networks). It has the important job of pulling out features from an image. This is done using special filters that process the image, automatically finding important details. Unlike older methods where features had to be manually identified, CNNs learn these directly from the data, making them more adaptable and accurate. Also, because the convolutional layer can understand the spatial layout of an image, it's really good at recognizing patterns no matter where they appear in the image.

After the convolutional layer in CNNs (Convolutional Neural Networks), there's the pooling layer, which is great at compressing data. This layer uses methods like max-pooling or average-pooling to make the size of the feature maps smaller. This reduction helps in two ways: it cuts down the amount of computing needed, speeding up the training of the network, and it also helps prevent overfitting, making sure the model can work well with new, unseen data. An important aspect of the pooling layer is that even though it

simplifies the data, it keeps the most important information, ensuring that the key parts of the image are still captured.

The Fully Connected Layer is found deeper in the network, and it plays the role of a major integrator. In this layer, every neuron is connected to every neuron in the previous layer. This dense interconnection allows the fully connected layer to bring together all the processed information from earlier layers and use it to produce the final result. This layer is crucial whether the task is to classify an image into a category or to predict a value in a regression task. It's here that the network makes its final decisions based on the data it has received and processed.

CNNs (Convolutional Neural Networks) are especially good at learning features in a hierarchical way. The initial, or shallow, layers of a CNN focus on simple features like edges, gradients, or colors. As we go deeper into the network, these layers start to recognize more complex structures, patterns, and objects. This process is similar to building a structure: the early layers are like laying the bricks, and the deeper layers use these bricks to create walls, rooms, and eventually the entire building. The evolution of deep learning has seen many different CNN architectures, each marking a step forward in the field. For example, AlexNet was a significant development that began a new era in deep learning, showing how effective deep networks are at recognizing images.

### 3.2.2 Transfer Learning

In the broad field of machine learning, Transfer Learning (TL) stands out as a highly efficient approach. It operates on a compelling premise: why start from scratch when one can capitalize on the reservoir of knowledge accumulated from a prior learning experience? This paradigm-shifting approach involves harnessing the insights from a pre-trained Deep

Neural Network (DNN) and then repurposing this foundational knowledge for a distinct, yet often related, task [68]. The transformative impact of Transfer Learning is especially pronounced within the domain of image processing, where vast datasets and computationally intensive training regimens are the norm. By tapping into pre-trained models, practitioners can sidestep the hurdles of extensive training, making the technique not only computationally economical but also time-saving. To grasp how Transfer Learning works, it's important to first look at the structure of Convolutional Neural Networks (CNNs). The layers in a CNN work together like a coordinated symphony, where each layer has its specific function:

The initial layers in a CNN serve as foundational elements, functioning like attentive guards. They focus on identifying basic visual components such as textures, colors, and edges. Since these simple patterns are common across various visual datasets, the knowledge gained in these layers is broadly applicable. This is similar to learning the basic grammar of a language, which can then be used in a wide array of conversations and contexts.

In a CNN, as we move to the deeper layers, the complexity increases. The advanced layers are tasked with interpreting more complex patterns and learning attributes specific to the dataset they are trained on. If the initial layers are like learning the basic grammar of a language, the advanced layers are equivalent to understanding idiomatic expressions and nuanced phrases that are unique to specific dialects or contexts.

Transfer Learning excels in its ability to effectively combine different layers of a CNN. It recognizes that the knowledge held in the initial layers of a pre-trained CNN is widely useful. In Transfer Learning, these initial layers are usually left unchanged, or

"frozen." This means the broad understanding of generic visual patterns these layers have developed is kept intact, ready for new tasks. However, for the model to effectively work with a new dataset's unique characteristics, some adaptability is necessary. This is where "fine-tuning" comes in. By "unfreezing" the top, more advanced layers and training them further, the model is fine-tuned. This process recalibrates the model to better align with the specifics of the new dataset. During this phase, the model fine-tunes its advanced layers to better capture the unique attributes of the new dataset [68]. This iterative process of adaptation ensures that while the model benefits from the foundational knowledge of the pre-trained layers, it also evolves to meet the specific demands of the new task. In a broader perspective, Transfer Learning is emblematic of a paradigm where knowledge is not siloed but is fluid, transferable, and evolutionary. By leveraging prior knowledge, it enables swift model development, potentially improving performance metrics and providing a practical solution, particularly in situations where data may be limited or computational resources are constrained. In the ever-changing field of deep learning, Transfer Learning serves as evidence of the strength of accumulated knowledge and continuous learning.

### 3.2.3 Spatial Attention Module

The Spatial Attention Module [69] is an essential part of deep neural networks, especially in the field of computer vision. Its main role is to focus on the most important spatial areas within a feature map. Feature maps are complex graphical representations created by convolutional layers during image processing tasks. They display a range of visual patterns and characteristics found in the input data, capturing the subtle visual details of the input. The key strength of the Spatial Attention Module lies in its ability to understand the relationships between different spatial areas within a feature map. This helps in emphasizing and processing the most relevant information from the input. By

understanding how different spatial locations within the feature map interrelate, the module can identify and prioritize regions abundant with pertinent information while de-emphasizing less relevant areas. To accomplish this, the Spatial Attention Module employs global pooling operations, typically global max pooling or global average pooling. These operations are designed to condense and summarize the spatial information present in each channel of the feature map. Once condensed, this summarized spatial data is processed through one or more fully connected layers, which then generate spatial-wise attention weights. These weights are indicative of the relative importance of each spatial location in the feature map. Utilizing these attention weights, the Spatial Attention Module modulates the original feature map. Through operations like addition or element-wise multiplication, the module can enhance critical regions and diminish those of lesser importance. This adjustment ensures the neural network hones its attention on key visual patterns, enhancing accuracy in tasks such as object localization and refining the overall representation of features. The integration of the Spatial Attention Module into neural networks offers significant advantages. In domains such as object detection, visual question answering, and image segmentation, pinpointing crucial spatial regions becomes invaluable. By enhancing the neural network's ability to discern and utilize spatial data, the module amplifies the discriminative capabilities of the network. Furthermore, the Spatial Attention Module aids the network in identifying meaningful relationships across the input data, leading to more robust and insightful feature representations. For a more visual representation of the Spatial Attention Module's operations, reference is made to Figure 3.2 This figure delineates the process flow of the module, starting with the F channel undergoing both maximum and

average pooling. The pooled outputs are then fused and undergo a convolutional operation. Following a sigmoid activation, the resultant is the spatial attention map, labeled as Ms.

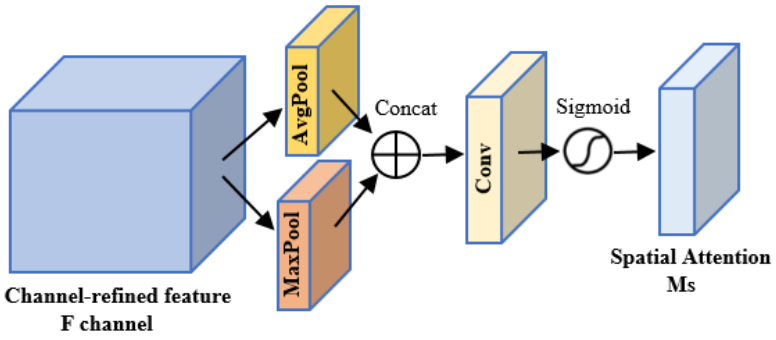


Figure 3.2: The structure of the Spatial Attention Module. [69]

### 3.2.4 Capsule Network

In the constantly evolving field of neural networks, Capsule Networks have emerged as an innovative development. They were created to address the limitations of Convolutional Neural Networks (CNNs) regarding the understanding of spatial hierarchies in visual data. Capsule Networks bring a more detailed and refined approach to interpreting visual information. A key feature of these networks is the use of "capsules," which are groups of neurons working together. Each capsule is designed to carefully analyze and understand specific details and characteristics present in visual objects.

To appreciate the full depth and breadth of Capsule Networks, it's essential to unpack their core components:

- **Capsules**: As foundational pillars of Capsule Networks, capsules are tasked with the responsibility of encoding diverse object attributes. Their spectrum of detection is vast, encompassing concrete traits like an object's size, hue,

and texture to more conceptual elements like its spatial orientation and relative positioning in a visual frame. The sophistication of capsules is further accentuated by the presence of an activation vector in each of them. This vector not only serves as a beacon, signaling the detection of a feature but also delineates the intricate properties and specifics of the detected feature, painting a comprehensive picture of the visual input.

- **Dynamic Routing**: Breaking away from conventional neural network processes, Capsule Networks introduce dynamic routing, a mechanism that promotes synergistic interactions among capsules. This inter-capsule collaboration is no mere exchange; it's a meticulous process that enables capsules to converge on a shared understanding of more complex, high-level features. This is achieved by leveraging the insights and consensus drawn from the contributions of their lower-level brethren. The beauty of dynamic routing lies in its facilitation of the network's capacity to intuitively understand and map out both hierarchical relationships and intricate spatial hierarchies, leading to a more refined and efficient learning trajectory.
- **Capsule Voting**: Seamlessly interwoven with the dynamic routing process is the concept of capsule voting. Here, capsules engage in a democratic process, casting votes to determine the activation of their senior, higher-level counterparts. This voting isn't arbitrary; it's predicated on the coherence between predicted poses. This meticulous voting process ensures that the Capsule Network maintains its resilience and accuracy, even when faced with

challenges like diverse poses or potential deformations in the visual data presented to it.

While the introduction of Capsule Networks into the neural network landscape holds significant promise, particularly with their enhanced robustness and potential for greater interpretability, their journey is still in its early chapters. Tapping into their full potential remains a fervent area of exploration. The effectiveness and ability of Capsule Networks are intricately tied to various factors, encompassing the finer details of the network's architectural design, the precision in hyperparameter tuning, and the judicious choice of training techniques befitting the specific tasks at hand. The forefront of current research is ardently focused on refining and optimizing Capsule Networks, with aspirations to deploy their advanced capabilities across a vast array of machine learning and computer vision domains.

## 3.3 Proposed Model

The model I propose is structured around two fundamental phases: preprocessing and a combined process of feature extraction and classification. In the preprocessing phase, I treat all texture images in a three-step approach. Firstly, data augmentation techniques are applied to enhance the dataset's diversity. Following this, these images are resized to maintain consistent $n \times n$ dimensions, ensuring uniformity in processing. Lastly, normalization is performed, which helps in scaling the input data and making the model less sensitive to the scale of features. Transitioning to the feature extraction phase, my approach is rooted in drawing deep features by integrating a spatial attention module with

pre-trained models, namely DensNet201 and InceptionV3. The core of transfer learning involves using pre-trained models to extract relevant features without having to begin the learning process from the beginning. In this architecture, the convolutional neural network (CNN) models work together with the spatial attention module. They act as robust feature extractors that are particularly adept at discerning the subtle and complex patterns as well as spatial relationships found in fabric images. It's important to emphasize that the overarching strategy based on transfer learning plays a crucial role in ensuring the accurate and efficient extraction of these deep features. As a result, this significantly enhances the accuracy of the fabric defect classification process. To provide a comprehensive understanding of the proposed model, Figure 3.3 provides a visual representation that encapsulates the flow and interrelation of its various components.

### 3.3.1 Preprocessing Phase

In the preprocessing phase, deep learning techniques are known for their accuracy, a recognition largely due to the extensive datasets they handle [70]. Yet, a challenge arises when faced with inadequate data. In such circumstances, there's a heightened risk of overfitting, which can, in turn, compromise the model's accuracy in real-world scenarios. To counteract this potential pitfall, data augmentation techniques are often introduced. In the context of this research, the foundational dataset comprised seven distinct balanced classes and totaled 349 images. Recognizing the limitations of such a dataset, I turned to data augmentation as a strategic solution. Techniques employed encompassed horizontal and vertical image flips and rotations across various degrees. The primary advantage of these techniques lies in their ability to preserve critical image attributes without

compromising the integrity of the data. Thus, they serve as a valuable remedy to the challenge posed by limited datasets. Following the augmentation, the data undergoes several crucial preprocessing steps before being ingested by the model. Firstly, dimension reduction is employed, wherein all images are resized to a more manageable size of 224 × 224 pixels. Subsequent to this resizing, normalization is implemented. This process recalibrates the pixel values, centering them around zero and scaling to achieve a standard deviation of one. Such an approach ensures a standardized dataset, promoting data consistency and enhancing comparability - factors instrumental in optimizing the performance of deep learning models during their training phase and amplifying their resilience. The final step in the preprocessing phase involves dividing the data into two crucial subsets: training and testing datasets. The former, as the name suggests, is employed for the primary training of the model. In contrast, the latter is set aside for the evaluation phase and contains data previously unseen by the model. It's noteworthy that both datasets encompass a mix of all image classes and are presented in a randomized sequence, ensuring comprehensive representation and unbiased evaluation.

### 3.3.2 Feature Extraction and Classification Phase

During the feature extraction and classification phase, intricate image classification methods, such as CNNs and their advanced variations, predominantly rely on expansive datasets with labeled training data. This reliance ensures the fine-tuning of parameters, thereby elevating their overall performance. Nevertheless, manually labeling a large dataset is a labor-intensive task. While increasing the depth of the network can potentially improve recognition abilities, it also extends training times due to the larger number of parameters that need to be optimized. A notable challenge encountered with the traditional CapsNet is its limited proficiency in defect detection accuracy. CapsNet, employing just a

single convolution layer for feature extraction, often misses out on capturing the deep multiscale features intrinsic to complex fabric images. Recognizing this limitation, I created a model specifically for fabric defect classification. The goal was to overcome the limitations of conventional methods. To do this, my approach harnesses the capabilities of both the InceptionV3 and DenseNet201 models. When coupled with the Spatial Attention Module, these models demonstrate enhanced ability in extracting features from texture images. InceptionV3 [71] and Densnet201 [72] adeptly extract hierarchical features, revealing the intricate details within fabric images. Concurrently, the Spatial Attention Module accentuates this extraction process by homing in on pivotal spatial regions that signify fabric anomalies. This collaboration between InceptionV3, Densnet201, and the Spatial Attention Module results in a robust feature extraction. Consequently, the proposed model stands out in delivering accurate and reliable defect classification. The next step involves classification, where the extracted features are passed into the Capsule Network. Capsule Networks are known for their dynamic routing mechanism, which enables connections between capsules across different layers. In this dynamic paradigm, upper-layer capsules concur on the presence and nuances of high-level visual attributes. This consensus is predicated on the harmonious agreement amongst the capsules in the foundational layers. Such an arrangement arms the CapsNet with the capability to astutely recognize intricate patterns while adeptly managing variations in positioning and perspectives of fabric defects. During the classification phase, the Capsule Network refines its comprehension of fabric anomalies. It does so by tapping into the hierarchical representations embedded within the capsule layers. As it undergoes training, the network meticulously adjusts its parameters, striving for pinpoint accuracy across a spectrum of

defect classes. The culmination of this process sees the Capsule Network offering a probabilistic prediction for each defect category. Classification is then determined based on the class with the preeminent probability score.

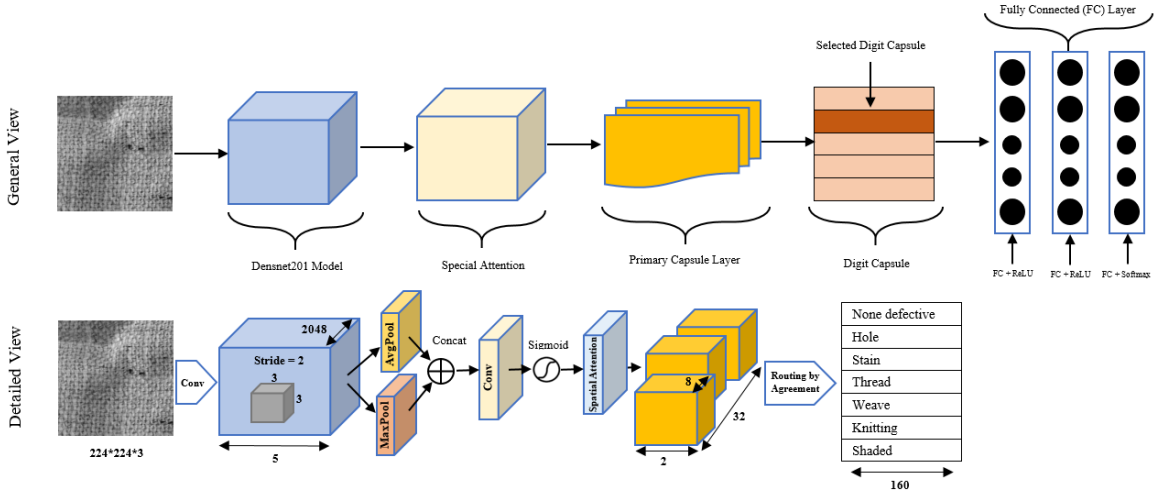


Figure 3.3: An overview of the proposed model.

## 3.4 Experimental Work and Results

I conducted experimental tests on the model I proposed using the TensorFlow framework, which served as the computational foundation. These tests were carried out on a computing machine equipped with an NVIDIA 80 GB GPU card and a substantial 90 GB of RAM. The machine's computational infrastructure was situated on the Paperspace platform. Figure 3.4 illustrates the 10-fold cross-validation process, highlighting the individual fold performances (Ki values) and the cumulative assessment (Kort).

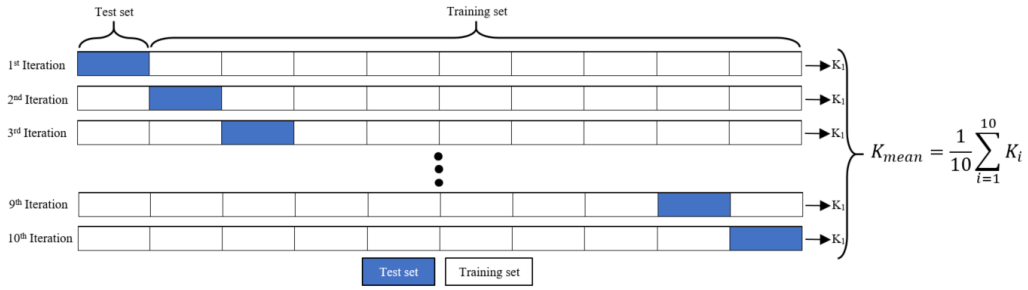


Figure 3.4: The diagram depicts the process of ten-fold cross-validation. In this approach, the dataset is divided into ten parts for assessing the model's precision. Nine parts are cyclically used for training, with one part reserved for testing. The mean value 'E' derived from the outcomes of the ten tests indicates the model's efficacy. This mean acts as the performance indicator for this K-fold cross-validation approach.

### 3.4.1 Dataset

The TILDA fabric dataset, a product of the DFG's Texture Analysis program, comprises eight categories of fabric defects for training, testing, and evaluating models. [73]. Notably, this segmentation also includes a class devoted to items devoid of any defects. Each of these categories contains a collection of 50 images, all of which are in TIFF format. The standard dimensions for these images stand at 768 x 512 pixels, and each image is represented in a grey-level resolution to capture the intricate details of fabric textures. A selection of sample images from these categories can be seen in Figure 3.5. For my research, I strategically used a subset of the TILDA dataset, resulting in a collection of 349 images. These images then underwent classification, resulting in a division into seven distinct categories. These categories, tailored to the research focus, were: No defects, Holes/Cuts, Stains/Colors, Thread issues, Foreign objects, Wrinkles, and Lighting changes. This categorization provided a robust foundation for the model training and evaluations.

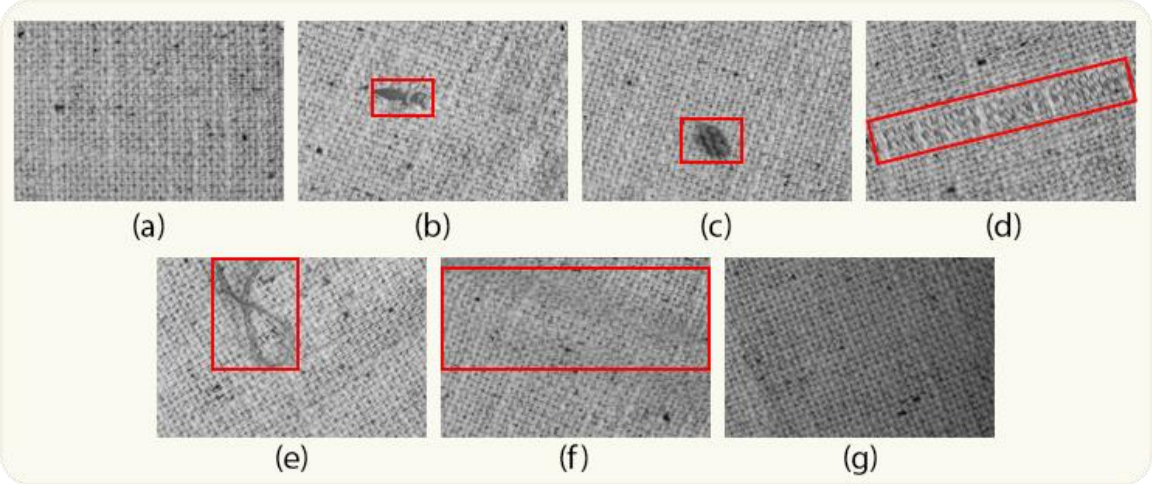


Figure 3.5: Depictions of various defect categories: (a) Flawless (b) Perforations/Cuts (c) Discolorations/Spots (d) Yarn anomalies (e) Extraneous items (f) Creases (g) Alterations in illumination.

### 3.4.2 Evaluation Metrics

To comprehensively understand the effectiveness and reliability of the classification system within the proposed model, I incorporated several key performance metrics. The chosen metrics, which include accuracy, precision, recall, specificity, and the F1 score, are used to evaluate the model from different perspectives:

- **Accuracy**: This metric serves as a benchmark of the model's overall performance, indicating the fraction of instances that the model correctly classified out of the total instances.
- **Precision**: Precision provides insight into the model's capability in classifying defects correctly. It quantifies the proportion of true defects among all the items that the model identifies as defects.

- **Recall (or Sensitivity**): This metric, in contrast to precision, assesses the model's proficiency in detecting actual defects. It measures the fraction of real defects that the model correctly identifies.
- **Specificity**: Given the multiple defect classes, specificity becomes essential. It evaluates the model's effectiveness in correctly classifying the non-defective items, indicating the proportion of correctly identified non-defective samples.
- **F1 Score**: The F1 score, also known as the F-measure, strikes a balance between precision and recall. It is the harmonic mean of these two metrics, ensuring a unified evaluation that emphasizes both defect detection and correct categorization.

