

POLITECNICO DI TORINO

POLITECNICO DI TORINO

Master Degree course in Building Engineering

Master Degree Thesis

# Advance Deep Learning Approaches for Defect Detection in Tunnels in Infrastructure Asset Management


**Supervisors**
Prof. Valentina Villa
Prof. Jelena Ninic

**Candidate**
Mohammadhamed Mozafarian


Academic Year 2024-2025

# Acknowledgements

I want to express my deepest gratitude to Professor Valentina Villa for her expert guidance, and to Professor Jelena Ninic from University of Birmingham for her insightful tutorship. Your thoughtful questions and encouragement profoundly shaped the direction of this thesis—I could not have asked for better mentors.

To my supervision team from Politecnico di Torino and University of Birmingham — Paola, Zehao, Kamil, and Giuseppe — thank you for being not only brilliant PhD researchers but also incredibly generous mentors. Your guidance, your time, and your patience meant more than I can express. You listened, challenged my ideas, shared your knowledge, and supported me at every step. I feel fortunate to have had such passionate and dedicated supervisors by my side. A thousand thank-yous.

To my love, Mahta. You've been my anchor and my light, and I could not have done this without your love by my side.

Lastly, I would be remiss not to thank my parents and my sisters. Their unwavering belief in me has been a constant source of strength, lifting my spirits and keeping me motivated through every challenge. Their love and encouragement carried me further than they know.

**Abstract**

Infrastructure relies on periodic monitoring to maintain functionality and reduce risks. Developing strategies for monitoring and managing aged infrastructure assets while respecting time-efficient methods has become a challenge in developed countries. This research presents a novel approach for infrastructure health monitoring by introducing a dataset developed on Italian concrete tunnels concerning defect detection, suitable for training state-of-the-art deep learning computer vision algorithms. The database contains five types of defects suggested by Italian regulations, including Seepage, Spalling, Damaged Joints, Cracks, and Corrosion, and three non-defect classes presenting the tunnel equipment, repair parts, and signs. Consequently, the database was evaluated and compared with three deep learning instance segmentation algorithms to check the efficacy. Additionally, the database was evaluated with semantic segmentation methods to represent the compatibility of multiple usages to distinguish between defect and non-defect classes. Subsequently, a damage report interface was developed to help specialists generate the tunnel defects report satisfactorily. This database was developed to help professionals overcome the problems related to the scarcity of data in the tunnel asset monitoring methods and enhance time efficiency and accuracy.

**Keywords:** Defect detection, Tunnel, Structural health monitoring, Deep learning, Instance segmentation, Semantic Segmentation

# Contents

**List of Tables** 4

**List of Figures** 5

**1 Introduction** 9
- 1.1 Context and Background . . . 10
  - 1.1.1 Infrastructure Asset Management . . . 10
  - 1.1.2 Transportation Asset Management . . . 11
  - 1.1.3 Tunnel Asset Management . . . 12
  - 1.1.4 Italian Guidelines and Standards . . . 14
- 1.2 Problem Statement and Objectives . . . 15

**2 Deep Learning Algorithms** 17
- 2.1 Different Learning Modes . . . 17
- 2.2 Various Deep Learning networks . . . 19
  - 2.2.1 Convolutional Neural Network . . . 19
  - 2.2.2 Faster R-CNN . . . 20
  - 2.2.3 Mask R-CNN . . . 21
  - 2.2.4 Mask Scoring R-CNN . . . 21
  - 2.2.5 Cascade Mask R-CNN . . . 22
  - 2.2.6 Hybrid Task Cascade . . . 23
  - 2.2.7 PointRend . . . 24
  - 2.2.8 SOLOv2 . . . 26
  - 2.2.9 CondInst . . . 26
  - 2.2.10 Transformer . . . 27
  - 2.2.11 Vision Transformer . . . 29
  - 2.2.12 SWIN Transformer . . . 29
  - 2.2.13 QueryInst . . . 31
  - 2.2.14 Mask2Former . . . 32

**3 Structural Health Monitoring** 35
- 3.1 Overview and background . . . 35

3.2 Deep Learning Based Structural Health Monitoring . . . . . 37
3.3 Computer Vision-based Approaches for Surface Damage Detection . 37
3.3.1 Damage Classification . . . . . 38
3.3.2 Bounding Box Level Damage Detection . . . . . 39
3.3.3 Pixel Level Damage Segmentation . . . . . 40

**4 Methodology** 43
4.1 Methodology . . . . . 43
4.1.1 Data Acquisition Method . . . . . 43
4.1.2 Image Processing . . . . . 48
4.1.3 Labeling and Creation of the Dataset . . . . . 50
4.1.4 Neural Network Training and Evaluation . . . . . 53
4.1.5 Evaluation Metrics . . . . . 54

**5 Result and Discussion** 59
5.1 Parameter Setting and Hardware . . . . . 59
5.2 Result of instance segmentation algorithms . . . . . 59
5.2.1 Detailed Analysis . . . . . 60
5.2.2 Discussion . . . . . 61
5.3 Result of semantic segmentation algorithm . . . . . 65
5.3.1 Discussion . . . . . 66

**6 Damage Report** 73

**7 Conclusion** 77

**Bibliography** 79

# List of Tables


3.1 CV-based method comparative analysis. . . . . 42

4.1 Taxonomy of Defects . . . . . 52
4.2 Instance segmentation models under investigation. . . . . 54
4.3 Semantic segmentation models under investigation. . . . . 54

5.1 Instance segmentation models result . . . . . 60
5.2 Comparative Analysis of CNN-based Instance Segmentation Algorithms on Our Dataset . . . . . 64
5.3 Comparative Analysis of Transformer-based Instance Segmentation Algorithms on Our Dataset . . . . . 65
5.4 Result of Mask R-CNN across different backbone sizes. . . . . 67
5.5 Result of Cascade Mask R-CNN across different backbone sizes. . . 67
5.6 Result of Mask2Former across different backbone sizes. . . . . 68
5.7 Comparative Analysis of Instance Segmentation Algorithms with Swin-Large Backbone on Our Dataset . . . . . 68
5.8 Semantic segmentation models under investigation. . . . . 70
5.9 Comparative Analysis of Semantic Segmentation Algorithms on Our Dataset . . . . . 71

# List of Figures


1.1 Overview of proposed framework . . . . . 16

2.1 Evolution of the benchmark network over the years. The blue rectangles model is used in this research to benchmark the proposed dataset. . . . . 19
2.2 First CNN architecture for concrete crack detection [55, 56] . . . . . 20
2.3 Faster R-CNN architecture [56] . . . . . 21
2.4 Architecture of Mask R-CNN for instance segmentation [41] . . . . 22
2.5 Architecture of Cascade Mask R-CNN. The "C" is classification, "B" is bounding box, and “S” denotes a segmentation branch. . . . . 23
2.6 The architecture evolution from Cascade Mask R-CNN to Hybrid Task Cascade [44]. . . . . 24
2.7 Architecture of PointRend [46] . . . . . 25
2.8 Architecture of SOLOv2 [47] . . . . . 27
2.9 The overall architecture of CondInst [48] . . . . . 28
2.10 Architecture of a typical transformer [56] . . . . . 28
2.11 Vision Transformer model overview [45] . . . . . 30
2.12 overview of Swin Transformer architecture [49] . . . . . 31
2.13 Architecture of QueryInst. The red arrows indicate mask branches [50]. 32
2.14 Overview of Mask2Former [51] . . . . . 33

4.1 Schematic of data acquisition equipment . . . . . 44
4.2 Vehicle equipped with Spacetec laser scanner collecting data in tunnels 47
4.3 Methodology of image capturing . . . . . 48
4.4 Location of the tunnels under investigation in Italy . . . . . 49
4.5 a) Report file of one section annotated by experienced engineers through visual inspection. b) The captured image of the corresponding section. . . . . 50
4.6 comparison of a tunnel section before and after processing. The raw captured image is shown on the left, while the processed version, after applying gamma correction and removing the road section, is on the right. . . . . 50

4.7 The ISAT software user environment used during the labeling phase. 51
4.8 Example of categories represent: a) Equipment, b) Repair part, c) Traffic Sign, d) Seepage, e) Spalling, f) Corrosion, g) Damaged joint, h) Crack . . . . . 52
4.9 Overall process of image processing from a large ultra-high resolution image obtained directly during the inspection to a labeled processed image for the dataset. . . . . 53
4.10 Explanation of the Intersection Over Union (IoU) . . . . . 55
4.11 Precision-recall curves for AP: (a) the area under the curve (AUC) represents the AP. (b) The larger AUC represents the higher AP. In this curve blue line has a higher AP than the orange line. . . . . 56

5.1 Prediction visualization of instance segmentation models with confidence score of 0.3 . . . . . 61
5.2 Prediction visualization of instance segmentation models with confidence score of 0.3 . . . . . 62
5.3 Confusion matrix of instance segmentation models with confidence score of 0.3 . . . . . 63
5.4 Normalized Confusion Matrix of Instance Segmentation models with Swin-Large backbone with multiple confidence Thresholds . . . . . 66
5.5 Some example of the instance segmentation models prediction with Swin Large backbone with confidence score of 0.3 . . . . . 69
5.6 Example of prediction of the Semantic Segmentation test on our dataset. Green instances represent the Non-defect class while the red ones show the Defects. . . . . 70
5.7 Example of prediction of the Semantic Segmentation test on our dataset. Green instances represent the Non-defect class while the red ones show the Defects. . . . . 72

6.1 Interactive interface for tunnel panoramic image damage detection. 74
6.2 First example of automatically generated damage report from two different tunnels based on 20m tunnel local panoramic images. . . . 75
6.3 Second example of automatically generated damage report from two different tunnels based on 20m tunnel local panoramic images. . . . 76

