Railway Track Fault Detection using AI

By

Muhammad Haroon

(Registration No: 00000329026)

Department of Robotics & Artificial Intelligence

School of Mechanical and Manufacturing Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

Railway Track Fault Detection using AI

By

Muhammad Haroon

(Registration No: 00000329026)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in

Robotics and Intelligent Machine Engineering

Supervisor: Dr. Muhammad Tauseef Nasir

Co Supervisor: Dr. Hammad Mehmood Cheema

School of Mechanical and Manufacturing Engineering

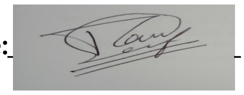
National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

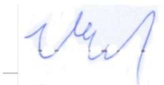
THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by **Regn No. 00000329026 Muhammad Haroon** of **School of Mechanical & Manufacturing Engineering (SMME)** has been vetted by undersigned, found complete in all respects as per NUST Statues/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis titled. **Railway Track Fault Detection using AI**

Signature: 

Name (Supervisor): Muhammad Tauseef Nasir

Date: 26 - Aug - 2024

Signature (HOD): 

Date: 26 - Aug - 2024

Signature (DEAN): 

Date: 26 - Aug - 2024

Form TH-4

National University of Sciences & Technology (NUST)

MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: Muhammad Haroon (00000329026) Titled: Railway Track Fault Detection using AI be accepted in partial fulfillment of the requirements for the award of MS in Robotics & Intelligent Machine Engineering degree.

Examination Committee Members

1. Name: Hammad Mehmood Cheema Signature:
2. Name: Shahid Ikram Ullah Butt Signature:
3. Name: Muhammad Safdar Signature:

Supervisor: Muhammad Tauseef Nasir Signature:

Date: 26 - Aug - 2024

26 - Aug - 2024

Head of Department Date

COUNTERSINGED

26 - Aug - 2024

Date Dean/Principal

<u>CERTIFICATE OF APPROVAL</u>

This is to certify that the research work presented in this thesis, entitled "**Railway Track Fault Detection Using AI**" was conducted by Mr. **Muhammad Haroon** under the supervision of **Dr. Muhammad Tauseef Nasir**.

No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Robotics & Artificial Intelligence** in partial fulfillment of the requirements for the degree of Master of Science in the Field of **Robotics and Intelligent Machine Engineering** at **School of Mechanical and Manufacturing Engineering,** National University of Sciences and Technology, Islamabad.

Student Name: <u>Muhammad Haroon</u> Signature: ________________

Examination Committee:

a) External Examiner 1: Dr. Shahid Ikram Ullah Signature: ________________
(Professor & HoD DME)
……………………………….

b) External Examiner 2: Dr. Muhammad Safdar Signature: ________________
(Associate Professor)
………………………………

Supervisor Name: Dr. Muhammad Tauseef Nasir Signature: ________________

Co-Supervisor Name: Dr. Hammad Mehmood Cheema Signature: ________________

Name of Dean/HOD: Dr. Kunwar Faraz Signature: ________________

AUTHOR's DECLARATION

I certify that this research work titled "*Railway Track Fault Detection using AI"* is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Signature of Student

Muhammad Haroon

2020-NUST-MS-R&AI-329026

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Railway Track Fault Detection using AI**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: ____________________________

Name: Muhammad Haroon

DEDICATION

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

ACKNOWLEDGEMENTS

In the infinite grace and wisdom of Allah Subhana-Watala, I find my strength and guidance. It is with His divine light that I have been blessed and guided throughout this academic Endeavor and in all facets of my life. To Him, I owe all gratitude, for it is through His will that I have been fortunate to be surrounded by people who have been pillars of support and guidance.

My deepest appreciation goes to my parents who have been an ocean of love, resilience, and support. Their unconditional faith in me and prayers have been the bedrock on which I have built my aspirations.

To my dear siblings, who have walked with me through thick and thin, your belief in me and your constant encouragement have been invaluable. I am also profoundly grateful to my supervisor, Dr. Muhammad Tauseef Nasir, who has been a mentor for excellence. His expertise, insights, and dedication have significantly shaped this work. Throughout this thesis, he provided invaluable guidance, constructive criticism, and a reservoir of patience. His unwavering support and wisdom have been pivotal in turning my aspirations into reality.

I would also like to extend my heartfelt thanks to my previous supervisor, Dr. Muhammad Jawad Khan, who taught me the foundational skills of conducting research, and to my Co-Supervisor, Dr. Hammad Mehmood Cheema, whose valuable insights have guided me from the very start of this thesis. Their combined guidance has been instrumental in shaping the direction and quality of my research.

A special note of thanks to my friends who have been an enduring source of safety, laughter, and reassurance. Their constant presence and belief in my capabilities, especially during moments of self-doubt, have been immensely uplifting.

Lastly, everybody who has contributed, even in the slightest way, to my academic journey has my heartfelt gratitude. Whether it was a word of encouragement, a gesture of support, or a piece of advice, I cherish it deeply. To all, may Allah shower His countless blessings upon you, and may He guide us in all our endeavors.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS 8

TABLE OF CONTENTS 9

LIST OF FIGURES 12

LIST OF TABLES 13

ABSTRACT 14

Chapter 1: Introduction 15

- **1.1 Importance of Rail Transport** 15
- **1.2 Railway Track and Components** 17
- **1.3 Rail Track Defects** 18
- **1.4 Motivation** 19
- **1.5 Object Detection** 20
  - **1.5.1 Supervised Learning** 20
  - **1.5.2 Self-Supervised Learning** 20
- **1.6 Problem Statement** 21
- **1.7 Objectives** 21
- **1.8 Proposed Solution** 21
- **1.9 Thesis Organization** 22

Chapter 2: Literature Review 23

- **2.1 Image Processing Based Approaches** 23
- **2.2 Supervised Learning Approaches** 23
- **2.3 Self and Semi Supervised Learning Approaches** 24
- **2.4 Research Gap** 25

Chapter 3: Proposed System 26

- **3.1 Dataset** 26
- **3.2 Deep Learning Models** 26
  - **3.2.1 Convolutional Neural Networks (CNNs)** 26
  - **3.2.2 YOLOv8** 27
  - **3.2.3 Faster R-CNN** 29
  - **3.2.4 EfficientNet** 30
  - **3.2.5 U-Net** 31

3.3 Proposed System Diagram 32
Chapter 4: Implementation 34
4.1 Data Collection and Preprocessing 34
4.1.1 Supervised Learning Dataset 34
Custom Collected Dataset: 34
Supplementary Dataset: 35
Dataset Preprocessing: 35
4.1.2 Self-Supervised Learning Dataset 37
Dataset Creation: 37
Image Selection: 38
Dataset Statistics: 38
Fine Tuning and Evaluation Dataset: 38
4.2 Supervised Learning Model 39
4.2.1 Model Selection 40
4.2.2 Model Training Parameters 41
4.2.3 Post Processing 43
4.3 Self Supervised Learning Model 44
4.3.1 Rail Extractor 45
4.3.2 Model Training 46
Stage 1: Pre-Training on Normal Rail Surface Images 46
Stage 2: Fine-Tuning on Rail Surface Defect Dataset (RSDDs) 47
4.3.3 Model Parameters 47
4.3.4 Evaluation Metrics 49
Dice Loss 49
Intersection over Union (IoU) 50
Chapter 5: Results and Discussion 51
5.1 Supervised Learning Model Results 51
5.1.1 Comparison with Other Object Detector Models 51
5.1.2 Comparison with Custom Trained Models 52
5.1.3 Model Validation 53
5.1.4 Performance on Each Class 54
Bounding Box Metrics: 54
Segmentation Metrics: 54

5.1.5 Plots 55
Training and Validation Losses 55
Precision, Recall, and mAP Metrics 56
5.1.6 Confusion Matrix 57
5.2 Self-Supervised Learning Model Results 59
5.2.1 Comparison with Custom Trained Models on Defect Dataset 60
5.2.2 Comparison with Other Research on RSDDs Dataset 62
5.2.3 Outputs on Custom Dataset 63
5.2.4 Plots 64
Training and Validation Loss 64
Training and Validation IoU 65
Chapter 6: Conclusion and Future Work 66
6.1 Conclusion 66
6.2 Future Work 67
References 68

LIST OF FIGURES

Figure 1: Various Railway Accident Pictures. 15

Figure 2: Railway Accidents in Pakistan Source: Pakistan Railways. 16

Figure 3: A Railway Track (Rawalpindi). 17

Figure 4: Components of Rail Track. 18

Figure 5: Different Defects in Rail Tracks. 19

Figure 6: Supervised Learning. 20

Figure 7: Self-Supervised Learning. 21

Figure 8: The simplest representation of CNN Architecture. 27

Figure 9: Simplified YOLOv8 Architecture. 29

Figure 10: U-Net Architecture. 32

Figure 11: Proposed System Diagram. 33

Figure 12: Images from Custom Dataset. 34

Figure 13: Images from Kaggle Dataset. 35

Figure 14: Images with their Augmented Pairs. 36

Figure 15: a) Image from Custom Dataset, b) Extracted Rail Part of the Image. 38

Figure 16: Rail surface images with their ground-truth images showing defected regions from RSDDs Dataset. 39

Figure 17: Workflow of Supervised Learning Model. 40

Figure 18: YOLOv8 – Segment Model Result Before NMS (Left) and After NMS (Right). 44

Figure 19: Self Supervised Model Workflow. 45

Figure 20: Segmentation results of Custom Trained YOLOv8 Segment model on Custom Dataset. 53

Figure 21: Custom trained YOLOv8 Segment Model Performance on Each Class Detection. 54

Figure 22: Training and Validation Plots of Custom Trained YOLOv8 Segment Model. 55

Figure 23: Confusion Matrix of Validation Dataset. 58

Figure 24: Normalized Confusion Matrix of Custom Trained YOLOv8 Segment Model. 59

Figure 25: (1) Input Image, (2) Groundtruth, (3) U-Net Output, (4) U-Net (50 Epoch) Output, (5) U-Net (250) Epoch Output. 61

Figure 26: (Left): Input Custom Rail Surface Image, Fig (Right): U-Net (250 Epochs) Segmentation Output. 63

Figure 27: Training and Validation Dice Loss and IOU of 250 Epochs Fine Tuned U-Net Model... 64

LIST OF TABLES

Table 1: Dataset Detail. **36**
Table 2: Class Distribution. **37**
Table 3: YOLOv8x – Segment Model Training Parameters. **42**
Table 4: U-Net Model Training Parameters. **48**
Table 5: Comparison of Selected Model with other state-of-the-art Object Detectors. **51**
Table 6: Comparison of Selected Model with other YOLO variants. **52**
Table 7: Comparison of different variants of Custom Trained U-Net Models. **60**
Table 8: Comparison using RSDDs Dataset with Other Research. **62**
Table 9: Comparison of Dice Coefficient and IoU between Different Studies. **63**

ABSTRACT

Railway infrastructure is critical for the safe and efficient transportation of goods and people. However, defects in rail tracks can lead to severe accidents, resulting in significant human and financial losses. This thesis presents the development of an end-to-end railway track defect detection system utilizing advanced deep learning techniques. The proposed system is designed to detect various defects, including rail surface anomalies and component defects such as missing fasteners, bolts, and fishplates. The approach combines a supervised YOLOv8x Segment Model and a self-supervised U-Net model. The YOLOv8x-segment model, trained on a curated dataset, achieved a mean Average Precision (mAP) of 94% and demonstrated its capability for real-time detection with high frames per second of 30 FPS. To overcome the issue of scarcity of labeled rail surface defects dataset, the U-Net model was pre-trained on normal rail images and fine-tuned using a subset of Rail Surface Defect Dataset (RSDDs), resulting in significant improvements in defect segmentation accuracy. The system demonstrated strong performance across various metrics and datasets, providing a reliable tool for enhancing railway safety through automated defect detection. Future work includes expanding detection capabilities to include sleepers and ballast and real-time implementation with live video feeds.

Key Words: *Railway Track Defect Detection, Deep Learning, YOLOv8, U-Net, Rail Surface Anomalies, Rail Component Defects, Automated Inspection, Self-Supervised Learning, Object Detection, Object Segmentation, Infrastructure Safety*

Chapter 1: Introduction

1.1 Importance of Rail Transport

Rail transport is a backbone of economy for many nations, providing an essential means of moving goods and people efficiently across vast distances. Its role in supporting the economy cannot be overstated, as it enables trade, connects remote regions, and supports the daily commute of millions. For countries like Pakistan, the railway network is not just a mode of transport; it is a lifeline that binds the nation together. Spanning over 11,881 kilometers (about 7382.51 mi), railway system of Pakistan is crucial for both economic activity and national connectivity (Naimat Khan, 2023). However, the safety and reliability of this system are paramount, as any disruptions can have far-reaching consequences. Ensuring the maintenance and integrity of railway infrastructure is, therefore, not just a matter of operational efficiency but also of public safety.

Figure 1: Various Railway Accident Pictures (Naimat Khan, 2023).

Tragically, the railway system of Pakistan has been plagued by numerous accidents over the years as shown in Figure 1, highlighting the critical need for improved safety measures. One of the most recent and devastating incidents was a train fire that resulted in the loss of over 30 lives, sparking widespread outrage and calls for reform (Naimat Khan, 2023). To give the overview, between August 2018 and June 2019, the frequency of railway accidents remained alarmingly high, with several incidents leading to significant fatalities and injuries, as shown in Figure 2 (BBC Reality Check, 2021). The incident underscored the vulnerabilities within the railway system and the urgent need for more stringent safety protocols.

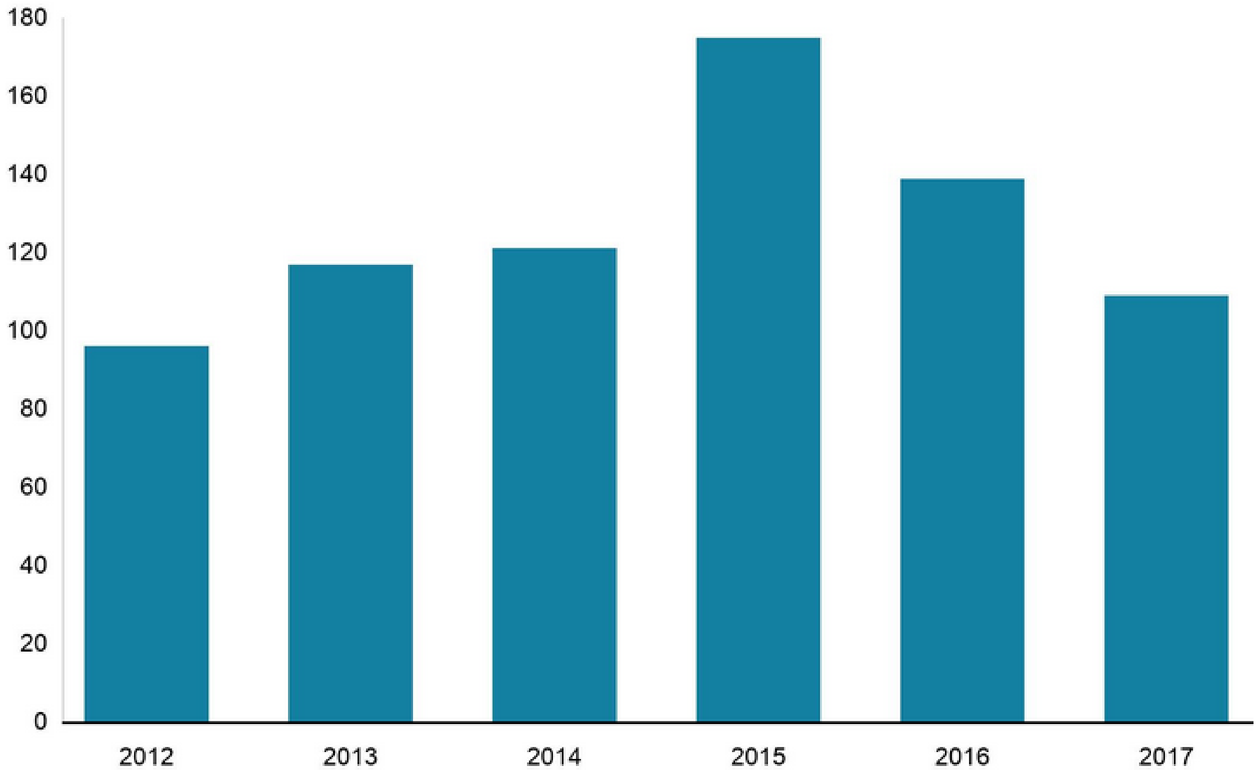


Figure 2: Railway Accidents in Pakistan Source: Pakistan Railways (BBC Reality Check, 2021).