The respective equations for these metrics are detailed below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

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

$$\text{Specifity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

$$\text{F1 score} = 2 \times \text{Precision} \times \frac{\text{Recall}}{\text{Recall} + \text{Precision}} \tag{5}$$

To further clarify, the equations for these metrics employ specific terminologies derived from the outcomes of the classification process:

- **True Positive (TP)**: This term denotes defects that were correctly classified by the model.

- **False Positive (FP)**: This represents instances where the model incorrectly identified a non-defect as a defect.
- **False Negative (FN)**: This term covers defects that the model missed or incorrectly classified as non-defective.
- **True Negative (TN)**: This indicates non-defects that were correctly recognized and excluded from the defect categories by the model.

These classifications form the foundation for equations (1)–(4). Importantly, for the evaluations, the number of predictions made by the model that matched the true labels was found to be equivalent to the count of test samples that were correctly labeled. This synchronization is pivotal to ensure the validity of the evaluation.

### 3.4.3 Implementation Details and Training

In this part, I explain how the model was built, focusing on the special settings, called hyperparameters, used during training. Choosing the right hyperparameters is very important because they greatly affect how well the model works. Since the best hyperparameters can vary depending on the dataset, it's crucial to adjust them specifically for each dataset to get the best results. Given the distinctive architecture and functionality of Capsule Networks (CapsNets), they have specialized methodologies for fine-tuning these hyperparameters, which come into play during the training phase. For the model's activation function, I incorporated the Softmax activation, well-suited for multi-class classification problems. I opted for a batch size of 8 instances during training, striking a balance between computational efficiency and gradient accuracy. The entire training process extended across 150 epochs, and for optimization, I employed the widely used Adam algorithm, which offers adaptive learning rates for individual parameters.

Further technical details of the model include:

- **Kernel Size**: I built the model using a kernel size of 3x3, a common choice that enables the model to capture spatial patterns efficiently.
- **Capsule Routing Time**: The process by which information is passed between capsules. For the model, I set this to 3 iterations to ensure adequate information routing without causing excessive computation.
- **Capsule Dimensions**: Each individual capsule within the network was designed to have eight dimensions, providing a richer representation of data.
- **Length of Capsules**: Both the primary capsules and the digit capsules were established with a length of seven. This determines the depth and complexity of information each capsule can hold.

Lastly, I made use of the ReduceLROnPlateau [74] callback function during the training process. This function is instrumental in adjusting the learning rate, a pivotal hyperparameter, based on the model's performance. Specifically, if the model doesn't register an improvement in its loss value over a pre-defined number of epochs (termed 'patience'), the learning rate is multiplied by a certain factor to facilitate convergence. For this study, I set this factor to 0.2 and the patience to 10 epochs, ensuring that the learning rate is modified adaptively to foster efficient and effective training.

### 3.4.4 Experimental Results

In the comprehensive exploration centered around fabric texture defect classification, the primary focus was the TILDA dataset, a collection of fabric textures showcasing varied defects. The ambition was to identify the Convolutional Neural Network (CNN) model that resonates best with the peculiarities of the TILDA dataset and delivers unmatched

performance in classifying fabric defects. To ensure the fidelity of my experiment, I employed the 10-fold cross-validation strategy during training. This approach ensured that I could confidently assess each model's adaptability to different data divisions. DenseNet201 emerged as the frontrunner, achieving an accuracy of 95.7% paired with an F1-score of 95.6%. This performance highlights its ability to strike a balance between precision and recall. Another contender, ResNet50, also demonstrated capabilities, achieving an accuracy of 93.7%. Although it delivered high accuracy, its F1-score slightly trailed DenseNet201. On the other hand, the Xception model, known for its distinctive architecture, performed well, achieving an accuracy of 93.1% and a closely matched F1-score. Slightly behind in the ranking was ResNet152V2, which, while effective, couldn't surpass the performance of its counterparts in this classification challenge. EfficientNetV2B0 and InceptionV3 delivered similar performances, both achieving approximately 90% accuracy. This close match in their performance underscores the idea that, despite having different architectural foundations, models can exhibit similar behavior when applied to a particular dataset. On the other end of the spectrum, MobileNet and EfficientNetB0, known for their computational efficiency, demonstrated respectable performances, with accuracies just of 89%. A consistent thread running through the observations was the perfect specificity of 100% that was mirrored across all models. This unanimous result is a testament to the TILDA dataset's clarity of features, which provides an environment conducive for models to make clear distinctions between defective and non-defective textures. In essence, the journey through this experimental landscape underscores the brilliance of DenseNet201 in navigating the intricacies of the TILDA dataset. The varied performance metrics across the models emphasize the pivotal role the

right architecture plays in harmonizing with a dataset's idiosyncrasies, ensuring peak classification efficacy.

| Model | Accuracy | Specificity | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| Xception [75] | 93.11 | 98.00 | 94.94 | 93.14 | 92.92 |
| EfficientNetV2B0 [76] | 89.97 | 100 | 93.29 | 89.92 | 90.02 |
| MobileNet [77] | 88.80 | 100 | 92.35 | 88.78 | 88.88 |
| ResNet152V2 [78] | 91.10 | 100 | 93.43 | 91.07 | 90.72 |
| EfficientNetB0 [79] | 88.24 | 100 | 91.89 | 88.21 | 88.14 |
| ResNet50 [78] | 93.69 | 100 | 95.45 | 93.64 | 93.63 |
| InceptionV3 [71] | 89.68 | 100 | 92.21 | 89.64 | 89.73 |
| DenseNet201 [72] | 95.70 | 100 | 96.40 | 95.71 | 95.62 |

Table 3.1: Performance of the Common CNN models.

In my subsequent study, I aimed to boost the defect identification in textures by adopting a new architectural method known as capsule networks. These networks excel at maintaining the structural order of features, making them apt for my purpose. I incorporated the capsule network architecture with the earlier discussed CNN models and retrained them on the dataset. As shown in Table 3.1, the outcomes were very promising. Impressively, all models showed marked enhancements in various metrics. For instance, the DenseNet201, already a top-performer in the initial study, increased its accuracy from 95.7% to a remarkable 98.2%. Its F1-score, crucial for the study, also rose to 98.2%. Importantly, all models maintained a 100% specificity, indicating their consistent skill at correctly recognizing non-defective textures. Moreover, precision and recall of each model grew considerably upon adding the capsule network. For example, the InceptionV3 model improved its accuracy from 89.6% to 97.1%, with its F1-score also at 97.1%. This uplift was not restricted to just one model but was seen universally across all tested models, underscoring the effectiveness of the capsule network in this domain. The structure of the capsule network, which is adept at interpreting spatial relationships in data, seems to work

harmoniously with the CNNs' feature-detection ability. This synergy is evident in models like MobileNet, which saw an accuracy increase from 88.8% to 92.5%. Notably, DenseNet201 and InceptionV3 models stood out. With F1-scores of 98.2% and 97.1%, respectively, they demonstrate their capability to accurately identify defective textures with minimal errors. This indicates that when these models are combined with capsule networks, they might set new standards in texture defect identification. To conclude, merging capsule network architecture with the selected CNN models resulted in a significant enhancement in performance. The consistent improvement in accuracy, precision, recall, and F1-score across all models, without compromising specificity, speaks volumes about the value of this approach.

| Model | Accuracy | Specificity | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| Xception [75] | 95.68 | 100 | 96.42 | 95.71 | 95.61 |
| EfficientNetV2B0 [76] | 95.70 | 100 | 96.51 | 95.71 | 95.73 |
| MobileNet [77] | 92.53 | 100 | 94.60 | 92.50 | 92.50 |
| ResNet152V2 [78] | 94.52 | 100 | 95.76 | 94.50 | 94.22 |
| EfficientNetB0 [79] | 92.25 | 100 | 94.76 | 92.21 | 92.29 |
| ResNet50 [78] | 95.41 | 100 | 96.56 | 95.42 | 95.33 |
| InceptionV3 [71] | 97.11 | 100 | 97.75 | 97.07 | 97.13 |
| DenseNet201 [72] | 98.28 | 100 | 98.52 | 98.28 | 98.27 |

Table 3.2: Performance of the Common CNN models as backbone of Capsule Network.

In the final phase of the experimental series, I sought to further boost the model's efficiency by adding a spatial attention module to the system. This type of module is created to assist the model in focusing on the most important areas of an image, which can potentially improve its capacity to detect defects within textures. For the purpose of this study, I combined this module with the InceptionV3 model. This model had already shown impressive results in the prior assessments. After combining the spatial attention module with the InceptionV3, I saw notable improvements in its performance. The model's

accuracy shot up to an impressive 98.28%. Additionally, its specificity stood at 100%, while precision climbed to 98.64%. I recorded a recall rate of 98.21%, leading to an F1-score of 98.24%. These figures indicate that the spatial attention module played a pivotal role in directing the model's attention to significant patterns and features within the texture images. Driven by these outstanding outcomes, I aimed higher for the last experiment. I decided to adopt a combined approach, utilizing the power of two distinct models - DenseNet201 and InceptionV3. I theorized that merging these two potent models would create a network that benefits from the strengths of both, potentially resulting in even more dependable and precise classifications. The theory turned out to be accurate. When I combined DenseNet201 and InceptionV3, the result was enhanced. This model set a new gold standard for identifying defects in textures, boasting an accuracy of 99.42% and preserving the perfect 100% specificity. Furthermore, it achieved a precision rate of 99.52% and a recall of 99.36%, leading to an F1-score of 99.38%. For a visual representation, Figure 3.6 provides a confusion matrix for this model using the TILDA dataset.

To wrap it up, the last experiment, which involved the use of a spatial attention module, brought about significant advancements, especially when combined with the InceptionV3 model. The ultimate achievement, a combined model of DenseNet201 and InceptionV3, set new industry standards for classifying texture defects. This highlights the vast potential in merging neural network designs and techniques to address intricate image classification challenges.

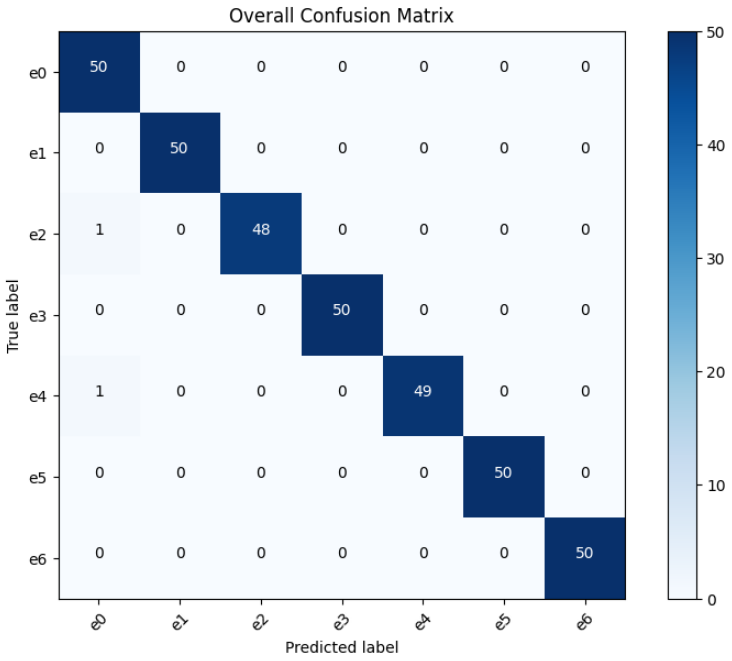


Figure 3.6: Confusion matrices of the proposed model.

## 3.5 Discussion

### 3.5.1 Comparison with state of the art CNN models

In the realm of texture defect classification, the accuracy of the model is a crucial determinant of its effectiveness. This emphasis on accuracy has driven us to conduct a thorough analysis, comparing the newly proposed model to a selection of well-established standalone CNN models. This comparison is detailed in Table 3.1. The accuracies achieved by these standalone CNN models, as recorded in Table 3.1, vary notably. The range begins with MobileNet, which achieves an accuracy of 88.21%, and extends up to 95.71% achieved by DenseNet201. These figures serve as a critical benchmark, allowing us to assess the effectiveness of different models and, more importantly, to measure the performance of the new approach. The innovative model, which synergistically combines the strengths of DenseNet201 and InceptionV3, pushes the boundaries of this benchmark. It achieves an accuracy of 99.42%, marking a significant improvement - a boost of 3.71

percentage points - over the previous best result from DenseNet201. For a more intuitive understanding of this comparative performance, I direct attention to Figure 3.7. Here, since I evaluated the model a on balanced dataset only the accuracy is compared against the accuracies of the standalone CNN models. In this graphical representation, the X-axis denotes the different model architectures, while the Y-axis displays the corresponding accuracy percentages. A cursory look at Figure 3.7 reveals that the proposed model surpasses all other individual CNN models in accuracy, cementing its position as a solution in the field.

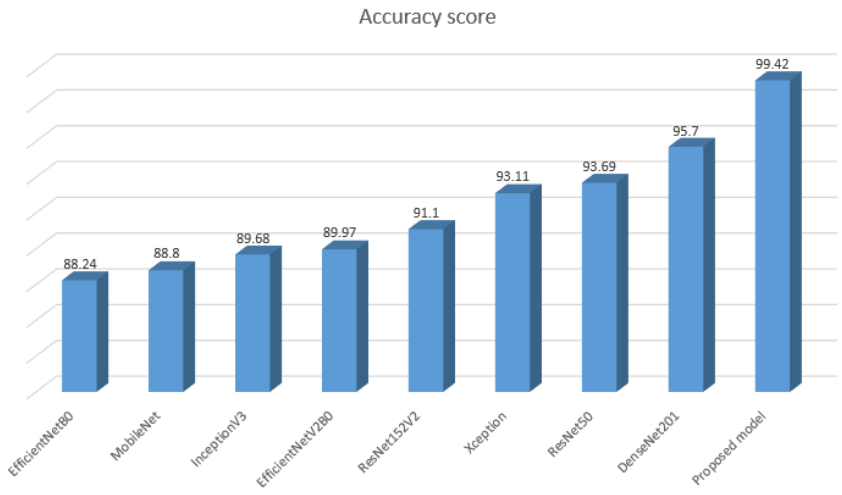


Figure 3.7: Comparative analysis of the accuracy percentage between the proposed model and various pre-trained CNN models on TILDA dataset.

### 3.5.2 Comparison with the Previous Studies

Texture classification is a field that has been extensively studied in the literature, as evidenced by the numerous research papers and studies available on the topic. A notable number of these studies have based their research on datasets crafted by the researchers themselves. Yet, it's the TILDA dataset, renowned for its diverse assortment of fabric defect textures, that has captured the interest of numerous scholars. To offer an objective

evaluation of the various studies, I can turn to Table 3.3, which showcases accuracy scores of different methods when applied to the TILDA dataset. This table also facilitates a direct comparison of the accuracy scores between the model introduced in the present study and those from prior research that employed the TILDA dataset for examination. The spectrum of methodologies in texture defect detection research is vast. Earlier approaches, highlighted by studies like those from references [80], [81], and [82], were built around feature extraction techniques. These include methods such as the Gray-Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP), among others. They also leveraged various classification techniques, including Support Vector Machines (SVM), Neural Networks, and the like. Of these classic methodologies, the study by [81] in 2020 is particularly noteworthy, achieving an accuracy of 97.25%—the pinnacle among traditional techniques. However, the landscape of texture classification research has been transformed with the emergence of pre-trained deep learning models. This new era, exemplified by studies like the one from [83] in 2019, has adopted advanced models, with AlexNet being a prime example. Delving deep into this research evolution, a discernible pattern emerges: deep learning models seem to consistently eclipse traditional methods when it comes to texture classification tasks.

| Reference | Year | Method | Accuracy Score |
|---|---|---|---|
| [80] | 2011 | Gray-Level Co-occurrence Matrix, Local Binary Patterns, and Support Vector Machines | 86.7% |
| [82] | 2019 | Random Decision Forest, Gray-Level Co-occurrence Matrix and Gabor Wavelet | 84.5% |
| [83] | 2019 | Deep CNN utilizing Multi-scaling and based on AlexNet | 96.55% |
| [81] | 2020 | Support Vector Machines, Local Binary Patterns, Gray-Level Co-occurrence Matrix and SVM | 97.25% |

| | | | |
|---|---|---|---|
| **Proposed model** | 2023 | Capsule network, spatial attention block | **99.42**% |

Table 3.3: Comparison of the proposed model's results with previous studies in the literature.

## 3.5 Conclusion

In textile manufacturing, quality control is essential to ensure that the fabrics produced meet the highest quality standards. This research article introduces a texture defect classification mechanism, based on a capsule-based neural network model. Such a design takes into account the nuanced challenges associated with fabric anomalies, which typically manifest as erratic patterns over intricate backgrounds. I recognized the limitations of conventional Capsule Networks, which employ single convolutional layers for feature extraction. This arrangement could potentially limit their ability to detect subtle and complex fabric defects. To address this issue, I combined state-of-the-art convolutional neural networks (CNNs) with a spatial attention component, all within the capsule architecture. This combination is dual-pronged in its benefits: Firstly, it taps into the merits of transfer learning, augmenting the model's learning and generalizing capabilities. Secondly, it provides the model with an improved feature discernment capability, which is essential for detecting the subtle patterns and spatial intricacies present in fabric images. For assessment purposes, the model underwent a 10-fold cross-validation on the multi-category TILDA dataset. The derived outcomes were commendable, with the model showcasing a accuracy of 99.42%. This research not only emphasizes the outstanding efficiency of the proposed model but also underlines its consistent superiority when juxtaposed with other methodologies.

# Chapter 4. DepthCrackNet: Pavement Crack Segmentation Model Using 3D Spatial Features and a Multi-Head Attention Mechanism

## 4.1 Introduction and Problems

The condition of road surfaces is a crucial aspect of traffic safety, and it can be jeopardized by the presence of cracks. The formation of these cracks can be attributed to various factors, including moisture content, the quality of road construction, and the volume of traffic it accommodates [84]. An investigation in 2006 illuminated the economic repercussions of accidents in the U.S., which were precipitated by inferior road conditions, amounting to a staggering $217.5 billion [85]. Neglecting timely repair of these cracks can exacerbate the damage, endangering road users, reducing road lifespan, and potentially resulting in both human and material losses. With the surge in road usage, the risks multiply, and if unchecked, can culminate in tragic fatalities. Hence, it becomes imperative for the departments overseeing transportation maintenance to ensure roads are kept in optimum condition. A cornerstone of such endeavors is the detection of cracks. Traditional manual detection methods are fraught with issues: they are labor-intensive, disrupt regular traffic flow, consume a significant amount of time, and pose potential hazards to workers [86]. To streamline the inspection process and reduce the workload on professionals, the introduction of automated crack detection systems is essential.

With the advancements in computer vision, there has been a growing interest in leveraging these technologies for the automation of crack detection [87], [88]. However,

designing an efficient model tailored for identifying pavement cracks isn't devoid of challenges. Figure 4.1 delineates three primary obstacles that such models are likely to confront:

- **Variations of Cracks**: Cracks aren't uniform. They manifest in a multitude of forms - differing in length, breadth, orientation, and curvature. This diversity complicates the development of a universally applicable segmentation strategy.
- **Different Pavement Types**: The material composition of pavements, such as asphalt or concrete, introduces variability in textures and crack morphologies. This demands flexible and adaptable segmentation methodologies.
- **Assorted Objects and Irregularities**: Images of road surfaces often feature miscellaneous items and anomalies like lane markings, depressions, inadvertent paint splatters, variable lighting effects, skid marks, and random debris. Some of these can mimic the appearance of cracks in terms of their shape, dimension, or texture, which introduces the possibility of false detections or unclear interpretations.

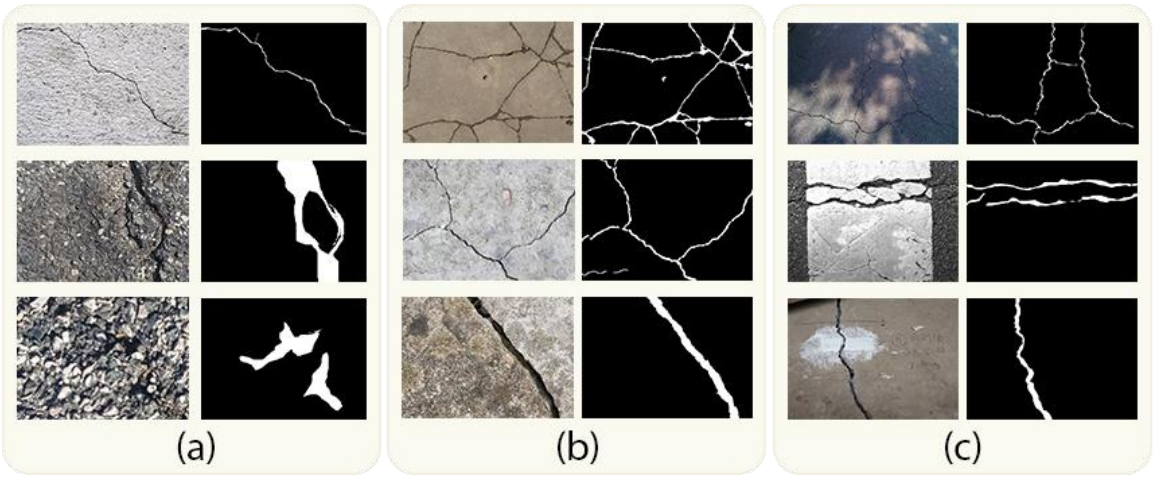


Figure 4.1: Example pictures used to identify cracks on pavement surfaces: (a) Variations of Cracks, (b) Different Pavement Types, (c) Assorted Objects and Irregularities.

The advancement of deep learning and improvements in image processing have accelerated the development of numerous automated methods for identifying cracks in pavements. In the foundational stages of this exploration, researchers like [86] and [89] favored threshold-based techniques, premised on the notion that pixels representative of cracks were perceptibly darker than their immediate surroundings. A myriad of features, encompassing wavelet characteristics [30], Histogram of Oriented Gradients (HOG) [35], and Gabor filters [90], were integrated for the detection of cracks. These methodologies, while adept at isolating the immediate attributes of cracks, often missed out on understanding the wider context within which the crack exists. Aiming for a holistic perception of crack detection, various investigations [91], [92] incorporated an amalgamation of photometric and geometric elements inherent in pavement crack visuals. The overarching goal of these strategies was to meticulously diminish noise and accentuate the connectivity of the detected cracks. However, these models encountered hurdles, particularly when deciphering cracks with irregular intensities or complex topological

nuances. To overcome these challenges, CrackForest [93] melded multi-tiered complementary features, tapping into the rich structural details embedded within crack segments. This approach surpassed its peers, notably Minimal Path Selection (MPS) [94], Free-Form Anisotropy (FFA) [95], CrackIT [96], and CrackTree [92] in terms of efficacy. Yet, a notable limitation of CrackForest [93] lies in its dependency on manually designed features, which may falter when discerning cracks against intricate backdrops punctuated with nuanced indicators.