# List of Acronyms

- **AI:** Artificial Intelligence
- **DL:** Deep Learning
- **CNN:** Convolutional Neural Network
- **DT:** Digital Twin
- **TEN-T:** Trans-European Transport Network
- **IAM:** Infrastructure Asset Management
- **EEA:** European Environment Agency
- **TAM:** Transportation Asset Management
- **TuAM:** Tunnel Asset Management
- **UHR:** Ultra High Resolution
- **GPU:** Graphic Processing Unit
- **SHM:** Structural Health Monitoring
- **NDT:** Nondestructive Testing
- **FEM:** Finite Element Model
- **ANN:** Artificial Neural Network
- **SVM:** Support Vector Machine
- **RGB:** Red, Green and Blue
- **UAV:** Unmanned Aerial Vehicle
- **UGV:** Unmanned Ground Vehicle
- **GPS:** Global Positioning System

- **CCTV:** Closed-Circuit Television
- **CBS:** Complex Background Scene
- **IoU:** Intersection over Union
- **AP:** Average Precision
- **AR:** Average Recall
- **LiDAR:** Light Detection and Ranging
- **SAM:** Segment Anything Model
- **MS COCO:** Microsoft Common Objects in Context
- **GT:** Ground Truth

# Chapter 1

# Introduction

Transportation infrastructure plays a vital role in everyday human life. While these kinds of structures are designed to serve for a long period, periodic maintenance and monitoring of them is significantly important to enhance their functionality and also decrease the risk of any incident and probable dangers. Among different kinds of transportation infrastructures, the importance of tunnels in mountainous areas is undeniable. Turning to Italy, with its Alps and Apennine mountains role of tunnels is crucial in transportation.

The Italian road network is hierarchically structured, from a managerial and administrative point of view, into motorways, state roads, regional, provincial, and municipal roads. Part of the Italian road network belongs to the Trans-European Transport Network (TEN-T), which is a planned network of roads, railways, airports, and water infrastructure in the European Union [1]. TEN-T demonstrates plans and guidelines for managing and maintaining the transportation network, including tunnels. The extension of the TEN-T falling within the national Italian level is equal to 9481 km since the last update; it therefore has an extension of less than 1% of the entire Italian road network. Across Europe, Italy is the EU country with the highest number of tunnels belonging to the TEN-T network, with a total of 465 tunnels. However, just about 19% (49 tunnels) of them in operation are categorized as "compliant" based on minimum safety requirements suggested by Legislative Decree no. 264 of 5 October 2006 [2]. Although the total number of Italian tunnels belonging to the TEN-T is limited compared to all the tunnels in Italy, it reflects the critical condition of the Italian tunnels and the necessity for their maintenance.

## 1.1 Context and Background

### 1.1.1 Infrastructure Asset Management

Infrastructure Asset Management (IAM) is the methodical, coordinated process of carrying out all necessary procedures and activities for an organization to effectively and sustainably manage its infrastructure assets throughout the course of their full life cycles. This includes the methods of acquisition, execution, maintenance, and disposal, all aimed at enhancing performance at the lowest possible cost and risk. It also contains financial, engineering, and environmental considerations that assist firms in making proper decisions on their assets. It is crucial to remember that different companies and sectors may have varying definitions of IAM [3].

The idea of IAM is crucial for addressing several issues that governments and organizations face around the world, particularly in metropolitan areas where more responsive and sustainable structures must be developed. Additionally, IAM integrates all of these, including risk control and technology utilization to improve asset performance and lifetime [4].

Over the past few years, IAM has grown in importance for a number of reasons. First of all, in many affluent nations, deteriorating infrastructure leads to regular failures that endanger public safety and result in significant losses. The American Society of Civil Engineers estimates that by 2029, engineers estimate that the US will require an additional $2.59 trillion for infrastructure [5]. Second, the impact of climate change on infrastructure resiliency and life expectancy is understood, and this situation emphasizes the importance of IAM strategies in this era. Severe flooding, sea level rise, extreme weather events, and an increase in the instances of severe and more frequent weather occurrences have a huge impact on the longevity and functionality of several systems and services, and that could jeopardize the usability and security of any infrastructure. IAM plays a vital role as a systematic monitoring and improvement approach in the face of climate change and also any severe natural disaster [6, 7].

IAM encompasses more than just maintenance; it also involves taking care of all aspects of infrastructure systems. It includes determining the assets' present condition, determining future needs, and making decisions based on assessment data. As mentioned, IAM applies to all types of infrastructure, such as public facilities, utilities, and transportation networks, highlighting the significance of the overall framework that should accommodate various asset types and their conditions [8].

Among different kinds of infrastructure, the transportation network has long been regarded as a vital part of a society's infrastructural development because it provides essential services necessary for economic growth, in addition to ensuring security and the safety of individuals and property. IAM served as an essential function in monitoring, periodic maintenance, and optimization of the functional

physical transportation entities such as roads, bridges, tunnels, and rail lines. Effective management of transport assets guarantees availability and reliability, and at the same time lowers the overall cost of interventions, reduces the risks of any hazardous incidents, and increases the safety of users and operators [9].

Transportation assets and facilities are extremely susceptible to slow degradation and sometimes catastrophic loss due to natural disasters. As a result, IAM strategies are crucial for managing the risks associated with asset failure within the construction and service process. Systematic periodic inspections of bridges and tunnels can detect areas of weakness that may cause devastating collapses by providing evidence for maintenance and replacement. In the same way, taking preventive measures to extend the life of the transport asset can significantly reduce costly repairs and unscheduled downtime [10].

Transportation infrastructure is exposed to the consequences of climate change disasters such as storms, frequent flooding, and sea level rise. According to the European Environment Agency (EEA), it is calculated that in the period 1980-2022, the losses in the economy due to the consequences of weather related natural disasters in the European Union (EU) amounted to 650 billion euros, the consequences of which threaten critical infrastructure [11]. To overcome these risks, IAM develops plans and strategies for risk management and resiliency approaches. In this way, asset managers can determine which link in the transport network is most susceptible to climate change impacts by completing a vulnerability assessment. This information allows them to allocate resources for adaptation strategies like building flood defense structures, promoting better drainage systems, using advanced materials, real-time data gathering and analyzing, etc.

Therefore, IAM is relevant to transportation systems because it ensures the safety, functionality, and resilience of transportation networks in the face of climate change, population growth, and aging infrastructure.

### 1.1.2 Transportation Asset Management

Transportation Asset Management (TAM) is essential for maintaining, operating, and improving transportation infrastructure throughout its life cycle. Transportation infrastructure plays a fundamental role in a country's economic growth and quality of life, including highways, bridges, tunnels, transit systems, etc. The purpose of TAM is to maximize the value of serviceability, safety, and utilization for transportation assets while minimizing their life cycle cost [12]. TAM can be thought of as the tool for regulating the productivity of physical assets, establishing service requirements, and determining the best prices for maintenance, repair, and replacement programs. It comprises choosing from a wide variety of decision-making methods within the scope of risk management, performance evaluation, and life-cycle cost analysis. One of the key tenets of TAM is the application of condition-based metrics, which assess the present condition of assets to inform maintenance

choices. TAM offers concrete proof of how improved resource utilization and efficient use of available resources may be accomplished while lowering the overall life-cycle cost and further extending the life of these systems through the ongoing evaluation of the current state of transportation systems [13].

The recognition of infrastructure management's importance in modern businesses was the primary factor in TAM's rise to prominence. Particularly in the US, TAM's adoption has been caused by the necessity that all transportation agencies include the value of infrastructure on their financial balance sheets to comply with modern regulations. This rule forced agencies to concentrate on evaluating asset performance and incorporate quantitative financial management techniques into future infrastructure development. Several nations with varying economic standing have realized the potential of TAM concepts for expansion and modification to fit and improve the current and future local and international demand for transportation [14].

Developed countries around the world are investigating various platforms to maintain TAM strategies for their assets. Like KUBA for road management in Switzerland, DANBRO for bridge management system in Denmark, BaTMan bridge and tunnel management in Sweden, and MRWA and NSW for bridge management systems in Australia [15]. Infrastructure managers progressively rely on infrastructure management platforms to enhance their decision-making procedures. Those responsible for developing and owning these systems stand to gain valuable insights into the latest capabilities of the most advanced systems and a comparative analysis of their system against others. This understanding serves as a valuable resource for shaping the future development of their systems. Additionally, it facilitates the identification of contacts who can provide in-depth insights into how others have successfully executed or are in the process of implementing plans similar to their own [16].

This research mainly focuses on the tunnels as part of transportation assets, and in the following parts, key challenges and opportunities for the development of the tunnel asset management will be discussed.

### 1.1.3 Tunnel Asset Management

Tunnel Asset Management (TuAM) can be identified as a subcategory of Infrastructure and Transportation Asset Management. TuAM has unique complications mainly because tunnels are highly specific structures with complex environments and with considerable consequences in the event of their failure. This is because tunnels serve as the foundation for transportation networks, particularly in mountainous areas where utilities and goods are transported by rail and roads. Due to the aging tunnel infrastructure and the increasing effects of climate change, which affect natural hazards like earthquakes, landslides, and floods, TuAM is comparatively important. Assessing condition, maintaining, controlling risk, and integrating

technology to improve tunnel performance and safety, as well as longevity, are the main components of an efficient TuAM.

Tunnels are built to function in conditions that are highly stressful and include forces like the earth pressure, hydrostatic pressure, and dynamic loads from the vehicles. Thus, more frequent inspections and more rigorous maintenance regimes are necessary. A major problem specific to TuAM is the interaction between soil and structure. Increasing urbanization and the need to use space more efficiently have resulted in the construction of more tunnels.