Although the number of accidents during this period was not unusually high compared to previous years, the severity and frequency of these incidents have highlighted the underlying systemic flaws in safety management practices of Pakistan Railways. These incidents are reminders of the potential human and economic costs associated with railway accidents, and they underscore the necessity of maintaining the highest standards of track safety and infrastructure maintenance.

By incorporating advanced technologies such as Artificial Intelligence (AI) for track defect detection, there is potential to significantly reduce the likelihood of such tragic events, ensuring safer journeys for passengers and more reliable operations for the railway network.

1.2 Railway Track and Components

A railway track is more than just a pair of steel rails; it is a carefully engineered system designed to withstand immense loads while providing a smooth, stable path for trains as shown in Figure 3. The structure and layout of the track are meticulously designed to support the weight and speed of trains while ensuring safety and durability over time.

Figure 3: A Railway Track (Rawalpindi).

The key components of a railway track, as labeled in Figure 4, include (The Engineering Community, 2021):

Rails: The steel rails form the primary surface on which train wheels run. They are designed to be durable and provide a low-friction path to ensure efficient movement.

Sleepers: Also known as crossties, these are laid perpendicular to the rails to maintain consistent spacing and distribute the load. Sleepers can be made of wood, concrete, or steel.

Rail Gauge: The distance between the inner sides of the two rails is known as the rail gauge. Maintaining a precise gauge is essential for the safe operation of trains.

Fasteners: These components secure the rails to the sleepers, preventing any lateral or vertical movement that could destabilize the track.

Fishplates: Metal bars used to join two rails at their ends, ensuring continuity and smooth transitions across rail joints.

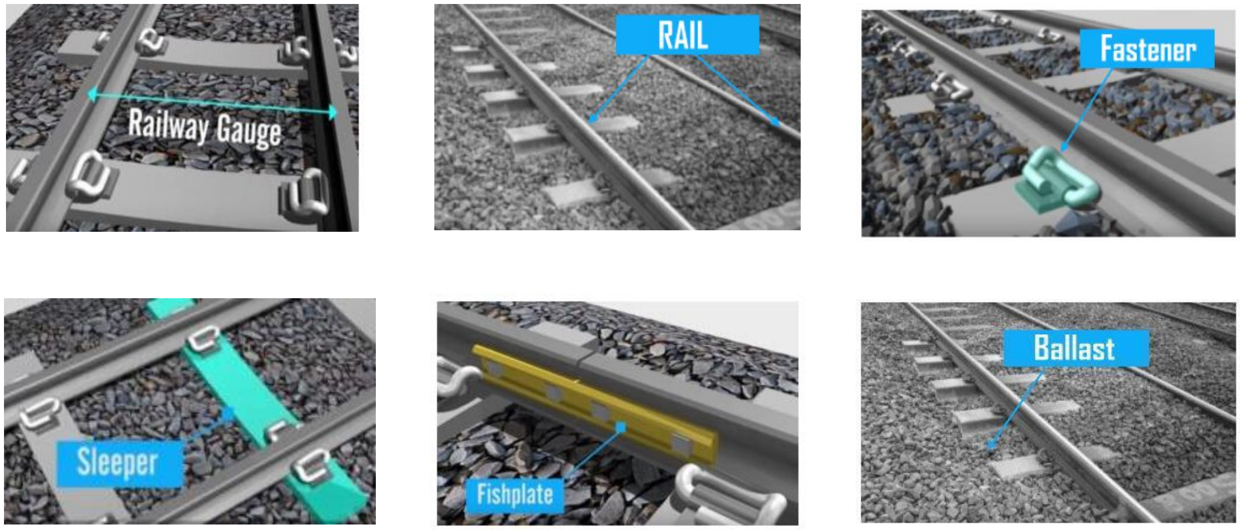


Figure 4: Components of Rail Track (The Engineering Community, 2021).

Ballast: A layer of crushed stone or gravel that supports the sleepers, absorbs vibrations, facilitates drainage, and prevents the growth of vegetation that could compromise the track.

These components work together to form a robust and resilient track system, capable of supporting the continuous and heavy loads imposed by train operations.

1.3 Rail Track Defects

Railway track defects, displayed in Figure 5 are a major concern in the operation and maintenance of rail networks (Mishra & Bhawar, 2022). These defects can lead to accidents, derailments, and costly repairs if not detected and addressed promptly (Chellaswamy et al., 2017). Defects can be categorized into several types, each presenting unique challenges:

Rail Defects:

- **Cracks/Defects:** Cracks can develop internally or externally on the rail surface. Internal cracks are particularly dangerous as they are not visible and can propagate without detection.
- **Deformation/Rail Buckling:** Rails can buckle due to thermal expansion, heavy loads, or inadequate maintenance, leading to misalignment and increasing the risk of derailment.

Sleeper Defects:

- **Wear and Tear:** Over time, sleepers degrade due to constant load and environmental exposure, leading to reduced effectiveness in maintaining rail gauge and alignment.

- **Hanging Sleepers:** When sleepers lose contact with the ballast, they can no longer provide adequate support, compromising track stability.

Fasteners/Fittings Defects:

- **Missing or Damaged Components:** Missing fasteners, bolts, or fishplates weaken the track structure and increase the likelihood of rail movement under load.

Ballast Defects:

- **Quantity/Level:** Insufficient or uneven ballast can lead to poor track stability, inadequate drainage, and increased stress on the rails and sleepers.

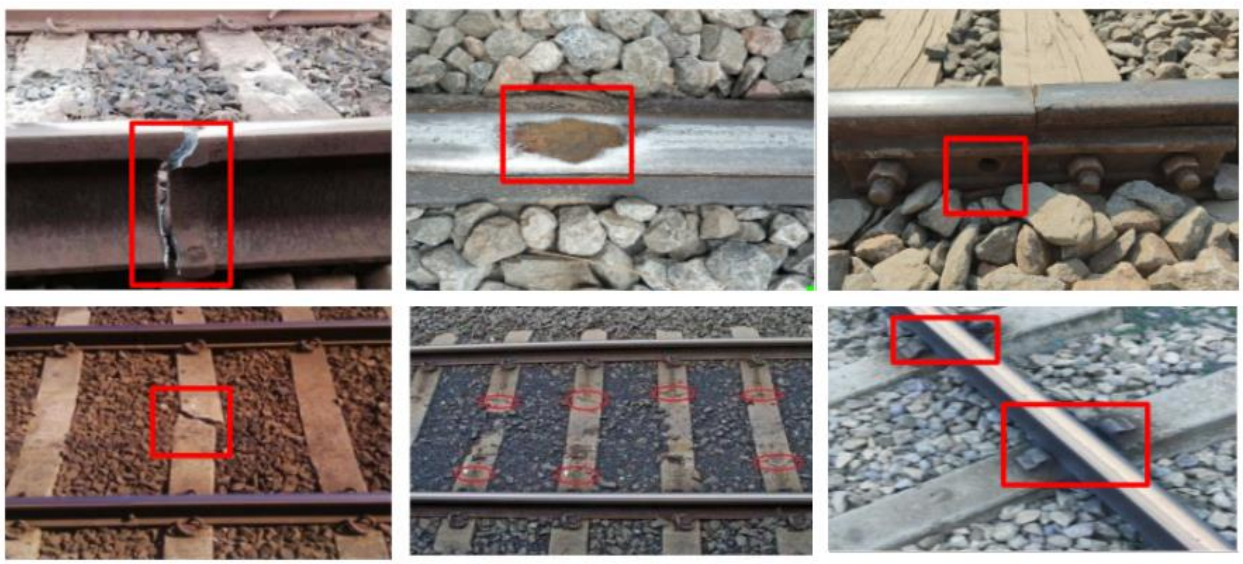


Figure 5: Different Defects in Rail Tracks (Mishra & Bhawar, 2022).

1.4 Motivation

With the above discussion in view, the motivation for this research is rooted in the need to enhance the safety and efficiency of railway network of Pakistan. The frequency of railway accidents, coupled with the significant human and economic toll they take, highlights the importance of developing more effective methods for detecting track defects. Traditional inspection methods are labor-intensive, error-prone, and often fail to detect subtle or internal defects that can lead to catastrophic failures. The advancements in artificial intelligence and machine learning present a unique opportunity to automate and improve the railway track inspection accuracy, thereby reducing the risk of accidents and enhancing the overall safety of rail transport.

1.5 Object Detection

To highlight, object detection plays a vital role in modern computer vision, allowing machines to identify and locate different objects within an image or video. In the context of railway maintenance, object detection technologies can be used to identify defects on railway tracks and components, ensuring timely repairs and maintenance.

1.5.1 Supervised Learning

In supervised learning, the model is trained on labeled data. The input to the model is both training samples and their labels (Figure 6). In railway defect detection, supervised learning involves training algorithms on annotated images of tracks, where each defect type is labeled (Cunningham et al., 2008). This approach allows the model to learn the characteristics of different defects and accurately identify them in new images. The success of supervised learning depends on the availability and quality of labeled data.

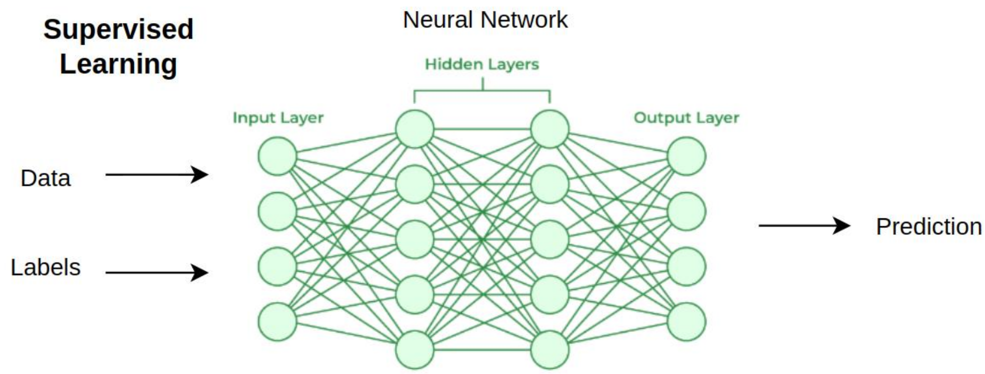


Figure 6: Supervised Learning.

1.5.2 Self-Supervised Learning

Self-supervised learning is an innovative approach that utilizes enormous amounts of unlabeled data to train models (Zhai et al., 2019) as shown in Figure 7. This method generates pseudo-labels from the data itself, which are used for training. In railway defect detection, where labeled data may be scarce, self-supervised learning can be particularly useful. It allows the model to learn from the inherent patterns in the data, improving its ability to detect and classify defects across various conditions and environments.

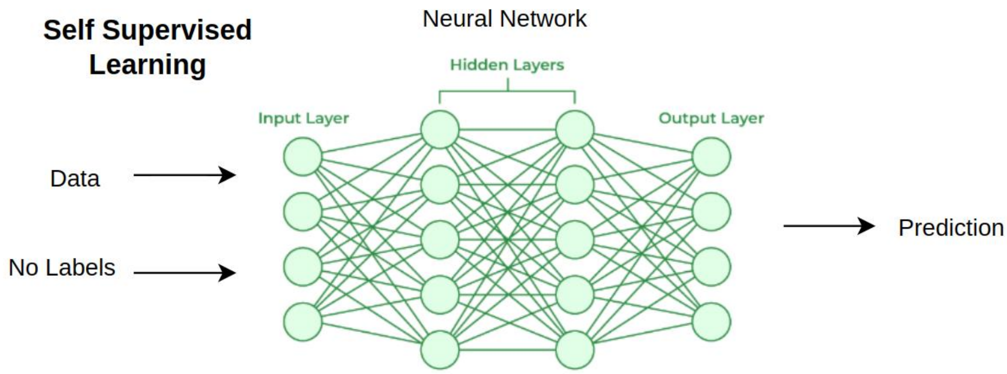


Figure 7: Self-Supervised Learning.

1.6 Problem Statement

The challenge of developing accurate models for railway track defect detection is compounded by several factors. First, there is a scarcity of labeled images depicting rail defects, which hinders the training of supervised learning models. Additionally, there is a significant class imbalance between images of defect-free tracks and those with defects, which complicates the training process. Finally, the variability in defect morphology, influenced by environmental conditions and rail material, further complicates the detection efforts.

1.7 Objectives

This research aims to:

- Develop an AI-based system capable of detecting various railway track defects, including missing fasteners, missing fishplates, and rail surface defects.
- Create a comprehensive dataset of railway track images, combining self-collected data with publicly available datasets, to train and fine-tune object detection models.
- Evaluate the performance of the developed system in real-world scenarios to ensure its effectiveness in enhancing railway safety and reliability.

1.8 Proposed Solution

The solution proposed in this thesis involves a two-stage pipeline for detecting railway track defects using advanced AI techniques. The first stage employs a YOLOv8 segmentation model, fine-tuned on a dataset of railway track images, to identify and segment various track components.

The segmented images are then processed to extract the rail surface, which is converted to grayscale and used to pre-train a U-Net-based model on a dataset of normal rail surfaces. This model is further fine-tuned on a subset of the Railway Surface Defects Dataset (RSDDs) to enhance its ability to detect surface irregularities. The combination of these two stages into a unified detection system offers a scalable and efficient solution for railway track inspection.

1.9 Thesis Organization

The structure of this thesis is as follows:

- **Chapter 2: Literature Review**
  Provides an overview of existing research on railway track defect detection, highlighting the strengths and limitations of current methods.
- **Chapter 3: Proposed System**
  Details the design and development of the AI-based detection system, including dataset preparation, model training, and system architecture.
- **Chapter 4: Implementation**
  Discusses the implementation of the proposed system, covering the tools and technologies used and the challenges encountered during development.
- **Chapter 5: Results and Discussion**
  Present the results of the performance evaluation of the system, analyzing its effectiveness in detecting railway track defects across different scenarios.
- **Chapter 6: Conclusion and Future Work**
  Summarizes the key findings of the research, discusses its implications for railway safety, and outlines possible directions for future research.

Chapter 2: Literature Review

2.1 Image Processing Based Approaches

There have been notable advancements in the detection of defects in Railway tracks by using several image processing approaches. These approaches make use of computer vision algorithms for analyzing and identifying railway tracks, giving rise to real-time monitoring and automated systems. In this aspect, the authors of a study (Karakose et al., 2018), devised a technique for detecting faults in the components of a railway by making use of cameras which have been mounted 3 on experimental vehicles. They used Canny edge approach along with Hough transform method for correctly identifying not only the components but also the faults in them. Their work manifested the true potential of plain computer vision techniques in improving the monitoring systems and railway up-keep. In addition to this, a real-time system for identification of defects on railway track was proposed, using Canny edge along with 2D discrete wavelet transform (Shah et al., 2020). This methodology, when used on a Raspberry Pi platform, proved to be quite effective in the detection of track surface faults in real-time, making it easier to highlight the practical applicability of image processing in the field. Moreover, in another study, MobileNetV2, a novel technique for recognition of railway track faults was proposed. It focused more on the reduction of maintenance cost as well as detection process automation (Ragala et al., 2022). The study demonstrated that by optimizing the MobileNetV2 network with Rmsprop, the model achieved improved accuracy and performance in classifying railway track faults

2.2 Supervised Learning Approaches

Supervised machine learning has emerged as a powerful tool in defect detection of railway tracks, making it easier for more efficient and accurate recognition of faults through the use of large datasets. In machine learning, both supervised and self-supervised learning approaches have been used to detect different defects in railway tracks. In this particular section, supervised learning approaches are discussed. Previously, a study was conducted by making use of image enhancement and improved YOLOX. It suggested the detection of defects related only to rail surface (Zhuang et al., 2018). It involved a mixture of image enhancement algorithms in the HSV space for processing the images of rail surface and making the YOLOX model better with the incorporation of attention mechanisms and feature fusion. The outcome manifested a drastic improvement in mAP by 2.42% in contrast to the original YOLOX model, showing the accuracy of this approach in real-time defect detection. Similarly, another study focused on fastener fault detection by processing images with deep learning models such as Faster R-CNN and VGG16 (Wei et al., 2019). This hybrid approach improved the fastener defect detection, pressing on the role of deep learning in automation of the railway track inspections. Apart from this, research conducted showcased the accuracy and speed of single-stage deep learning models, specifically the YOLO series, in disclosing key railway components such as rails, bolts, and clips (Wang et al., 2020). This technique underscored the accuracy of deep learning in streamlining the railway inspection technique.