In the current wave of research, the adoption of deep learning paradigms has become widespread, particularly in computer vision projects. A plethora of studies [88], [41], [97], [98] aspired to exploit the robust feature extraction capabilities of deep learning tailored for crack detection. For instance, works like [88], [41], and [97] utilized deep learning to perform patch-oriented classification, a tactic efficient but somewhat unwieldy and sensitive to the size of the patches in question. On the other hand, [88] approached crack detection as a segmentation task, utilizing deep learning to categorize individual pixels as either part of cracks or background elements. Yet, as articulated by [41], the endeavor of crack detection possesses inherent complexities, different from standard semantic segmentation, especially when considering the stark contrast between the primary subject (cracks) and the surrounding environment. To counter the intricate challenges of securing robust feature representations and navigating the distinct class imbalances inherent in automated crack detection, I introduce DepthCrackNet, an innovative model meticulously designed to autonomously detect cracks on pavements.

The core contributions of this research are delineated as follows:

- Incorporation of Double Convolutional Encoder (DCE): I integrate the Double Convolutional Encoder (DCE) into the segmentation framework. Constructed with successive convolution layers, the DCE is tailored to augment feature extraction capabilities while ensuring optimal utilization of parameters.
- Introduction of TriInput Multi-Head Spatial Attention (TMSA): This research introduces the TMSA mechanism, an attention module capable of simultaneously processing three input feature maps. By employing multi-head attention, it aims to extract deeper contextual understanding, enhancing the accuracy of segmentation.
- Adoption of the Spatial Depth Enhancer (SDE): The Spatial Depth Enhancer (SDE) module is another notable component of the model. It ingeniously transforms two-dimensional feature maps to a three-dimensional milieu, which serves to accentuate the model's depth discernment and spatial articulation.
- Superiority Demonstrated in Empirical Tests: The empirical evaluations, conducted using prominent crack datasets including Crack500, DeepCrack, and GAPS, consistently highlight the superiority of DepthCrackNet over contemporary state-of-the-art models in the field of crack detection.

Through this research, I mark a significant leap forward in the field of automated pavement crack detection.

## 4.2 Proposed Model

In the framework of my research, I conceptualize crack detection as an exercise in pixel-specific binary categorization. When presented with an input image that may contain a crack, the deep learning model aims to produce a predictive map that highlights the presence of cracks. In this generated map, areas identified as potentially containing cracks

are assigned higher probability scores, indicating a greater confidence in detecting real cracks in those areas. On the other hand, areas without cracks are indicated by lower probability scores, indicating a lower likelihood of actual cracks being present. A detailed representation of the model's architecture is presented in Figure 4.2. As shown in Figure 4.2, the model is designed in a similar fashion to the U-Net framework. The encoder segment integrates a Double Convolution Encoder (DCE) module. This element encompasses a series of 2D convolutions, followed by batch normalization (BN) and ReLU activation layers, all meticulously crafted to optimally extract features from pavement-centric images. To enhance this extraction, we've incorporated the Spatial Depth Enhancer (SDE) module within the encoder. This module employs a 3D convolution technique on the images, refining the model's feature identification capabilities. Diving into the decoder portion, we've embedded the TriInput Multi-Head Spatial Attention (TMSA) module, a mechanism responsible for fusing diverse feature maps. It's imperative to note that each attention head within this module functions autonomously, encompassing a spectrum of spatial interrelationships. The fruits of this intricate process are subsequently funneled to the Convolution Transpose Decoder (CTD). This decoder, built from 2D transpose convolution, batch normalization (BN), and ReLU layers, expands both the horizontal and vertical dimensions of the image. By harnessing these enriched feature sets, the CTD yields highly accurate crack identification outcomes.

In summary, the DepthCrackNet is architecturally rooted in four pivotal components. Each of these cornerstones -1) Double Convolution Encoder (DCE), 2) Spatial Depth Enhancer (SDE), 3) TriInput Multi-Head Spatial Attention (TMSA), and 4) Convolution Transpose Decoder (CTD) - will be dissected in depth in the ensuing sections of the study.

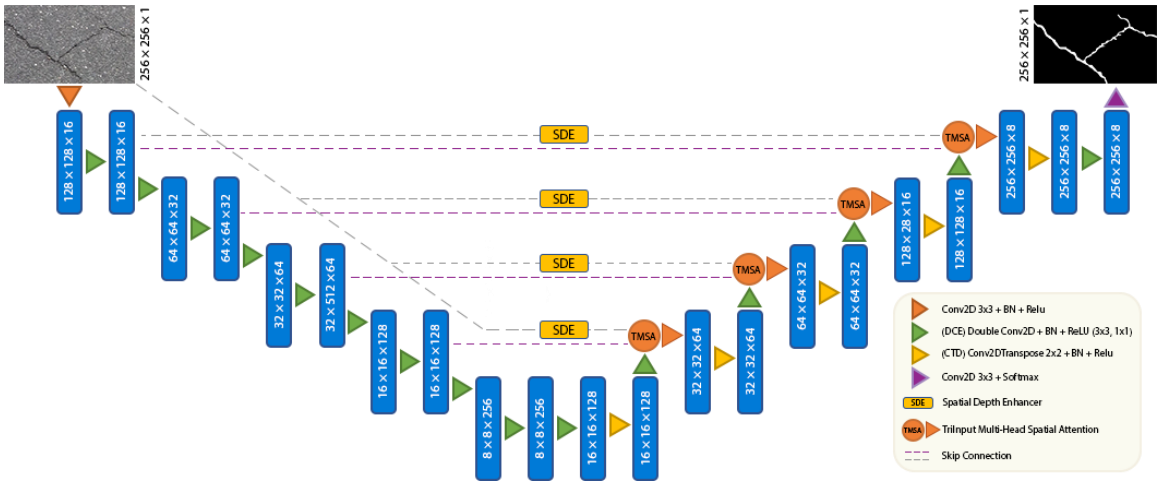


Figure 4.2: The architecture of the DepthCrackNet.

### 4.2.1 Double Convolution Encoder (DCE)

Traditional Convolutional Neural Network (CNN) architectures rely on a structured sequence of layers, with each layer carrying out operations like convolution, Rectified Linear Unit (ReLU) activation, and batch normalization to effectively extract important features from images. However, a significant challenge arises: as these networks delve deeper to uncover more intricate semantic details, they often encounter the well-known problem of vanishing gradients [78], [79], [99]. To overcome this challenge, my investigation introduces the Double Convolution Encoder (DCE). Drawing from the architectural of the Inception V3 model [71], the DCE incorporates dual convolutional layers, each armed with distinct filter configurations. This architectural decision is carefully designed to capture spatial details meticulously without significantly increasing the model's parameter numbers. Furthermore, this architectural configuration enables the Deep Crack Extractor (DCE) to effectively mitigate the challenges commonly associated with deep Convolutional Neural Networks (CNNs). This ensures robust feature extraction,

particularly when dealing with a restricted training dataset [70]. In a broad context, Convolutional Neural Networks (CNNs) are fundamentally built upon three key pillars: the convolutional layers, batch normalization procedures, and activation functions [100].

- **Convolutional Layer**

At the heart of a Convolutional Neural Network (CNN) lies the convolutional layer, a pivotal component that executes a convolution operation on the input data. This operation essentially acts as a specialized filter mechanism. A visual representation of this convolution operation can be observed in Figure 4.3 as the model undergoes the training process, it meticulously adjusts the filter weights. This adjustment endows the filters with the capability to discern and accentuate features that are of utmost relevance to the specific task in question. In the given context, the term 'w' denotes the weight, 'x' stands for the input data, 'b' is indicative of the bias value, and Noutput portrays the resultant output, all of which can be gleaned from Eq. (6).

$$N\ output = w \times x + b \tag{6}$$

- **Batch Normalization Layer**

Batch normalization stands as a pivotal mechanism in bolstering the consistency and efficacy of neural networks. At its core, this procedure entails the normalization of outputs originating from prior layers. Such normalization guarantees that the inputs directed to every ensuing layer maintain a uniform mean and variance. A significant merit of this process is its ability to mitigate the internal covariate shift, which subsequently facilitates a more rapid training cycle and lessens the reliance on initial weight setups. Moreover, the implementation of batch normalization often translates to more streamlined loss function terrains, which simplifies the overarching optimization journey.

- **Activation Layer**

Following the convolutional procedure, the data progresses to the activation layer. In this phase, a specific transformation is imposed upon the output data emanating from the preceding layer. This transformation introduces a non-linear dynamic into the model's computations. The Rectified Linear Activation Unit (ReLU) stands out as a prevalent activation function for this stage, and its primary action is to negate any negative values by relegating them to zero. While ReLU enjoys widespread usage, there are other alternatives like "tanh" and "sigmoid." These counterparts similarly operate by confining the input data within a defined boundary.

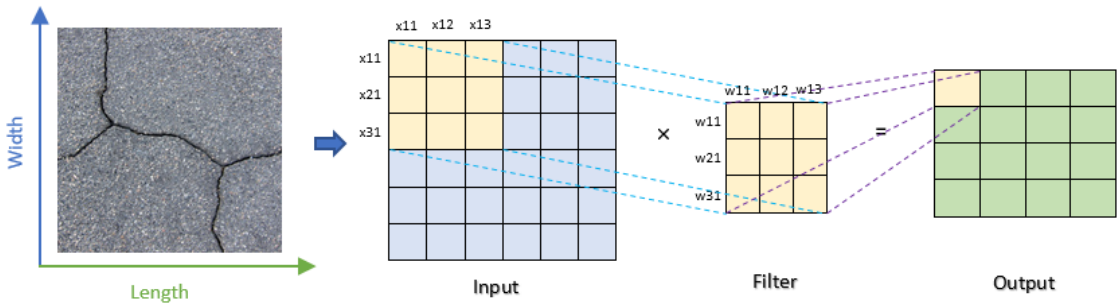


Figure 4.3: Illustration of the 2D convolution process in a convolutional layer.

### 4.2.2 Spatial Depth Enhancer (SDE)

Within the segmentation model's encoder section, we've incorporated the Spatial Depth Enhancer (SDE) to amplify feature extraction and emphasize the depth-oriented characteristics evident in pavement crack images. At its core, the SDE aims to provide a more profound spatial interpretation of features, all the while being computationally efficient. This is actualized by transitioning from the customary 2D convolution approach to the more intricate 3D convolution methodology. To materialize this, the initial image is subdivided into segments of N×N, which subsequently serve as an additional dimensional

layer. The resulting architecture exhibits a notable resemblance to the 3D biomedical imagery structure, as referenced in [101]. Upon entry, the SDE module promptly modifies the input tensor's dimensions to make them amenable to 3D convolutional activities. This strategic alteration primes the tensor for ensuing depth-centric transformations. Post-adjustment, the 3D convolution mechanism is set in motion. By infusing an extra depth dimension, the process is better poised to discern spatial structures and depth-pertinent traits, as compared to its 2D counterpart. Figure 4.4 offers an illustrative snapshot of the mechanics of a 3D convolution maneuver. It's noteworthy to mention that the convolution filter's depth is fluidly determined by the channels present in the input, ensuring a harmonious balance between adaptability and computational thriftiness. In the concluding stages, the tensor is reverted to a 2D framework after undergoing 3D convolution and subsequent activation. This transition ensures that the processed tensor meshes smoothly with ensuing model layers, specifically the TMSA module.

In summary, the Spatial Depth Extraction (SDE) module provides the segmentation model with an enhanced spatial understanding of the input image. This enhancement is especially crucial when the model is tasked with identifying subtle and intricate crack patterns on pavements, patterns that may go unnoticed by models limited to 2D convolutional layers.

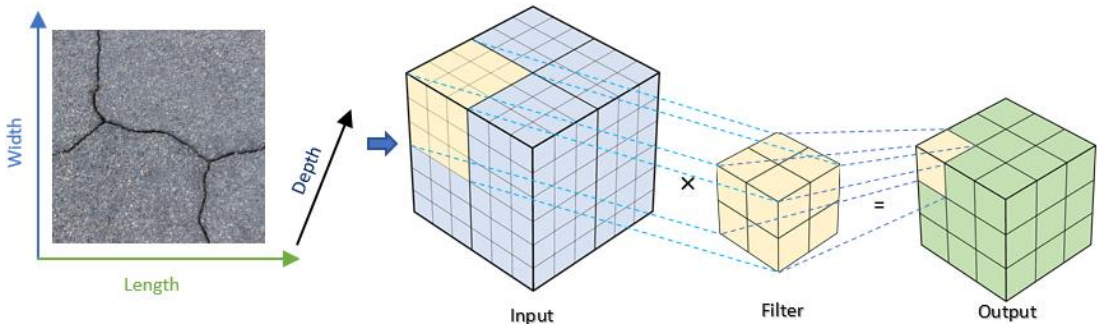

Figure 4.4: Illustration of a 3D convolution operation.

### 4.2.3 TriInput Multi-Head Spatial Attention (TMSA)

Attention mechanisms have provided a shift in the way deep learning architectures perceive and prioritize data [5], [102]. These mechanisms, rather than evenly distributing attention across all data points, enable the model to hone in on pertinent segments related to a task. The introduction of multi-head attention augments this by facilitating multiple focus points or 'perspectives' [103], [104], ensuring that a diverse range of spatial information is captured from various perspectives. In the domain of pavement crack detection via segmentation methodologies, the TriInput Multi-Head Spatial Attention (TMSA) stands out as a module crafted for the decoder segment. Intricacies like hue, contour, and texture are enmeshed within spatial nuances, while semantic details overflow with context-rich data—stellar for classification but often lacking in locational or morphological precision [105]. On the flip side, spectral aspects illuminate spatial ties across points in the input image, achieved via reshaping and 3D convolutional techniques, as delineated in [106]. Given that these facets are sourced via disparate channels, these maps inherently contain diverse content. This study introduces a novel approach where, instead of simply combining these maps, I design a 3D-input multi-head attention structure. The TMSA's blueprint is deeply rooted in the squeeze and excitation paradigm [57]. This module adeptly combines three distinct feature maps: the foundational feature maps extracted via double Conv2D layers (DCE) and transported through the encoder's skip pathways; the spatially-rich maps from the Spatial Depth Enhancer (SDE) that underscore depth-oriented subtleties; and maps from the Conv2DTranspose (CDC) module in the decoder, with a focus on recovering spatial details that may have been lost during the encoding process.

The core of the TMSA module lies in its utilization of multi-head attention, a mechanism designed to concurrently focus on different spatial regions. Each head in the TMSA initiates by sequentially merging all three input streams, culminating in a cohesive feature map. Following this, a Global Average Pooling 2D layer computes the mean across spatial dimensions in this unified map. This computation crafts a holistic portrayal of the amalgamated feature maps, subsequently channeled through back-to-back dense layers, interspersed with a ReLU activation. Emerging from these layers, a sigmoid activation molds the output, deriving weights correlating to the unified feature map's spatial locales. Post this, these weights are reformulated and expanded to resonate with the spatial dimensions of the initial feature map input. The culmination is an element-wise product of the enlarged weights and the cohesive feature map, ensuring each spatial point within the map is bequeathed a weight emblematic of its relative significance. Upon each head's processing conclusion, their outputs are fused to form the ultimate attention-augmented feature map. Through its multifaceted attention orchestration, the TMSA module ensures the resultant feature map accentuates pivotal regions, concurrently fusing a plethora of spatial and depth nuances. This integration of spatial intelligence significantly improves the model's ability to accurately identify pavement cracks.

### 4.2.4 Convolution Transpose Decoder (CTD)

The design of DepthCrackNet places significant emphasis on the central DCE and SDE modules within its encoder segment, which play a vital role in shaping robust feature maps. Nestled within the decoder partition of the DepthCrackNet, the Convolution Transpose Base Decoder (CTD) takes full advantage of these features to deliver exemplary crack detection outcomes. The CTD module, meticulously crafted, serves as the backbone of the decoding process. It incorporates a series of steps, including transpose convolution,

batch normalization, and the utilization of Rectified Linear Unit (ReLU) activation. These steps, working together, aim to recover the spatial details that may have been lost during the encoding process. At the core of the CTD is the Conv2DTranspose layer, which utilizes transposed convolutions, often known as deconvolutions, to expand the spatial dimensions of feature maps. To facilitate this spatial expansion, a (4,4) sized kernel partnered with a (2,2) stride is wielded, which effectively magnifies the spatial resolution twofold. A subsequent batch normalization is trailed by a ReLU activation, infusing the feature maps with non-linearity, which is pivotal for discerning intricate data patterns and relationships. An outstanding characteristic of the CTD module is its remarkable synergy with the preceding modules in the architecture, particularly the TriInput Multi-Head Spatial Attention (TMSA) and the Double Convolutional Encoder (DCE). This synergy ensures the seamless flow of information and the effective utilization of features for crack detection. With the integration facilitated by the TMSA, the module effectively combines the expanded tensor with the provided skip connections. Consequently, the decoder not only accesses the rich features from its deeper layers but also integrates them with spatial information from the network's early stages, ensuring the retention of essential details crucial for accurate crack detection. This integration enhances the model's overall performance and ability to detect cracks effectively. Structurally, the CTD network unfolds over five distinct levels. Within its architecture, the primary four levels encompass convolution transpose layers, TMSA, and DSC. As each tier unfolds, the feature map undergoes a metamorphosis. Conv2DTranspose layers are harnessed to augment its spatial dimensions. The TMSA, or Top-Down Skip Attention layer, plays a pivotal role by combining the initial high-level feature map with its low-level spectral and spatial

counterparts. After this integration, the fused feature map passes through the DSC (Deep Supervision Context) layer. This process enhances the model's ability to capture intricate spatial details and spectral information, contributing to more accurate crack detection. Advancing to the final stage of the CTD network, known as the output stage, the feature map undergoes further refinement through a sequence of operations including Conv2DTranspose, Batch Normalization (BN), and ReLU activations. These operations prepare the feature map for the crack detection task. The refined feature map then passes through the DSC layer, resulting in a feature map that matches the dimensions of the input image but contains 16 channels. In the pixel-wise classification step, the feature map undergoes a 1x1 convolution (Conv) operation followed by the application of a softmax function. This series of operations ultimately produces a 256x256x2 output matrix, which is used for crack detection.

## 4.3 Experimental Work and Results

The effectiveness of the DepthCrackNet model was assessed by comparing its performance on two publicly datasets, Crack500 [43] and DeepCrack [107]. This comparison was benchmarked against renowned architectures such as R2U-Net [108], Attention U-Net [109], TransUNet [110], and Swin-Unet [111], which frequently surface in related academic literature. In this exploration:

In Chapter 4, a meticulous exploration of the underpinnings and evaluations of DepthCrackNet is undertaken. Section 4.3.1 provides an exhaustive review of the selected datasets, detailing their content and establishing their pertinence. Moving forward, Section 4.3.2 delves into the selected evaluation metrics, which serve as the benchmark for assessing DepthCrackNet's performance in comparison to its counterparts. These metrics

provide a quantitative foundation for an objective assessment of the model's effectiveness. Going deeper, Section 4.3.3 unveils the intricate details of the model's development, from its inception to its training regimen. This discussion elucidates the reasoning behind crucial decisions related to the model's architecture, parameters, and optimization strategies. Finally, Section 4.3.4 serves as a culmination, where the experimental results undergo meticulous examination to determine the effectiveness of DepthCrackNet in relation to other models.

### 4.3.1 Dataset

To validate the effectiveness of the proposed methodology, I conducted extensive experiments using two renowned pavement crack datasets, namely Crack500 [43] and DeepCrack [107]. I divided these datasets into three separate sets, maintaining a distribution ratio of 6:2:2. [112]. Specifically, 60% of the data was earmarked for training, 20% served as a validation set to fine-tune model parameters, and the remaining 20% constituted the test set to objectively assess the model's performance. With this systematic approach, I aimed to ensure that the model was exposed to a diverse range of data during training, allowing it to learn effectively and adapt to various scenarios. The selected datasets, each with its distinctive attributes, served as a robust benchmark for evaluating the effectiveness of the proposed methodology.

| Dataset | Resolution | Images | Train | Validation | Test |
|---|---|---|---|---|---|
| Crack500 [43] | 640 × 360 | 3368 | 2020 | 674 | 674 |
| DeepCrack [107] | Variable | 537 | 429 | 54 | 54 |

Table 4.1: Overview of the crack datasets utilized in the experiments.

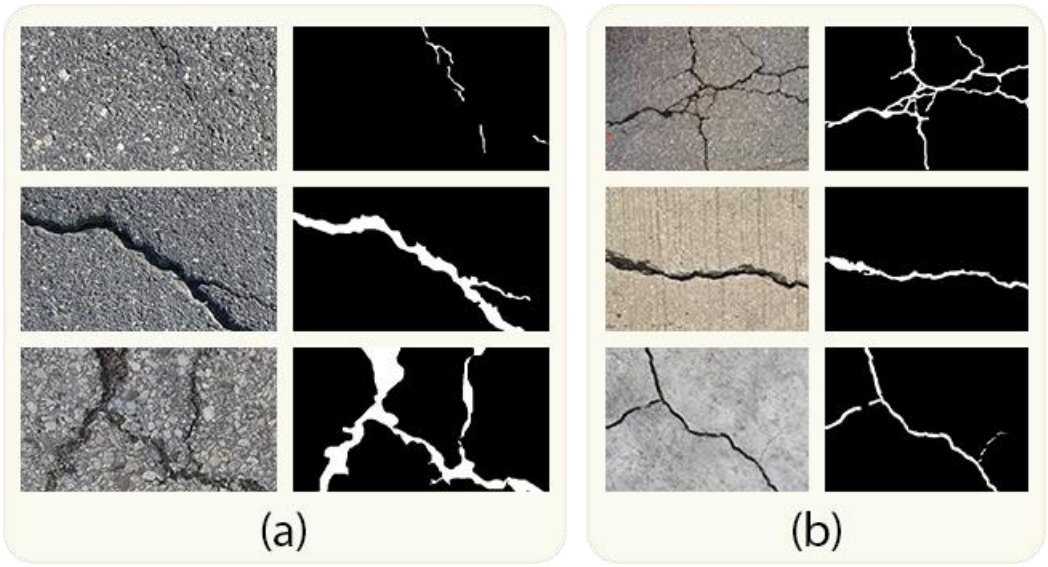


Figure 4.5: Sample images and their associated true data from the research datasets used. (a) features images from the Crack500 collection, and (b) showcases images from the DeepCrack collection.