Tunnels are built on a heterogeneous soil matrix, which means that their structural behavior is influenced by surrounding geological conditions. Because of these dynamics, it is difficult to assess tunnel response to loading conditions, which requires complex simulation tools during tunnel design, as well as geological explorations [17].

The ingress of ground and underground water is one of the main issues in tunnel design and maintenance. Insufficient drainage systems or high groundwater levels can accelerate tunnel lining degradation. Without proper control, it can lead to failure, expansion, cracks, and complete collapse of a building. As a result, drainage systems and waterproof membranes need to be built and installed properly, and moisture seeping into tunnel structures must be monitored periodically to ensure the safety [18].

To ensure the safety and serviceability of tunnels during their service life towards the mentioned risks, a strategic plan for periodic monitoring is essential. In addition, real-time monitoring for the prevention of unpredictable incidents such as fire outbreaks and traffic accidents is significantly needed. Therefore, stakeholders recognize and develop technologies to monitor tunnel assets and also plan strategies for periodic monitoring and maintenance to prevent and mitigate the risk of any hazardous incident. In the following sections, challenges and opportunities for tunnel monitoring strategies will be discussed.

### Periodic Monitoring and Maintenance Strategies for Tunnel Assets

Tunnel maintenance aims to respond to the current tunnel problems, as well as prevent future and probable problems with assets. As a result of so many assets, a comprehensive monitoring strategy is required in order to manage them effectively in terms of ordering, decision making, and budget allocation. Current development in data acquisition methods and technology opens up a new era for monitoring strategies for transportation assets, including tunnels [19]. These emerging acquisition technologies ease the use of tools such as Building Information Modeling (BIM) [20], Digital Twin (DT) [21] and Artificial Intelligence (AI) [22] approaches. These technological tools facilitate efficient management of the tunnel infrastructure since they provide details information from digital images and spatial data, respectively.

From various kinds of monitoring, Structural Health Monitoring (SHM) has a significant role in ensuring the long life of infrastructure assets. In the case of tunnels, SHM acts as a preventive method that will help to minimize the risk of structural failure in tunnels. SHM methodology, by exploring various data and information, can suggest the optimum and efficient time for intervention and maintenance of the assets. This activity can be dynamic, and it occurs both internally as a result of traffic loads and externally because of weather conditions, among others. SHM systems, in particular, allow providing constant updates on the state of the structure and thus minimizing operational risks since the structure is monitored via different methodologies such as real-time sensors or non-destructive techniques.

Consequently, SHM plays an important role in identifying infrastructure asset problems at an early stage before they turn into deformity, cracking, stress, or even collapse.

The next section is focused on machine learning- and deep learning-based structural health monitoring approaches in more detail.

### 1.1.4 Italian Guidelines and Standards

In 2022, the Italian Ministry of Infrastructure and Sustainable Mobility introduced new guidelines for the classification and management of risk, safety assessment, and monitoring of existing tunnels through Ministerial Decree 247/22 [23]. The Guideline illustrates a procedure for the management of activities aimed at the safety of existing road tunnels in order to prevent inadequate levels of damage, such as affecting the safety of the work and, more generally, of the entire infrastructure, making the risk acceptable. The approach aims to pursue a conduct of prevention with respect to the emergence of potentially dangerous situations, to plan the adoption of preventive maintenance interventions without incurring conditions of emergency intervention. The goal of the guideline is developing an adoption of awareness and prevention conducts for the entire area of road tunnels in operation, it being understood that if conditions that require the adoption of immediate interventions are already recognized, the operator will proceed with the identification and implementation of the interventions recognized as necessary regardless of the application of the procedures provided for by the guidelines.

The primary objective of these guidelines is to establish a quantifiable level of safety to guide maintenance decision-making, aiming to minimize the risk of hazardous situations and avoid the need for urgent interventions. The document proposes a framework structured into three main steps: collecting existing data (e.g., tunnel length, location, previous maintenance interventions, etc.), risk classification, and safety assessment. The guidelines divide tunnels into segments called "Conci", each 20 meters long. Every segment is analyzed and assigned a specific risk level called the "Class of Attention". These classes serve as the foundation for decision-making in planning maintenance interventions.

As described in the guidelines, the Classes of Attention are determined through a simplified assessment of the risk factors associated with tunnels based on available knowledge and inspections. However, the recording and labeling process of the defects to acquire the class of attention remains primarily manual and is done by experienced technicians.

## 1.2 Problem Statement and Objectives

In this thesis, I introduce a methodology for creating a multi-type defect dataset suitable for instance segmentation tasks derived from Panoramic Ultra High Resolution (UHR) images captured from Italian concrete tunnels, and consequently, benchmark this dataset using various state-of-the-art Convolutional Neural Networks (CNNs) and Transformer-based algorithms.

The work was conducted in collaboration with Tecne, part of *"Autostrade per l'Italia"*, which, as the manager of the tunnel infrastructure, has had to adapt to the new directives since 2022 by completing the analyses and inspections described in the guidelines for all the tunnels under its management. Among the documents provided by Tecne, the primary resources included .TIFF images, which were used as the basis for developing the dataset. Additional documents provided by the Tecne team were collected during on-site inspections conducted by experts, including detailed data and information on all defects identified within the tunnels. All documents were prepared for each 20-meter segment, with defects coded according to the defect classification system outlined in Ministerial Decree 247/22. Each defect was assigned a unique numerical code, allowing retrieval of detailed information from various documents, including: 1- Shape and position on the segment (via inspection sheets), 2- Geometric details (via Excel files), and 3- Photographic documentation (via photo reports).

An overview of the proposed methodology is represented in Figure 1.1. The first part demonstrates the data acquisition process and available reports from on-site inspection, including an illustration of the data capturing technique, an example of unprocessed raw images, and related reports. The second section details the image processing workflow and the SAM-based annotation technique used for labeling the defects, and lastly formats the dataset in compliance with the COCO standard. Finally, the last section describes the training and benchmarking procedures, utilizing various State-of-the-Art instance segmentation and semantic segmentation algorithms.

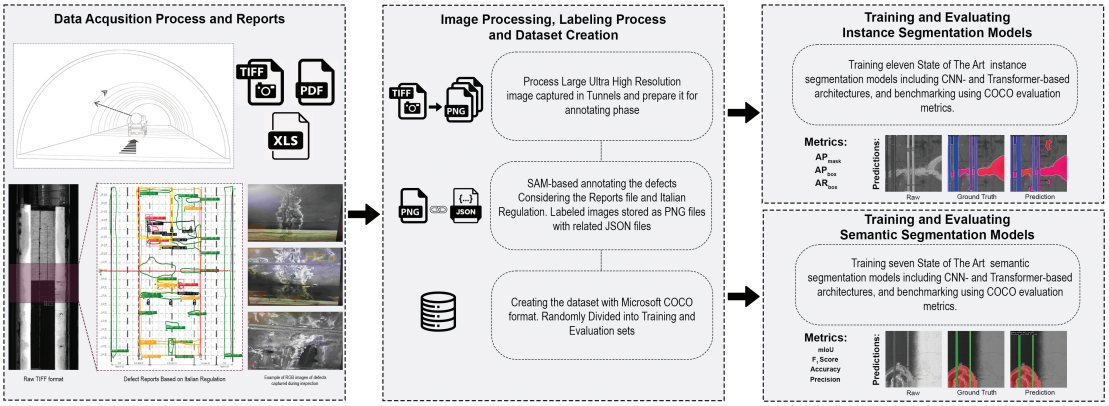



Figure 1.1: Overview of proposed framework

# Chapter 2

# Deep Learning Algorithms

The structure of DL networks is comparable to that of the human brain. These networks use several hidden layers to learn or identify patterns and extract features from the training data. In the DL architecture, an input is sent to each hidden layer, travels through the layers, and then gives the output in the last layer. DL's benefit is that it can automatically extract features during training, which allows the deep neural network to achieve its goal. This network's accuracy rises as the amount of training data increases. Advances in technology and the enhanced performance of graphic processing units (GPUs) and their parallel computing capabilities made this highly computational procedure possible.

DL has assumed a prominent position in the field of SHM in recent years. This change is a reaction to the growing difficulties that contemporary civil infrastructure presents. With sensors producing large, complex datasets, DL provides a potent remedy by automatically deriving insightful information. Furthermore, the capabilities of SHM have been enhanced by DL's incorporation of computer vision, which allows the analysis of 1D data, such as vibration or strain measurements, to 4D data, like RGB videos for damage assessment. These advancements, along with improvements in technology and user-friendly frameworks, have made DL a crucial instrument for improving structural safety and democratized its application in SHM.

In this chapter, different learning modes will be discussed in section 2.1, and after that, various deep learning networks used in DL-based SHM will be presented in section 2.2.

## 2.1 Different Learning Modes

Deep Learning (DL) networks are mostly regarded as a subfield of machine learning. Deep neural networks can learn in four different ways: supervised, unsupervised, semi-supervised, and reinforcement learning.

In supervised learning modes, the network's learnable parameters are changed via backpropagation by calculating a loss function that represents the discrepancy between the actual output and labeled target values or ground truth training data. The learnable parameters are updated via backpropagation based on the chain rule in order to minimize the computed gap. As a result, by using labeled data for training, the networks in this supervised mode have an opportunity to discover the patterns of the target objects. As a result, the network often needs a significant quantity of labeled data to train because of the nature of this learning mode. SHM uses these supervised modes to detect and localize damage from vibrations, and to detect and segment cracks and segment problems.

However, it can be difficult to gather ground truth data from actual structures that represent different damage scenarios. Every piece of civil infrastructure has different characteristics, such as material properties and associated dynamic behaviors, as well as unique boundary conditions. Consequently, information gathered for supervised learning from diverse damage scenarios could not be transferable to different training models [24]. Therefore, unsupervised learning has been employed to overcome these limitations.