In another study (Zheng et al., 2021), the author suggested a multi-object detection method by making use of deep convolutional neural networks like Mask R-CNN and YOLOv5 for detection of rail surface and fastener defects. Their contributions showcased the robustness of deep learning models in comprehending the railway inspection tasks. Furthermore, another study presented a comprehensive system for continuous monitoring of railway tracks using computer vision-based feature extractor and CNN based detector. Their approach detected between defective and non-defective using different deep learning models like ResNet, VGGNet and InceptionV3, among which ResNet-50 achieved 83.8% accuracy (Rakshit et al., 2022). A little while back, a comparative study of YOLOv5, Faster RCNN, EfficientNet for identifying railway track defects was performed (Minguell & Pandit, 2023). To validate their model, they tested it on their small dataset of clip, rail, and fishbolt to yield the best recall value of 0.93 for a Faster RCNN model. In another study, a new convolutional network-based ESA model to detect defective and non-defective rail tracks was proposed (ŞENER et al., 2022). Their model achieved 92.21% accuracy, whereas the state-of-the-art model at that time achieved accuracy of 94.9%. In addition to that, another research was conducted to detect rail surface defects like squat and cracks using YOLOv5 (Sangeetha et al., 2023). They perform a comparative analysis of different models on their rail surface dataset, among which the performance of YOLOv5 was better.

2.3 Self and Semi Supervised Learning Approaches

Recent advancements in self-supervised and semi-supervised learning approaches have further enhanced the capabilities of defect detection systems, particularly in scenarios where labeled data is scarce (Hajizadeh et al., 2016). The first study to tackle this problem was performed (Min and Li, 2022). They developed a self-supervised method, Defect Removal Variational Autoencoders (DR-VAE) to detect rail surface defects. It was a two-stage lightweight model, in which the first stage comprises InceptionNet based rail surface extractor followed by a self-supervised based U-Net model (Ronneberger et al., 2015) for rail surface defect detection. This study provided the validation of the idea of using self-supervised learning techniques to correctly recognize the defects with limited labelled data.

In another study (Xu et al., 2023), they developed a self-supervised learning-based defect representation learning framework (R-SSRL) to detect rail surface defects using label-limited data. They designed a CNN based encoder-decoder model to segment defects on the surface of the rail. Their model outperformed other models in detecting rail surface defects.

In another recent study (Min et al., 2023), the author presented an improved U-PerNet model designed for rail surface defect detection. They leverage a tiny version of Swin Transformer (Liu et al., 2021) for detection of rail surface defects using segmentation. They achieve notable performance metrics across multiple rail surface datasets, demonstrating significant improvements in detection accuracy.

2.4 Research Gap

While significant progress has been made in detecting railway track defects using various AI and machine learning techniques, there remains a gap in the literature. Most studies focus either on surface defects or on component defects individually, but not on both simultaneously. This study addresses this gap by developing a comprehensive detection system capable of identifying both surface and component defects, thereby offering a more holistic solution to railway track maintenance and safety. By integrating both supervised and self-supervised learning approaches, this research provides a robust framework that improves upon existing methods, offering enhanced detection capabilities across a range of defect types.

Chapter 3: Proposed System

In this chapter, the proposed system to detect different defects in the railway track system will be discussed. We will be going through how we gathered our dataset, and different architectures of deep learning models we used.

3.1 Dataset

The success of deep learning models in defect detection tasks depends heavily on the quality and diversity of the dataset used for training and evaluation. For this study, we designed two datasets, one was an object detection dataset that contained images of railway tracks, and the other was a segmentation dataset which contained images of the rail surface. The datasets included both normal and abnormal track conditions, with a focus on various rail components and surface defects. The images captured different types of defects and intact rail structures, providing a comprehensive basis for developing and evaluating our detection models. Detailed information about the datasets, including the types of images and their processing, are provided in the next chapter.

3.2 Deep Learning Models

Deep learning models have revolutionized the field of image processing and defect detection by enabling highly accurate and efficient analysis of complex visual data. In this study, several deep learning models were employed to detect and classify railway track defects.

3.2.1 Convolutional Neural Networks (CNNs)

CNNs are well known for extracting features from input data (O'Shea & Nash, 2015). They have been used previously for object detection problems in various research. They are widely used in image processing and image recognition problems. They achieved remarkable results in vision (Lopez Pinaya et al., 2020). Besides, CNNs are several convolution layers with non-linear activation functions at the end, as shown in Figure 8. Traditionally, each input neuron is connected to an output neuron. These are also known as Fully Connected Networks. In certain cases, this is not the case, and in other cases, these neurons are not connected to each output neuron. Different types of filters are applied in each layer and then their results are combined. They are called convolutional because they use a mathematical operation called convolutions to process data. CNNs are widely used in Computer Vision because they learn spatial features of the data very well (Ketkar & Moolayil, 2021). They are used in different tasks such as Machine Vision, Object Classification, and Object Segmentation. Filters are applied for CNNs so they can learn and recognize these patterns and features. Then, CNNs are used to predict these features and patterns. CNNs have the benefit of retaining the context of the input data (Alzubaidi et al., 2021). CNNs can understand the context of images because they use a fixed window size on the input data. CNN works by applying filters over data to learn the features and patterns present in the data. The filters are grouped in single or multiple convolutional layers. Each layer will apply filters to the input data and pass it through a non-linear activation function before passing it on to the next layer. This allows the models to learn complex patterns by reducing linearity. Filters are small square matrices that scan the input data for features and patterns that could make up a class. After the output, one or more convolutional layers use the patterns and features to make the predictions.

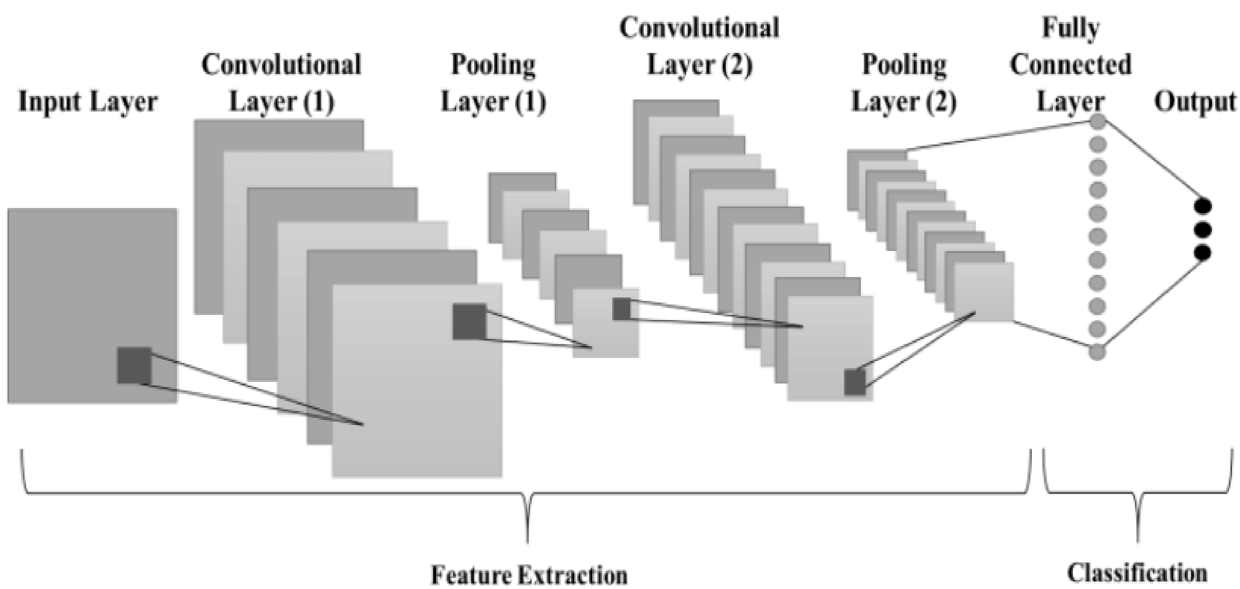


Figure 8: The simplest representation of CNN Architecture (O'Shea & Nash, 2015).

3.2.2 YOLOv8

The YOLOv8 (Jocher et al., 2023) (You Only Look Once, version 8) is the latest iteration of the YOLO object detection framework, known for its accuracy and speed in detecting objects within images. Unlike traditional object detection models that use a two-stage process—first identifying regions of interest and then classifying them—YOLOv8 uses a single stage network for predicting bounding boxes and their class probabilities directly from the images in a single pass. This architecture enables real-time object detection, making it very useful in applications where speed is critical, such as in railway track inspection.

YOLOv8 architecture is built upon a CNN backbone, typically a variant of CSP-Darknet (Karna et al., 2023), which extracts the features in the form of feature maps from input images. These maps are then passed through a series of convolutional layers, each designed to refine the features and predict bounding boxes and their class probabilities. The architecture is divided into three main components: the backbone, the neck, and the head (Karna et al., 2023). The backbone is responsible for feature extraction, the neck aggregates features at different scales (using methods like PANet or FPN), and the head performs the final detection and classification tasks.

One of the key innovations in YOLOv8 is the use of anchor-free detection, which simplifies the process of detecting objects of varying sizes and shapes (Hussain, 2024). YOLOv8 also incorporates advanced augmentation techniques and loss functions, such as CIoU loss, which enhances its performance in detecting objects under challenging conditions. The model can be trained end-to-end, making it efficient in learning from large datasets with diverse object categories.

In context of this study, YOLOv8 was used to detect and segment different components of rail track, including rail, fasteners, bolts, and fishplates. The speed of YOLOv8 allows for real-time monitoring, providing immediate feedback and enabling swift detection of railway track components. A simplified architecture of YOLOv8 is shown in Figure 9.

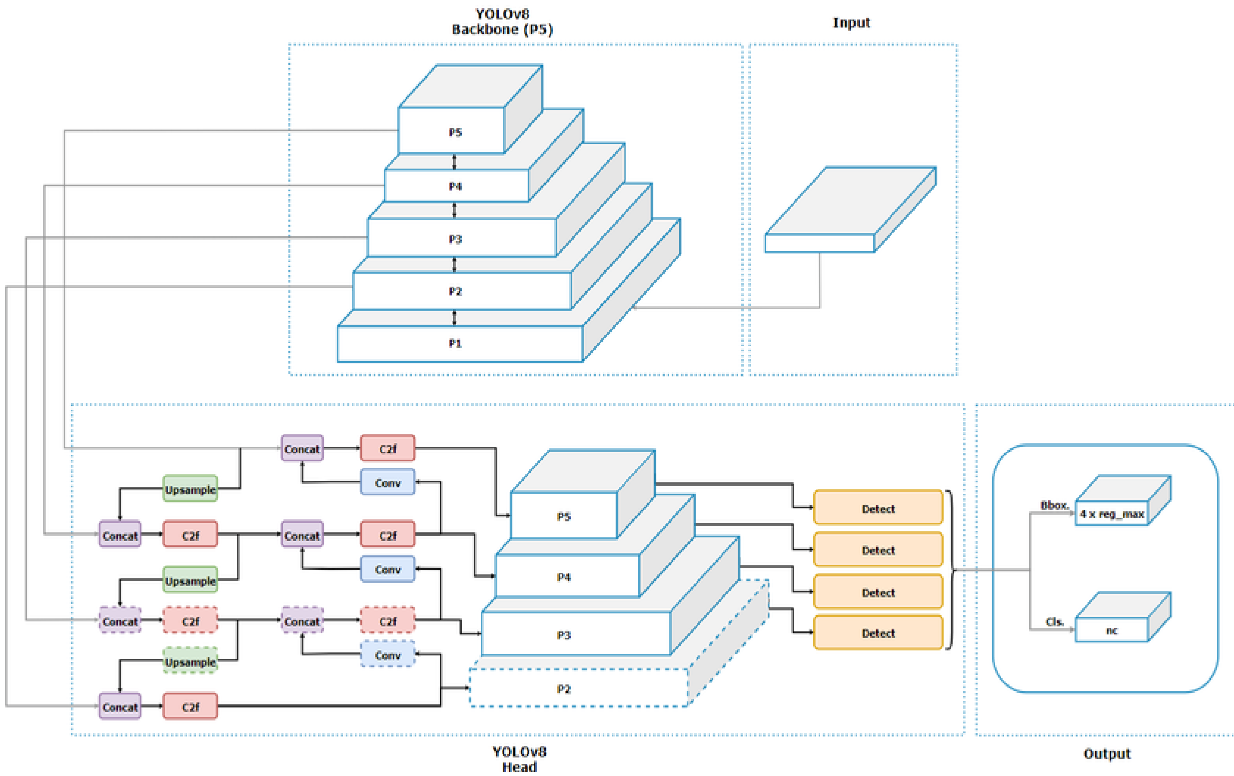


Figure 9: Simplified YOLOv8 Architecture (Karna et al., 2023).

3.2.3 Faster R-CNN

The Faster R-CNN (Ren et al., 2015) (which is a region-based CNN) is an object detection framework that extends the ideas of Fast R-CNN by incorporating a Region Proposal Network (RPN) that significantly speeds up the process of identifying objects in images. Faster R-CNN is designed to address the inefficiencies of previous methods, where the region proposal and detection tasks were handled separately, often leading to slower processing times.

Its architecture (Ren et al., 2015) is made up of three main components: the backbone, the region proposal network, and the detection network. The backbone is basically a pre-trained CNN like VGG or ResNet, which extracts feature maps from the image. The RPN then slides a small network over these feature maps to generate region proposals, which are potential areas where objects may be located. These proposals are then refined through non-maximum suppression to eliminate overlapping boxes.

Once the region proposals are generated, they are passed to the detection network, which consists of a fully connected layer which classifies the objects within the proposed regions and further refines the bounding box coordinates. Faster R-CNN uses a shared computation strategy, where the feature maps used by the RPN are also used by the detection network, making the process more efficient and faster.

Faster R-CNN is known for its high accuracy, especially in scenarios where precise object localization is crucial. However, it is computationally more intensive than single-stage detectors like YOLOv8 (Taye, 2023).

For the present research, Faster R-CNN was used to detect different components of railway track, and the results were compared with the YOLOv8 and EfficientNet model. It was found that Faster-RCNN accurately identifies and classifies defects, such as rail surface and missing track components, but the speed of detection was very low as compared to other models.

3.2.4 EfficientNet

The EfficientNet (Tan & Le, 2019) is a family of CNNs that are designed to be both computationally efficient and highly accurate. The main innovation in EfficientNet is using a compound scaling method, which scales up the depth, width, and resolution of the network in a balanced manner to optimize performance. This approach contrasts with traditional methods, which typically scale only one or two dimensions, often leading to suboptimal performance.

The architecture of EfficientNet starts with a baseline model, EfficientNet-B0, which is then scaled to create larger models (EfficientNet-B1 to B7) by increasing the number of layers (depth), the number of channels (width), and the image resolution. This scaling method is guided by a compound coefficient that determines how much to increase each dimension. The baseline model itself is built using mobile inverted bottleneck convolution layers, which are optimized for both accuracy and efficiency.

Furthermore, design features of EfficienctNet with its depthwise separable convolutions, greatly decrease the number of parameters and computational demands, while maintaining high performance (Tan & Le, 2019). Additionally, EfficientNet employs squeeze-and-excitation optimization blocks, which adaptively recalibrate feature maps to improve the representational power of the network.

In railway track defect detection, EfficientNet was employed to build a model with a balance between high accuracy and computational efficiency. It is particularly useful in applications where deployment resources are limited, such as in edge devices used for real-time track monitoring. But the performance and speed of YOLOv8 outperforms the performance and speed of EfficientNet.