#### 4.3.1.1 Crack500

The Crack 500 dataset, as referenced in [43], is a comprehensive collection of 500 images, each boasting a resolution approximately around 2000 x 1500 pixels. These images were taken within the boundaries of Temple University using a mobile phone camera. Due to constraints related to computing resources, these images were segmented into 16 unique, non-intersecting partitions. It's noteworthy to mention that only those segments with over 1000 pixels explicitly depicting cracks were kept for further analysis. An additional layer of detail was added through the careful inclusion of pixel-level annotations for every image that displayed cracks. As a result of this meticulous process, the Crack 500 dataset has evolved to include sum of 3368 images showcasing cracks.

#### 4.3.1.2 DeepCrack

The DeepCrack dataset, cited as [107], is an extensive collection of 537 images that are distinctive for their diverse crack scales and intricate backgrounds, providing a well-rounded depiction of varying crack features. This dataset is enriched with three distinct textures, namely, bare, dirty, and rough. Moreover, it showcases two different scene categories: concrete and asphalt. A notable characteristic of the cracks displayed in these images is their varied widths, which can be as narrow as a single pixel and can expand up to a broad 180 pixels. Intriguingly, in each image, the area covered by cracks is only a fraction of the overall space, reflecting typical scenarios one might encounter in the real world. The manual annotation of each image in the dataset, resulting in binary representations of the presence or absence of cracks, adds a layer of precision and value to the dataset. This meticulous annotation process ensures that the ground truth information for crack detection is accurate and reliable, further enhancing the quality of the dataset for training and evaluation.

### 4.3.2 Evaluation Metrics

For the assessment of the segmentation model's effectiveness, I relied upon several definitive metrics. These encompassed Precision, Recall, F1 Score, and mIoU. Precision gauges how adeptly defects are categorized, whereas Recall measures the proficiency in pinpointing negative samples. The F1 Score acts as a balance between Precision and Recall, providing an insight into the model's capability to discern and consistently differentiate between the segmented areas and the genuine target regions in the images. On the other hand, the Mean Intersection over Union (mIoU) measures the congruence between the model's predicted segmentation and the established ground truth. This metric serves as a testament to the model's spatial precision in demarcating objects or defects.

Below is the detailed equations for Mean Intersection over Union (mIoU):

$$\mathrm{mIoU} = \frac{\mathrm{pr} \cap \mathrm{GT}}{\mathrm{pr} \cup \mathrm{GT}} \tag{7}$$

### 4.3.3 Implementation Details and Training

In this part of the study, I explore the detailed settings of my model, focusing on the important hyperparameters that were key during the training stage. The advanced model was subjected to testing via the TensorFlow framework. This procedure was facilitated by a robust computing configuration boasting an NVIDIA 80 GB GPU card complemented by 90 GB RAM, all operating seamlessly on the Paperspace platform. During the training segment for the proposed networks, I established the batch size at 32, while the epoch count was set at 200. The Adam optimization algorithm was the chosen tool for refining the network parameters. Intricately woven into the model is the TriInput Multi-Head Spatial Attention (TMSA), for which I designated the number of heads as 4. The extensive set of component impact analysis, which delved into a myriad of loss functions, eventually converged on the adoption of a weighted hybrid loss function. This specific function played a crucial role in making the learning process more efficient. It helped maintain a balance between accurate pixel-level predictions and the overall spatial alignment between the predicted and actual segmentations. This function is mathematically expressed as: 0.9 × Binary Cross-Entropy Loss + 0.1 × Dice Loss 0.9×Binary Cross-Entropy Loss+0.1×Dice Loss. Here, while the Binary Cross-Entropy Loss zeroes in on individual pixel accuracy, The Dice Loss is designed to enhance the similarity between the predicted and actual segmentation areas, leading to more precise and accurate segmentation results. In the pursuit to adjust the learning rate and identify the most favorable number of training

epochs, I integrated the ReduceLROnPlateau and EarlyStopping callback mechanisms. The former, ReduceLROnPlateau, modulates the learning rate by scaling it with a designated factor, especially when there's a stagnation in the decrease of the loss value over a preset 'patience' epoch count. In tandem, the EarlyStopping mechanism intervenes to terminate the training when deemed necessary. For the purposes of this research, both the factor and patience parameters for ReduceLROnPlateau were firmly set at 0.5.

### 4.3.4 Experimental Results

In this section, I showcase the findings procured from the investigations using the Crack500 and DeepCrack datasets. These results are delineated into two categories: the visual interpretations and the quantitative analyses, which are comprehensively detailed in Subsections 4.3.4.1 and 4.3.4.2 respectively.

#### 4.3.4.1 Crack500:

In Figure 4.6, I provide a detailed visual comparison that places real ground-truth data from the Crack500 dataset side by side with segmentation outcomes from various methods, especially the DepthCrackNet. The layout of the figure is structured such that the initial two columns present the pristine images alongside their paired ground-truth segmentations. Intermediate columns, spanning from the third to the sixth, reveal the performances by distinct models like R2U-Net, Attention U-Net, TransUNet, and Swin-Unet. Concluding this visual array, the seventh column displays the segmentation ability of DepthCrackNet. This illustrative presentation vividly captures the intricate challenges of pavement crack detection as represented in the Crack500 dataset. In the top row, which emphasizes the detection of minute cracks, the model distinctly stands out, boasting an IoU of 63%. This model's advanced ability to capture subtle details stands out when compared to similar models like R2U-Net, Attention U-Net, TransUNet, and Swin-Unet, which have

Intersection over Union (IoU) scores of 57%, 56%, 53%, and 56%, respectively. In the second row, which deals with identifying patterns against similar backgrounds, the model shows precision with an Intersection over Union (IoU) of 58%. This is slightly better than Swin-Unet, which scores 56%, highlighting the model's skill in detecting subtle differences. In the subsequent row, where the spotlight is on faint cracks against textured pavements, the model's commendable IoU of 81% is hard to miss. The model's ability to detect slight irregularities in complex environments becomes even more evident when compared to the formidable R2U-Net, which has a slightly lower performance at 79%. Additionally, in the analysis of large crack segmentation, shown in the fourth sequence, the model's Intersection over Union (IoU) of 73.19% underscores its strength in identifying obvious defects. This performance stands out, especially when compared to other significant architectures that hover around the 70% mark. The last sequence highlights the difficulties encountered due to different pavement materials. In this scenario, the model's adaptability is notable with an Intersection over Union (IoU) of 56%. On the other hand, Swin-Unet faces significant challenges, achieving an IoU of 0. This considerable difference emphasizes the need for an adaptive architectural design that can accurately interpret

various material textures in pavement crack detection.

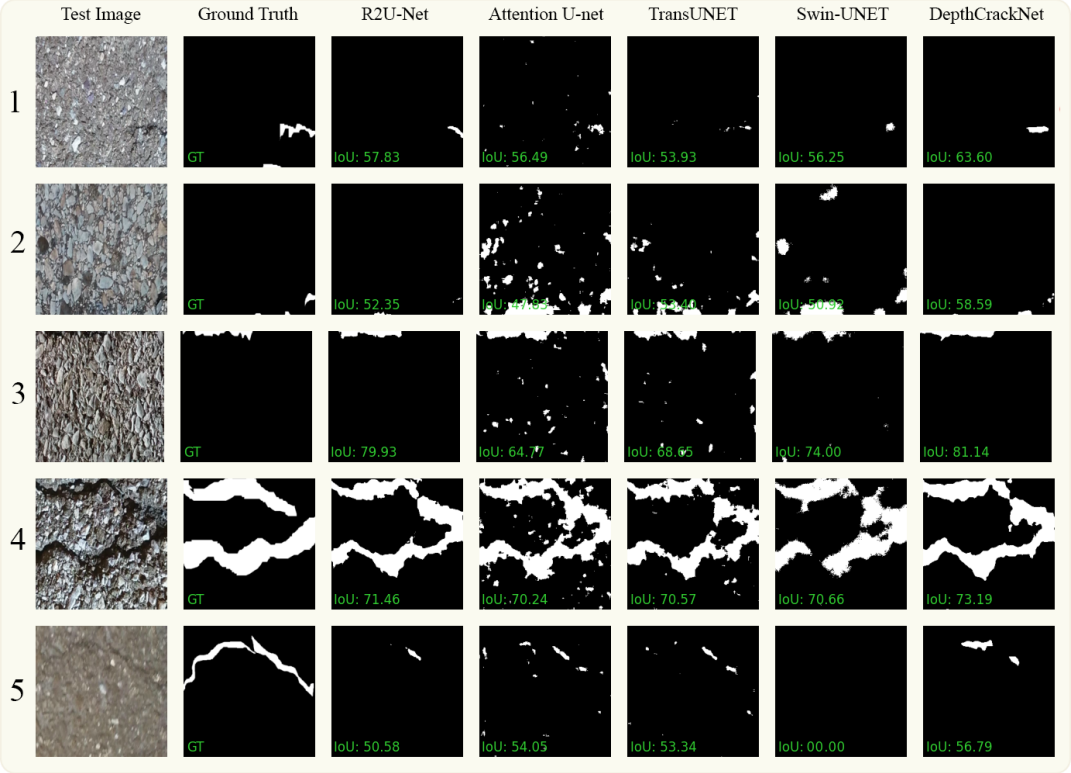


Figure 4.6: A graphical comparison of the DepthCrackNet model with multiple top-performing models using the Crack500 dataset.

Table 4.2 offers a side-by-side evaluation of the uniquely designed model against contemporary state-of-the-art alternatives, focusing specifically on their performances with the Crack500 dataset. While the table encompasses a myriad of performance metrics, the mean Intersection over Union (mIoU) stands out as a pivotal gauge of a model's proficiency in mapping the crack territories with precision. The model's performance is clearly demonstrated as it secures the highest position with a mean Intersection over Union (mIoU) of 77. This score highlights its proficiency in accurately identifying both areas with cracks and those without. Following closely, yet still behind, is the R2U-Net with a mIoU of 73.45. Although this is an admirable achievement, it still shows a noticeable gap in performance

compared to the top model. TransUNet also competes effectively, achieving a respectable mean Intersection over Union (mIoU) of 69.08. However, when compared with the model's score, it's clear that the innovative modifications we've incorporated into our model's architecture give it a significant advantage. The Attention U-Net, despite its specialized focus mechanisms, manages a mIoU of only 65.58. This somewhat underwhelming score indicates potential avenues for refining its approach to better grapple with Crack500's inherent intricacies. Contrarily, the Swin Transformer notches up a decent mIoU of 66.38. However, TransUNet's performance subtly suggests that there might be challenges in adapting transformer-based frameworks for this specific application. The model's leading mean Intersection over Union (mIoU) score is more than just a number—it confirms its robustness, flexibility, and adaptability, particularly in the complex task of pavement crack detection. This superiority isn't limited to the mIoU alone; its precision score of 87.00 further highlights its unparalleled performance, surpassing all other models under consideration.

| Models | Precision | Recall | F1 | mIoU |
|---|---|---|---|---|
| Attention U-Net [109] | 49.98 | 72.94 | 59.31 | 65.58 |
| R2U-Net [108] | 81.21 | 67.10 | 73.49 | 73.45 |
| TransUNet [110] | 67.10 | 64.17 | 65.61 | 69.08 |
| Swin-Unet [111] | 67.01 | 57.67 | 61.99 | 66.38 |
| Proposed DepthCrackNet | 87.03 | 64.11 | 73.83 | 77.00 |

Table 4.2: Assessment outcomes of the DepthCrackNet model compared to other models using the Crack500 dataset.

#### 4.3.4.2 DeepCrack:

Figure 4.7 highlights the effectiveness of the new segmentation model in comparison to well-known benchmarks such as R2U-Net, Attention U-Net, TransUNet, and Swin-

Unet, using the detailed DeepCrack dataset. In the section focusing on the challenges of similar backgrounds, the model takes the lead with an Intersection over Union (IoU) of 73%. This superior performance becomes even more notable when juxtaposed with the closely following Attention U-Net, which achieves an IoU of 71.82%. This difference highlights the model's remarkable ability to differentiate between actual crack defects and possible background distractions, a skill that is essential for practical, real-world applications. Exploring the challenge of identifying thin cracks, the model's capability becomes even more pronounced. It achieves an Intersection over Union (IoU) of 82.48%, showcasing its high level of precision. This is particularly evident when compared to TransUNet, which lags considerably with an IoU of only 57.25%. This vast discrepancy underscores potential shortcomings in the latter's ability to identify minute and subtle defects. Addressing a real-world concern, the third row of the figure delves into scenarios showcasing a mix of crack sizes. In this rigorous test, my model continues its streak, clinching an IoU of 81%. The R2U-Net, however, offers stiff competition, slightly edging out at 82%. In comparison, TransUNet struggles with detecting cracks of various sizes, as shown by its relatively modest Intersection over Union (IoU) of 70%. This demonstrates the superior adaptability of my model to handle the variability and unpredictability of crack sizes. Evaluating the domain of thick cracks, the supremacy of my model becomes almost unassailable. With an awe-inspiring IoU of 91.38%, it leaves other models, including the next best performer, Attention U-Net (with an IoU of 85.93%), in its wake. This performance attests to my model's finesse in mapping pronounced and distinct defect features. The grand finale, embodied in the fifth row, presents the Herculean challenge of detecting ultra-thin cracks. While many models struggle with this task, my model excels,

achieving an Intersection over Union (IoU) of 67.46%. In stark contrast, TransUNet fails completely, unable to detect any cracks and scoring a zero IoU. This significant difference in performance is a clear indication of the superior structure and design of my model, particularly in dealing with the most complex challenges of pavement crack detection.

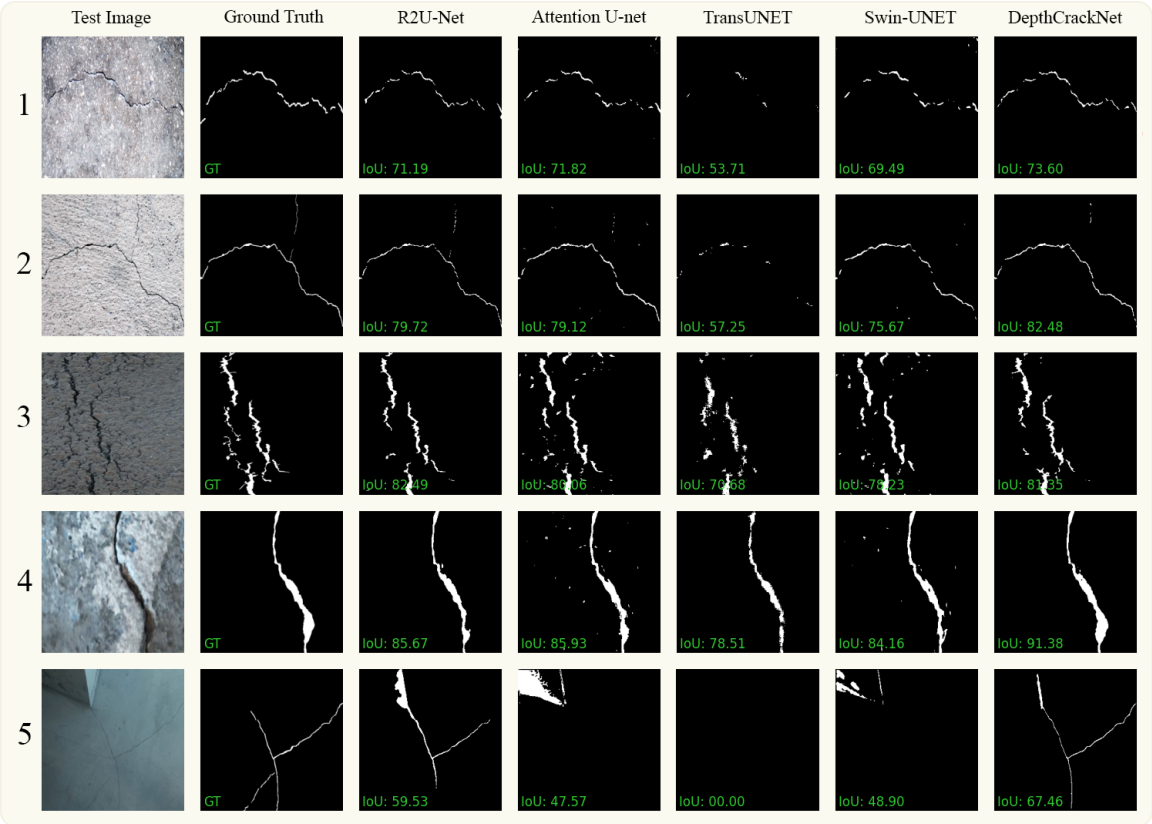


Figure 4.7: A side-by-side visual representation of the DepthCrackNet model and various top-tier models on the DeepCrack dataset.

Table 4.3 given the quantifiable performance metrics of various models on the DeepCrack dataset, offering a discerning perspective into their adeptness in pavement crack detection. My proposed model sets a commendable benchmark with an mIoU of 83.9. This score emphasizes the model's skill not only in identifying cracks but also in effectively differentiating them from surrounding non-crack areas. Its nearest competitor, R2U-Net,

registers a mIoU of 79.23. While commendable in its own right, this score underscores the enhanced detection ability of my model, particularly when the numbers are juxtaposed. Delving deeper into the metrics, my model's holistic performance is evident. Boasting a Precision of 81.9 and a Recall of 84.9, it manifests a judicious balance in minimizing both false positives and false negatives. This equilibrium is further validated by an enviable F1 score of 83.3, reflecting a harmonious blend of precision and recall. Attention U-Net, with its core rooted in enhancing feature discernment through attention mechanisms, yields a mIoU of 75.79. Despite its intrinsic architectural advantage geared towards refining feature understanding, it finds itself overshadowed by my model, especially when it comes to navigating the nuanced crack patterns endemic to the DeepCrack dataset. TransUNet, another contender in this evaluation, achieves a mIoU of 75.03. This demonstrates its reasonable ability in detecting cracks, even if it doesn't quite reach the zenith set by my model. In a somewhat surprising turn, Swin Transformer, despite its acclaimed ability to harness long-range interactions, ends up on the lower end of the spectrum with an mIoU of 69.01. A particular point of concern is its Recall of 54.48, which exposes the bottlenecks transformer-based models might encounter in this domain. This suggests potential challenges in consistently and comprehensively identifying crack instances, a task made even more intricate by the diverse range of pavement textures and varying crack dimensions present in the dataset.

| Models | Precision | Recall | F1 | mIoU |
|---|---|---|---|---|
| Attention U-Net [109] | 81.80 | 81.34 | 81.57 | 75.79 |
| R2U-Net [108] | 87.95 | 89.09 | 88.52 | 79.23 |
| TransUNet [110] | 82.31 | 76.68 | 79.40 | 75.03 |
| Swin-Unet [111] | 81.94 | 54.48 | 65.45 | 69.01 |
| Proposed DepthCrackNet | 81.93 | 84.91 | 83.39 | 83.93 |

Table 4.3: Comparison outcomes of the DepthCrackNet model with other models based on the DeepCrack dataset.

## 4.4 Discussion

This section is structured for an in-depth examination of three comprehensive subsections, each catering to a unique facet of the overarching topic. Section 4.4.1, titled "Component Impact Analysis" is dedicated to a detailed assessment of the component-wise contributions of the proposed model. In these studies, my aim is to gain a detailed understanding of the model's architecture. I'm trying to figure out which parts of the model are essential for its performance and which ones can be modified or removed without greatly affecting its accuracy. This methodical exploration allows for a profound appreciation of each component's significance within the model. Next, in Section 4.4.2, titled "Review of Related Research," I conduct a thorough examination of previous studies that are directly relevant to the datasets being analyzed. This evaluative process not only constructs a robust historical context but also underlines the advancements and gaps in knowledge that have taken shape over time. Such an assessment is paramount, as it provides a reference point against which the contemporary research can be gauged. Transitioning from the past to the present, Section 4.4.3, "Analysis of Model Limitations," pivots towards a focused investigation of the specific instances where the proposed model might show discrepancies. This section is crucial for highlighting areas that need more improvement, whether it's situations where the model doesn't meet expectations or errors in crack detection. By zeroing in on these limitations, the intent is to pave the way for enhancements in future iterations of the

model. This intricate organization of subsections assures a holistic exploration of the subject, balancing historical insights with practical analysis of the model's operational boundaries.

### 4.4.1 Component Impact Analysis

The automatic detection of pavement cracks is an essential and intricate task. Addressing this complexity, the DepthCrackNet model was developed, featuring three primary modules. To determine the influence and value of each of these modules, a detailed component impact analysis was performed. This analysis centered on three components: the Double Convolution Encoder (DCE), the TriInput Multi-Head Spatial Attention (TMSA) module, and the Spatial Depth Enhancer (SDE) module.

In DepthCrackNet's context, its performance was evaluated without the presence of each module, using the Crack500 and DeepCrack datasets, both established benchmarks in crack detection. Starting with the DCE component, removing it led to a significant drop in performance. The mIoU scores plummeted to 70.1% on Crack500 and 76.4% on DeepCrack. These figures, when compared to when DCE is active, highlight its essential role. The DCE, equipped with a series of convolution layers, offers a detailed interpretation of image data. This allows the model to identify cracks across different scenarios effectively. Next, the TMSA module's analysis showed that without it, mIoU scores reached 72.8% on Crack500 and 78.5% on DeepCrack. While this decline wasn't as dramatic as the DCE's removal, it still emphasizes TMSA's crucial role. The TMSA module, notable for its multiple attention heads, adeptly captures a diverse range of spatial dynamics. This feature bolsters the model's resilience, enabling it to adapt to varying pavement conditions and irregularities. Lastly, the study turned its focus to the SDE

module. In its absence, mIoU scores settled at 74.5% for Crack500 and 80.2% for DeepCrack. The noticeable change in these results affirms the significance of the SDE within the model. By providing a more profound insight into spatial aspects and improving feature extraction, the SDE guarantees DepthCrackNet's precise crack detection abilities.

In summary, the ablation analysis solidly confirms the critical role of each module within DepthCrackNet. The DCE, TMSA, and SDE modules collectively enhance the model's performance, ensuring both robustness and accuracy in detecting pavement cracks. It presents an exciting avenue for future research to refine these modules or introduce innovative ones to push the boundaries of crack detection further.