In unsupervised learning, the network examines the data and unlabeled datasets. Without the assistance of an outside human, these models uncover hidden patterns in the data. Only data from the baseline structures is used to train the networks for SHM applications. The challenges of supervised learning can be addressed by this unsupervised DL model. The input data can be successfully reproduced by the well-trained network using only data from the baseline structure. The input data may be regarded as outliers in comparison to the learned data if the trained network is unable to faithfully replicate the input. The autoencoder is one of the representative unsupervised DL networks [25–27].

Another mode of learning is semi-supervised learning. Both labeled and unlabeled data are used in the training dataset for semi-supervised learning. When there is a small quantity of ground truth labeled data or when preparing a large number of ground truth data for training requires less work, this semi-supervised mode can be used. Thus, to accomplish the network designer's objective, both supervised and unsupervised modes are used in combination.

The last mode is Reinforcement Learning. In order to accomplish a goal in a complicated or unpredictable environment, the algorithms in reinforcement learning use a series of decisions derived from the rewards through initial random trials and errors. To solve an issue, trial-and-error methods are used, and either rewards or punishments are applied for the activities taken. In reinforcement learning, maximizing the reward is the main objective.

## 2.2 Various Deep Learning networks

Over the years, specifically the last 10 years, many advanced DL algorithms have been developed to overcome the limitations of traditional Machine Learning (ML) methods. In the Figure 2.1, an overview of the most popular ML and DL networks over the years is described.

These influential research papers and architectures include LeNet [28], AlexNet [29], ZFNet [30], Generative Adversarial Network (GAN) [31], GoogleNet [32], VGGNet [33], R-CNN [34], ResNet [35], Unet [36], FCN [37], fast R-CNN [38], faster R-CNN [39], YOLO [40], SegNet, Mask R-CNN [41], MS R-CNN [42], Cascade Mask R-CNN [43], HTC [44], ViT [45], PointRend [46], SOLOv2 [47], CondInst [48], SWIN Transformer [49], QueryInst [50], Mask2Former [51]. Some of the most famous DL methods have also been adopted for DL- based structural health monitoring. In this section, some core DL algorithms, including algorithms that were employed in this research, will be reviewed.

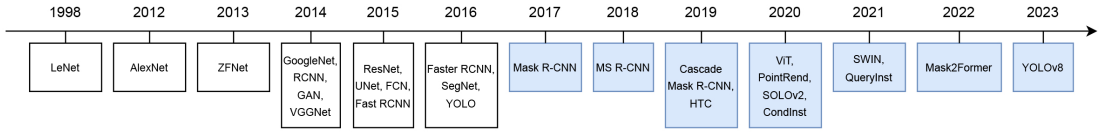



Figure 2.1: Evolution of the benchmark network over the years. The blue rectangles model is used in this research to benchmark the proposed dataset.

### 2.2.1 Convolutional Neural Network

The convolutional neural network (CNN), which was first created to identify handwritten zip code digits supplied by the USPS, is the most representative initial DL network [52]. Inspired by biological processes, a CNN's neuronal connection patterns mimic the structure of the animal visual cortex, which has shown promise in solving a variety of image recognition issues. CNNs effectively extract information from input images and require less computation and pooling because of their sparse connectivity. CNNs may also be able to distinguish between a wide range of classes. CNNs are an effective technique for image recognition because of these special benefits.

With CNNs, the main challenge was the need for a large amount of labeled data, but this challenge was overcome through pretraining for transfer learning using well-annotated databases, such as ImageNet [53], CIFAR-10, and CIFAR-100 datasets, and the MNIST Database [54].

The CNN architecture consists of three parts including convolution, pooling, and fully connected layers. Cha et al. [55] designed a new CNN for crack detection as

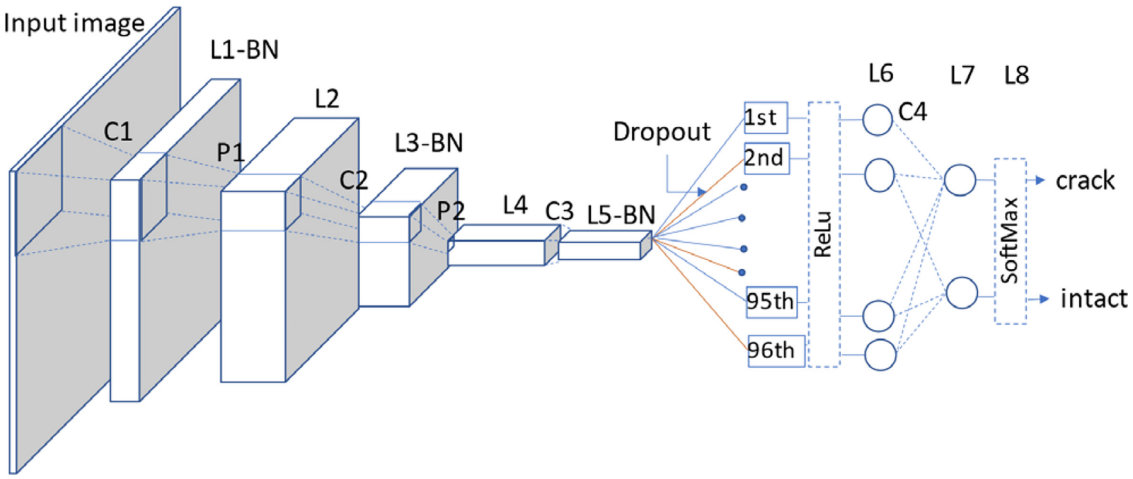



Figure 2.2: First CNN architecture for concrete crack detection [55, 56]

with use of CNN architecture. The number of layers and sequences depends on the types of data and the level of accuracy. In most cases, the filter size is smaller than the input size. In the pooling layer, down-sampling is used to reduce the dimensions of the inputs and reduce the computational cost.

### 2.2.2 Faster R-CNN

The faster R-CNN was proposed by Ren et al. [39] and consists of two parts: the region proposal network (RPN) and the fast R-CNN. Figure 2.3 shows the architecture of Faster R-CNN. Using input and object proposals from selective searches [57], a region-based CNN (R-CNN) [34] was created for multiple object detection and localization. A CNN was then utilized to extract features. Accuracy was greatly increased by the R-CNN in comparison to CNN-based techniques. However, because there were three distinct training procedures—a CNN, a regressor, and SVMs—it was computationally expensive and time-consuming.

Background of faster R-CNN enriched with creation of fast R-CNN [38] to overcome the drawbacks of R-CNNs, and it performed better in terms of speed and accuracy. Fast R-CNN's accuracy and speed remained poor despite the improvement because of the laborious external selective search strategy. Ren et al. [39] created the faster R-CNN by combining an RPN and fast R-CNN to enhance training accuracy to address these problems. The quick R-CNN uses region of interest (RoI) pooling to extract features from the candidate bounding boxes that the RPN suggests. Classification and bounding box regression are then performed. Through end-to-end network training and feature sharing between the RPN and fast R-CNN, the faster R-CNN improves accuracy. Faster R-CNN enables researchers to investigate real-time object detection and has found applications in various fields. In the field of SHM, for the first time, Cha et al. [58] utilized this architecture to detect

multiple structural damages.

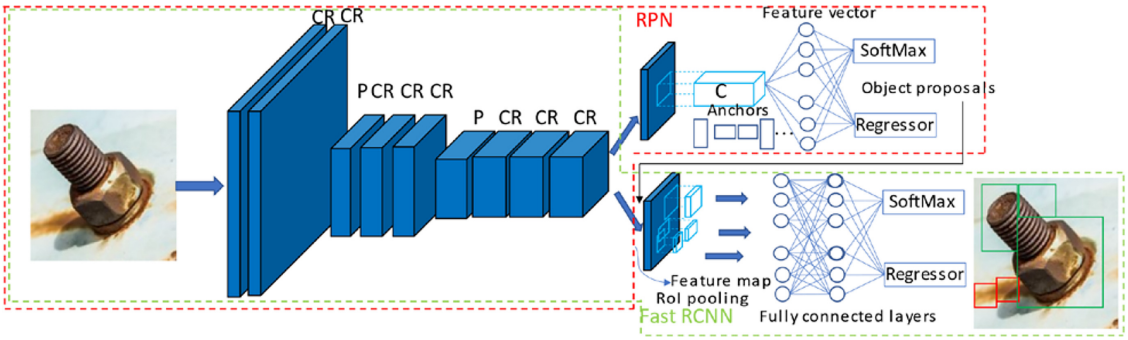



Figure 2.3: Faster R-CNN architecture [56]

### 2.2.3 Mask R-CNN

An expanded form of the faster R-CNN, the Mask R-CNN [41], is produced by including an additional branch, as illustrated in Figure 2.4. In parallel with the current branch for bounding box regression, this additional branch makes predictions about the object mask. The Mask R-CNN has a third branch called a fully convolutional network (FCN) that gives the object mask, whereas the faster R-CNN just produces two outputs: a class label and a bounding box offset. Pixel-to-pixel alignment, which was absent from the faster R-CNN, is introduced by the Mask R-CNN. The features and ROI are not aligned because Faster R-CNN harvests features with coarse spatial quantization. When compared to classification tasks, this misalignment has a major impact on pixel-to-pixel mask predictions. The Mask R-CNN suggests the RoIAlign layer, which is quantization-free, as a solution to this problem [59]. Bilinear interpolation is used in place of crude quantization to achieve ROIAlign.

The introduction of Mask R-CNN marked a significant breakthrough in the field of computer vision, as it enabled precise object detection at the pixel level by generating segmentation masks for individual objects.