3.2.5 U-Net

Another neural network architecture termed U-Net (Ronneberger et al., 2015) is a CNN architecture designed specifically for image segmentation tasks. It was initially developed for use in biomedical image segmentation but has since been widely adopted across various fields, including defect detection in industrial and infrastructure settings. The architecture of U-Net is particularly well-suited for tasks where the precise localization of features within an image is required (Ronneberger et al., 2015).

The U-Net architecture also consists of two main components: the encoder and the decoder (Leng et al., 2018). The encoder is a typical convolutional network that captures context by down sampling the input image through a number of convolutional and max-pooling layers. Each step in the encoder reduces the spatial dimensions of the image while increasing the depth of the feature maps, allowing the model to learn complex features and patterns.

The decoder mirrors the encoder but in reverse. It upscales the feature maps using transposed convolutions, gradually restoring the spatial dimensions while preserving the learned features. A key feature in U-Net is the use of skip connections, which directly transfer feature maps from the contracting path to the corresponding layers in the expansive path (Siddique et al., 2021). These connections allow the model to combine high-level abstract features with low-level spatial information, enabling precise and accurate segmentation. Figure 10 displays the U-Net architecture, with each blue box representing a multi-channel feature map. The number of channels is indicated at the top of each box, while the x-y dimensions are shown in the lower left corner. The white boxes signify copied feature maps.

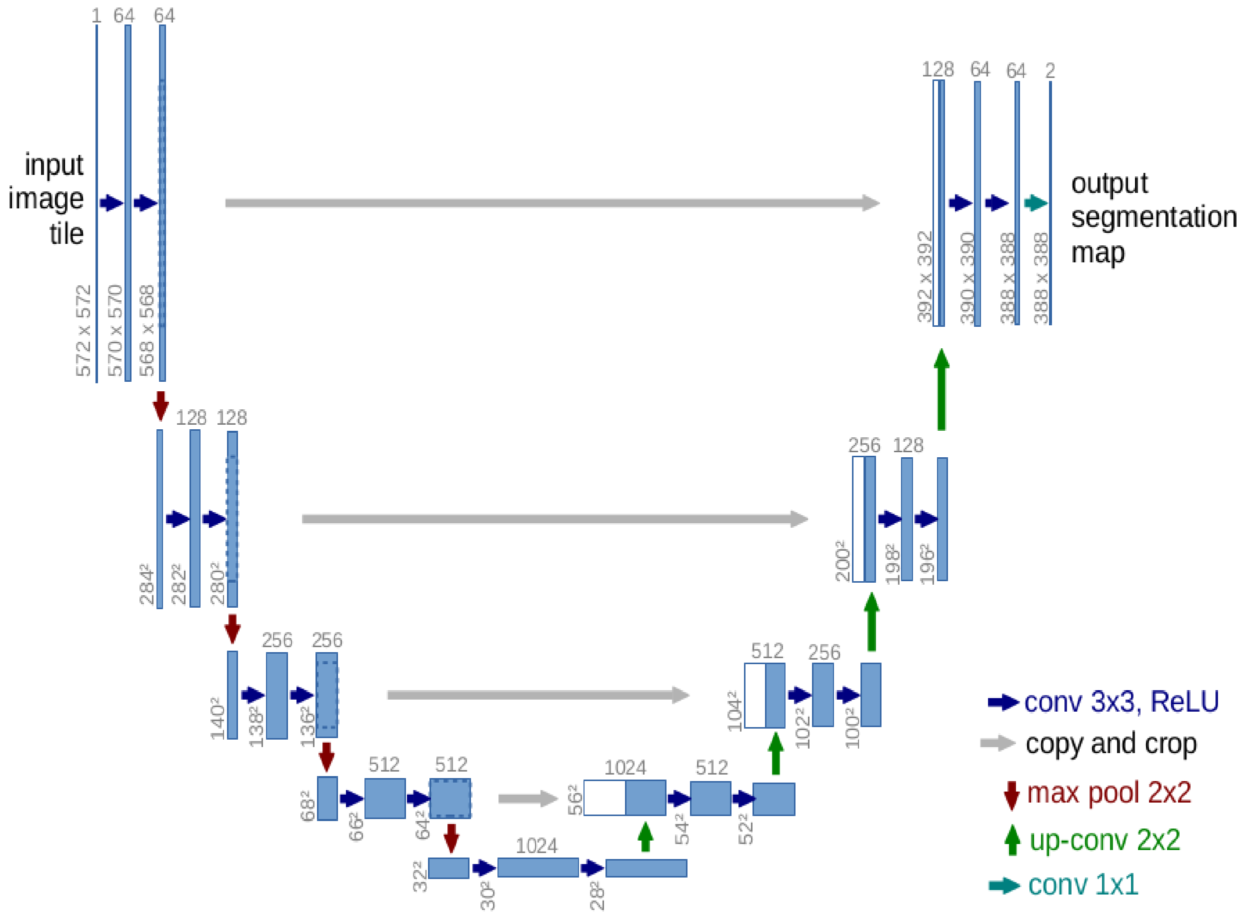


Figure 10: U-Net Architecture (Siddique et al., 2021).

Presently, U-Net is used to segment rail surface defects. It can segment rail surfaces from the background, isolating the areas of interest for further analysis. This capability is essential for detecting surface defects, such as cracks or wear, as it ensures that the model focuses on relevant regions of the image.

3.3 Proposed System Diagram

In this chapter, the author discussed their datasets, including the data they gathered for training and its statistics. The author then discussed the different architectures used for training and fine-tuning their datasets. In Figure 11, the proposed system diagram is presented to better understand the processes. In the next chapter, the author discusses how they joined all these components and created their system.

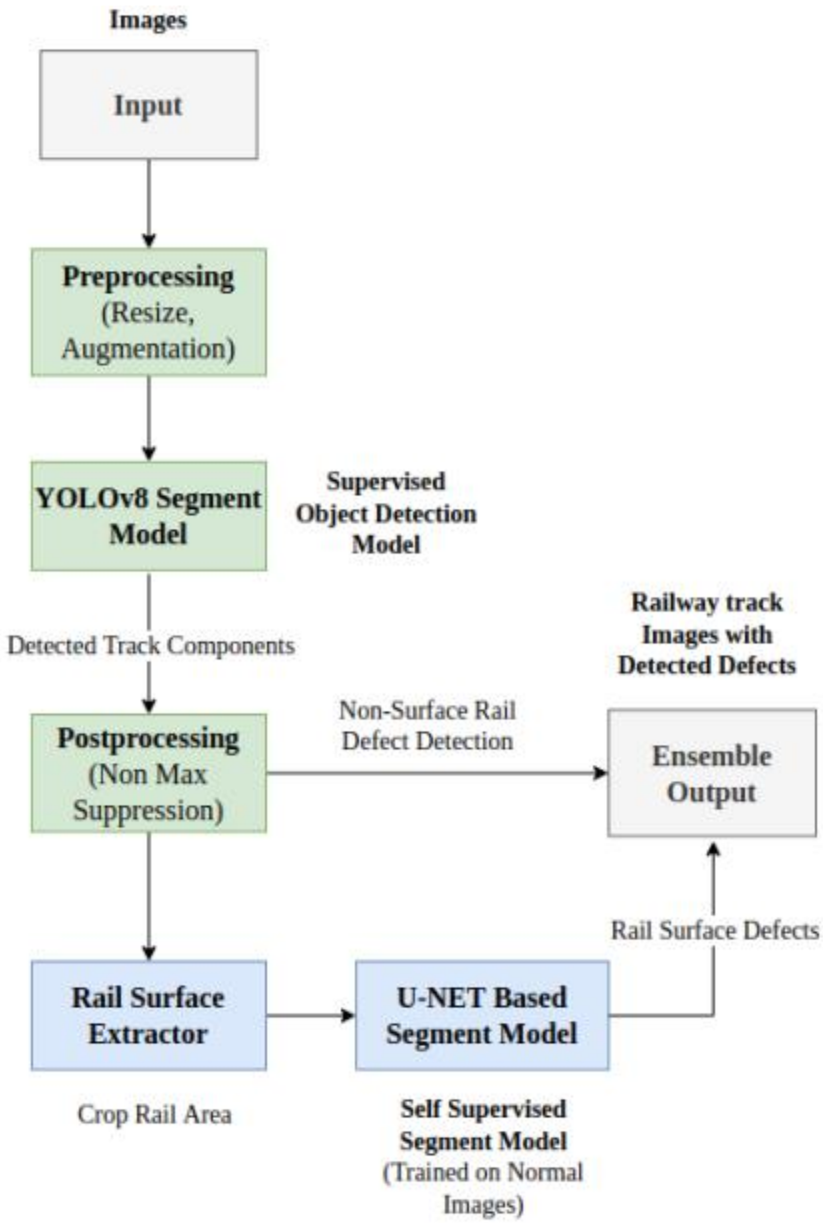


Figure 11: Proposed System Diagram.

Chapter 4: Implementation

This chapter will discuss how the author achieved the goal of creating an end-to-end automated railway track defect detection system. Additionally, the models were compared to determine the most appropriate model at hand. Furthermore, in this chapter, the author discusses how the solutions were implemented, the steps taken, and the methodology to complete those steps.

4.1 Data Collection and Preprocessing

4.1.1 Supervised Learning Dataset

Custom Collected Dataset:

Locations: The primary dataset was collected from two locations: H13 Railway Line in Islamabad and Saddar Railway Line in Rawalpindi. These tracks were chosen due to their accessibility and an approximate representation of typical railway conditions in Pakistan. The chosen tracks contained both defected and normal tracks. Some sample images from this dataset are provided in Figure 12.

Figure 12: Images from Custom Dataset.

Devices: The dataset was gathered using an OAK-D POE Camera, known for its high-resolution imaging and depth perception, and supplemented with mobile cameras for broader coverage. The images captured varied conditions and defects commonly encountered on railway tracks.

Annotation: The images were annotated using Roboflow (Dwyer, 2024), a powerful annotation tool that supports efficient labeling of large datasets. The images were annotated for two different types of tasks, one is object detection, and the other is instance segmentation. The annotated dataset was then downloaded in multiple data formats like YOLOv8, YOLOv8 OBB, YOLOv8 – Segmentation, Coco format and in the form of torch tensors.

Supplementary Dataset:

To enhance the robustness of the supervised learning model, the custom-collected dataset was integrated with Kaggle "Railway Track Fault Detection" dataset (Eunus et al., 2024). This supplementary dataset provided additional diversity in defect types and rail conditions, which is crucial for training a model that generalizes well to new data. We carefully selected only 300 images from this dataset based on their similarity with our custom dataset with different types of railway track components like fasteners and fishplates. Some samples images from this dataset are provided in Figure 13.

Figure 13: Images from Kaggle Dataset (Eunus et al., 2024).

Dataset Preprocessing:

Data Cleaning: Initial cleaning steps involved removing duplicate images, correcting mislabeled data, and ensuring that all images were relevant to the task.

Dataset Merging: The custom-collected dataset was merged with the Kaggle dataset to form a comprehensive dataset containing a wide variety of railway track defects. The length of the dataset before and after merging is provided in Table 1.

Image Resizing: All the images in the curated dataset were resized to a uniform dimension of 640x640x3 to ensure consistency during the training of the model and to optimize computational efficiency.

Data Augmentation: To deal with overfitting and improve the generalization ability of the model, different data augmentation techniques were applied. These included horizontal flips, tilts, and adjustment of brightness to simulate different environmental conditions as shown in Figure 14.

Figure 14: Images with their Augmented Pairs.

Dataset Splitting: The final dataset, consisting of approximately 3000 images after augmentation, was split into training (75%), validation (15%), and testing (10%) datasets.

Table 1: Dataset Detail.

Datasets	**Custom Dataset**	**Kaggle Dataset**	**Combined Dataset**	**Final Dataset After Augmentation**

Images	1500	300	1800	3000

Classes: The dataset was categorized into five classes representing different types of defects and rail components. These included:

1. Rail.
2. Fastener.
3. Defective Fastener.
4. Fishplate.
5. Defective Fishplate.

Class Distribution:

The class distribution is provided in the Table 2:

Table 2: Class Distribution.

Class Name	Rail	Fastener	Defective Fastener	Fishplate	Defective Fishplate
No of Samples	3100	6234	640	600	350

4.1.2 Self-Supervised Learning Dataset

Dataset Creation:

The self-supervised learning dataset was created by extracting rail surface area from the output images of the YOLOv8 segment model used in the initial stage of the supervised learning process. This extraction focused on isolating the rail surface to train a model that could distinguish between normal and defective surfaces.

Figure 15: a) Image from Custom Dataset, b) Extracted Rail Part of the Image.

Image Selection:

Only images where the rail surface was normal and free of scratches or other defects were included in this dataset. This approach aimed to train the model to recognize what a "healthy" rail surface looks like, which could then be used to identify anomalies in the defect detection phase.

Dataset Statistics:

The final dataset for the self-supervised learning model consisted of 400 normal images, providing a focused dataset for training the model to recognize deviations from the norm.

Fine Tuning and Evaluation Dataset:

Another dataset was used in this study to fine-tune and validate the normal rail surface trained model, which was Rail Surface Defects Detection (RSDDs) dataset (Gan et al., 2017). The Railway Surface Defect Detection Dataset (RSDDs) is a specialized dataset curated to facilitate the detection of surface-level defects on railway tracks (Niu et al., 2020). It comprises a diverse collection of images that capture different types of surface defects, such as wear, corrosion, and cracks which can reduce the structural integrity and safety of railways. The RSDDs dataset is widely used in research for training and benchmarking models focused on detecting and classifying these defects, providing a valuable resource for advancing automated railway maintenance systems.

A subset of this dataset was used to fine tune the normal rail surface model, and the other subset was used to evaluate this fine-tuned model. Sample images from this dataset along with its groundtruth is shown in the Figure 16:

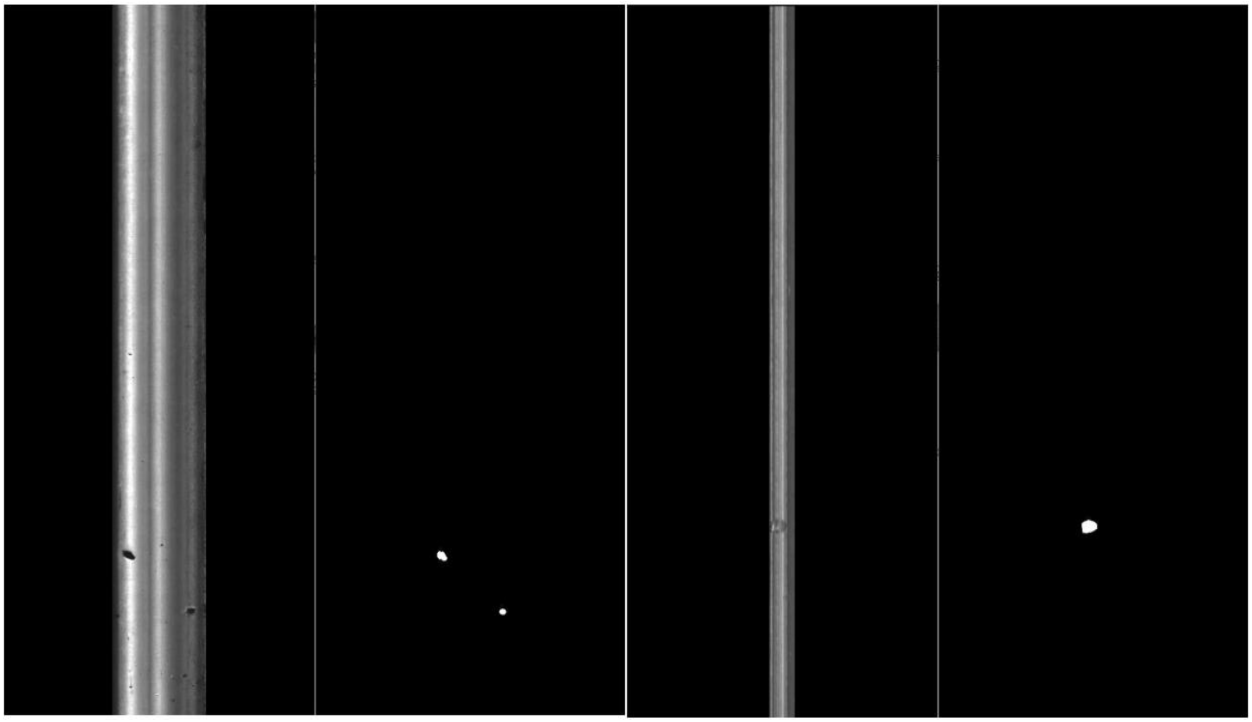


Figure 16: Rail surface images with their ground-truth images showing defected regions from RSDDs Dataset (Gan et al., 2017).