### 4.4.2 Comparison with the Previous Studies

The identification of cracks in pavements plays a pivotal role in maintaining quality standards. Over the years, a plethora of techniques rooted in image processing and machine learning have been developed with the aim to simplify and fully automate the crack detection mechanism. In the scope of this research, a unique model, named DepthCrackNet, has been introduced, with its primary objective being the automatic identification of surface defects. The efficacy of this model was rigorously tested using two renowned public datasets frequently mentioned in scholarly literature: Crack500 and DeepCrack. From the data presented in Table 4.4, several observations can be made regarding DepthCrackNet's performance on the Crack500 dataset. DepthCrackNet excels with a precision rate of 87, setting it apart from other models. The closest competitor, a model from [113] which employs a Feature Pyramid Network enhanced by a self-guided attention refinement module, trails with a precision of 83. While the model showcases ability in precision, it faces challenges in recall, scoring 64.1. In comparison, models like the one mentioned in

[114], which achieves a recall of 80 using DeepLab enhanced with Multi-Scale Attention, outperform it. DepthCrackNet records an F1 score of 73.8, which is competitive but falls short of the 79.4 score achieved by the model in [113]. Furthermore, the model stands out in terms of mIoU, achieving 77, thereby surpassing all peers and setting a benchmark. Previously, the best mIoU was 65.3 as recorded by the model in [115]. Shifting focus to its performance on the DeepCrack dataset, the model exhibits a respectable precision of 81.9. However, it doesn't top the charts, as the model in [116] clinches a higher precision of 88.3, utilizing an Attention Module combined with a Focal Loss Function. DepthCrackNet showcases its strength with a commendable recall of 84.9, outperforming all its contemporaries, signaling its ability in pinpointing genuine crack occurrences. While the model boasts an F1 score of 83.3, it's marginally overshadowed by the model in [117] which achieves an F1 score of 87.5, owing to the incorporation of a Morphology Branch paired with a Shallow Detail Branch. In this category, DepthCrackNet reigns supreme with an mIoU of 83.9, indicating its ability in correlating predicted segmentation with actual data. Upon analyzing the above metrics, it becomes evident that the incorporation of 3D spatial attributes and a sophisticated multi-head attention mechanism substantially bolsters DepthCrackNet's proficiency in accurately discerning pavement cracks. Despite certain setbacks, especially concerning recall in the Crack500 dataset, its enhanced precision and mIoU metrics underscore its potential in precise segmentation and minimizing false positives—factors of paramount importance for practical applications in pavement upkeep systems.

| References | Methods | Dataset | Precision | Recall | F1 | mIoU |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| [118] | CNN, Pyramid Attention Network | Crack500 | 81.6 | 76.5 | - | 62.35 |
| [113] | Feature pyramid network, self-guided attention refinement module | | 83 | 79.6 | 79.4 | - |
| [114] | DeepLab With Multi-Scale Attention | | 69.5 | 800 | 74.4 | 55.9 |
| [119] | Unet based method | | - | - | - | 60 |
| [115] | CNN model | | 80.7 | 77.3 | - | 65.3 |
| [120] | Self-Attention-based Efficient U-Net | | - | - | 77.5 | 66.3 |
| [121] | ECA Channel Attention Module and FCNhead Decoding Dock | | - | - | - | 63.97 |
| [116] | RUC-Net with scSE Attention Module | | 69.8 | 76.1 | 72.9 | 57.3 |
| [122] | Joint Topology-preserving and Feature-refinement Network | | 68.81 | 69.0 | 65.7 | - |
| **Proposed DepthCrackNet** | 3D Spatial Features and Multi-Head Attention Mechanism | | 87.0 | 64.1 | 73.8 | 77 |
| [116] | Attention Module and Focal Loss Function | DeepCrack | 88.3 | 81.2 | 84.6 | 73.3 |
| [117] | Morphology Branch and Shallow Detail Branch | | - | - | 87.5 | 77.9 |
| **Proposed DepthCrackNet** | 3D Spatial Features and Multi-Head Attention Mechanism | | 81.9 | 84.9 | 83.3 | 83.9 |

Table 4.4: Results from prior research comparing performances on the Crack500 and DeepCrack datasets.

### 4.4.3 Error Analysis

The proposed model works well in identifying pavement cracks in various datasets, but it's important to carefully examine the cases where it didn't perform as expected. This will help us find ways to improve it. Figure 4.8 visually presents three instances where the model encountered difficulties in accurate crack detection. In the initial row of Figure 4.8, the model was entirely unsuccessful in discerning any cracks. This shortcoming might be due to certain complex elements in the images that could have confused the model. For example, backgrounds with similar textures or a lack of clear contrast between the cracks and their surrounding environment might have contributed to this issue. Understanding the subtleties of this limitation is crucial because it sets the foundation for strengthening the model's robustness in future iterations. Transitioning to the second row of Figure 4.8, I observe a situation where the model did detect cracks, reflected in an Intersection over Union (IoU) score of 56.32. Despite showcasing its capability to some extent, the IoU metric underscores the existence of ample scope to amplify the model's proficiency. Moving on to the third row of Figure 4.8, I once again encounter a situation akin to the first row where the model failed to discern any cracks. This reiterates the model's present limitations in contending with certain crack types or specific visual contexts within the image. Comprehending the root causes behind these lapses is instrumental in finetuning the model to ensure heightened accuracy and dependability. Future research efforts should be directed towards addressing these identified challenges. Possible strategies may include expanding the training dataset to include a wider range of crack examples, enhancing the model's ability to discern features, and experimenting with advanced post-processing techniques. These adjustments aim to curtail false negatives, thereby augmenting the model's overarching efficacy in crack identification.

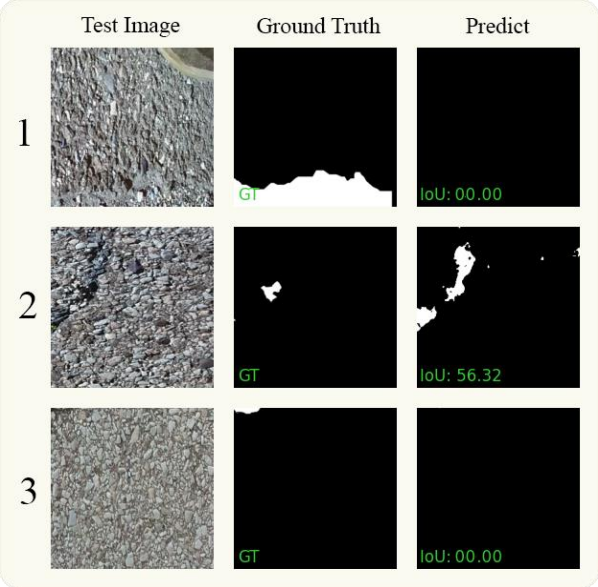


Figure 4.8: Part (a) displays the visual results of failures for the Crack500 dataset, while part (b) shows the same for the DeepCrack dataset.

## 4.5 Conclusion

In this study, I introduced DepthCrackNet, a model inspired by the U-Net framework. The primary goal of this model is to transform the task of detecting pavement cracks, which is a significant step toward enhancing road safety. DepthCrackNet boasts a unique architectural composition, blending the Double Convolution Encoder (DCE), the TriInput Multi-Head Spatial Attention (TMSA) module, and the Spatial Depth Enhancer (SDE) module. This structure is meticulously curated to adeptly navigate the intricate challenges presented by the diverse nature of cracks and the assortment of irregularities found on road surfaces. To ascertain its efficacy, DepthCrackNet underwent stringent testing using two esteemed public datasets: Crack500 and DeepCrack. The results were encouraging, with the model registering mIoU scores of 77.0% on the Crack500 dataset and an even more 83.9% on the DeepCrack dataset. A side-by-side evaluation with prevailing models

highlighted DepthCrackNet's commendable performance, amplifying its viability for practical integration within pavement management frameworks.

# Chapter 5. E-UNet3+: Steel Surface Defect Segmentation Model Using an Enhanced UNet3+ with Multiscale Feature Learning and Attention Mechanisms

## 5.1 Introduction and Problems

Surface imperfections in manufactured products not only reduce the overall quality of the goods but also have a substantial financial impact on production [70], [123], [124]. While technological advancements have transformed many sectors, a considerable segment of the manufacturing industry remains dependent on manual, human-driven inspections to identify these imperfections. These manual methods, while traditional, are labor-intensive and often lack the precision and efficiency of automated systems [5]. Recognizing these drawbacks, there is a growing momentum towards integrating automated techniques for surface defect detection, which are proving their worth in various industrial scenarios. Strip steel is integral to numerous sectors, spanning from construction engineering to aerospace and the automotive industry. Ensuring the quality of this pivotal material directly influences the durability and reliability of the finished products within these industries. Thanks to technological advancements, a range of computer vision and machine learning techniques have emerged to transform the process of detecting surface defects in steel. [125]–[127]. Yet, as illustrated in Figure 5.1, there exist three predominant challenges that any prospective model for defect detection needs to confront:

a) **Diversity of Defects**: The surface of steel can manifest various defect types, including but not limited to corrosion, pitting, and scratches. Given this variety, it's

imperative for detection systems to be versatile, capable of recognizing and categorizing a myriad of defect forms.

b) **Ambiguous Background Textures**: Frequently, the natural textures found on steel surfaces can resemble defects, creating significant challenges in accurately distinguishing them. This underscores the need to integrate advanced image processing methods to differentiate real defects from deceptive background patterns.

c) **Range in Defect Sizing**: The dimensions of defects on steel surfaces can span a broad spectrum, from minuscule inclusions to more pronounced dents. Thus, it's crucial for the detection system to employ a multiscale approach, ensuring accurate detection across all sizes of defects.

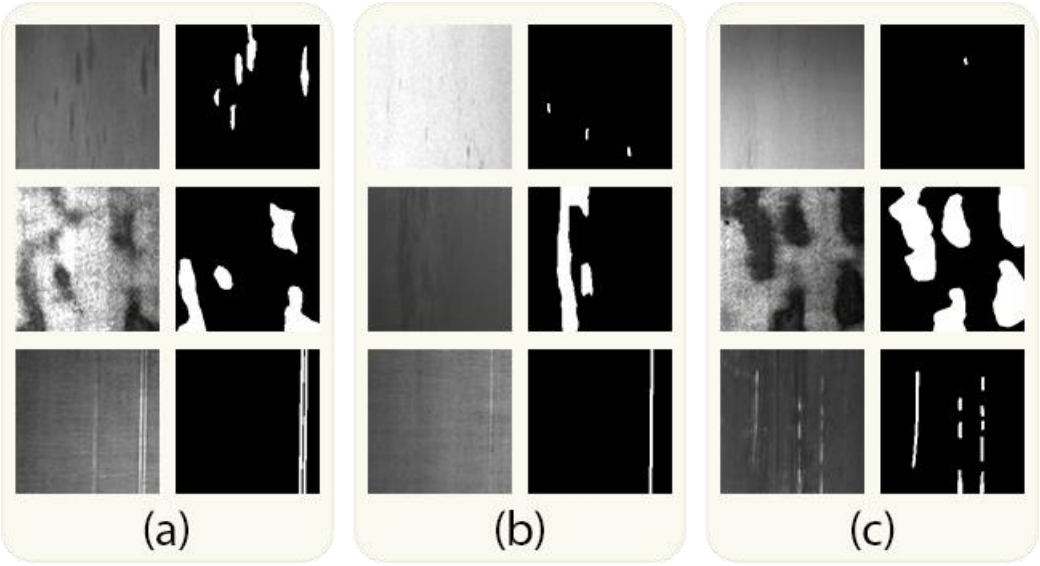


Figure 5.1: Difficulties in identifying imperfections on steel surfaces (defects are indicated by white areas in the reference image): a) Diversity of defects, b) Ambiguous background textures, c) range in defect sizing.

Recent studies underscore the value of computer vision techniques and models in identifying surface anomalies in strip steel. These techniques can be broadly grouped into two primary sectors: conventional methods and deep learning-based strategies. When discussing traditional techniques, I see three primary subclasses: statistical, spectral, and model-centric methodologies. To clarify, statistical techniques identify defects by analyzing variations in pixel distributions between the anomalous region and the standard background texture [124]. A study by [128] dived into pixel distribution patterns within defects on hot-rolled steel, aiming to fine-tune segmentation thresholds. Similarly, [129] utilized a region-specific, adjustable threshold to counterbalance inconsistent surface attributes induced by scaling. However, these statistical methods tend to perform well with images that have high contrast but struggle when dealing with the frequently encountered low-contrast images in industrial settings. The muted luminance blurs defect edges, thereby diminishing the efficacy of these techniques. To tackle this issue, [130] amalgamated Gabor filters with morphological traits to spot pinholes in flawed steel. In a related vein, [131] harnessed optimized Gabor filters to pinpoint the least energy divide between defects and their background, assisting in better segmentation. Expanding on this, [49] proposed a wavelet-centric algorithm tailored to separate defects from grainy backgrounds. While spectral techniques can be highly efficient, they often demand intricate adjustments to parameters like Gabor and wavelet filters. Bypassing these intricacies, several model-driven techniques have been spotlighted. One notable instance is [51]'s use of the Markov random field model to refine sheet-metal imagery for a streamlined analysis. Nevertheless, these model-based strategies grapple with the challenge of pinpointing defects marked by slight intensity variations or those with muted contrasts, irrespective of their supervised or

semi-supervised nature. In contrast, methods rooted in deep learning stand out for their superior ability, mainly due to their innate capability to autonomously discern crucial features. A case in point is [132]'s introduction of a dedicated VSD network tailored for steel defect sorting, which surpassed benchmarks set by renowned architectures like VGG19 and ResNet. Further enhancing this domain, [133] incorporated a semi-supervised learning paradigm with generative adversarial networks, aiming to produce unlabeled defect specimens to bolster classification accuracy. While such classification-focused methods tout accuracy metrics, they often omit granular details regarding defects' exact locations and morphologies. Bridging this knowledge gap, pioneering research has zeroed in on automated techniques for defect detection. A highlight includes [134]'s adoption of deformable convolutions as a replacement for standard convolutions within the Faster R-CNN framework, amplifying its efficacy in spotting diminutive object defects. But, these object detection-centric techniques, although adept at pinpointing defect locales, often stumble when delineating defect perimeters and configurations. Mitigating this limitation, [135] unveiled a saliency detection framework harnessing channel-weighted and residual decoder segments for sharper defect spotting. While strides have been made in semantic segmentation and saliency detection concerning capturing edge shapes, there remains an urgent call for deeper research to fine-tune detection boundaries, augment feature extraction, and reinforce model resilience.

In my research, I present an augmented version of the UNet3+ model [136], enhanced with Multiscale Feature Learning and Attention Mechanisms, tailored explicitly for steel defect segmentation. This advanced E-UNet3+ model, comprised of 6.7 million parameters, showcases good outcomes, largely credited to its streamlined yet potent

architecture. This represents a notable departure from its predecessor, the original UNet3+ model[136], which boasts a hefty 26.9 million parameters. Moreover, when juxtaposed with contemporary benchmarks, my model distinctly surpasses them in overall efficiency.

The main highlights of my research encompass:

a) The inception of an innovative encoder framework, weaving in varied dilation rates and DropBlock regularization. This design captures features across different scales, amplifying the model's flexibility and resilience.

b) I enhanced the conventional max-pooling procedure inherent in UNet3+ by integrating strided Conv2D layers. This, when paired with subsequent concatenation and a 1x1 convolution, preserves pivotal feature details, boosting the model's predictive ability.

c) By embedding the Convolutional Block Attention Module (CBAM) within the skip pathways of UNet3+, I fine-tuned the model's concentration on vital attributes pivotal for defect identification.

d) Assessments revealed that the proposed E-UNet3+ model stands superior to previous models. Furthermore, this revamped architecture exhibited proficiency in surface defects, accounting for the variety in defect nature, resemblance to the background, and the broad spectrum of defect dimensions.

## 5.2 Proposed Model

In this study, I introduce an enhanced version of the UNet3+ model, finely tuned for the specific purpose of detecting defects in steel. This model incorporates three crucial improvements that enhance its abilities in feature extraction, down-sampling functions, and

strengthening skip connections. A visual depiction of the entire framework of the E-UNet3+ model can be found in Figure 5.2.

Delving deeper into the core components employed:

- Within the encoder segment of the structure, I infuse a Multiscale Feature Learning Module (MSFLM). This specialized module is crafted to assimilate multiscale contextual data. Its efficacy is further amplified by integrating DropBlock, a regularization strategy which enhances the model's adaptability to diverse datasets. In this context, my model utilizes filter dimensions of 16, 32, 64, and 128.
- In contrast to the traditional down-sampling techniques, typically executed via max-pooling layers in the original UNet3+ model, my refined version blends max-pooling with strided convolutions in a novel hybrid approach. The ensuing feature sets, derived from both techniques, are seamlessly concatenated and processed through a 1x1 convolution. This not only retains but also magnifies the model's ability to represent intricate features.
- A pivotal addition to my architecture is the integration of the Convolutional Block Attention Module (CBAM) within the model's skip pathways. This sophisticated module meticulously refines the feature maps, zeroing in on key areas, rendering the model acutely attuned to subtle defect intricacies on steel facades.

Collectively, these modifications result in the E-UNet3+ model surpassing its predecessor, the original UNet3+, and competing with other state-of-the-art models in the field of steel defect detection.

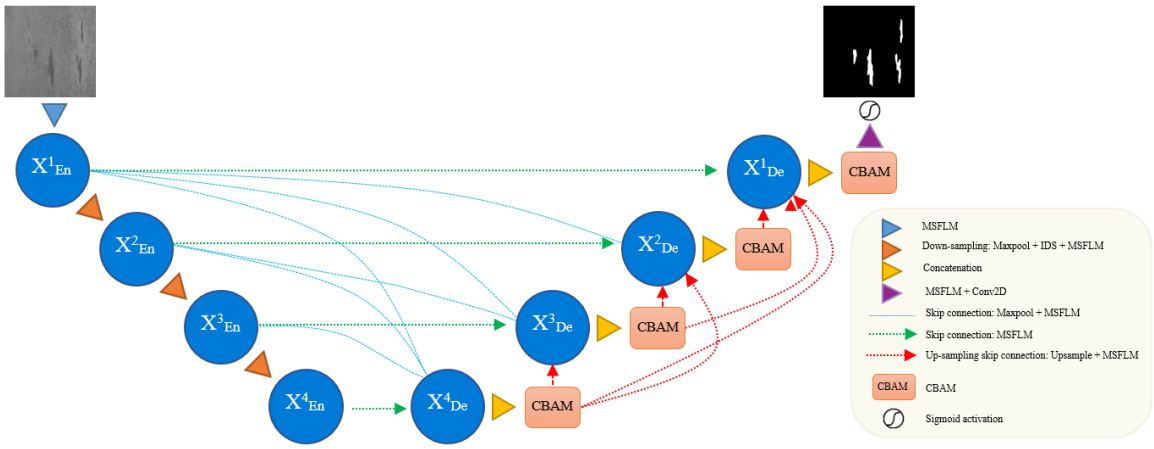


Figure 5.2: The architecture of the proposed E-UNet3+ model.

### 5.2.1 Multiscale Feature Learning Module (MSFLM)

In the realm of Convolutional Neural Network (CNN)-driven designs, especially when targeting defect detection, There is an ongoing balancing act between the complexity of a system and its functional effectiveness. Conventional frameworks like the classic U-Net and its derivatives predominantly harness a blend of convolutional tiers, ReLU activations, and batch normalization for feature derivation. To enhance these designs, researchers often increase the number of layers to access richer and more semantic features. [78], [79], [99]. However, such augmentation comes with its own set of challenges:

- **Vanishing Gradient Dilemma**: Amplifying the layer count, while theoretically beneficial for feature capture, accentuates the notorious vanishing gradient issue, impeding effective network learning.
- **Computational Strain**: Every added convolutional operation inflates the parameter count, ratcheting up the computational demands, making these deep structures increasingly taxing.

- **Data Dearth**: The field of defect detection frequently faces challenges related to dataset limitations [[59], [109]]. This shortage of data makes training these complex networks more challenging, as they typically require a substantial amount of training data.

Given these challenges, my research introduces the Multiscale Feature Learning Module (MSFLM), which is seamlessly integrated into the UNet3+ framework. The core strength of MSFLM lies in its ability to incorporate context from various scales, thereby enhancing the defect detection capabilities of my UNet3+-based model. MSFLM replaces the conventional convolutional segment found in the original UNet3+ architecture. Its primary purpose is to extract multiscale information from the input using dilated convolutions and then integrate these insights into a comprehensive feature map. A visual representation of this configuration can be seen in Figure 5.3.

As illustrated in Figure 5.3, the MSFLM is structured around three core elements:

#### 5.2.1.1 Foundational Convolution Layer

Initially, the input navigates through a convolutional layer equipped with a $1 \times 1$ kernel dimension. This process yields a feature map, serving as a cornerstone for the ensuing stages. Within the figure, this is labeled as the 'initial conv2D'.

#### 5.2.1.2 Dilated Convolutional Assemblies

The MSFLM features a trio of primary assemblies, each encompassing a collection of Conv2D layers distinguished by unique dilation rates. Every assembly is tailored to implement dilated convolutions on the input, striving to harvest features across varied scales without substantially amplifying the receptive field dimensions or inflating the parameter count. To bolster generalization and reinforce the model's resilience, DropBlock

regularization is embedded within each assembly. In addition to this, each pathway includes batch normalization combined with ReLU activation patterns. These jointly work to standardize feature value distributions while simultaneously infusing the system with necessary non-linear properties. Furthermore, residual pathways emerge as pivotal. Following the synthesis of feature maps after every assembly, these maps undergo element-wise fusion with the preliminary feature map—a methodology termed 'residual connection'. This strategy plays a crucial role in addressing the vanishing gradient problem, thereby strengthening the network's learning capabilities.

#### 5.2.1.3 Feature Integration

In the final stage, the feature maps obtained from each assembly are merged along the channel axis, followed by the application of a ReLU activation pattern. This culminates in an integrated feature map saturated with multiscale insights. By assimilating Multiscale feature learning within the encoder segment of UNet3+ blueprint, the ambition is to augment the model's competency in adeptly identifying an array of steel defects, ensuring both precision and resilience.