### 2.2.4 Mask Scoring R-CNN

The Mask Scoring R-CNN (MS R-CNN) [42] is an enhanced variant of the Mask R-CNN framework that addresses a fundamental inconsistency between mask quality and classification confidence. As illustrated in Figure X, the standard Mask R-CNN assigns the same classification score to both the predicted class and the corresponding segmentation mask. However, this approach assumes a direct correlation between classification confidence and mask quality, which often does not hold in practice. Poor-quality masks may receive high scores, negatively impacting

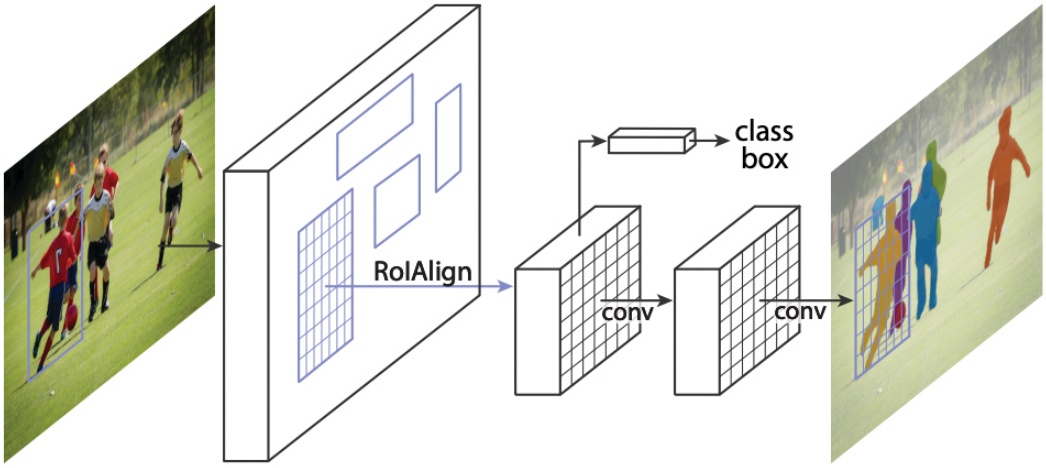



Figure 2.4: Architecture of Mask R-CNN for instance segmentation [41]

the overall performance of instance segmentation metrics such as Average Precision (AP).

To overcome this limitation, MS R-CNN introduces an additional branch—termed the MaskIoU head—which is designed to predict the Intersection over Union (IoU) between the predicted mask and the ground truth. This component operates in parallel with the original mask head and is trained to estimate the quality of the predicted mask independently of the classification confidence. The final mask score is computed as the product of the classification score and the predicted MaskIoU, providing a more reliable measure of mask accuracy.

This decoupling of classification and mask quality assessment leads to more accurate ranking of instance predictions, especially under strict IoU thresholds. By integrating the MaskIoU prediction branch, MS R-CNN effectively refines the scoring mechanism of Mask R-CNN, resulting in improved performance on standard instance segmentation benchmarks.

### 2.2.5 Cascade Mask R-CNN

Cascade Mask R-CNN [60] is an extension of the Mask R-CNN framework that addresses the limitations of single-stage object detection and segmentation pipelines, particularly in terms of localization accuracy and detection quality at higher Intersection over Union (IoU) thresholds. As illustrated in Figure 2.5, Cascade Mask R-CNN introduces a multi-stage refinement process by sequentially connecting several detection heads, each trained with an increasingly strict IoU threshold. This cascading structure enables the model to progressively improve the quality of bounding box regression and classification across stages.

Each stage in the cascade leverages the output of the previous stage as input, refining object proposals more precisely. The final mask prediction branch remains structurally similar to that of Mask R-CNN but benefits from the improved object localization provided by the cascade of detectors. Importantly, the use of multiple stages mitigates overfitting to lower-quality proposals and enhances robustness to localization errors.

To maintain alignment between features and regions of interest throughout the stages, Cascade Mask R-CNN also adopts the RoIAlign operation, as introduced in Mask R-CNN, ensuring accurate pixel-level mask prediction. Overall, Cascade Mask R-CNN achieves state-of-the-art performance on challenging benchmarks by combining the strengths of multi-stage detection refinement and high-quality instance segmentation.

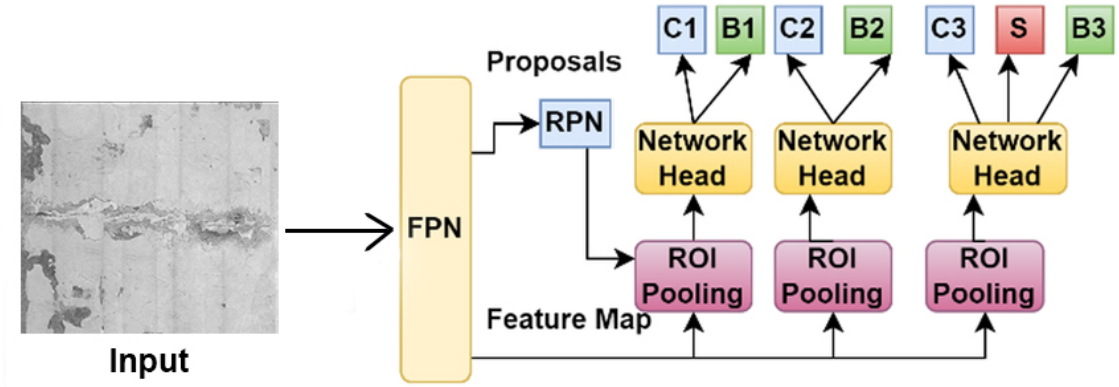



Figure 2.5: Architecture of Cascade Mask R-CNN. The "C" is classification, "B" is bounding box, and “S” denotes a segmentation branch.

### 2.2.6 Hybrid Task Cascade

Hybrid Task Cascade (HTC) [44] is an advanced instance segmentation framework that extends the Cascade R-CNN architecture by introducing a deeply integrated multi-task learning strategy. As illustrated in Figure 2.6, HTC enhances the traditional cascade design by not only performing progressive refinement of detection results across stages but also by interleaving mask prediction branches within each cascade stage. This interleaving enables simultaneous optimization of object detection and instance segmentation in a tightly coupled manner.

Unlike Cascade Mask R-CNN, where the segmentation task is treated as a parallel branch disconnected from the cascade structure, HTC introduces stage-wise mask information flow—each mask head receives inputs not only from the current detection features but also from the refined mask features of previous stages. This

design encourages rich feature reuse and enhances the quality of mask predictions by incorporating contextual and hierarchical cues across stages.

In addition to this architectural refinement, HTC incorporates a semantic segmentation branch that operates globally over the image, producing a semantic feature map that is fused with both bounding box and mask branches. This semantic context helps to guide both object localization and mask refinement, particularly in complex scenes where visual ambiguity exists.

By combining cascade regression, interleaved mask prediction, and global semantic segmentation, HTC achieves a higher level of cross-task synergy. This hybrid strategy leads to significant improvements in both detection and segmentation accuracy, particularly under challenging conditions where precise localization and mask quality are critical.

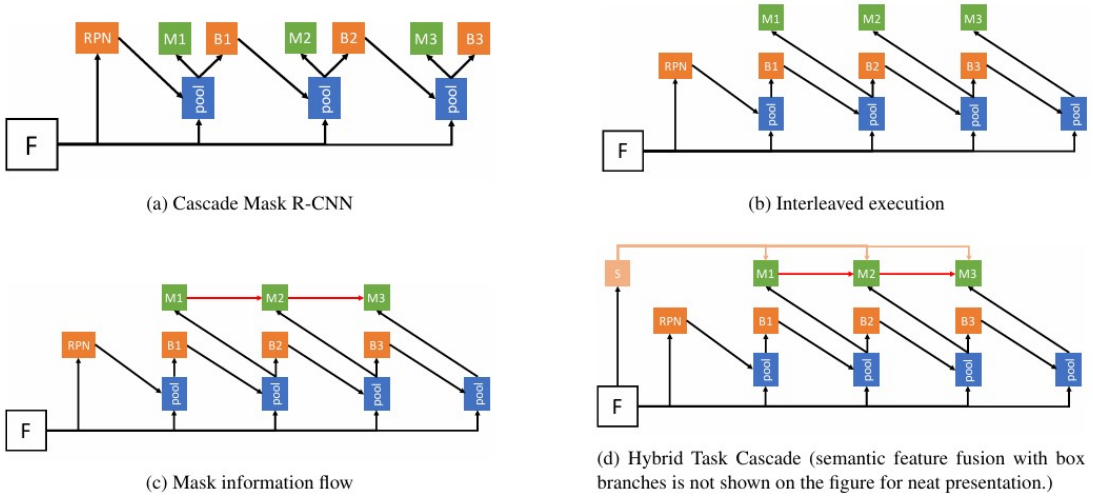



Figure 2.6: The architecture evolution from Cascade Mask R-CNN to Hybrid Task Cascade [44].

### 2.2.7 PointRend

PointRend (Point-based Rendering) [46] is an extension to conventional instance segmentation frameworks such as Mask R-CNN, designed to address the limitations of coarse, low-resolution mask predictions typically generated by standard fully convolutional networks (FCNs). As illustrated in Figure 2.7, traditional mask heads operate on fixed-resolution feature maps (e.g., 28×28), which leads to overly smooth and imprecise object boundaries, particularly for objects with fine-grained structures or high-frequency details.

To overcome this challenge, PointRend formulates mask prediction as a rendering problem, inspired by techniques in computer graphics. Rather than predicting

masks over a regular grid, PointRend iteratively selects and refines a set of points based on prediction uncertainty. Specifically, the algorithm begins by producing a coarse, low-resolution mask and then progressively samples points with high uncertainty—often located near object boundaries—for fine-grained refinement.