4.2 Supervised Learning Model

This model is responsible for detecting different components of railway track. The model will provide the segmented output with each railway track component detected with both segmentation and bounding boxes. As this model is trained on annotated label dataset, therefore this part is known as Supervised Learning Model. This approach involved training an object detector model, which in the case of this study is YOLOv8 segment, on a labeled dataset to accurately identify and classify various types of railway track components. This model detected all the non-rail surface defects like missing fastener, missing fishplate and will pass the rail part to the next model.

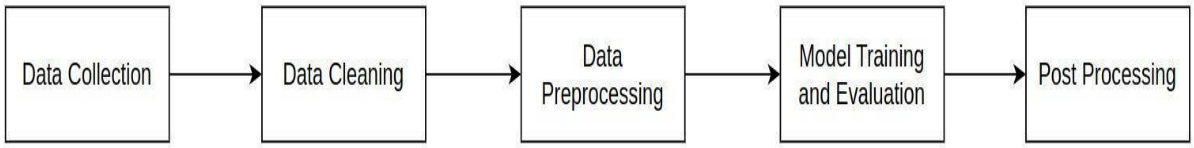


Figure 17: Workflow of Supervised Learning Model.

The overall process, as illustrated in Figure 17, follows a systematic workflow starting from data collection, through data cleaning and preprocessing, model training and evaluation, and finally to post-processing. This section details the model selection, training parameters, and post-processing techniques employed in this phase.

4.2.1 Model Selection

For the supervised learning model, YOLOv8x Segment Model was chosen as the primary model due to its exceptional performance in object detection and segmentation tasks, especially in scenarios that require real-time analysis. The selection process involved a thorough comparison with other established models, such as Faster R-CNN and EfficientNet, both of which are renowned for their accuracy and efficiency in different domains. However, YOLOv8x stood out for several compelling reasons which are discussed below that make it particularly well-suited for railway track defect detection.

One of the primary reasons for selecting YOLOv8x was its superior Frames Per Second (FPS) rate when applied to the custom dataset. In real-time defect detection, the ability to process images quickly is critical. High FPS ensures that the model can handle video streams or large batches of images efficiently, providing near-instantaneous feedback that is crucial for timely maintenance decisions in railway operations. Faster R-CNN, while highly accurate, tends to have a lower FPS due to its two-stage architecture, which involves generating region proposals before classification. This additional step, although beneficial for accuracy, slowed the detection process, making it less ideal for applications where speed is paramount. Similarly, EfficientNet, known for its balance between accuracy and efficiency, still could not match the real-time processing capabilities of

YOLOv8x, particularly when dealing with the high-resolution images typical in railway defect detection.

In addition to speed, YOLOv8x achieved a higher mean Average Precision (mAP), which is a crucial metric for evaluating the performance of any object detection model. mAP measures the accuracy of the model by calculating the average precision for each class and then averaging these values. A higher mAP indicates that the model is not only precise but also reliable across different types of objects or, in this case, defects. The higher mAP of YOLOv8x reflects its robust feature extraction capabilities and its anchor-free detection mechanism, which allows it to adaptively select relevant features and regions of interest within the images. This adaptability is particularly important in the context of railway defect detection, where defects can vary significantly in size, shape, and appearance.

Moreover, the architecture of YOLOv8x is optimized to balance both speed and accuracy without sacrificing one for the other. Its design incorporated several advanced techniques, such as optimized convolutional layers and efficient use of computational resources, which enabled it to deliver high performance even on large and complex datasets. This efficiency makes it possible to deploy the model in real-world environments where computational resources may be limited, yet high accuracy and real-time processing are still required.

Overall, the choice of YOLOv8x Segment Model was driven by its ability to provide a high FPS rate and superior mAP, making it highly effective for real-time railway track defect detection. Its advanced architecture and optimization for both speed and accuracy ensured that it outperformed other models such as Faster R-CNN and EfficientNet in the specific context of this study, thus providing reliable and timely detection of various rail defects that are critical for maintaining the safety and integrity of railway operations.

4.2.2 Model Training Parameters

Before training the YOLOv8x model, specific parameters were defined to optimize its performance on the defect detection task. These parameters were carefully selected based on the nature of the dataset and the available computational resources.

Below is an overview of the key training parameters used for the YOLOv8x Segment Model:

Table 3: YOLOv8x – Segment Model Training Parameters.

Parameter	Value
Input Shape	640x640x3
No of Layers	365
Parameters	68 million
Batch Size	8
Loss Metric	Bounding Box Loss
Accuracy Metric	mAP, IoU
Training Epochs	100
GPUs	1
Training Time	3 Hours
Trained Model Size	144 MB
Inference Time	20 ms

These parameters were set to ensure that the model could learn effectively from the dataset while maintaining computational efficiency. The input shape of 640x640 pixels was selected to balance detail and processing speed. The model was trained over different epoch settings from 50 to 200 epochs, but the optimal results were obtained at 100 epochs, with a batch size adjusted according to the GPU memory availability, ensuring stable and efficient training. The GPU used for training this model was the NVIDIA RTX 3090 with 24 GB VRAM.

The loss metric used, Bounding Box Loss, was critical for optimizing the ability of the model to accurately predict the location of defects within an image. The mAP and Intersection over Union (IoU) were used as accuracy metrics to evaluate the performance of model in correctly identifying and localizing defects.

4.2.3 Post Processing

After the YOLOv8x model produces its predictions, post processing is necessary to refine the results and ensure that only the most accurate predictions are retained. For this purpose, Non-Maximum Suppression was performed on the output (Hosang et al., 2017).

Non-Maximum Suppression (NMS): This technique was used to eliminate the redundant and overlapping bounding boxes, which often occurred in object detection tasks. The goal of NMS was to retain only the most accurate bounding box for each detected object, henceforth decreasing the chance of multiple detections of the same defect.

Process:

- **Assign Scores to Each Bounding Box:** Each predicted bounding box is assigned a confidence score based on the likelihood that it contains a defect.
- **Sort Boxes by Descending Confidence:** The bounding boxes are then sorted in descending order of their confidence scores, with the highest scoring boxes considered first.
- **Keep the Highest-Scoring Box:** The box with the highest confidence score is retained, and any other boxes that overlap significantly (based on IoU) with this box are suppressed.
- **Repeat for Remaining Boxes:** This process is repeated for all remaining boxes, ensuring that only the most relevant detections are kept.

By applying NMS, the final output of the YOLOv8x model is refined to include only the most accurate and non-overlapping bounding boxes, thereby improving the reliability of the defect detection system. A comparison between the results of the YOLOv8 segment model before and after applying NMS is provided in Figure 18.

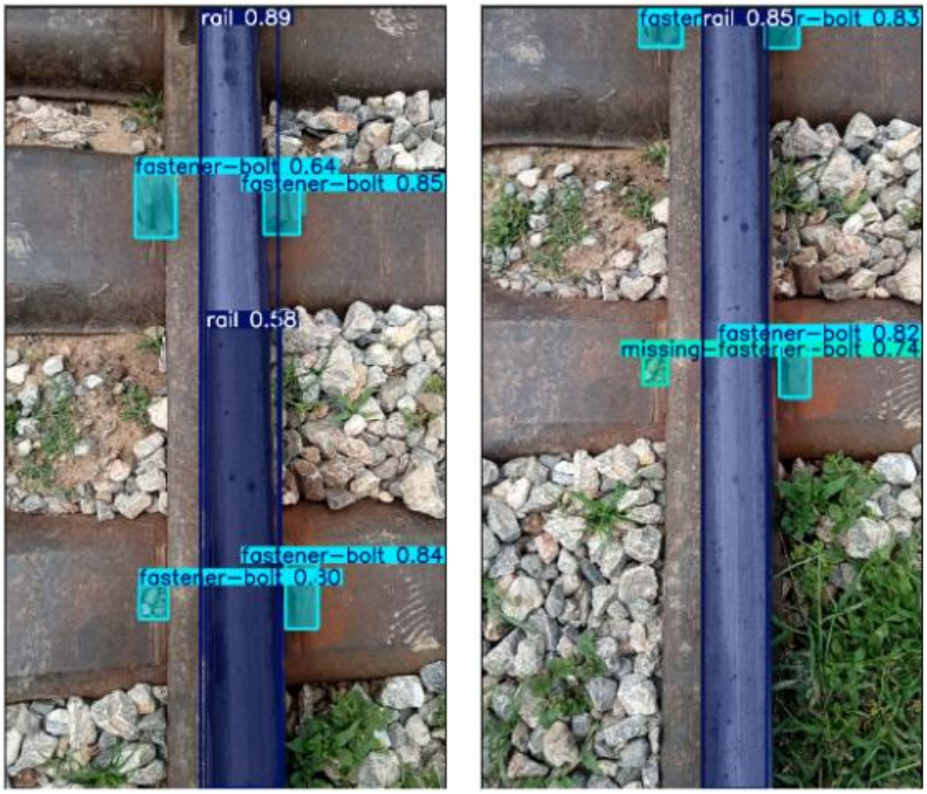


Figure 18: YOLOv8 – Segment Model Result Before NMS (Left) and After NMS (Right).

4.3 Self Supervised Learning Model

The self-supervised learning model in this project was designed to address the challenge of detecting rail surface defects through a two-stage training process. This approach leveraged a U-Net based architecture to effectively segment and identify defects from rail surface images. The input of this model was the output segmented image of the supervised learning model, and the output of this model was the segmented image of the rail surface with defects in white and background in black color. Figure 19 illustrates the flow of this process, beginning with the extraction of the rail surface from images processed by the YOLOv8 Segment Model, followed by preprocessing, model training, and fine-tuning on specific datasets.

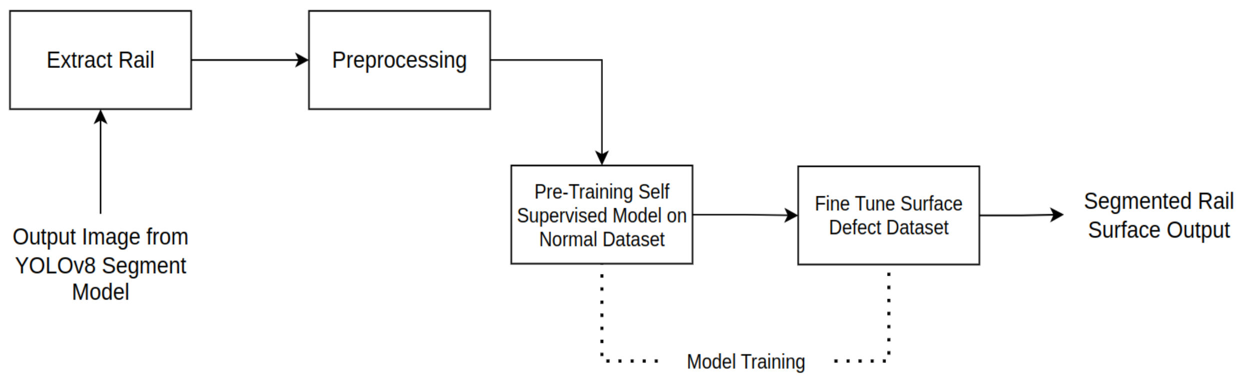


Figure 19: Self Supervised Model Workflow.

4.3.1 Rail Extractor

Before diving into model training, an essential step in the process was the extraction of the rail component from the output of the YOLOv8 segmentation model. The YOLOv8 model was initially used to detect and segment different components of the railway track, including the rail itself, fasteners, and other elements. Among these, the rail class was of interest for further analysis. The rail component was extracted by utilizing the segment mask generated by the YOLOv8 model specifically for the rail class. This segment mask identified the exact pixels in the image that corresponded to the rail surface. The extraction process involved the following steps:

Segmentation Mask Application: The segment mask for the rail class was applied to the original image, isolating the rail component from the rest of the track. This step effectively removed any extraneous elements, such as fasteners or ballast, that were not relevant to the surface defect analysis.

Region of Interest (ROI) Extraction: After applying the mask, the region of interest (ROI) corresponding to the rail was extracted. This ROI was then resized and normalized to a consistent format suitable for input into the self-supervised learning model.

Preprocessing for Consistency: The extracted rail surface images underwent preprocessing to maintain consistency throughout the dataset. This preprocessing included resizing the images to a fixed dimension and converting them to grayscale to focus on texture and structural features rather than color.

This rail extraction step was crucial in preparing the data for the subsequent self-supervised learning phase, as it ensured that the input to the model was focused solely on rail surface, thereby improving the accuracy of defect detection.

4.3.2 Model Training

The model training was conducted in two distinct stages, allowing for the effective adaptation of the model to the task of detection of defects on rail surface.

Stage 1: Pre-Training on Normal Rail Surface Images

In the first stage of training, the U-Net model was pre-trained on a dataset composed exclusively of normal rail surface images. These images were carefully curated to represent defect-free conditions, providing the model with a clear understanding of what constitutes a healthy rail track. The pre-training aimed to teach the model the standard features and patterns of normal rail surfaces, such as texture, color gradients, and structural uniformity, without any interference from defects.

Several experiments were conducted to optimize this pre-training phase. Initially, the model was run on different settings, including:

Baseline U-Net Model: The standard U-Net model was trained purely on normal rail surface images to establish a baseline performance.

Fine-Tuned U-Net on Defect Dataset Subset: The pre-trained U-Net model was then fine-tuned on a subset of the Rail Surface Defect Dataset (RSDDs) to allow the model to adjust its understanding to include defect characteristics.

Incorporation of Pseudo-Random Defects: Another approach involved introducing pseudo-random defects into the pre-training stage. These artificially generated defects were designed to mimic real-world anomalies, enabling the model to encounter a wider variety of conditions during training.

Despite these variations, the results indicated that the U-Net model trained for 250 epochs on normal rail surface images and then fine-tuned on a subset of the defect dataset outperformed all other configurations. This model demonstrated superior generalization capabilities, effectively distinguishing between normal and defective rail surfaces while maintaining high segmentation accuracy. The extended training duration allowed the model to refine its feature extraction abilities, leading to more accurate and reliable predictions in subsequent testing.

Stage 2: Fine-Tuning on Rail Surface Defect Dataset (RSDDs)

In the second stage of training, the pre-trained U-Net model was fine-tuned using a subset of the Rail Surface Discrete Defects (RSDDs) (Gan et al., 2017). The RSDDs comprise various images depicting different types of rail surface defects, such as cracks, wear patterns, and other structural anomalies. Fine-tuning the model on this dataset allowed it to adapt its learned features from the pre-training phase to more effectively identify and classify defects.

Fine-tuning was performed under various settings of training epochs, with the most promising results emerging after 250 epochs of training. The fine-tuning process was crucial for bridging the gap between normal and defective conditions, enabling the model to refine its understanding of defects while maintaining the context of a healthy rail surface learned during pre-training. This stage ensured that the model could accurately identify subtle defects that might not have been as apparent in the normal images used in the first stage.

The extended fine-tuning process allowed the model to adjust its weights more precisely, leading to improved performance on both the remaining subset of the RSDDs and a custom rail track dataset used for final evaluation. The successful application of fine-tuning demonstrated the capacity of the model to generalize well to unseen data, making it a robust tool for real-world defect detection scenarios.

By carefully managing these two stages of training, the model developed a comprehensive understanding of rail surface characteristics, leading to a high level of accuracy in defect detection. The combination of extensive pre-training on normal images and targeted fine-tuning on defect dataset proved to be the most effective strategy for this task.

4.3.3 Model Parameters

The training and fine-tuning of the U-Net model were conducted with specific parameters optimized for the task. These parameters were carefully selected to balance training time, model accuracy, and computational efficiency.

Table 4: U-Net Model Training Parameters.