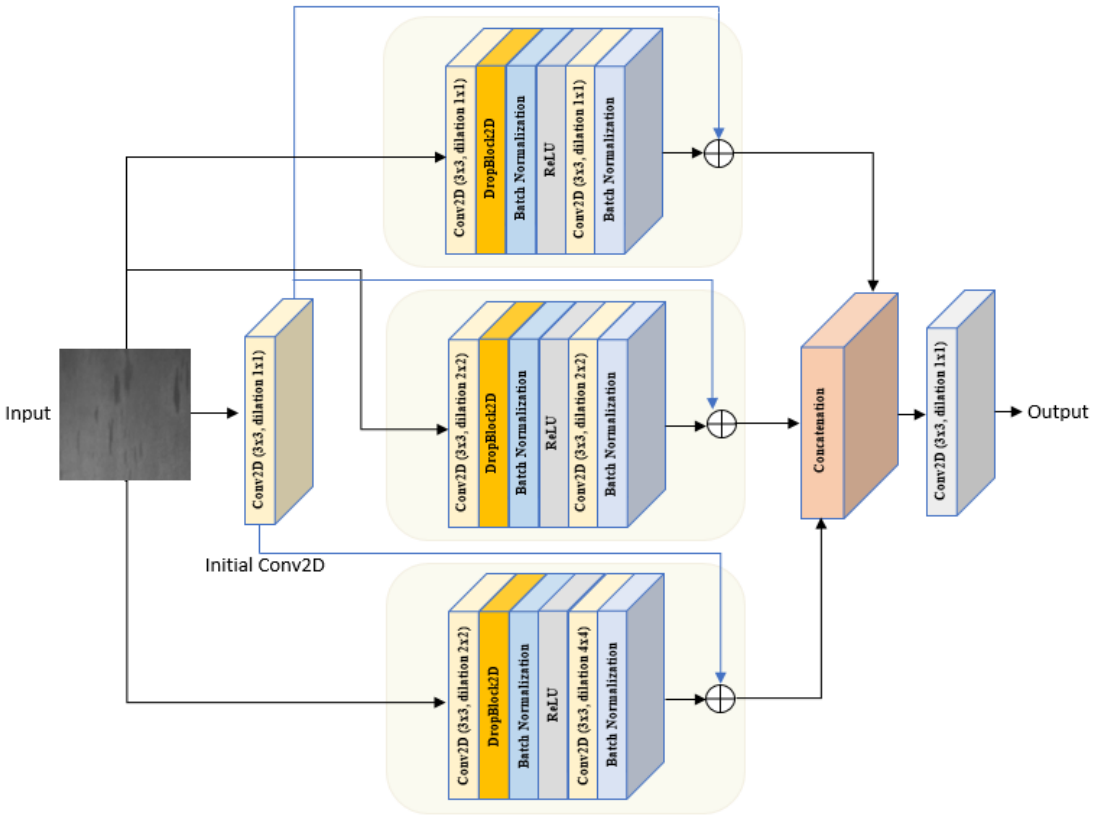


Figure 5.3: The structures of Multiscale Feature Learning Module (MSFLM).

### 5.2.2 Improved Down-Sampling Module (IDS)

Conventional down-sampling strategies predominantly employ pooling layers, such as max-pooling or average pooling, to diminish the spatial dimensions of an image. While these approaches facilitate computational efficiency, they present two primary shortcomings:

- There is a tendency to exclude essential spatial information, which can lead to reduced performance, particularly in complex tasks such as object detection or segmentation.
- These layers are static, indicating they remain unaltered during the learning phase. This non-adaptive nature can culminate in less than optimal outcomes.

To confront these issues, particularly the potential erosion of essential feature details during the down-sampling phase, the research introduces the Improved Down-Sampling (IDS) mechanism. Crafted to execute the down-sampling function with heightened efficiency relative to its traditional counterparts, the IDS module forgoes typical pooling layers. Instead, it harnesses the power of a 3×3 convolution with 2×2 strides. This is seamlessly followed by Batch Normalization and a ReLU activation process. This design doesn't simply reduce the spatial dimensions of feature maps; it also allows the convolution process to evolve and capture representations, thereby enhancing the expressiveness of the feature map for subsequent layers.

### 5.2.3 Convolutional Block Attention Module (CBAM)

In CNN-inspired architectures like U-Net and its derivatives, skip connections serve a pivotal role. They channel lower-level features to the network's more profound depths, thus facilitating a more refined reconstruction of the resultant output. Yet, a notable limitation of traditional skip connections lies in their indiscriminate transmission of features. Rather than prioritizing, they transfer all features uniformly. This non-selective amalgamation can occasionally overshadow and water down paramount features, potentially hindering optimal outcomes in specialized operations such as image segmentation and object detection. To navigate this impediment, the research harnesses the capabilities of the Convolutional Block Attention Module (CBAM) [59]. This module is meticulously crafted to refine the efficacy of the E-UNet3+ model's skip connections. CBAM operates with a dual focus: it engages both the channel and spatial dimensions of feature maps. This dual attention mechanism ensures that the most vital channels and

regions within the feature maps are emphasized. Such a deliberate concentration on pivotal channels and spatial regions translates to a superior quality of feature transmission, which in turn accentuates the ability of E-UNet3+ model. The intricacies of CBAM's architecture are vividly laid out in Figure 5.4.

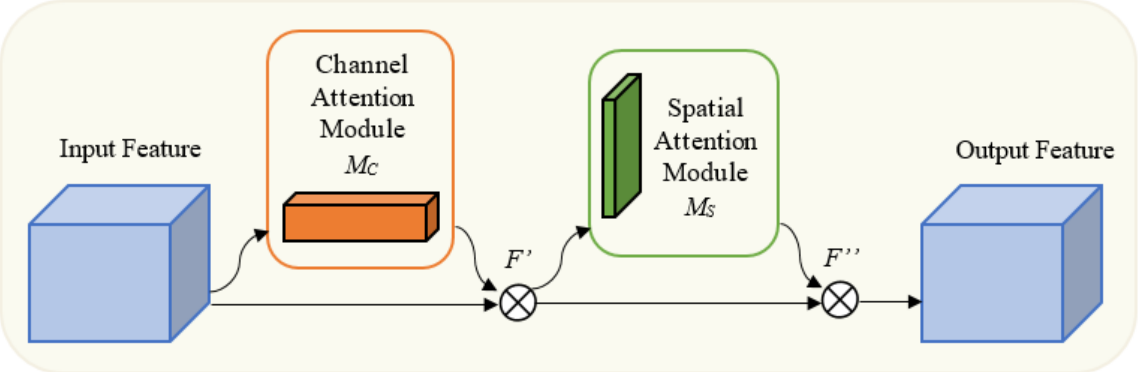


Figure 5.4: Convolutional block attention module (CBAM) structure.

The entire attention procedure can be summarized as follows [59]:

$$F' = Mc(F) \otimes F \quad (8)$$

$$F'' = Ms(F') \otimes F' \quad (9)$$

Where the symbol $\otimes$ denotes element-wise multiplication. $F'$ is the outcome when the feature map is multiplied by the channel attention map, and $F''$ is the result when $F'$ is further multiplied by the spatial attention map, producing the final output.

#### 5.2.3.1 Channel Attention Module

Critical to the improvement of feature extraction in convolutional networks is the channel attention module. Its primary role is to carefully evaluate channels, emphasizing those that are essential for effective feature extraction. A fundamental challenge that this module addresses is maintaining data integrity and minimizing data loss during the feature

selection process. To overcome this challenge, the module cleverly combines the capabilities of two distinct pooling layers: global average pooling and global max pooling. These layers work together to compress the feature map in the spatial dimension. Delving deeper, as depicted in Figure 5.5, the global average pooling layer operates as a detector of overarching, common features in the feature map. In stark contrast, the global max pooling layer hones in on the intricate discrepancies and variations peppered throughout the feature map. Remarkably, the combined might of these two layers transcends their individual capacities. Their synergistic interplay ensures that feature extraction is both comprehensive and nuanced, showcasing superior performance in comparison to either of the pooling layers functioning autonomously.

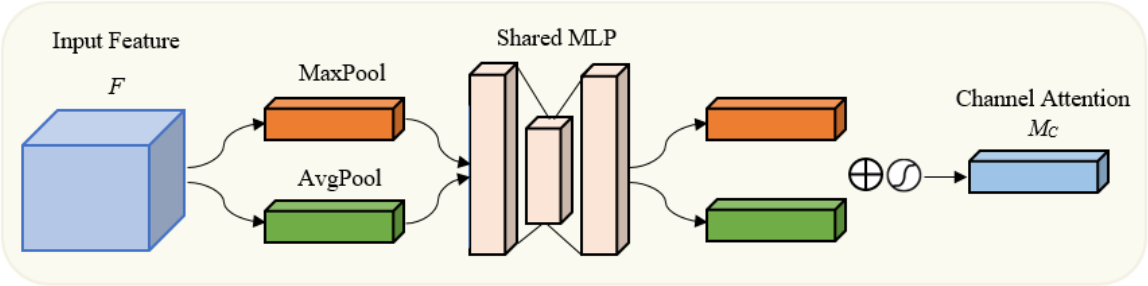


Figure 5.5: Channel attention module.

Following the pooling layers, the squeezed feature maps $F\mathrm{C}avg$ and $F\mathrm{C}\max$ are passed through a shared Multi-Layer Perceptron (MLP) network consisting of a single hidden layer. The MLP operates at a predetermined compression ratio to lower computational complexity and the number of parameters. A sigmoid activation function is then applied to produce the channel attention map $M\mathrm{C}(F) \in R\mathrm{C}\times1\times1$ the process of which was as follows [59]:

$$MC(F) = \sigma(MLP(Avgpool(F)) + MLP(Maxpool(F))) \quad (10)$$

$$= \sigma(W1(W0(FCavg)) + W1(W0(FCmax)))$$

In this scenario, $FCavg$ and $FCmax$ are the feature maps obtained after applying the global average pooling and global max pooling layers, respectively. The sigmoid function is represented by σ. The weight matrices W0 and W1 of the MLP have dimensions RC/r×C and RC×C/r, respectively, where r is the compression ratio.

#### 5.2.3.2 Spatial Attention Module

As shown in Figure 5.6, the spatial attention module is shown to focus more on certain regions of the feature map that are more responsive compared to what the channel attention module targets. For the feature maps outputted by the spatial attention module, both the global average pooling and global max pooling layers are employed to squeeze the feature maps into two 2D maps, $FSavg$ and $FSmax$, along the channel dimension. This is done to accentuate regions of the feature map that contain crucial information.

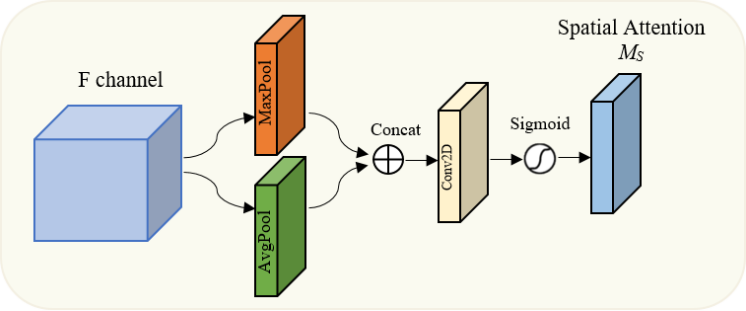


Figure 5.6: Spatial attention module.

After concatenating the two 2D feature maps, $FSavg$ and $FSmax$, an effective feature map is formed, which then undergoes a convolution operation. Subsequently, a

sigmoid function is applied to this convolved feature map to compute the spatial attention map, denoted as MS (F) with dimensions 1×H×W [59].

$$Ms(F) = \sigma(f7\times7([AvgPool(F); MaxPool(F)])) \\ = \sigma\ (f7\times7([FSavg; F\text{Smax}])) \tag{11}$$

In this setup, $FSavg$ and $F$Smax are the channel-dimension-squeezed feature maps, while $\sigma$ signifies the sigmoid function. The introduction of CBAM into the CNN brings an attention mechanism that enables both the channel and spatial modules to work collaboratively.

## 5.3 Experimental Work and Results

In the delineation of the research methodology, I begin with Section 5.3.1, which provides an in-depth understanding of the dataset I employed for this study, touching upon its composition, source, and how it aligns with the research objectives. Following this, Section 5.3.2 shifts focus to the evaluation metrics. These metrics serve as essential tools, enabling us to gauge the effectiveness and precision of my model's performance in real-world scenarios. Furthering the exploration, Section 5.3.3 delves into the intricate details of my model's implementation. This section provides insights into the details of the training process, including hyperparameter configurations and other crucial settings that have a significant impact on the model's learning progress. Lastly, I transition to Section 5.3.4, where I present the culmination of my efforts—the experimental results. Here, readers are introduced to a comprehensive evaluation, enriched with both visual demonstrations and quantitative assessments. This twofold analysis offers a well-rounded perspective on the model's capabilities and performance benchmarks. By progressing through these

systematically structured sections, readers can acquire a comprehensive and granular understanding of the research methodology and its subsequent results.

### 5.3.1 Dataset

During the experimental phase, I selected the SD-saliency-900 [135] dataset, which is a publicly accessible resource, to evaluate the efficacy of the proposed model. Comprising 900 steel surface defect images, this dataset has been segmented into three primary defect categories: inclusion, patches, and scratches. A visual representation of some of these images is displayed in Figure 5.7. With each category containing an equal distribution of 300 images, the standard resolution for these images stands at 200 × 200 pixels. To enhance the research methodology, the dataset also furnishes pixel-level labels specific to each defect type. This granularity facilitates not only the training of E-UNet3+ model but also its in-depth evaluation. To ensure compatibility with the UNet3+ architecture, I undertook a standardization process wherein all images from the dataset were resized to a consistent 224 × 224 pixel resolution. Following this, I divided the standardized dataset into distinct segments: training, validation, and testing, with the specific distributions and details elucidated in Table 5.1.

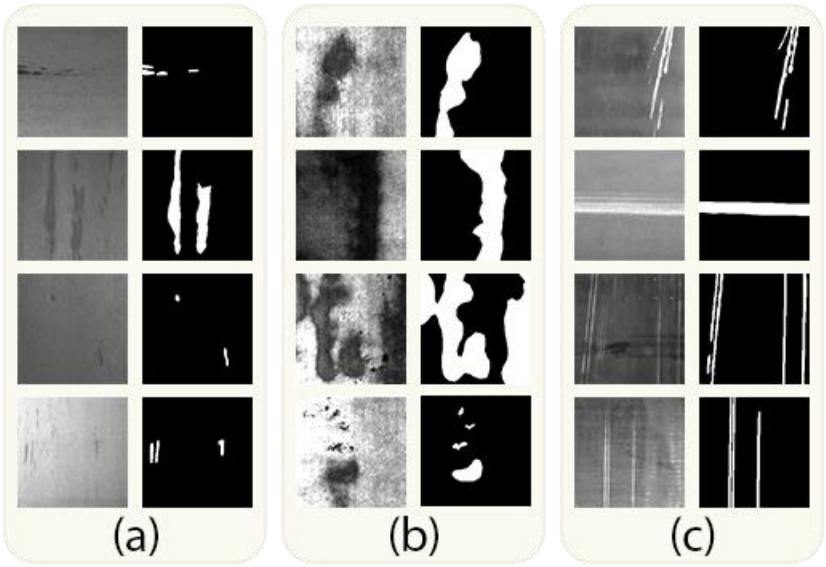


Figure 5.7: Example pictures along with their respective reference data from the SD-saliency-900 dataset, highlighting the three main types of steel surface imperfections: a) inclusions, b) patches, and c) scuffs.

| Dataset | Resolution | Images | Train | Validation | Test |
|---|---|---|---|---|---|
| SD-saliency-900 [135] | 200 × 200 | 900 | 624 | 156 | 120 |

Table 5.1: Overview of the SD-saliency-900 dataset utilized in the experiments.

### 5.3.2 Evaluation Metrics

When measuring the performance of the proposed segmentation model, I chose several vital metrics: Precision, Recall, F1 Score, and Mean Intersection over Union (mIoU). Precision focuses on the model's ability to make accurate predictions when classifying defects, ensuring minimal false alarms. Recall, on the other hand, emphasizes the model's proficiency in identifying all genuine defects, thereby reducing missed detections. The F1 Score provides a comprehensive understanding of the model's overall accuracy by striking a balance between Precision and Recall. Mean Intersection over Union

(mIoU) measures the overlap between the model's predicted segmentation and the actual ground truth, effectively gauging the model's spatial accuracy in defect detection.

### 5.3.3 Implementation Details and Training

In this section, I discuss the choices made regarding hyperparameters during the training phase. My model's architecture was developed using the TensorFlow and Keras frameworks, both of which are renowned open-source platforms dedicated to deep learning. During training, I employed a batch size of 8 and trained the model for a total of 100 epochs. To update the network's parameters, I used Adam's optimizer. As for the computational setup, the experiments were executed on an Ubuntu 20.04 system. This system was augmented with the power of an NVIDIA 80 GB GPU card and equipped with 90 GB of RAM. All these computations were performed within the Paperspace environment, ensuring a seamless and efficient training process.

### 5.3.4 Experimental Results

In this section, I present the research findings, which include both visual and quantitative insights. We have compared the model's performance with top models in the field using the SD-saliency-900 dataset as a benchmark. To ensure a fair comparison, it's crucial to note that each model underwent identical training and testing conditions. Additionally, I maintained consistency during the training phase of all models in the study by using the same set of parameters.

In Figure 5.8, I offer a visual comparison using sample ground-truth images from the SD-saliency-900 dataset's test set, placing them side by side with segmentation results from the proposed model and several existing models in the field. The initial two columns display the original images and their corresponding ground-truth segmentations. Columns

ranging from the third to the eighth feature segmentation results from renowned models, including UNet [54], LinkNet [137], FPN [138], ResUnet-a [139], Attention U-net [109], and PSPNet [55]. In contrast, the segmentation outcomes of the novel UNet3+ model are displayed in the ninth column. A closer look at Figure 5.8 reveals the diverse challenges inherent to the SD-saliency-900 dataset, especially when it comes to detecting steel surface defects. This complexity arises from a myriad of factors, such as varying types of defects, similarities in the background textures, and the sheer diversity in defect sizes. To offer a comprehensive view, the first three rows of Figure 5.8 showcase sample images that illustrate the various types of defects encountered. By evaluating the model performance using Intersection over Union (IoU) percentages, it becomes evident that UNet3+ model surpasses its counterparts in detecting all three key defect categories, namely small inclusions, patches, and linear scratches, marking its definitive edge over other state-of-the-art models. Upon analyzing various models' performance on the SD-saliency-900 dataset, certain trends and insights emerge. Both UNet and Attention U-Net, while showcasing versatility across different defect types, exhibit substantial variability in their performance metrics. This variation implies that while these models can adapt to various tasks, they may not be the optimal choice for the broad spectrum of steel defects present. On the other hand, models like LinkNet and FPN demonstrate only a moderate performance, especially when compared to architectures that are more specialized in their approach. An exception to this trend is FPN's notable excellence in detecting small inclusion defects. Such a performance underscores FPN's adeptness at integrating multiscale features, thereby effectively capturing and representing different defect scales. Further observations bring ResUnet-a into focus, a model that consistently performs well

across all defect categories. However, even with its strong performance backed by residual connections, it still doesn't match up to the superiority of the proposed model in terms of the Intersection over Union (IoU) scores. Similarly, while the PSPNet stands out with its pronounced efficiency in identifying lines and patches, it doesn't offer the same level of versatility as my model when encountering varied defect types. The consistent top performance of my model across different defect types is a testament to its robustness and adaptability. This adaptability becomes even more crucial when dealing with the diverse nature of steel defects, which can range from inclusions to patches or scratches. To further emphasize the model's capabilities, one only needs to examine rows 4, 5, and 6 of Figure 5.8. These rows provide a deep dive into the challenges posed by the background similarity, a factor that significantly elevates the intricacy of defect detection tasks. A case in point is the inclusion defect category highlighted in row 4. In this particularly challenging category, the model excels with an IoU score of 90%, highlighting its unmatched ability in defect detection even in complex backgrounds. The model's ability becomes particularly evident when compared to other top-performing models. It significantly outperforms the second-best performer, LinkNet, by a substantial margin of 7 percentage points. When delving into the nuances of patches and scratches defects, the consistent superiority of the model is unmistakable, recording IoUs of 89%. This score not only underscores the model's adeptness but also its unwavering capability to manage a diverse array of defect types, even when set against intricate backgrounds. While models like LinkNet and FPN do carve out commendable performances in some categories, they falter when it comes to delivering consistent excellence across the spectrum. Consistent and outstanding performance like this becomes the hallmark of the model, setting it apart in the field of defect detection.

Additional insights from Figure 5.8, specifically from rows 7 through 9, provide valuable information about the model's ability to deal with defects of different sizes. An example of this is the model's performance with small inclusion defects in row 7, where it achieves an IoU of 87%. This performance, when set side by side with FPN's 84%, highlights the model's superior architectural nuances, optimized for grasping features spanning different scales. Even in the patches defect category, while formidable models like LinkNet, FPN, and ResUnet-a do stake out scores in the high-80s realm, the model transcends this benchmark, reaching an IoU of 91%. This performance trajectory continues in the scratches domain as well, with the model achieving an 88% IoU, distinctively outpacing PSPNet's 86%. The key takeaway from these results is the model's consistent superiority, regardless of the type or size of defects. Such consistent supremacy suggests that the model architecture has not only been meticulously crafted but also fine-tuned to seamlessly adapt to myriad scales. The inference here is that the model likely embodies advanced multiscale feature integration techniques, rendering it more proficient than its counterparts in the comparison.

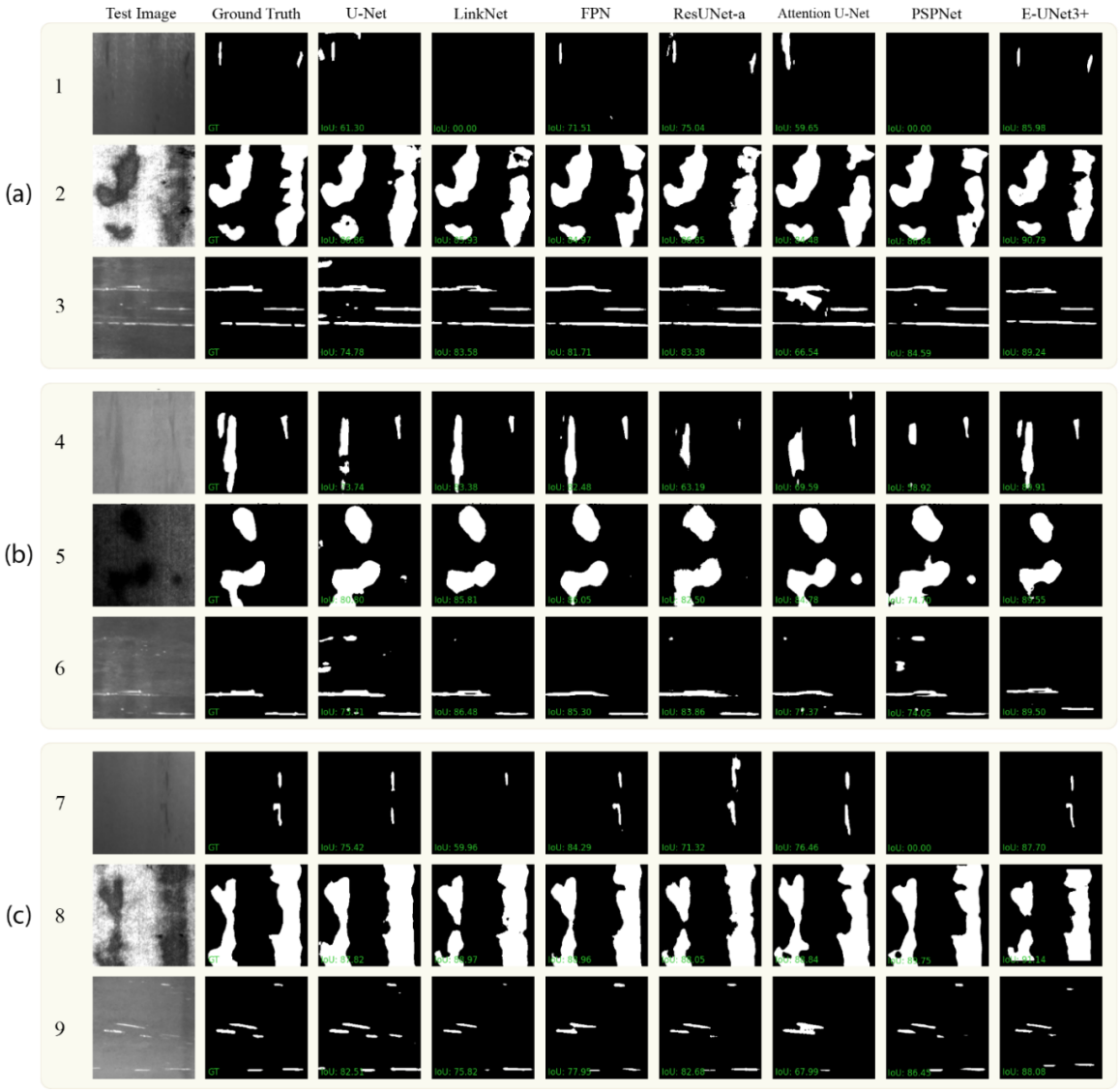


Figure 5.8: Visual representations from the SD-saliency-900 sample test set, divided into three segments: a) various defect classifications, b) resemblance in background, and c) a range in defect dimensions.