Each selected point is processed using point-wise feature extraction, combining coarse features from the backbone and fine-grained features from earlier convolutional layers. These features are then passed through a shared multilayer perceptron (MLP) to predict the final mask logits at each point. This adaptive and resolution-agnostic approach allows PointRend to produce high-quality masks with crisp and accurate boundaries.

By decoupling mask resolution from the fixed grid structure and dynamically focusing computational resources on ambiguous regions, PointRend significantly improves segmentation quality while maintaining computational efficiency. Its modular design allows seamless integration with existing detection frameworks like Mask R-CNN, enabling enhanced boundary accuracy without major architectural modifications.

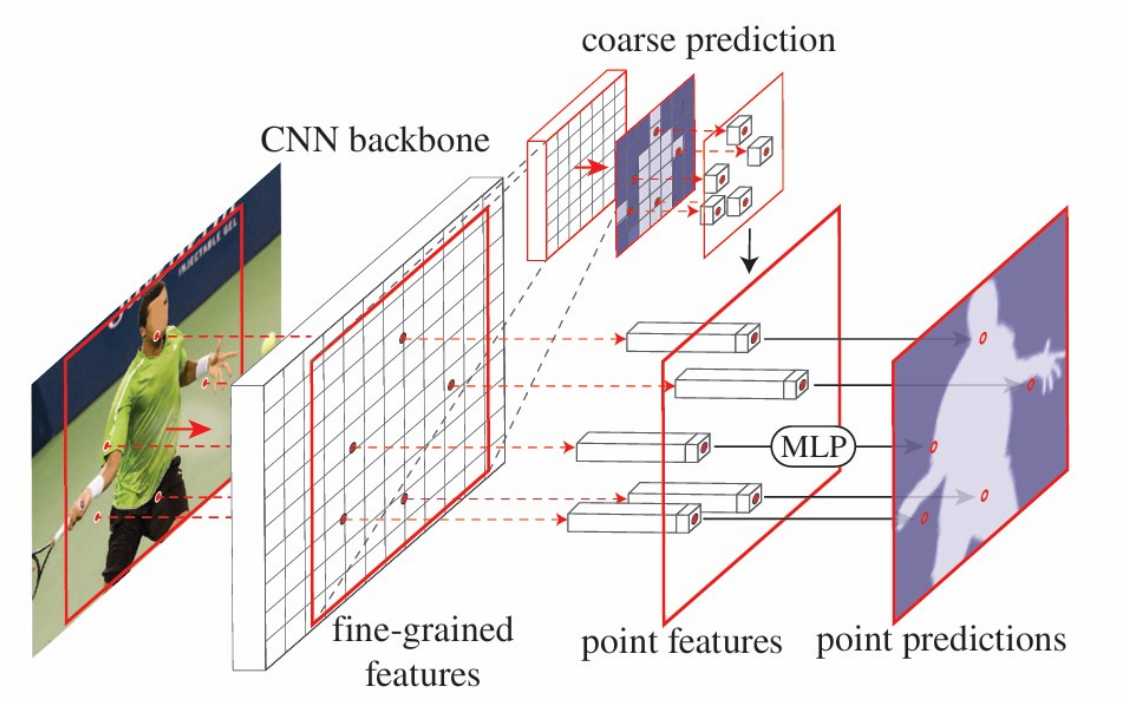



Figure 2.7: Architecture of PointRend [46]

### 2.2.8 SOLOv2

SOLOv2 (Segmenting Objects by Locations v2) [47] is a refined instance segmentation framework that builds upon the original SOLO architecture by further optimizing the process of object segmentation without relying on region proposals, bounding box regression, or post-processing steps such as non-maximum suppression. As illustrated in Figure 2.8, SOLOv2 adopts a fully convolutional, single-stage design that formulates instance segmentation as a category-aware, per-pixel classification problem over spatial locations.

In contrast to two-stage approaches like Mask R-CNN, which decouple detection and segmentation, SOLOv2 directly predicts instance masks by treating each grid cell as responsible for segmenting an object at a specific spatial location and scale. The feature map is dynamically partitioned into grids of different sizes across pyramid levels, enabling scale-aware prediction. For each cell that falls within an object's center region, SOLOv2 predicts both the object category and the corresponding segmentation mask using a mask kernel branch and a dynamic convolution mechanism.

A key innovation in SOLOv2 is the introduction of dynamic instance-aware kernels that are generated per instance and applied to a shared, high-resolution feature map. This design allows the model to produce high-quality, detailed instance masks with sharp boundaries and strong spatial consistency. Additionally, SOLOv2 introduces an improved alignment strategy and a simplified loss formulation, enhancing training stability and mask accuracy.

By eliminating the reliance on predefined anchors, region proposals, and post-processing heuristics, SOLOv2 achieves a more elegant and computationally efficient pipeline. It demonstrates competitive performance on standard instance segmentation benchmarks while offering significant speed advantages over traditional region-based methods.

### 2.2.9 CondInst

CondInst (Conditional Convolutions for Instance Segmentation) [48] is a proposal-free instance segmentation framework that departs from the traditional paradigm of using static, per-category mask heads. As illustrated in Figure 2.9, CondInst introduces a novel mechanism where instance-specific dynamic convolution kernels are generated on-the-fly, enabling the model to condition the mask prediction on each detected instance dynamically, without requiring a separate mask head for each category.

Built upon a standard object detection backbone such as FCOS, CondInst retains a fully convolutional structure and avoids region-wise operations like RoIAlign, which are typically used in methods like Mask R-CNN. The framework consists of two major components: first, a detection branch that predicts category labels and

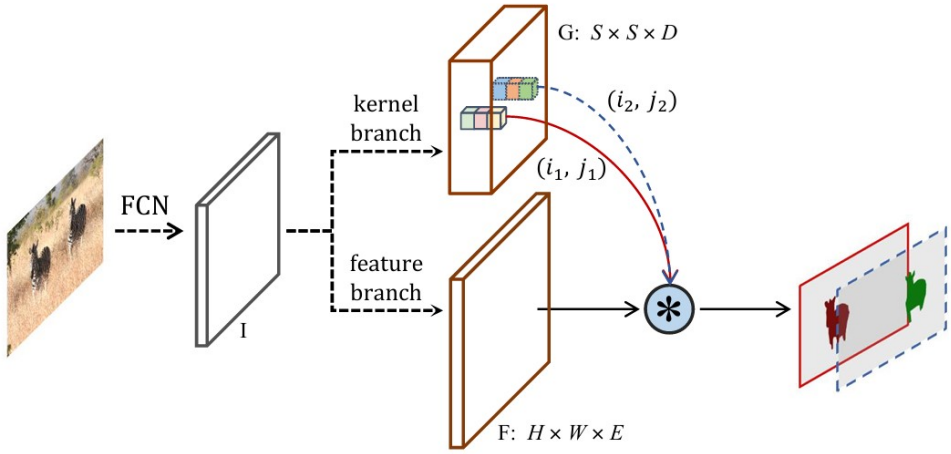



Figure 2.8: Architecture of SOLOv2 [47]

bounding boxes using dense anchors or anchor-free approaches, and secondly, a dynamic mask branch that outputs convolutional weights (referred to as conditional kernels) for each instance.

These dynamic kernels are applied to a shared, high-resolution feature map, using convolution operations to produce an instance-specific mask. This design allows the network to generate flexible and expressive masks tailored to the geometry of each object, while maintaining a compact and efficient architecture. Moreover, CondInst enables end-to-end training without explicit mask supervision at intermediate stages, reducing complexity compared to cascaded or multi-branch designs.

By combining dense object detection with conditional convolution-based mask generation, CondInst achieves a strong balance between accuracy and efficiency. Its fully convolutional nature and the elimination of hand-crafted post-processing steps like RoI pooling or proposal selection contribute to its superior scalability and simplicity, making it well-suited for real-time and large-scale applications.

### 2.2.10 Transformer

Multi-head self-attention (MHA), the most important part of the transformer network, is an attention mechanism used by the feed-forward neural network-based model known as the transformer [45]. When creating the output feature map, the network should concentrate on the pertinent area of the input feature map, which is determined by this attention mechanism. As shown in Figure 2.10, the transformer's encoder-decoder architecture uses additional methods in addition to self-attention, such as positional encoding and layer normalization.

In short, attention functions as a crucial operator inside the transformer model,

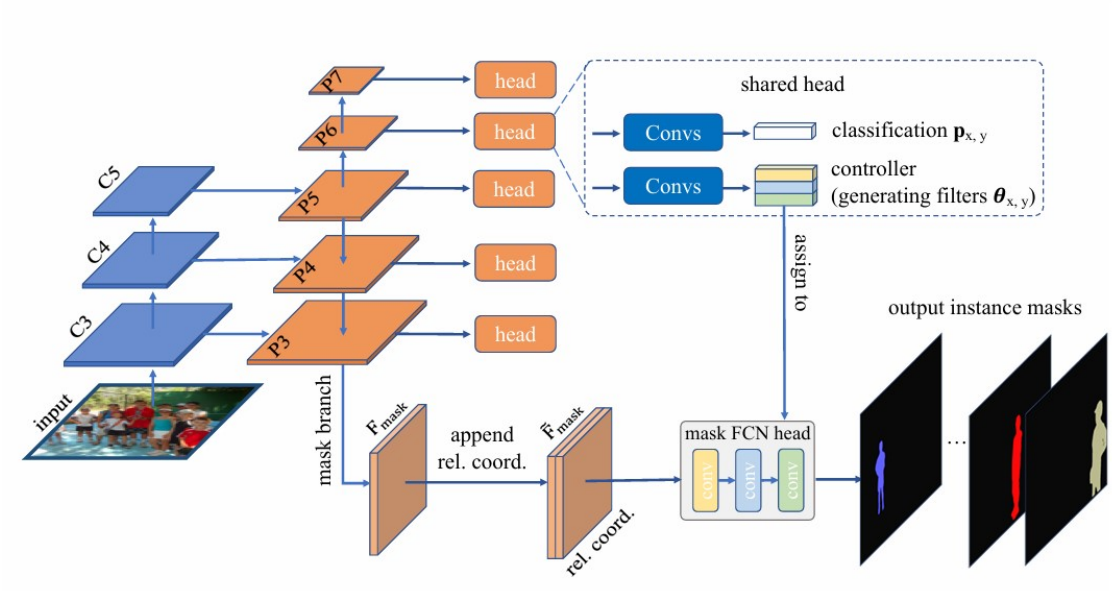



Figure 2.9: The overall architecture of CondInst [48]

which is the fundamental model. The transformer is used in many different fields, such as sound signals, pictures, and natural language processing (NLP). An input is transformed into two sequences by the transformer network upon receipt: a sequence of positional encodings and a sequence of vector embeddings. The inputs and outputs are essentially converted into dense vectors first. Since the model lacks an RNN that can remember how it learned the input sequences, it is essential to assign each word in a sentence a specific position in a sequence [61].