Model Parameters	Fine Tuned U-Net (50 Epochs)	Fine Tuned U-Net (250 Epochs)
Input Shape (grayscale)	635x206	635x206
Learning Rate	0.001	0.0001
Epochs	50	250
Kernel Size	3	3
Padding	1	1
Batch Size	16	16
Model Size	124 MB	124 MB
Inference Time	1.5 s	1.5 s
Training Time	2 Hours	~8 Hours

The model was trained with an input shape of 635x206 pixels in grayscale format, which was suitable for capturing the details necessary for defect detection. The learning rate was adjusted between the pre-training and fine-tuning stages, with a lower rate during fine-tuning to ensure that the model could make more precise adjustments based on the new data.

The training process for the model with 250 epochs took approximately 8 hours, compared to 2 hours for the 50-epoch version. This extended training time allowed the model to develop a deeper understanding of the defect patterns, ultimately leading to improved performance during inference. The model structure of U-Net, with a kernel size of 3 and padding of 1, ensured that the convolutional layers could effectively capture the key features of the rail surface while maintaining the spatial integrity of the image features. The batch size of 16 was chosen to balance the need for sufficient gradient updates while maintaining efficient GPU utilization.

4.3.4 Evaluation Metrics

In evaluating the performance of our self-supervised learning model, U-Net based segmentation model, it was crucial to select metrics that accurately reflect the ability of the model to segment and identify defects on rail surfaces. Two primary evaluation metrics were used: Dice Loss and Intersection over Union (IoU). These metrics are commonly employed in segmentation tasks due to their effectiveness in measuring the overlap and accuracy of the predicted segmentation masks against the ground truth.

Dice Loss

It is a metric derived from Dice Coefficient, which measures the similarity between two sets (Li et al., 2019). It is specifically useful in segmentation tasks to evaluate how well the predicted segmentation mask overlaps with the actual ground truth mask. The formula used to calculate the Dice Coefficient is:

$$\text{Dice Coefficient} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Where X is predicted set of pixels and Y is ground truth set. Dice Loss, therefore, is:

$$\text{Dice Loss} = 1 - \text{Dice Coefficient}$$

The Dice Loss penalizes both false positives and false negatives, making it particularly sensitive to the overlap between the predicted and true segmentation regions. In the context of rail surface defect detection, using Dice Loss as an evaluation metric allows us to measure how accurately the

U-Net model can predict the exact regions of defects on the rail surface. Since defects on rail surfaces can vary greatly in size and shape, Dice Loss is beneficial in ensuring that the model not only detects the presence of a defect but also accurately outlines its boundaries.

Intersection over Union (IoU)

IoU is another essential evaluation metric used for evaluating the accuracy of object detection and image segmentation models (Rezatofighi et al., 2019). It is calculated by dividing the area of overlap between predicted segmentation mask and ground truth mask by the area of their union.:

$$\mathrm{IoU} = \frac{|X \cap Y|}{|X \cup Y|}$$

Where X is predicted segmentation and Y is ground truth. It ranges from 0 to 1, with a higher value indicating a better overlap between the predicted and actual segmentation.

In segmentation tasks, IoU is critical because it provides a direct measure of how well the predicted mask covers the actual object or defect. For the U-Net based model used in this study, IoU was an important metric to assess the precision of model in defect detection. A higher IoU indicates that the model is more accurate in segmenting the defect regions, reducing false positives (incorrectly identifying non-defect areas as defects) and false negatives (missing actual defects).

Chapter 5: Results and Discussion

In this chapter, the results of the defect detection models are presented, focusing on the performance of both the supervised and self-supervised learning approaches. We analyze the outcomes based on key metrics such as Frames Per Second (FPS), mean Average Precision (mAP), and class-specific performance metrics like precision and recall. The results demonstrate the effectiveness of the YOLOv8x Segment Model in detecting various types of railway track defects and the comparative performance against other models.

5.1 Supervised Learning Model Results

5.1.1 Comparison with Other Object Detector Models

The YOLOv8x Segment Model was evaluated against other prominent object detection models, including Faster R-CNN and EfficientNet, to assess its relative performance. The evaluation metrics included FPS and mAP, which were critical for determining the efficiency of model and accuracy in defect detection.

Table 5: Comparison of Selected Model with other state-of-the-art Object Detectors.

Model	FPS	mAP
Faster-RCNN	18	88.2%
EfficientNet	21	84.6%
YOLOv8-Segment	**30**	**94%**

The YOLOv8-Segment model demonstrated better performance with an FPS of approximately 30, significantly higher than both Faster R-CNN (18 FPS) and EfficientNet (21 FPS). This higher FPS indicated that YOLOv8-Segment is more capable of processing images rapidly, which is crucial for real-time railway defect detection. In terms of accuracy, YOLOv8-Segment also achieved a higher mAP of 94%, surpassing the 88% mAP of Faster R-CNN and 84% mAP of EfficientNet. The combination of speed and accuracy makes YOLOv8-Segment the most effective model for this application, as it can reliably detect defects while maintaining the throughput necessary for real-time deployment.

5.1.2 Comparison with Custom Trained Models

Further comparisons were made between various custom-trained models to evaluate the effectiveness of the YOLOv8-Segment model. The metrics used include Bounding Box Precision, Bounding Box Recall, mAP at IoU thresholds of 50% (i.e. mAP50), and the average mAP across IoU thresholds ranging from 50% to 95% (i.e. mAP50-95).

Table 6: Comparison of Selected Model with other YOLO variants.

Model Name	**Bounding Box Precision**	**Bounding Box Recall**	**mAP50**	**mAP50-95**
YOLOv8	86.8%	86%	91%	78.5%
YOLOv9 – Segment	91.2%	**90.5%**	93.7 %	**81%**
YOLOv8 - Oriented Bounding Box	**93.7%**	88%	93.9%	79.6%
YOLOv8 - Segment	92.2%	89.4%	**94%**	**81%**

The YOLOv8-Segment model outperformed other custom-trained models, achieving a Bounding Box Precision of 92.2% and Bounding Box Recall of 89.4%. These metrics indicate that the YOLOv8-Segment model not only accurately identifies defects but also does so with high reliability across various defect types. The mAP50 model having a value of 94% reflects its high accuracy at the IoU threshold of 50%, while its mAP50-95 of 81% suggested strong performance across a range of IoU thresholds. Compared to the other models, the ability of YOLOv8-Segment to maintain high precision and recall, coupled with its superior mAP scores, underscores its robustness and effectiveness for detecting and segmenting defects on railway tracks.

5.1.3 Model Validation

The validation of the YOLOv8-Segment model involved testing its performance on a separate validation set and analyzing the output images. The model was able to accurately detect all the components of the rail track with high accuracy, providing clear and precise bounding boxes and segmentation masks for each identified defect.

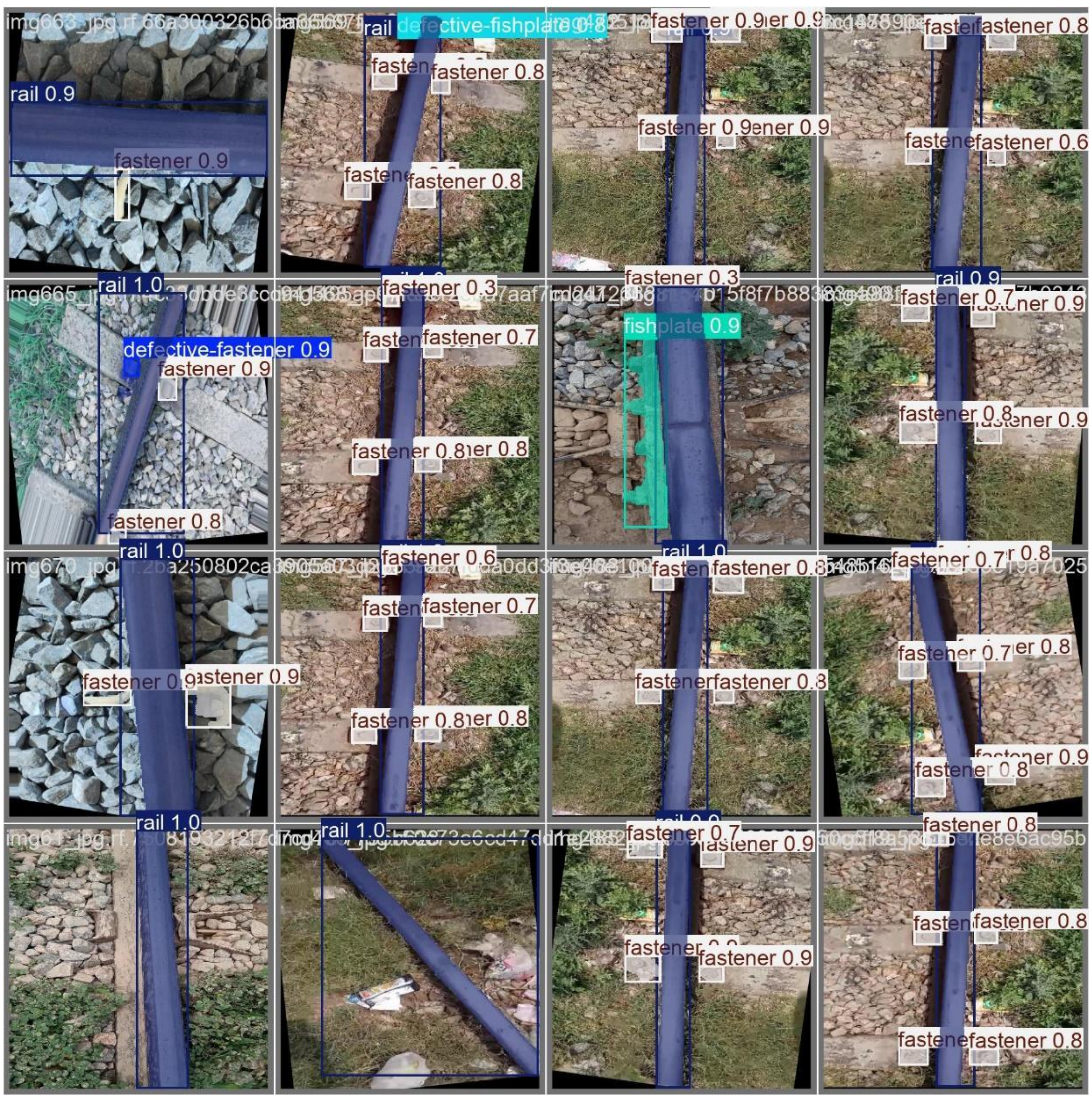


Figure 20: Segmentation results of Custom Trained YOLOv8 Segment model on Custom Dataset.

Example output images from the model showcase its ability to accurately detect and segment different rail components, such as fasteners, fishplates, and rail surface in Figure 20. These results are consistent across various environmental conditions and rail track scenarios, further validating the performance of the model in real-world applications.

5.1.4 Performance on Each Class

The performance of the YOLOv8-Segment model was further broken down by class to evaluate its precision, recall, and mAP for each detected defect type. The analysis included both bounding box and segmentation metrics, providing a detailed view of the capabilities of the model across different classes. The results are shown in Figure 21.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95)
all	182	743	0.922	0.894	0.94	0.81	0.918	0.891	0.938	0.725
defective-fastener	34	58	0.901	0.793	0.899	0.768	0.883	0.778	0.889	0.622
defective-fishplate	19	20	0.881	0.85	0.916	0.774	0.88	0.85	0.916	0.679
fastener	149	434	0.952	0.915	0.966	0.771	0.947	0.912	0.963	0.719
fishplate	28	31	0.884	0.935	0.936	0.791	0.881	0.935	0.936	0.693
rail	182	200	0.992	0.975	0.98	0.948	0.997	0.98	0.984	0.912

Figure 21: Custom trained YOLOv8 Segment Model Performance on Each Class Detection.

Bounding Box Metrics:

Precision: The model achieved high precision across all classes, indicating that most detected defects were true positives.

Recall: The recall was also consistently high, demonstrating the ability of the model to detect most defects present in the images.

mAP: The mean Average Precision for bounding boxes across different classes was robust, reflecting the accuracy of the model in both localization and classification.

Segmentation Metrics:

Precision and Recall: Like the bounding box metrics, the segmentation metrics showed high precision and recall, confirming the effectiveness of the model in segmenting the defect regions accurately.

mAP: The segmentation mAP scores further validated the ability of the model to produce high-quality segmentation outputs, crucial for detailed defect analysis.

These class-specific performance metrics highlight the versatility of the YOLOv8-Segment model and accuracy in handling various types of railway track defects, making it a reliable tool for comprehensive track inspection.

5.1.5 Plots

The following set of plots provides a comprehensive visualization of the training and validation performance of the YOLOv8-Segment model across various metrics. These plots are crucial for understanding the learning dynamics of the model and its effectiveness in defect detection on railway tracks.

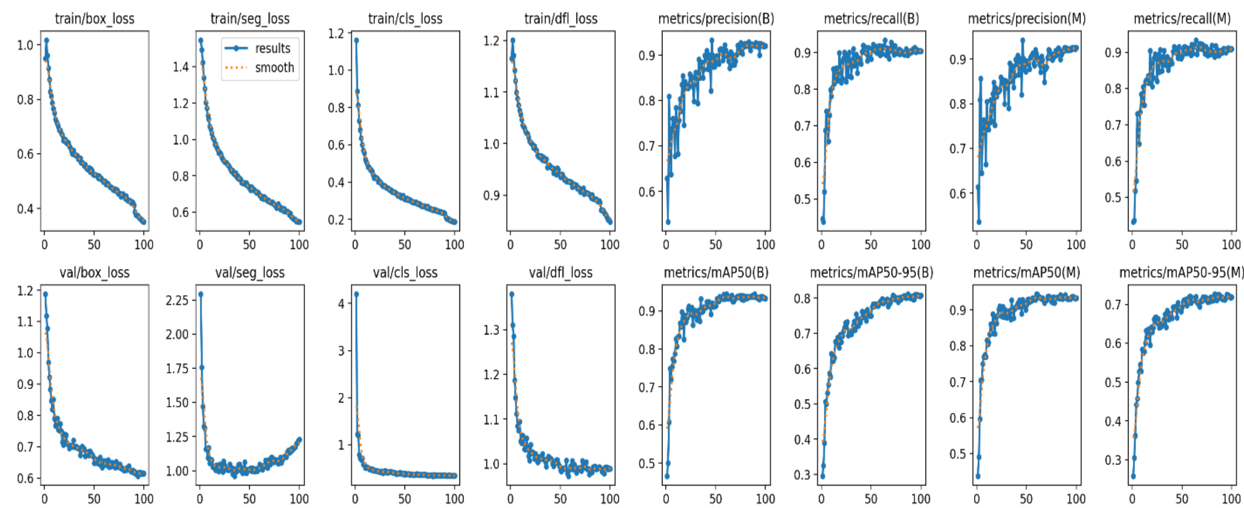


Figure 22: Training and Validation Plots of Custom Trained YOLOv8 Segment Model.

Training and Validation Losses

The description of Figure 22 are as follows:

train/box_loss: This plot shows the bounding box loss during training. A consistent downward trend is observed, indicating that the model is increasingly precise in predicting the bounding boxes around detected defects. The gradual reduction in loss signifies that the model is effectively learning the spatial properties of the defects.

train/seg_loss: This plot represents the segmentation loss during training. The smooth decline in segmentation loss reflects the ability of the model to accurately delineate the boundaries of defects within the rail track images. Lower segmentation loss corresponds to better performance in segmenting defects from the background.

train/cls_loss: The classification loss during training is shown here, with a steady decrease over time. This reduction suggests that the model is improving in its ability to correctly classify the detected objects (e.g., fasteners, rails, bolts) into the appropriate categories.

train/dfl_loss: The Distribution Focal Loss (DFL) is depicted in this plot, showing a continuous reduction during training. DFL helps in refining the localization of bounding boxes, ensuring that the predictions are more aligned with the true locations of the defects.

val/box_loss: This plot shows the bounding box loss during validation. Although it follows a similar downward trend as the training loss, it fluctuates slightly more, which is typical as the model encounters more varied data in the validation set. This plot is crucial for assessing the generalization ability of the model.

val/seg_loss: The segmentation loss on the validation data also decreases over time, also with more variability. This indicates the robustness of the model in handling different segmentation challenges across the validation images.

val/cls_loss: The classification loss for the validation set shows a significant decrease, demonstrating the capability of the model to correctly identify and classify different defect types even when presented with unseen data.

val/dfl_loss: Like the training loss, the DFL for validation decreases, suggesting that the model maintains its localization accuracy even on the validation set.