Table 5.2 offers a detailed breakdown of performance metrics emanating from the explorative ventures into steel defect detection. Standing tall amidst an array of state-of-the-art models, E-UNet3+ model notches an mIoU score of 86.19%. This metric, which evaluates both false positives and negatives across categories, holds particular weight in

the realm of segmentation tasks. Its importance is further amplified in contexts like steel defect detection, where accurate delineation of defects is more critical than just detection. While the LinkNet model showcases its ability with an admirable F1-score of 88.47%, it doesn't quite match up to my model in terms of mIoU, clocking in at 83.99%. Despite its precise detections as evinced by its high F1-score, LinkNet's performance isn't consistently exemplary across all evaluation yardsticks. This suggests that it might be grappling with the multifaceted challenges inherent to steel defect detection—varying defect typologies, sizes, and intricate background patterns. Another contender, the FPN—renowned for managing multiscale information adeptly—posts an F1-score of 86.10% and an mIoU of 82.40%. Even with its robust architectural underpinnings, it doesn't eclipse the model. Furthermore, the ResUnet-a, known for its residual connections and feature discernment capabilities, falls behind with an mIoU of 81.52%. Both Attention U-Net and PSPNet, despite their well-regarded stature, manifest mIoUs of 78.21% and 80.26% in succession, underlining certain constraints in this particular use-case. The foundational UNet architecture, while pioneering, reflects a subdued performance, registering the least mIoU at 76.89%. Its heightened recall signifies sensitivity, but this doesn't necessarily translate to pinpoint defect identification. This observation highlights the idea that models like E-UNet3+ are better suited to address contemporary challenges. In essence, E-UNet3+ combines the architectural strengths of various models, enhancing them with innovative features that enhance feature integration and representation. This assertion is supported by its outstanding performance metrics, establishing it as the most comprehensive and skilled model designed for steel defect identification.

| Model | Precision | Recall | F1-score | mIoU |
|---|---|---|---|---|
| UNet [54] | 69.36 | 95.39 | 80.32 | 76.89 |

| | | | | |
|---|---|---|---|---|
| LinkNet [137] | 87.95 | 89.00 | 88.47 | 83.99 |
| FPN [138] | 84.89 | 87.36 | 86.10 | 82.40 |
| ResUnet-a [139] | 83.29 | 88.11 | 85.63 | 81.52 |
| Attention U-net [109] | 79.58 | 86.14 | 82.73 | 78.21 |
| PSPNet [55] | 81.49 | 85.48 | 83.44 | 80.26 |
| Proposed E-UNet3+ | 86.26 | 89.63 | 87.91 | 86.19 |

Table 5.2: Numerical test results of the SD-saliency-900 dataset.

## 5.4 Discussion

This section is organized into three parts to facilitate a thorough examination. Section 5.1 delves deep into the network's intrinsic components, dissecting the influence each has on the cumulative outcome. Moving forward, Section 5.2 shifts its lens to a retrospective look at prior investigations associated with the dataset in use. The Section 5.3 engages in a critical discourse on occasions when the proposed model may falter or misidentify defects.

### 5.4.1 Component Impact Analysis

In this ablation study, I systematically assess the modifications implemented in the foundational UNet3+ model, tailored for steel defect detection. These modifications were carefully examined and their effects were analyzed using the available sd-saliency-900 dataset. The results were detailed across metrics such as Precision, Recall, F1-score, and mIoU. Table 5.3 summarizes these ablation results, highlighting a progressive improvement in the model's performance with each adjustment. My analysis highlighted that each strategic change elevated the UNet3+ model's competence, adeptly navigating the intricacies of steel defect detection. For instance, the CBAM integration marked a notable upswing in precision, albeit with a slight recall dip. However, this didn't deter the mIoU's growth, emphasizing CBAM's ability in channeling the model's attention towards pivotal features. The subsequent integration of IDS not only balanced precision and recall

but also increased the mIoU. This underscores its efficacy in preserving intricate details during the downsampling phase. The subsequent fusion of MSFLM witnessed a significant recall surge, with minimal trade-offs in precision or mIoU. MSFLM emerges as an instrumental feature, adept at identifying diverse steel defects, spanning different sizes and intricacies. DropBlock's subsequent integration marginally pulled down precision but accentuated recall and F1-score. This adjustment indicates DropBlock's proficiency in refining the model's adaptability, fostering a well-rounded performance metrically. In essence, the advanced UNet3+ model, embedding all these enhancements, triumphantly registers an unparalleled mIoU of 86.19%.

| Methods | Precision | Recall | F1-score | mIoU |
|---|---|---|---|---|
| UNet3+ | 85.43 | 89.03 | 87.20 | 84.86 |
| UNet3 + CBAM | 88.70 | 84.80 | 86.70 | 85.03 |
| UNet3 + CBAM + IDS | 87.17 | 88.10 | 87.63 | 85.87 |
| UNet3 + CBAM + IDS + MSFLM | 86.37 | 89.21 | 87.85 | 85.61 |
| UNet3 + CBAM + IDS + MSFLM + DropBlock | 86.26 | 89.63 | 87.91 | 86.19 |

Table 5.3: Ablation experimental results of SD-saliency-900 dataset.

### 5.4.2 Comparison with the Previous Studies

In my investigation, I meticulously contrasted the advanced E-UNet3+ model against benchmarks set by previous research undertakings, all leveraging the SD-saliency-dataset. The comparative insights are encapsulated in Table 5.4. It's evident that E-UNet3+ model posted a mIoU of 86.19%. Additionally, the strategy presented by [140], integrating a Residual Attention Network with bidirectional convolutional LSTM, clocked in a commendable mIoU of 82. Another technique that employed non-convex total variation regularized RPCA with kernelization exhibited an accuracy rate of 88.64 and a notable

AUC at 92.55. Significantly, none of the past endeavors on the SD-Saliency-900 dataset managed to match or exceed the mIoU achieved by E-UNet3+. This accentuates the unmatched ability of the proposed structure in steel defect detection. In a nutshell, set against prior research on the SD-Saliency-900 dataset, E-UNet3+ stands unparalleled, registering the most commendable mIoU amidst all appraised techniques.

| Reference | Methods | mIoU | Dice | Accuracy | AUC | F1-score |
|---|---|---|---|---|---|---|
| [141] | Chained atrous spatial pyramid pooling network | 78.20 | - | - | - | - |
| [142] | Depth-wise separable convolution | - | 80.80 | 95.42 | - | - |
| [140] | Residual attention network, bidirectional convolutional long short-term memory | - | 82.0 | 96.20 | - | - |
| [143] | Nonconvex total variation, regularized RPCA with kernelization | - | - | 88.64 | 92.55 | 57.87 |
| Proposed E-UNet3+ | Multiscale feature learning, attention mechanisms | 86.19 | 87.93 | 97.37 | 99.36 | 87.91 |

Table 5.4: Results of previous studies using SD-saliency-900 dataset.

### 5.4.3 Error Analysis

In E-UNet3+, we observed significant progress in detecting steel imperfections using the SD-saliency-900 dataset. Nevertheless, it's essential to examine cases where the model faced challenges and identify areas for improvement. Figure 5.9 provides insights into three scenarios where the model's detection capability seemed limited. The images in the first two rows of Figure 5.9 illustrate situations where the model detected a higher number of defects than what was documented in the reference ground truth. These detections occurred

in areas that bore a strong resemblance to genuine defects. Crucially, it merits attention that the defect variants showcased in the first couple of images are somewhat infrequent within the dataset. Such limited exposure during training might have spurred the model's amplified alertness towards these sporadic configurations. A potential remedy to this hiccup might be enriching future training cycles with more of these rare defect instances, honing the model's aptitude to discern between these atypical formations and innocuous variations. In Figure 5.9's third row, I illustrate an instance related to patches where the model clocked an IoU score of 73, pointing towards areas needing further refinement. While the proposed model has showcased proficiency, there are instances, as evidenced in Figure 5.9, where it overestimates defect regions compared to the ground truth annotations. Tackling this hurdle, future endeavors might emphasize refining the model's discernment between authentic defects and analogous image patterns. Additionally, integrating post-processing mechanisms could elevate the precision of patch-level forecasts. Infusing the training set with a broader assortment of rare defects might also bolster model generalization, thereby trimming down erroneous detections. As advancements in deep learning for defect identification continue to progress, addressing these nuances can pave the way for even more robust and accurate detection frameworks, particularly in the field of industrial applications.

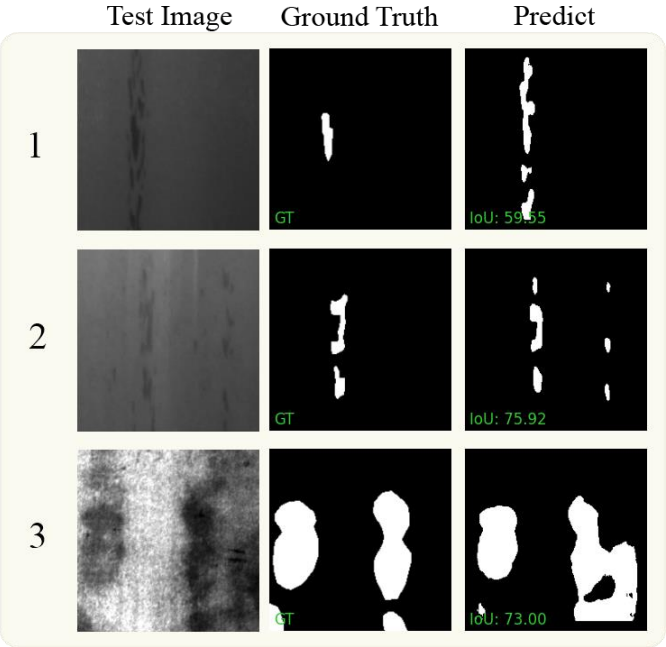


Figure 5.9: Sample challenging cases on the SD-saliency-900 dataset.

## 5.5 Conclusion

The journey towards devising adept, precise, and economical deep learning models tailored for detecting defects on steel surfaces remains a pivotal challenge, quintessential for elevating the standards and dependability of production methodologies. This research introduces E-UNet3+, a revamped variant of UNet3+, melding Multiscale Feature Learning with Attention Mechanisms, specifically designed to navigate the intricate terrain of automatic steel surface flaw identification. The input stems from meticulous refinements applied to the UNet3+ encoder. The infusion of the Multiscale Feature Learning Module capacitates the model to assimilate defect characteristics across varying scales, facilitating a nuanced grasp of multifaceted defect motifs. By deploying Conv2D layers with diverse dilation metrics, paired with DropBlock stabilization and batch normalization, the model acquires a versatile feature discernment ability. By transitioning from classic max-pooling to a novel downscaling technique utilizing strided Conv2D layers, the model achieves

superior feature discernment and portrayal. One feature of E-UNet3+ is its incorporation of the Convolutional Block Attention Module (CBAM) within the skip pathways. This enhances the model's ability to focus on relevant defect detection features. These combined improvements result in a model that excels in performance while keeping its network parameters compact. E-UNet3+'s capabilities are confirmed using the SD-saliency-900 dataset, where it outperforms contemporary models, achieving an mIoU of 86.19%. This is a testament to its proficiency in accurately identifying steel defects with unmatched precision. To encapsulate, E-UNet3+ stands as a monumental leap in automated steel surface defect identification, heralding significant cost and time savings in production contexts, ultimately ushering in augmented product caliber and streamlined industrial operations.

# Chapter 6. Conclusion and Future Work

## 6.1 Conclusion

In this thesis, I have explored and developed advanced deep learning models for quality control in manufacturing and infrastructure maintenance, focusing on fabric production, pavement crack detection, and steel surface defect detection. The culmination of this work is presented in three pivotal papers, each contributing uniquely to the field.

In Chapter 3, I addressed the complexities of fabric defect detection. The development of a texture defect classification system using a capsule-based neural network marked a significant advancement in the field. This system, enhanced by state-of-the-art convolutional neural networks (CNNs) and a spatial attention module, demonstrated an enhanced accuracy in identifying intricate and subtle defects in fabrics. The integration of these technologies not only improved the model's learning and generalization capabilities but also its feature extraction efficiency, as evidenced by a 99.42% accuracy rate on the TILDA dataset.

In Chapter 4, I introduced DepthCrackNet, a novel model designed for the critical task of pavement crack detection. This U-Net shaped model, featuring a Double Convolution Encoder, TriInput Multi-Head Spatial Attention module, and Spatial Depth Enhancer, was specifically tailored to navigate the challenges posed by crack variability and miscellaneous on-road anomalies. DepthCrackNet's performance, validated on the Crack500 and DeepCrack datasets, showed promising mIoU scores, underscoring its potential for real-world deployment in pavement maintenance and road safety.

Chapter 5 detailed the development of E-UNet3+, an enhanced version of the UNet3+ architecture for steel surface defect detection. This model, incorporating Multiscale Feature Learning and Attention Mechanisms, underwent significant architectural modifications. Notably, the introduction of the Convolutional Block Attention Module in the skip connections and the use of Conv2D layers with different dilation rates, provided a comprehensive understanding of complex defect patterns. E-UNet3+ achieved a mIoU score of 86.19% on the SD-saliency-900 dataset, outperforming existing models and demonstrating its effectiveness in high-precision defect identification.

Together, these chapters represent a significant advancement in the application of deep learning models to quality control in various industrial and infrastructural contexts. Each model exhibits a unique blend of innovative architectural features and practical application potential, setting a new standard for future research in automated defect detection and quality assurance. These advancements contribute towards more efficient, accurate, and cost-effective manufacturing and maintenance processes, ultimately ensuring higher standards of quality and safety in these critical sectors.

## 6.2 Future Work

The path ahead, illuminated by the findings and achievements of the three studies, hints at an era where I not only refine the models but also venture into territories uncharted, aligning technological advancements with pressing industry needs.

- Beginning with the textile defect detection model, real-time analysis stands out as an imperative goal. Delving deeper, it's not just about real-time feedback but also about integrating the model within the manufacturing ecosystem. This

integration involves making sure that the machinery and software communicate seamlessly, potentially pausing production lines instantaneously when a defect is detected. Further exploration could also look into multi-modal data inputs – perhaps combining visual data with sensor data from the machinery itself, creating a more comprehensive defect detection system. There's also a promising avenue in exploring adaptive learning, where the model continuously refines its accuracy by learning from any defects it might miss initially, essentially evolving with each production cycle.

- For DepthCrackNet, the future landscape is expansive. Beyond real-time operations, there's an emergent need for developing an end-to-end pavement health monitoring system. Such a system would not just detect cracks but predict their progression based on various external factors like traffic load, weather conditions, and material quality. This predictive maintenance approach could fundamentally alter urban planning and maintenance schedules. Another intriguing avenue is the potential integration with autonomous vehicles. As self-driving cars become more prevalent, they could be equipped with DepthCrackNet, transforming every vehicle into a mobile pavement inspection unit, offering continuous feedback to city maintenance departments.
- E-UNet3+ paves the way for a plethora of advancements in the realm of industrial manufacturing. The initial steps would involve diversifying the model's training with a wider array of manufacturing materials, ensuring its robustness across different production scenarios. This universality can be complemented by embedding feedback mechanisms into industrial machinery, allowing for

instantaneous corrections during manufacturing processes. Moreover, the realm of defect detection can be expanded to encompass predictive analytics. By analyzing patterns of defects over time, combined with data on production processes and raw material quality, the model could potentially predict defect occurrences, allowing preemptive measures. Furthermore, the integration of augmented reality (AR) tools could provide technicians with real-time visual insights into defects, streamlining repair and maintenance tasks.

In essence, the forward trajectory for these models transcends mere refinements. It's about creating interconnected ecosystems where deep learning models operate in harmony with machinery, human operators, and overarching industrial objectives. This holistic approach, which blends detection, prediction, and prevention, promises a future where quality assurance is not just a checkpoint but an integrated, evolving entity, driving industries towards unprecedented levels of efficiency and excellence.

# Reference


[1] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep Learning for Visual Tracking: A Comprehensive Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, May 2022, doi: 10.1109/TITS.2020.3046478.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[3] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Visual Servoing from Deep Neural Networks." arXiv, Jun. 07, 2017. Accessed: Oct. 25, 2023. [Online]. Available: http://arxiv.org/abs/1705.08940

[4] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan. 2000, doi: 10.1109/34.824819.

[5] A. Saberironaghi, J. Ren, and M. El-Gindy, "Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review," *Algorithms*, vol. 16, no. 2, p. 95, Feb. 2023, doi: 10.3390/a16020095.

[6] J. Landgraf, "Computer vision for industrial defect detection," presented at the Sheet Metal 2023, Apr. 2023, pp. 371–378. doi: 10.21741/9781644902417-46.

[7] P. M. Bhatt *et al.*, "Image-Based Surface Defect Detection Using Deep Learning: A Review," *Journal of Computing and Information Science in Engineering*, vol. 21, no. 4, p. 040801, Aug. 2021, doi: 10.1115/1.4049535.

[8] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–21, 2022, doi: 10.1109/TIM.2022.3196436.

[9] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao, "Surface Defect Detection Methods for Industrial Products: A Review," *Applied Sciences*, vol. 11, no. 16, p. 7657, Aug. 2021, doi: 10.3390/app11167657.

[10] D. Ai, G. Jiang, S.-K. Lam, P. He, and C. Li, "Computer vision framework for crack detection of civil infrastructure—A review," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105478, Jan. 2023, doi: 10.1016/j.engappai.2022.105478.

[11] D. Yapi, M. Mejri, M. S. Allili, and N. Baaziz, "A Learning-Based Approach for Automatic Defect Detection in Textile Images," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 2423–2428, Jan. 2015, doi: 10.1016/j.ifacol.2015.06.451.

[12] M. Makaremi, N. Razmjooy, and M. Ramezani, "A new method for detecting texture defects based on modified local binary pattern," *SIViP*, vol. 12, no. 7, pp. 1395–1401, Oct. 2018, doi: 10.1007/s11760-018-1294-9.

[13] Kuo, C.-Y. Shih, and J.-Y. Lee, "Automatic Recognition of Fabric Weave Patterns by a Fuzzy C-Means Clustering Method," *Textile Research Journal*, vol. 74, no. 2, pp. 107–111, Feb. 2004, doi: 10.1177/004051750407400204.

[14] M. T. N. Truong and S. Kim, “Automatic image thresholding using Otsu’s method and entropy weighting scheme for surface defect detection,” *Soft Comput*, vol. 22, no. 13, pp. 4197–4203, Jul. 2018, doi: 10.1007/s00500-017-2709-1.
[15] A. Latif-Amet, A. Ertüzün, and A. Erçil, “An efficient method for texture defect detection: sub-band domain co-occurrence matrices,” *Image and Vision Computing*, vol. 18, no. 6, pp. 543–553, May 2000, doi: 10.1016/S0262-8856(99)00062-1.
[16] L. Zhang, “Fabric Defect Classification Based on LBP and GLCM,” *JFBI*, vol. 8, no. 1, pp. 81–89, Jun. 2015, doi: 10.3993/jfbi03201508.
[17] R. A. Lizarraga-Morales, F. E. Correa-Tome, R. E. Sanchez-Yanez, and J. Cepeda-Negrete, “On the Use of Binary Features in a Rule-Based Approach for Defect Detection on Patterned Textiles,” *IEEE Access*, vol. 7, pp. 18042–18049, 2019, doi: 10.1109/ACCESS.2019.2896078.
[18] L. Yang *et al.*, “Hyperspectral image classification using wavelet transform-based smooth ordering,” *Int. J. Wavelets Multiresolut Inf. Process.*, vol. 17, no. 06, p. 1950050, Nov. 2019, doi: 10.1142/S0219691319500504.
[19] B. G. Osgood, *Lectures on the Fourier Transform and Its Applications*. American Mathematical Soc., 2019.
[20] K. L. Mak and P. Peng, “An automated inspection system for textile fabrics based on Gabor filters,” *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 359–369, Jun. 2008, doi: 10.1016/j.rcim.2007.02.019.
[21] A. Bodnarova, M. Bennamoun, and S. Latham, “Optimal Gabor filters for textile flaw detection,” *Pattern Recognition*, vol. 35, no. 12, pp. 2973–2991, Dec. 2002, doi: 10.1016/S0031-3203(02)00017-1.
[22] X. Z. Yang, “Discriminative fabric defect detection using adaptive wavelets,” *Opt. Eng*, vol. 41, no. 12, p. 3116, Dec. 2002, doi: 10.1117/1.1517290.
[23] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, “Fabric defect detection systems and methods—A systematic literature review,” *Optik*, vol. 127, no. 24, pp. 11960–11973, Dec. 2016, doi: 10.1016/j.ijleo.2016.09.110.
[24] Y. Zhang, B. Liu, X. Ji, and D. Huang, “Classification of EEG Signals Based on Autoregressive Model and Wavelet Packet Decomposition,” *Neural Process Lett*, vol. 45, no. 2, pp. 365–378, Apr. 2017, doi: 10.1007/s11063-016-9530-1.
[25] F. S. Cohen, Z. Fan, and S. Attali, “Automated inspection of textile fabrics using textural models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 803–808, Aug. 1991, doi: 10.1109/34.85670.
[26] S. Mei, Y. Wang, and G. Wen, “Automatic Fabric Defect Detection with a Multi-Scale Convolutional Denoising Autoencoder Network Model,” *Sensors*, vol. 18, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/s18041064.
[27] K. Hanbay, S. Golgiyaz, and M. F. Talu, “Real time fabric defect detection system on Matlab and C++/Opencv platforms,” in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017, pp. 1–8. doi: 10.1109/IDAP.2017.8090180.
[28] J. Jing, H. Ma, and H. Zhang, “Automatic fabric defect detection using a deep convolutional neural network,” *Coloration Technol*, vol. 135, no. 3, pp. 213–223, Jun. 2019, doi: 10.1111/cote.12394.