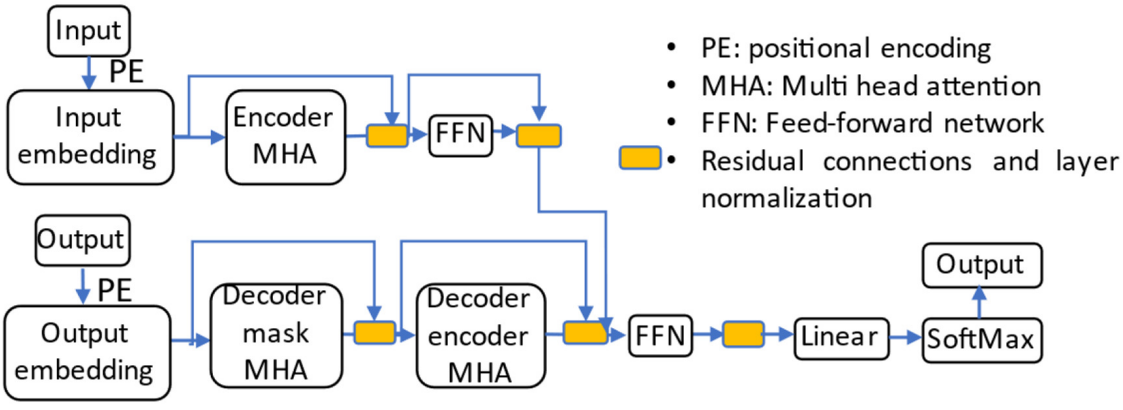



Figure 2.10: Architecture of a typical transformer [56]

### 2.2.11 Vision Transformer

The Vision Transformer (ViT) is a pioneering deep learning architecture that adapts the transformer model—originally developed for natural language processing (NLP)—to the domain of computer vision [45]. Unlike conventional convolutional neural networks (CNNs), which use local receptive fields to capture spatial hierarchies, ViT relies solely on the multi-head self-attention (MHA) mechanism to model global context across the entire image. As illustrated in Figure 2.11, this approach enables ViT to capture long-range dependencies between image regions from the earliest layers of the network.

To process image data in a transformer framework, ViT first divides the input image into a sequence of fixed-size non-overlapping patches (e.g., 16×16 pixels), which are then flattened and linearly projected into embedding vectors. These patch embeddings are analogous to word embeddings in NLP and serve as the input tokens for the transformer encoder. To retain spatial information—otherwise lost due to the permutation-invariant nature of self-attention—ViT adds learnable positional encodings to each patch embedding.

The core of ViT consists of a series of transformer encoder blocks, each composed of MHA and feed-forward layers, along with residual connections and layer normalization. Unlike CNNs, which inherently impose an inductive bias of locality and translation equivariance, ViT learns spatial relationships through self-attention, allowing greater flexibility in modeling complex visual patterns but requiring significantly more data to train effectively.

ViT has demonstrated competitive and often superior performance on large-scale vision benchmarks, especially when trained on massive datasets. Its architecture reflects a shift toward general-purpose, attention-based models in computer vision, capable of unifying modeling approaches across modalities such as text, audio, and images.

### 2.2.12 SWIN Transformer

The Swin Transformer (Shifted Window Transformer) is a hierarchical vision transformer architecture that introduces locality and scalability into the transformer design, addressing key limitations of the original Vision Transformer (ViT) [49]. While ViT treats an image as a flat sequence of patches and models global dependencies from the beginning, the Swin Transformer incorporates a shifted window-based self-attention mechanism that enables both local feature modeling and efficient computation, making it more suitable for dense prediction tasks such as object detection and semantic segmentation.

As shown in Figure 2.12, the Swin Transformer first partitions the input image into non-overlapping patches, which are projected into patch embeddings. These

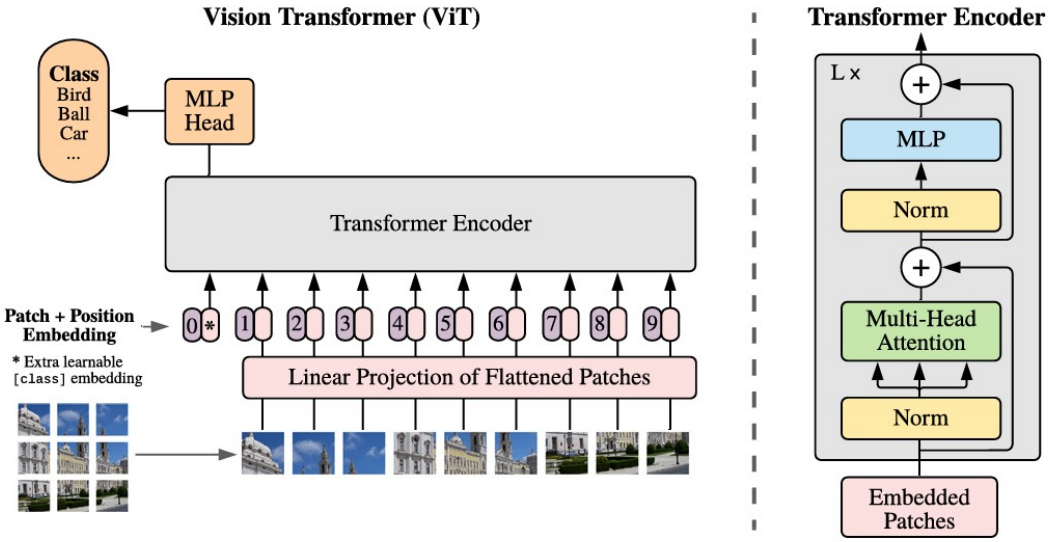



Figure 2.11: Vision Transformer model overview [45]

embeddings are then processed through multiple stages, forming a hierarchical representation that is structurally similar to CNN-based backbones. Within each stage, self-attention is applied within local windows—fixed-size subregions of the feature map—rather than across the entire image. This window-based attention significantly reduces computational complexity, scaling linearly with image size as opposed to quadratically.

To allow cross-window connections and enhance information flow between local regions, the Swin Transformer introduces a shifted window scheme in alternating layers. This design enables the model to capture long-range dependencies while maintaining computational efficiency. Each stage doubles the number of channels and halves the spatial resolution, creating a multi-scale feature hierarchy akin to traditional convolutional architectures.

The Swin Transformer also integrates patch merging layers for downsampling and layer normalization throughout the network. These architectural choices make it highly adaptable as a unified backbone for a variety of computer vision tasks, including classification, detection, and segmentation. Unlike ViT, which requires extensive pretraining on large datasets, the Swin Transformer demonstrates strong performance even with limited data, due to its built-in inductive biases and hierarchical structure.

Overall, the Swin Transformer bridges the gap between transformers and CNNs by combining the flexibility of attention mechanisms with the efficiency and inductive strength of convolutional hierarchies.

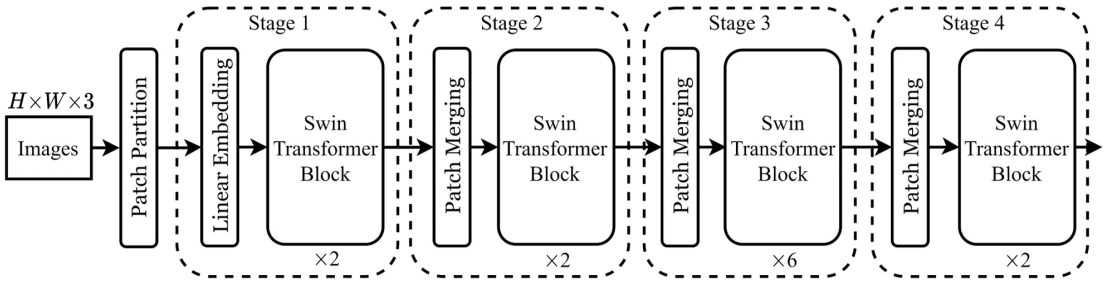



Figure 2.12: overview of Swin Transformer architecture [49]

### 2.2.13 QueryInst

QueryInst (Query-Based Instance Segmentation) is a transformer-inspired instance segmentation framework that unifies object detection and mask prediction through a query-based paradigm, eliminating the need for region-wise operations such as RoIAlign. Introduced as an extension to the dynamic instance segmentation family, QueryInst integrates the strengths of conditional convolution and query-driven modeling to improve mask quality and inference efficiency [50].