Precision, Recall, and mAP Metrics

metrics/precision(B) and metrics/precision(M): These plots display the precision of the model for the bounding box and segmentation tasks, respectively. Precision improves steadily over the training epochs, indicating a reduction in false positives and showing that the model is becoming more confident and accurate in its defect detections.

metrics/recall(B) and metrics/recall(M): These recall plots illustrate how well the model is identifying all relevant defects. An upward trend is observed, showing that the model is progressively capturing a higher proportion of actual defects, with fewer instances of missed detections.

metrics/mAP50(B) and metrics/mAP50(M): These plots depict the mAP at an IoU threshold of 50%, which measures how well the model balances precision and recall. The increase in mAP50 over time highlights the growing competence of the model in accurate defect detection.

metrics/mAP50-95(B) and metrics/mAP50-95(M): The mAP at varying IoU thresholds (from 50% to 95%) is shown here, providing a more stringent evaluation of the performance of the model. The gradual rise in mAP50-95 indicates that the model is consistently improving across a range of IoU thresholds, underscoring its reliability in both easy and challenging detection scenarios.

These plots collectively provide a detailed overview of the performance of the model during training and validation. The consistent improvement across all metrics demonstrates the effectiveness of YOLOv8-Segment model in detecting and classifying railway track defects, making it a powerful tool for enhancing railway maintenance and safety through automated inspection systems.

5.1.6 Confusion Matrix

A normalized confusion matrix was generated to evaluate the classification performance of the model, providing insight into the ability of the model to correctly classify each defect type while minimizing misclassifications. The matrix visualizes the predictions of the model against the actual defect classes, with the intensity of the color indicating the proportion of correct or incorrect classifications on validation set (Figure 23).

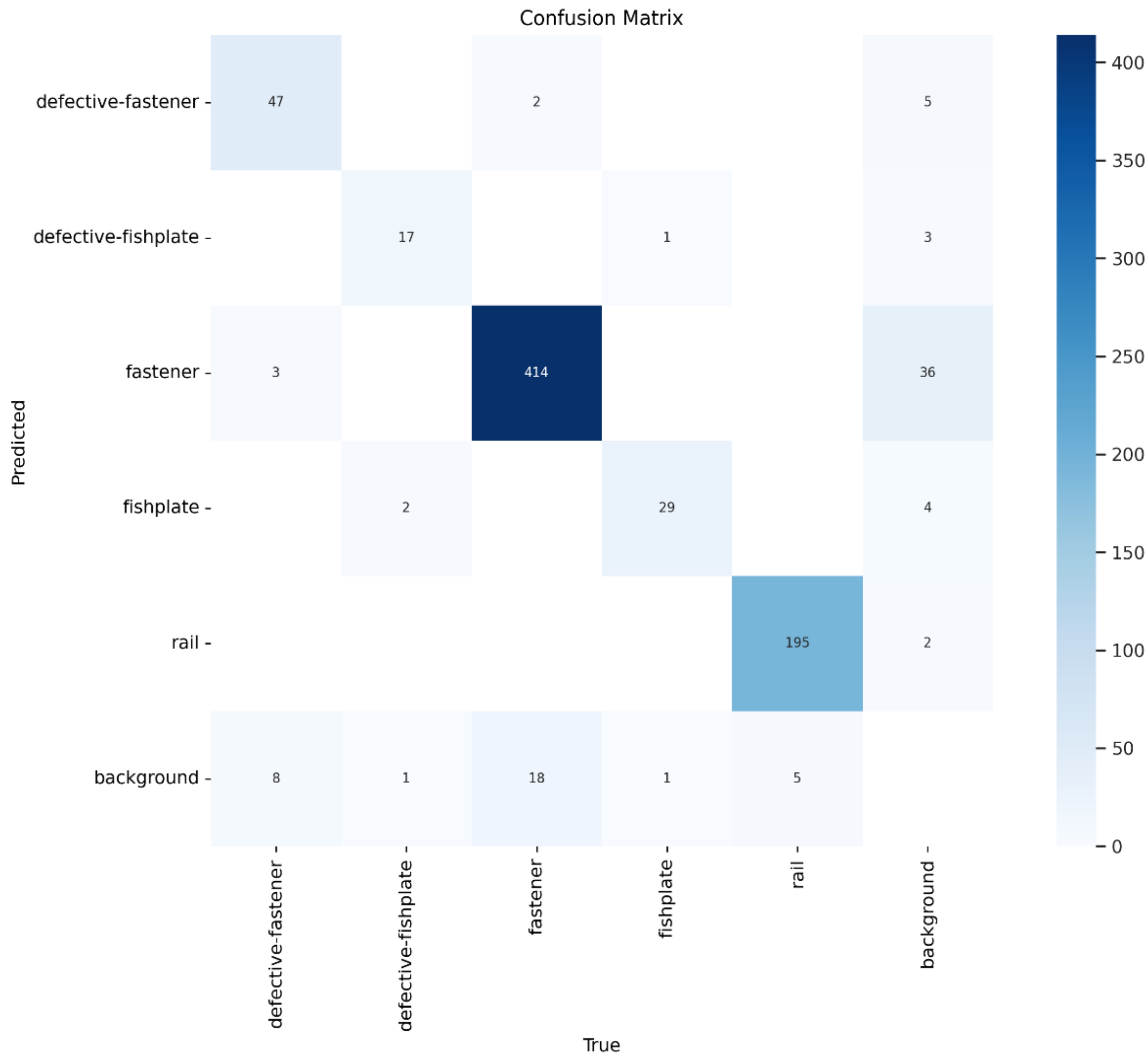


Figure 23: Confusion Matrix of Validation Dataset.

The confusion matrix demonstrates that the YOLOv8-Segment model achieved a high true positive rate across several defect types, particularly for categories like "fastener," "defective-fastener," and "rail," which all showed perfect classification accuracy with values of 0.95. This indicates that the model was highly effective in detecting and correctly classifying these defect types with minimal errors (Figure 24).

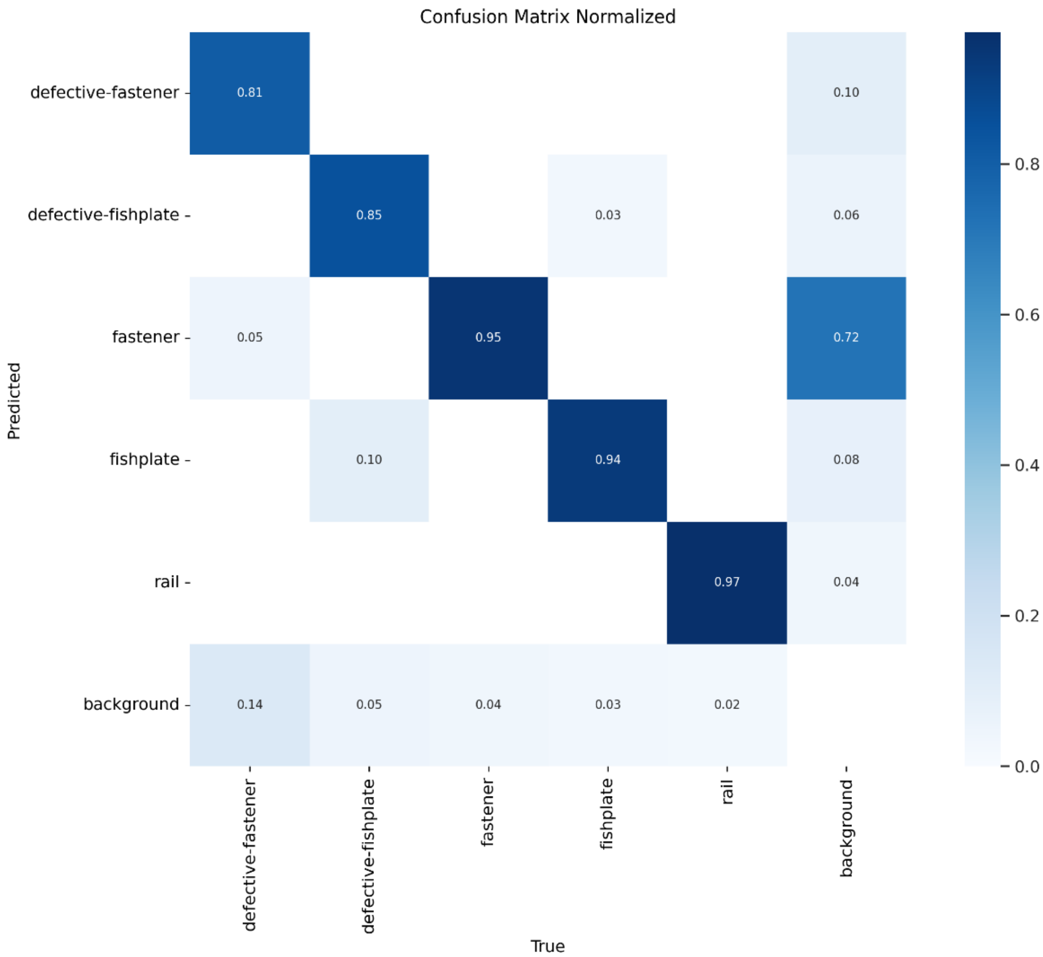


Figure 24: Normalized Confusion Matrix of Custom Trained YOLOv8 Segment Model.

These results underscore the accuracy and reliability of the YOLOv8-Segment model in differentiating between various railway track defects, further supporting its application in real-time railway maintenance and safety operations. The high true positive rates of the model across most classes highlight its potential for enhancing railway safety through automated inspection systems, reducing the likelihood of undetected defects that could lead to operational failures.

5.2 Self-Supervised Learning Model Results

In this section, the results obtained from the self-supervised learning model are provided, which was trained and fine-tuned on the Rail Surface Defect Dataset (RSDDs) and evaluated on both the RSDDs and a custom dataset. This section focuses on the performance of the U-Net-based model across various stages of training, including pre-training and fine-tuning, and comparisons with other research studies and models.

5.2.1 Comparison with Custom Trained Models on Defect Dataset

The self-supervised learning model was benchmarked against various configurations of custom-trained models to evaluate its performance on the defect dataset. The key metric used for this evaluation was the Intersection over Union (IoU) on both the training and validation sets.

Table 7: Comparison of different variants of Custom Trained U-Net Models.

Model	Train IoU	Val IoU
U-Net	42.3 %	31.1 %
U-Net Fine Tuned 50 Epochs	87.4 %	83.2 %
U-Net Fine Tuned 250 Epochs	**91.38 %**	**85.1%**

The initial U-Net model, which was trained solely on normal rail surface images, achieved a relatively low IoU of 42.3% on the training set and 31.1% on the validation set. These results indicated that the model struggled to generalize from normal conditions to defect detection without further training.

Fine-tuning the U-Net model for 50 epochs on the defect dataset significantly improved its performance, with the Train IoU increasing to 87.4% and the Val IoU to 83.2%. This improvement demonstrated that the model effectively adapted to the defect data during the fine-tuning process, enhancing its ability to detect and segment defects accurately.

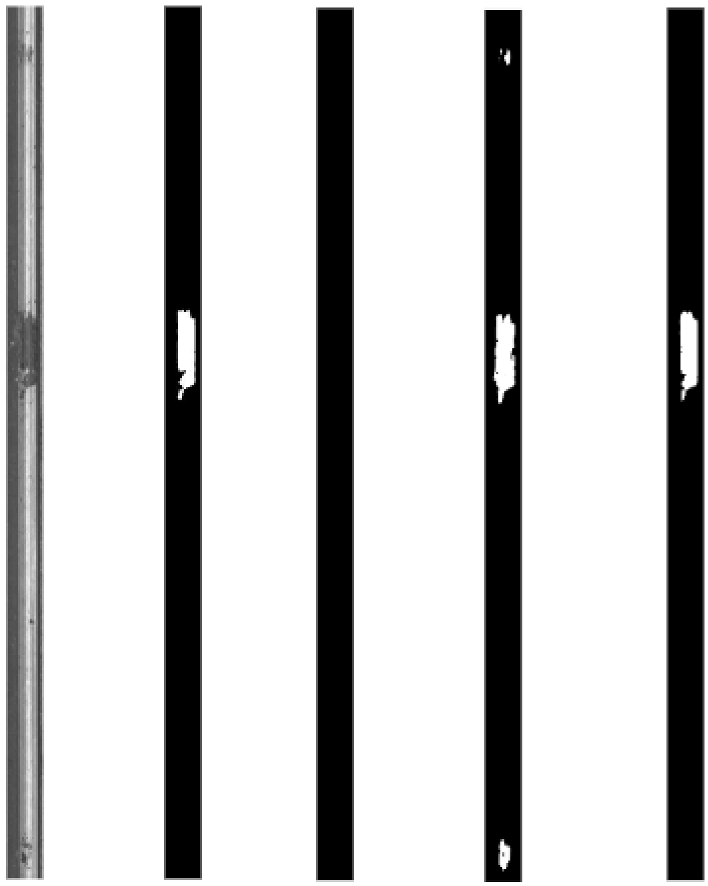


Figure 25: (1) Input Image, (2) Groundtruth, (3) U-Net Output, (4) U-Net (50 Epoch) Output, (5) U-Net (250) Epoch Output.

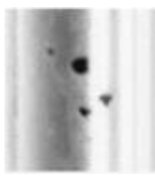
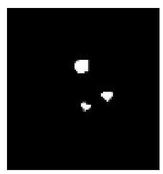
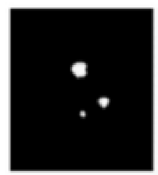
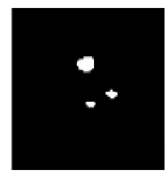
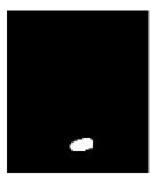
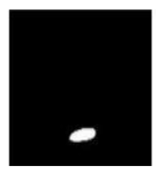
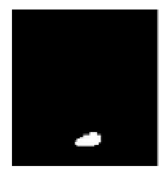
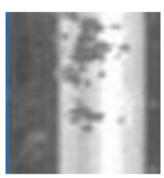
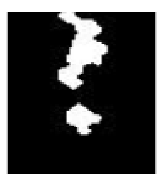
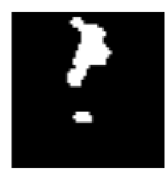
The best performance was observed when the model was fine-tuned for 250 epochs, achieving a Train IoU of 91.38% and a Val IoU of 85.1%. This prolonged training allowed the model to further refine its understanding of defect characteristics, resulting in improved accuracy and generalization across the validation set.

These results highlight the importance of fine-tuning in self-supervised learning models, particularly for tasks that involve significant variation between training and target datasets. The consistent increase in IoU with more epochs of fine-tuning underscores the capacity of the model to learn complex patterns associated with rail surface defects.

5.2.2 Comparison with Other Research on RSDDs Dataset

To further validate the performance of our self-supervised learning model, its results were compared with those from other research studies using the RSDDs dataset. This comparison provided insights into how the proposed model stacks up against established methods in the field.

Table 8: Comparison using RSDDs Dataset with Other Research.

Ref	Input Image	Ground Truth	Result in Ref.	Present Study
(Min & Li, 2022)				
(Xu et al., 2023)				
(Xu et al., 2023)				

The comparison involved analyzing output results where the U-Net model fine-tuned for 250 epochs was evaluated against other state-of-the-art models (Table 8). The results demonstrated that the model presented in this study produced relatively more accurate and detailed segmentation masks for the detected defects. A quantitative comparison between the present study and other state of the art studies is provided in Table 9. These findings suggest that the self-supervised approach, combined with extensive fine-tuning, is at par with other state-of-the-art models.

Table 9: Comparison of Dice Coefficient and IoU between Different Studies.

Models	Dice Do-efficient Performance	Ground Truth
(Min & Li, 2022)	72.6	82
(Xu et al., 2023)	75.5	85.7
Present Study	74.3	84.5

5.2.3 Outputs on Custom Dataset

The performance of the self-supervised learning model was also tested on a custom-collected rail dataset. This dataset contained images that were not part of the RSDDs, providing a unique challenge for the model to generalize its learned features to new, unseen data.