[29] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, “Automation of Pavement Surface Crack Detection using the Continuous Wavelet Transform,” in *2006 International Conference on Image Processing*, Oct. 2006, pp. 3037–3040. doi: 10.1109/ICIP.2006.313007.
[30] J. Zhou, P. S. Huang, and F.-P. Chiang, “Wavelet-based pavement distress detection and evaluation,” *OE*, vol. 45, no. 2, p. 027007, Feb. 2006, doi: 10.1117/1.2172917.
[31] W. Huang and N. Zhang, “A novel road crack detection and identification method using digital image processing techniques,” in *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, Dec. 2012, pp. 397–400.
[32] “Research on Crack Detection Method of Airport Runway Based on Twice-Threshold Segmentation | IEEE Conference Publication | IEEE Xplore.” Accessed: Aug. 15, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7406145
[33] W. Xu, Z. Tang, J. Zhou, and J. Ding, “Pavement crack detection based on saliency and statistical features,” in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 4093–4097. doi: 10.1109/ICIP.2013.6738843.
[34] A. Akagic, E. Buza, S. Omanovic, and A. Karabegovic, “Pavement crack detection using Otsu thresholding for image segmentation,” in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2018, pp. 1092–1097. doi: 10.23919/MIPRO.2018.8400199.
[35] R. Kapela *et al.*, “Asphalt surfaced pavement cracks detection based on histograms of oriented gradients,” in *2015 22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)*, Jun. 2015, pp. 579–584. doi: 10.1109/MIXDES.2015.7208590.
[36] H. Zakeri, F. M. Nejad, A. Fahimifar, A. D. Torshizi, and M. H. F. Zarandi, “A multi-stage expert system for classification of pavement cracking,” in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, Jun. 2013, pp. 1125–1130. doi: 10.1109/IFSA-NAFIPS.2013.6608558.
[37] “Vision for road inspection | IEEE Conference Publication | IEEE Xplore.” Accessed: Aug. 15, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6836111
[38] Y. Maode, B. Shaobo, X. Kun, and H. Yuyao, “Pavement Crack Detection and Analysis for High-grade Highway,” in *2007 8th International Conference on Electronic Measurement and Instruments*, Aug. 2007, pp. 4-548-4–552. doi: 10.1109/ICEMI.2007.4351202.
[39] V. Kaul, A. Yezzi, and Y. Tsai, “Detecting Curves with Unknown Endpoints and Arbitrary Topology Using Minimal Paths,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1952–1965, Oct. 2012, doi: 10.1109/TPAMI.2011.267.
[40] H. Li, D. Song, Y. Liu, and B. Li, “Automatic Pavement Crack Detection by Multi-Scale Image Fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2025–2036, Jun. 2019, doi: 10.1109/TITS.2018.2856928.

[41] "Crack Segmentation by Leveraging Multiple Frames of Varying Illumination | IEEE Conference Publication | IEEE Xplore." Accessed: Aug. 15, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7926704
[42] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.
[43] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection," *IEEE Trans. Intell. Transport. Syst.*, vol. 21, no. 4, pp. 1525–1535, Apr. 2020, doi: 10.1109/TITS.2019.2910595.
[44] D. Mazzini, P. Napoletano, F. Piccoli, and R. Schettini, "A Novel Approach to Data Augmentation for Pavement Distress Segmentation," *Computers in Industry*, vol. 121, p. 103225, Oct. 2020, doi: 10.1016/j.compind.2020.103225.
[45] "Semi-Supervised Semantic Segmentation Using Adversarial Learning for Pavement Crack Detection | IEEE Journals & Magazine | IEEE Xplore." Accessed: Aug. 15, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9032091
[46] J. S. Lee, S. H. Hwang, I. Y. Choi, and Y. Choi, "Estimation of crack width based on shape-sensitive kernels and semantic segmentation," *Structural Control and Health Monitoring*, vol. 27, no. 4, p. e2504, 2020, doi: 10.1002/stc.2504.
[47] S. Wang, X. Wu, Y. Zhang, X. Liu, and L. Zhao, "A neural network ensemble method for effective crack segmentation using fully convolutional networks and multi-scale structured forests," *Machine Vision and Applications*, vol. 31, no. 7, p. 60, Sep. 2020, doi: 10.1007/s00138-020-01114-0.
[48] "Sensors | Free Full-Text | Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique." Accessed: Aug. 15, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3452
[49] W. Liu and Y. Yan, "Automated surface defect detection for cold-rolled steel strip based on wavelet anisotropic diffusion method," *IJISE*, vol. 17, no. 2, p. 224, 2014, doi: 10.1504/IJISE.2014.061995.
[50] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440. Accessed: Aug. 15, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html
[51] F. Gayubo, J. L. Gonzalez, E. de la Fuente, F. Miguel, and J. R. Peran, "On-line machine vision system for detect split defects in sheet-metal forming processes," in *18th International Conference on Pattern Recognition (ICPR'06)*, Aug. 2006, pp. 723–726. doi: 10.1109/ICPR.2006.902.
[52] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818. Accessed: Sep. 12, 2023. [Online]. Available:

https://openaccess.thecvf.com/content_ECCV_2018/html/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.html
[53] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
[54] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.
[55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890. Accessed: Sep. 12, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html
[56] "An End-to-End Neural Network for Road Extraction From Remote Sensing Imagery by Multiple Feature Pyramid Network | IEEE Journals & Magazine | IEEE Xplore." Accessed: Sep. 12, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8410903
[57] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141. Accessed: Aug. 15, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html
[58] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 286–301. Accessed: Sep. 12, 2023. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Yulun_Zhang_Image_Super-Resolution_Using_ECCV_2018_paper.html
[59] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19. Accessed: Sep. 12, 2023. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Sanghyun_Woo_Convolutional_Block_Attention_ECCV_2018_paper.html
[60] J. Fu *et al.*, "Dual Attention Network for Scene Segmentation," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3146–3154. Accessed: Sep. 12, 2023. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Fu_Dual_Attention_Network_for_Scene_Segmentation_CVPR_2019_paper.html
[61] F. Liu, G. Lin, and C. Shen, "CRF learning with CNN features for image segmentation," *Pattern Recognition*, vol. 48, no. 10, pp. 2983–2992, Oct. 2015, doi: 10.1016/j.patcog.2015.04.019.

[62] A. Rasheed *et al.*, “Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–24, Nov. 2020, doi: 10.1155/2020/8189403.
[63] V. Tiwari and G. Sharma, “Automatic Fabric Fault Detection Using Morphological Operations on Bit Plane,” *International Journal of Engineering Research*, vol. 2, no. 10, 2013.
[64] Y. Li and C. Zhang, “Automated vision system for fabric defect inspection using Gabor filters and PCNN,” *SpringerPlus*, vol. 5, no. 1, p. 765, Jun. 2016, doi: 10.1186/s40064-016-2452-6.
[65] L. Bissi, G. Baruffa, P. Placidi, E. Ricci, A. Scorzoni, and P. Valigi, “Automated defect detection in uniform and structured fabrics using Gabor filters and PCA,” *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 838–845, Oct. 2013, doi: 10.1016/j.jvcir.2013.05.011.
[66] Y. Li, W. Zhao, and J. Pan, “Deformable Patterned Fabric Defect Detection With Fisher Criterion-Based Deep Learning,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, Apr. 2017, doi: 10.1109/TASE.2016.2520955.
[67] T. Tuncer and S. Dogan, “Pyramid and multi kernel based local binary pattern for texture recognition,” *J Ambient Intell Human Comput*, vol. 11, no. 3, pp. 1241–1252, Mar. 2020, doi: 10.1007/s12652-019-01306-1.
[68] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” arXiv, Nov. 06, 2014. doi: 10.48550/arXiv.1411.1792.
[69] “Spatial Attention - an overview | ScienceDirect Topics.” Accessed: Dec. 14, 2023. [Online]. Available: https://www.sciencedirect.com/topics/engineering/spatial-attention
[70] J. Ren, H. A. Gabbar, X. Huang, and A. Saberironaghi, “Defect Detection for Printed Circuit Board Assembly Using Deep Learning,” in *2022 8th International Conference on Control Science and Systems Engineering (ICCSSE)*, Jul. 2022, pp. 85–89. doi: 10.1109/ICCSSE55346.2022.10079777.
[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.
[72] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks.” arXiv, Jan. 28, 2018. doi: 10.48550/arXiv.1608.06993.
[73] “Computer Vision Group, Freiburg.” Accessed: Jul. 31, 2023. [Online]. Available: https://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html
[74] K. Mukherjee, A. Khare, and A. Verma, “A Simple Dynamic Learning Rate Tuning Algorithm For Automated Training of DNNs.” arXiv, Oct. 25, 2019. Accessed: Dec. 14, 2023. [Online]. Available: http://arxiv.org/abs/1910.11605
[75] F. Chollet, “Xception: Deep Learning With Depthwise Separable Convolutions,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258. Accessed: Aug. 17, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

[76] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training.” arXiv, Jun. 23, 2021. doi: 10.48550/arXiv.2104.00298.
[77] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” arXiv, Apr. 16, 2017. doi: 10.48550/arXiv.1704.04861.
[78] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. Accessed: Aug. 15, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
[79] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 6105–6114. Accessed: Aug. 15, 2023. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html
[80] Y. Ben Salem and S. Nasri, “Woven fabric defects detection based on texture classification algorithm,” 2011.
[81] Y. Ben Salem and M. N. Abdelkrim, “Texture classification of fabric defects using machine learning,” *IJECE*, vol. 10, no. 4, p. 4390, Aug. 2020, doi: 10.11591/ijece.v10i4.pp4390-4399.
[82] N. T. Deotale and T. K. Sarode, “Fabric Defect Detection Adopting Combined GLCM, Gabor Wavelet Features and Random Decision Forest,” *3D Res*, vol. 10, no. 1, p. 5, Jan. 2019, doi: 10.1007/s13319-019-0215-1.
[83] P. R. Jeyaraj and E. R. Samuel Nadar, “Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm,” *International Journal of Clothing Science and Technology*, vol. 31, no. 4, pp. 510–521, Jan. 2019, doi: 10.1108/IJCST-11-2018-0135.
[84] S. S. Adlinge and A. K. Gupta, “Pavement Deterioration and its Causes”.
[85] T. R. Miller and E. Zaloshnja, “Cost of crashes related to road conditions,” presented at the 53rd Annual Scientific Conference of the Association for the Advancement of Automotive Medicine, Baltimore, Maryland, 4-7 October 2009, 2009. Accessed: Oct. 25, 2023. [Online]. Available: https://trid.trb.org/view/1149962
[86] M. S. Kaseko and S. G. Ritchie, “A neural network-based methodology for pavement crack detection and classification,” *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 4, pp. 275–291, Dec. 1993, doi: 10.1016/0968-090X(93)90002-W.
[87] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, “An innovative methodology for detection and quantification of cracks through incorporation of depth perception,” *Machine Vision and Applications*, vol. 24, no. 2, pp. 227–241, Feb. 2013, doi: 10.1007/s00138-011-0394-0.
[88] R. Fan *et al.*, “Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France: IEEE, Jun. 2019, pp. 474–479. doi: 10.1109/IVS.2019.8814000.
[89] F. Liu, G. Xu, Y. Yang, X. Niu, and Y. Pan, “Novel Approach to Pavement Cracking Automatic Detection Based on Segment Extending,” in *2008*

*International Symposium on Knowledge Acquisition and Modeling*, Dec. 2008, pp. 610–614. doi: 10.1109/KAM.2008.29.
[90] R. Medina, J. Llamas, E. Zalama, and J. Gómez-García-Bermejo, "Enhanced automatic detection of road surface cracks by combining 2D/3D image processing techniques," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 778–782. doi: 10.1109/ICIP.2014.7025156.
[91] K. Fernandes and L. Ciobanu, "Pavement pathologies classification using graph-based features," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 793–797. doi: 10.1109/ICIP.2014.7025159.
[92] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, Feb. 2012, doi: 10.1016/j.patrec.2011.11.004.
[93] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic Road Crack Detection Using Random Structured Forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016, doi: 10.1109/TITS.2016.2552248.
[94] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016, doi: 10.1109/TITS.2015.2477675.
[95] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila, "Free-form anisotropy: A new method for crack detection on pavement surface images," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 1069–1072. doi: 10.1109/ICIP.2011.6115610.
[96] H. Oliveira and P. L. Correia, "Automatic Road Crack Detection and Characterization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, Mar. 2013, doi: 10.1109/TITS.2012.2208630.
[97] L. Pauly, D. Hogg, R. Fuentes, and H. Peel, "Deeper Networks for Pavement Crack Detection," Proceedings of the 34th ISARC. Accessed: Aug. 15, 2023. [Online]. Available: https://eprints.whiterose.ac.uk/120380/
[98] M. Eisenbach *et al.*, "How to get pavement distress detection ready for deep learning? A systematic approach," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA: IEEE, May 2017, pp. 2039–2047. doi: 10.1109/IJCNN.2017.7966101.
[99] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.
[100] S. K. G. Manikonda and D. N. Gaonkar, "A Novel Islanding Detection Method Based on Transfer Learning Technique Using VGG16 Network," in *2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)*, Feb. 2019, pp. 109–114. doi: 10.1109/ICSETS.2019.8744778.
[101] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3D Deep Learning on Medical Images: A Review." arXiv, Oct. 13, 2020. doi: 10.48550/arXiv.2004.00218.
[102] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An Attentive Survey of Attention Models." arXiv, Jul. 12, 2021. Accessed: Aug. 15, 2023. [Online]. Available: http://arxiv.org/abs/1904.02874

[103] C. Tao, S. Gao, M. Shang, W. Wu, D. Zhao, and R. Yan, *Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism*. 2018, p. 4424. doi: 10.24963/ijcai.2018/614.
[104] C. Xi, G. Lu, and J. Yan, "Multimodal sentiment analysis based on multi-head attention mechanism," in *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, in ICMLSC '20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 34–39. doi: 10.1145/3380688.3380693.
[105] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust Visual Tracking via Hierarchical Convolutional Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, Nov. 2019, doi: 10.1109/TPAMI.2018.2865311.
[106] M. Imani and H. Ghassemian, "An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges," *Information Fusion*, vol. 59, pp. 59–83, Jul. 2020, doi: 10.1016/j.inffus.2020.01.007.
[107] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019, doi: 10.1016/j.neucom.2019.01.036.
[108] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation." arXiv, May 29, 2018. doi: 10.48550/arXiv.1802.06955.
[109] O. Oktay *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," Apr. 2018.
[110] J. Chen *et al.*, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation." arXiv, Feb. 08, 2021. doi: 10.48550/arXiv.2102.04306.
[111] H. Cao *et al.*, "Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation." arXiv, May 12, 2021. doi: 10.48550/arXiv.2105.05537.
[112] J. Zhang, R. Ding, M. Ban, and T. Guo, "FDSNeT: An Accurate Real-Time Surface Defect Segmentation Network," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 3803–3807. doi: 10.1109/ICASSP43922.2022.9747311.
[113] J. C. Ong, S. L. Lau, M.-Z. Ismadi, and X. Wang, "Feature pyramid network with self-guided attention refinement module for crack segmentation," *Structural Health Monitoring*, vol. 22, no. 1, pp. 672–688, Jan. 2023, doi: 10.1177/14759217221089571.
[114] X. Sun, Y. Xie, L. Jiang, Y. Cao, and B. Liu, "DMA-Net: DeepLab With Multi-Scale Attention for Pavement Crack Segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18392–18403, Oct. 2022, doi: 10.1109/TITS.2022.3158670.
[115] G. Doğan and B. Ergen, "A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection," *Measurement*, vol. 195, p. 111119, May 2022, doi: 10.1016/j.measurement.2022.111119.
[116] G. Yu, J. Dong, Y. Wang, and X. Zhou, "RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation," *Sensors*, vol. 23, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/s23010053.

[117] J. Pang, H. Zhang, H. Zhao, and L. Li, “DcsNet: a real-time deep network for crack segmentation,” *SIViP*, vol. 16, no. 4, pp. 911–919, Jun. 2022, doi: 10.1007/s11760-021-02034-w.
[118] W. Wang and C. Su, “Convolutional Neural Network-Based Pavement Crack Segmentation Using Pyramid Attention Network,” *IEEE Access*, vol. 8, pp. 206548–206558, 2020, doi: 10.1109/ACCESS.2020.3037667.
[119] L. Jiang, Y. Xie, and T. Ren, “A DEEP NEURAL NETWORKS APPROACH FOR PIXEL-LEVEL RUNWAY PAVEMENT CRACK SEGMENTATION USING DRONE-CAPTURED IMAGES”.
[120] S. Gupta, S. Shrivastwa, S. Kumar, and A. Trivedi, “Self-attention-Based Efficient U-Net for Crack Segmentation,” in *Computer Vision and Robotics*, P. K. Shukla, K. P. Singh, A. K. Tripathi, and A. Engelbrecht, Eds., in Algorithms for Intelligent Systems. Singapore: Springer Nature, 2023, pp. 103–114. doi: 10.1007/978-981-19-7892-0_9.
[121] S. Jia, “Semantic segmentation of pavement cracks based on an improved U-Net,” *Journal of Computing and Electronic Information Management*, vol. 10, no. 3, Art. no. 3, May 2023, doi: 10.54097/jceim.v10i3.8672.
[122] M. Cheng, K. Zhao, X. Guo, Y. Xu, and J. Guo, “Joint Topology-preserving and Feature-refinement Network for Curvilinear Structure Segmentation,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 7127–7136. doi: 10.1109/ICCV48922.2021.00706.
[123] C. Gu, Y. Lu, M. Chen, G. Sun, and Z. Ni, “A reweighting offset bin classification network for surface defect detection and location of metal components,” *Measurement*, vol. 187, p. 110166, Jan. 2022, doi: 10.1016/j.measurement.2021.110166.
[124] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, “Automated Visual Defect Detection for Flat Steel Surface: A Survey,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 626–644, Mar. 2020, doi: 10.1109/TIM.2019.2963555.
[125] D. M. Sime, G. Wang, Z. Zeng, and B. Peng, “Deep learning-based automated steel surface defect segmentation: a comparative experimental study,” *Multimed Tools Appl*, May 2023, doi: 10.1007/s11042-023-15307-y.
[126] B. Guo, Y. Wang, S. Zhen, R. Yu, and Z. Su, “SPEED:Semantic Prior and Extremely Efficient Dilated Convolution Network for Real-Time Metal Surface Defects Detection,” *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2023, doi: 10.1109/TII.2022.3233674.
[127] D. M. Sime, G. Wang, Z. Zeng, W. Wang, and B. Peng, “Semisupervised Defect Segmentation With Pairwise Similarity Map Consistency and Ensemble-Based Cross Pseudolabels,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 9, pp. 9535–9545, Sep. 2023, doi: 10.1109/TII.2022.3230785.
[128] D. Djukic and S. Spuzic, “Statistical discriminator of surface defects on hot rolled steel”.
[129] J. P. Yun, S. Choi, and S. W. Kim, “Vision-based defect detection of scale-covered steel billet surfaces,” *OE*, vol. 48, no. 3, p. 037205, Mar. 2009, doi: 10.1117/1.3102066.

[130] D. Choi, Y. Jeon, J. P. Yun, and S. W. Kim, “Pinhole detection in steel slab images using Gabor filter and morphological features,” *Appl. Opt., AO*, vol. 50, no. 26, pp. 5122–5129, Sep. 2011, doi: 10.1364/AO.50.005122.
[131] J. P. Yun, S. Choi, J.-W. Kim, and S. W. Kim, “Automatic detection of cracks in raw steel block using Gabor filter optimized by univariate dynamic encoding algorithm for searches (uDEAS),” *NDT & E International*, vol. 42, no. 5, pp. 389–397, Jul. 2009, doi: 10.1016/j.ndteint.2009.01.007.
[132] “A Steel Surface Defect Recognition Algorithm Based on Improved Deep Learning Network Model Using Feature Visualization and Quality Evaluation | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Sep. 12, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9031427
[133] Y. He, K. Song, H. Dong, and Y. Yan, “Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network,” *Optics and Lasers in Engineering*, vol. 122, pp. 294–302, Nov. 2019, doi: 10.1016/j.optlaseng.2019.06.020.
[134] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, “A New Steel Defect Detection Algorithm Based on Deep Learning,” *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–13, Mar. 2021, doi: 10.1155/2021/5592878.
[135] G. Song, K. Song, and Y. Yan, “EDRNet: Encoder–Decoder Residual Network for Salient Object Detection of Strip Steel Surface Defects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9709–9719, Dec. 2020, doi: 10.1109/TIM.2020.3002277.
[136] H. Huang *et al.*, “UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 1055–1059. doi: 10.1109/ICASSP40776.2020.9053405.
[137] A. Chaurasia and E. Culurciello, “LinkNet: Exploiting encoder representations for efficient semantic segmentation,” in *2017 IEEE Visual Communications and Image Processing (VCIP)*, Dec. 2017, pp. 1–4. doi: 10.1109/VCIP.2017.8305148.
[138] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125. Accessed: Sep. 12, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.html
[139] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, “ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, Apr. 2020, doi: 10.1016/j.isprsjprs.2020.01.013.
[140] L. Yang, S. Xu, J. Fan, E. Li, and Y. Liu, “A pixel-level deep segmentation network for automatic defect detection,” *Expert Systems with Applications*, vol. 215, p. 119388, Apr. 2023, doi: 10.1016/j.eswa.2022.119388.
[141] Z. Zheng *et al.*, “CASPPNet: a chained atrous spatial pyramid pooling network for steel defect detection,” *Meas. Sci. Technol.*, vol. 33, no. 8, p. 085403, Aug. 2022, doi: 10.1088/1361-6501/ac68d2.

[142] Z. Huang, J. Wu, and F. Xie, “Automatic surface defect segmentation for hot-rolled steel strip using depth-wise separable U-shape network,” *Materials Letters*, vol. 301, p. 130271, Oct. 2021, doi: 10.1016/j.matlet.2021.130271.
[143] “Surface Defects Detection Using Non-convex Total Variation Regularized RPCA With Kernelization | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Sep. 12, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9346005