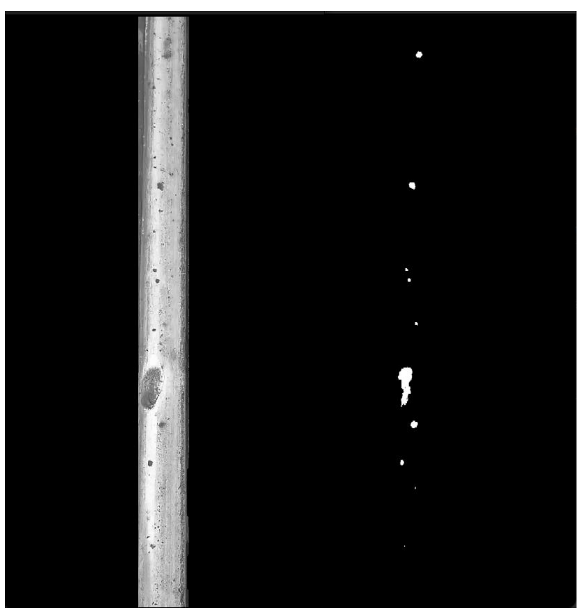


Figure 26: (Left): Input Custom Rail Surface Image, Fig (Right): U-Net (250 Epochs) Segmentation Output.

The 250-epoch fine-tuned U-Net model was applied to this custom dataset, and the results were impressive. The model accurately detected and segmented various rail surface defects, showcasing its ability to generalize beyond the datasets it was trained on. The outputs on the custom dataset confirmed that the model maintained high performance and reliability in real-world scenarios, validating its practical applicability for railway maintenance and safety inspections.

5.2.4 Plots

To visually analyze the training process and the performance of this fine-tuned model, the plots were generated for the Dice Loss and IoU metrics over the course of the training and validation phases. These plots are provided in Figure 27.

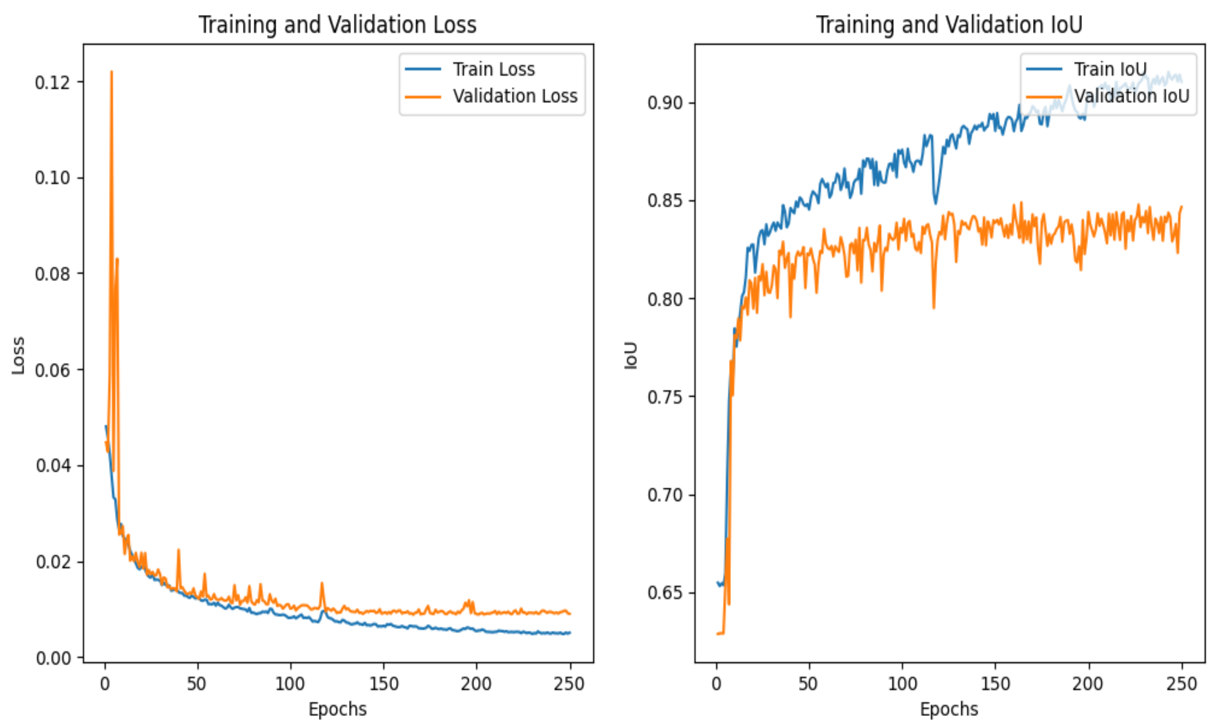


Figure 27: Training and Validation Dice Loss and IOU of 250 Epochs Fine Tuned U-Net Model.

Training and Validation Loss

Training Loss: The plot shows a consistent decrease in training loss across the epochs, suggesting that the model is successfully learning and refining its segmentation capabilities.

Validation Loss: The validation loss also decreases, though with some fluctuations, suggesting the model is generalizing well despite encountering varied data.

These trends demonstrate that the model improves consistently during training, with decreasing errors in both training and validation.

Training and Validation IoU

Training IoU: The IoU steadily increases, reflecting the growing accuracy of the model in segmenting defects, with the curve approaching a high level of performance.

Validation IoU: Although slightly lower than the training IoU, the validation IoU also shows improvement, indicating the ability of the model to generalize well to unseen data.

These plots confirm that the fine-tuning process was essential in achieving high segmentation accuracy, with the model showing strong performance across both training and validation phases.

Chapter 6: Conclusion and Future Work

In this chapter, the results achieved from the present study have been summarized. The chapter also discussed the limitations of the proposed system and mentioned the possible future work of this thesis.

6.1 Conclusion

This study focused on designing an end-to-end railway track defects detection system that detects different types of defects present on railway tracks, including rail surface and rail track component defects. The system was designed using a two-stage approach, incorporating both supervised and self-supervised learning models to achieve comprehensive defect detection.

In the supervised learning stage, the YOLOv8x Segment Model was selected for its superior performance in real-time object detection tasks over Faster-RCNN and EfficientNet. The model was trained on a curated dataset that included a combination of custom-collected images and publicly available dataset. The YOLOv8x Segment Model demonstrated high accuracy in detecting and classifying various track defects, such as missing fasteners, bolts, and fishplates, with a mean Average Precision (mAP) of 94% and a high FPS rate of 30, making it suitable for real-time applications. The robustness of the model was further validated by its strong performance across different metrics, including precision, recall, and IoU, underscoring its effectiveness in real-world railway maintenance scenarios.

In the self-supervised learning stage, a U-Net based model was employed to segment and identify rail surface defects. The model underwent extensive training, beginning with pre-training on normal rail surface images followed by fine-tuning on a specialized defect dataset (RSDDs). The fine-tuned U-Net model achieved significant improvements in IoU, with the best performance observed after 250 epochs. The results confirmed the capability of the model to accurately detect and segment rail surface defects, demonstrating its potential for use in detailed rail surface inspections.

The evaluation of both models involved rigorous testing on custom and validation datasets, with performance metrics such as Dice Loss and IoU used to assess accuracy and generalization ability. The results indicated that both the supervised and self-supervised learning models were highly effective in their respective tasks, providing a comprehensive solution for railway track defect detection.

In conclusion, the developed system presents a powerful tool for enhancing railway safety by automating the detection of various track defects. The integration of advanced deep learning techniques allows for real-time monitoring and precise identification of potential hazards, thereby reducing the risk of accidents and facilitating timely maintenance interventions. This research contributes to the field of railway maintenance by offering a scalable, efficient, and accurate solution for track defect detection, with potential applications in various railway networks worldwide.

6.2 Future Work

Future work could involve extending the current model to detect and analyze additional railway track components such as sleepers and ballast. While the current model can detect other elements, its performance would depend on training with a dataset that includes specific annotations for sleepers and ballast. By enhancing the dataset to cover these components, the model could be fine-tuned to accurately detect and analyze them, thereby providing a more comprehensive monitoring solution. This would ensure that the rail and its immediate components are monitored and the underlying structural elements, leading to a more holistic approach to rail infrastructure safety.

Another area of future development is the implementation of the system in a real-time environment using live video feeds. Integrating the model with cameras mounted on moving trains or dedicated inspection vehicles (carts) would allow for continuous, automated monitoring of the rail network. Real-time processing would enable immediate alerts to maintenance teams, reducing response times and enhancing overall rail safety by detecting defects as they occur.

Finally, the performance of the U-Net module could be improved by incorporating enhanced pseudo defects during the pretraining stage. These synthetic defects, designed to closely mimic real-world anomalies, would help the model better generalize to actual defect scenarios. By expanding the variety of pseudo defects used during pretraining, the model could learn more diverse features, leading to improved accuracy in detecting and segmenting rail surface defects under various real-world conditions.

References

[1] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

[2] BBC Reality Check. (2021). Pakistan train fire: Are accidents at a record high? *BBC*.

[3] Chellaswamy, C., Balaji, L., Vanathi, A., & Saravanan, L. (2017). IoT based rail track health monitoring and information system. *2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 1–6. https://doi.org/10.1109/ICMDCS.2017.8211548

[4] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised Learning.

[5] Dwyer, B., N. J., H. T., et. al. (2024). Roboflow (Version 1.0).

[6] Eunus, S. I., Hossain, S., Ridwan, A. E. M., Adnan, A., Islam, M. S., Karim, D. Z., Alam, G. R., & Uddin, J. (2024). ECARRNet: An Efficient LSTM-Based Ensembled Deep Neural Network Architecture for Railway Fault Detection. AI (Switzerland), 5(2), 482–503. https://doi.org/10.3390/ai5020024

[7] Gan, J., Li, Q., Wang, J., & Yu, H. (2017). A Hierarchical Extractor-Based Visual Rail Surface Inspection System. *IEEE Sensors Journal*, *17*(23), 7935–7944. https://doi.org/10.1109/JSEN.2017.2761858

[8] Hajizadeh, S., Núñez, A., & Tax, D. M. J. (2016). Semi - Supervised Rail Defect Detection from Imbalanced Image Data. *IFAC - Papers Online*, *49*(3), 78–83. https://doi.org/10.1016/j.ifacol.2016.07.014

[9] Hosang, J., Benenson, R., & Schiele, B. (2017). Learning non-maximum suppression. http://arxiv.org/abs/1705.02950

[10] Hussain, M. (2024). YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision. http://arxiv.org/abs/2407.02988

[11] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO. https://github.com/ultralytics/ultralytics

[12] Karakose, M., Yaman, O., Murat, K., & Akin, E. (2018). A New Approach for Condition Monitoring and Detection of Rail Components and Rail Track in Railway *. In *International Journal of Computational Intelligence Systems* (Vol. 11).

[13] Karna, N., Putra, A. P., Rachmawati, S. M., Abisado, M., & Sampedro, G. A. (2023). Toward accurate fused deposition modeling 3d printer fault detection using improved YOLOv8 with hyperparameter optimization Optimization. *IEEE Access*. https://doi.org/10.1109/ACCESS.2017.DOI

[14] Ketkar, N., & Moolayil, J. (2021). Convolutional Neural Networks. In Deep Learning with Python (pp. 197–242). *Apress*. https://doi.org/10.1007/978-1-4842-5364-9_6

[15] Leng, J., Liu, Y., Zhang, T., Quan, P., & Cui, Z. (2018). Context-Aware U-Net for Biomedical Image Segmentation. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2535–2538. https://doi.org/10.1109/BIBM.2018.8621512

[16] Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., & Li, J. (2019). Dice Loss for Data-imbalanced NLP Tasks. http://arxiv.org/abs/1911.02855

[17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. http://arxiv.org/abs/2103.14030

[18] Lopez Pinaya, W. H., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2020). Convolutional neural networks. In *Machine Learning* (pp. 173–191). Elsevier. https://doi.org/10.1016/B978-0-12-815739-8.00010-9

[19] Min, Y., Li, J., & Li, Y. (2023). Rail Surface Defect Detection Based on Improved UPerNet and Connected Component Analysis. *Computers, Materials and Continua*, *77*(1), 941–962. https://doi.org/10.32604/cmc.2023.041182

[20] Min, Y., & Li, Y. (2022). Self-Supervised Railway Surface Defect Detection with Defect Removal Variational Autoencoders. *Energies*, *15*(10). https://doi.org/10.3390/en15103592

[21] Minguell, M. G., & Pandit, R. (2023). TrackSafe: A comparative study of data-driven techniques for automated railway track fault detection using image datasets. *Engineering Applications of Artificial Intelligence*, *125*, 106622. https://doi.org/10.1016/j.engappai.2023.106622

[22] Mishra, M. R., & Bhawar, R. (2022). Design and Development of Device Used for Detection of Cracks on Railway Tracks Article in. In *International Journal of Research in Engineering and Technology*. www.irjet.net

[23] Naimat Khan. (2023). At least 30 killed, 70 injured after train derails in southern Pakistan. *Arab News Pakistan.*

[24] Niu, M., Song, K., Huang, L., wang, qi, Yan, Y., & Meng, Q. (2020). Unsupervised Saliency Detection of Rail Surface Defects using Stereoscopic Images. *IEEE Transactions on Industrial Informatics*, 1–1. https://doi.org/10.1109/TII.2020.3004397

[25] O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. http://arxiv.org/abs/1511.08458

[26] Ragala, Z., Retbi, A., & Bennani, S. (2022). RAILWAY TRACK FAULTS DETECTION BASED ON IMAGE PROCESSING USING MOBILENET. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *48*(4/W3-2022), 135–141. https://doi.org/10.5194/isprs-archives-XLVIII-4-W3-2022-135-2022

[27] Rakshit, S., Sandeep, B. S., & Eliza Femi Sherley, S. (2022). Railway Track Fault Detection using Deep Neural Networks. *2022 IEEE 6th Conference on Information and Communication Technology, CICT 2022*. https://doi.org/10.1109/CICT56698.2022.9997883

[28] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. http://arxiv.org/abs/1506.01497

[29] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression.

[30] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. http://arxiv.org/abs/1505.04597

[31] ŞENER, A., ERGEN, B., & TOĞAÇAR, M. (2022). Fault Detection from Images of Railroad Lines Using the Deep Learning Model Built with the Tensorflow Library. *Turkish Journal of Science and Technology*, *17*(1), 47–53. https://doi.org/10.55525/tjst.1056283

[32] Shah, A. A., Chowdhry, B. S., Memon, T. D., Kalwar, I. H., & Andrew Ware, J. (2020). Real time identification of railway track surface faults using canny edge detector and 2D discrete wavelet transform. *Annals of Emerging Technologies in Computing*, *4*(2), 53–60. https://doi.org/10.33166/AETiC.2020.02.005

[33] Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access*, *9*, 82031–82057. https://doi.org/10.1109/ACCESS.2021.3086020

[34] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. http://arxiv.org/abs/1905.11946

[35] Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, *11*(3), 52. https://doi.org/10.3390/computation11030052

[36] The Engineering Community. (2021). The Main Railway Track Components. *Engineering Articles.*

[37] Wang, T., Yang, F., & Tsui, K. L. (2020). Real-time detection of railway track component via one-stage deep learning networks. *Sensors (Switzerland)*, *20*(15), 1–15. https://doi.org/10.3390/s20154325

[38] Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Engineering Applications of Artificial Intelligence*, *80*, 66–81. https://doi.org/10.1016/J.ENGAPPAI.2019.01.008

[39] Xu, Y., Wang, H., Liu, Z., & Zuo, M. (2023). Self-Supervised Defect Representation Learning for Label-Limited Rail Surface Defect Detection. *IEEE Sensors Journal*, *23*(23), 29235–29246. https://doi.org/10.1109/JSEN.2023.3324668

[40] Zhai, X., Oliver, A., Kolesnikov, A., & Beyer, L. (2019). S 4 L: Self-Supervised Semi-Supervised Learning.

[41] Zheng, D., Li, L., Zheng, S., Chai, X., Zhao, S., Tong, Q., Wang, J., & Guo, L. (2021). A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network. *Computational Intelligence and Neuroscience*, *2021*. https://doi.org/10.1155/2021/2565500

[42] Zhuang, L., Wang, L., Zhang, Z., & Tsui, K. L. (2018). Automated vision inspection of rail surface cracks: A double-layer data-driven framework. *Transportation Research Part C: Emerging Technologies*, *92*, 258–277. https://doi.org/10.1016/j.trc.2018.05.007