As illustrated in Figure 2.13, QueryInst builds upon a standard object detector such as FCOS, where object proposals are generated in a dense, anchor-free manner. For each proposal, a learnable query feature is dynamically generated and used to guide the instance segmentation process. These query features serve as tokens in a transformer-style decoder, which attends to multi-scale features extracted from the backbone using multi-head attention, enabling each query to capture rich semantic context and instance-level details.

A key innovation in QueryInst is its use of iterative query refinement. Rather than predicting masks in a single forward pass, the model refines both detection and segmentation predictions across multiple stages. At each stage, the query feature is updated based on attended features from the previous step, allowing for progressive enhancement of object localization and mask accuracy. This mechanism is conceptually analogous to the cascade refinement used in Cascade R-CNN, but implemented through attention-guided updates instead of static feature pooling.

The mask head in QueryInst adopts a dynamic convolutional approach, similar to CondInst, where instance-specific kernels are predicted and applied to a shared, high-resolution feature map. This allows the framework to generate high-quality segmentation masks without requiring a fixed mask head for each object.

By combining query-based reasoning, transformer-style attention, and dynamic mask generation, QueryInst achieves a powerful balance between flexibility, accuracy, and computational efficiency. Its architecture demonstrates the potential of fully end-to-end, proposal-free instance segmentation pipelines and represents a shift toward unified, attention-driven models in vision tasks.

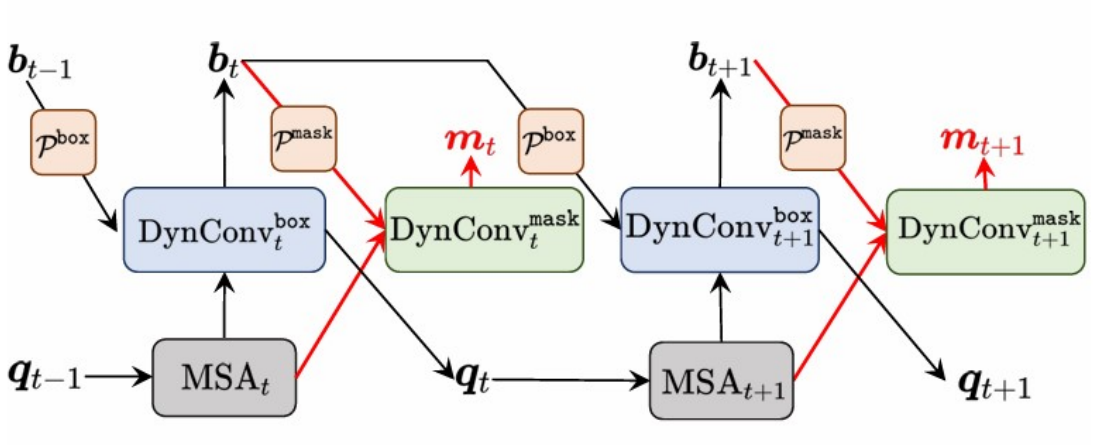



Figure 2.13: Architecture of QueryInst. The red arrows indicate mask branches [50].

### 2.2.14 Mask2Former

Mask2Former (Mask Transformer for Universal Segmentation) is a unified framework that re-casts instance, semantic, and panoptic segmentation as a common mask-classification problem, bridging the gap between region-based and dense-prediction paradigms [51]. Inspired by DETR-style [62] transformers yet tailored for dense tasks, Mask2Former couples a multi-scale pixel decoder with a transformer decoder to achieve high-resolution detail and long-range reasoning within a single architecture.

As sketched in Figure 2.14, the pixel decoder first ingests a hierarchical CNN—or Swin-Transformer—backbone and produces scale-aware feature maps using multi-scale deformable attention. These refined feature maps preserve fine spatial cues while remaining computationally tractable. The transformer decoder then operates on a fixed set of learned mask queries; through multi-head cross-attention, each query attends to the pixel-decoder features and iteratively generates two outputs: a class prediction and a binary mask embedding. The final segmentation map is obtained by linearly projecting each mask embedding onto the high-resolution feature grid, followed by a softmax-based bipartite matching loss that enforces a one-to-one assignment between ground-truth masks and queries.

Crucially, Mask2Former's design dispenses with hand-engineered post-processing (e.g., RoIAlign, NMS) and task-specific heads. A single set of parameters, trained end-to-end, seamlessly adapts to disparate segmentation tasks by merely changing the label space—"thing" classes for instance segmentation, "stuff" classes for semantic segmentation, or both for panoptic segmentation. Empirically, this task-agnostic formulation achieves state-of-the-art accuracy across COCO [63], ADE20K [64], and Cityscapes [65], while its modular pixel-and-query decoders maintain efficiency suitable for large-scale vision applications.

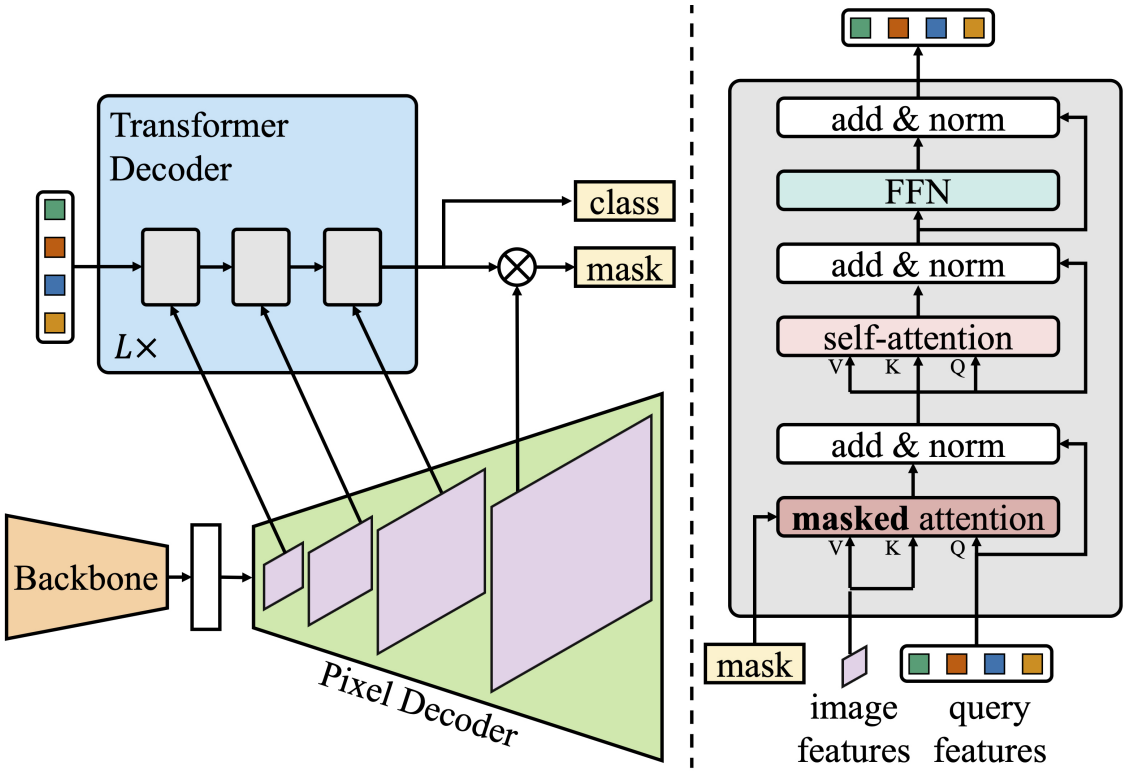



Figure 2.14: Overview of Mask2Former [51]

# Chapter 3

# Structural Health Monitoring

## 3.1 Overview and background

The broad topic of structural health monitoring (SHM) of civil infrastructures is concerned with identifying structural flaws, tracking structural conditions, and evaluating a structure's safety using long-term data from different kinds of sensors incorporated into the structural systems or during periodic inspection for monitoring of the assets. Through structural repair and disaster management, SHM is a crucial procedure for maintaining, identifying anomalies, and enhancing the serviceability of civil infrastructures. Continuous use causes civil infrastructures to gradually degrade and lose their intended function. The significance of early and routine maintenance is increased by this inevitable process [56].
Through vibration testing and ongoing measurements of the audio frequency dynamic modulus and damping of specimens subjected to tensile loading, researchers found that structural damage frequently appears as changes in various structural properties, such as strength (stiffness) and damping, in the early stages of SHM [66]. Dynamic modal features, such as natural frequencies, mode shapes, and damping, are also affected by these structural changes. Various global and local damage identification techniques have been developed in order to recognize these changes and evaluate damage.
Since these changes have significant effects on the dynamic vibration characteristics of the structure, global techniques concentrate on identifying changes in the monitored system's modal properties. Local approaches, on the other hand, include localized nondestructive testing (NDT) techniques and visual inspections. To further determine the extent of localized harm, local approaches typically rely on data gathered from global approaches [56].
Generally, global methods, due to their heavy reliance on vibration measurements, are frequently referred to as vibration-based approaches. There are three subfields of vibration-based approaches: hybrid approaches, physics model-based approaches, and data-driven approaches. The majority of these methods use established finite

element models (FEM) or observed vibrations to determine the modal characteristics in both the damaged and intact (baseline) structures. For example, some research [67] extracted natural frequencies from measured acceleration time series, while others [68] employed mode shapes for structural damage identification.
In order to extract damage-sensitive features from observed vibrations, a variety of techniques have since been put forth in vibration measurement-based damage detection systems. These damage-sensitive characteristics include spatiotemporal characteristics within the vibration time series for a variety of applications, such as buildings, rotating mechanical systems, offshore structures, civil bridges, and tunnels, in addition to changes in modal properties and their subsequent processing [56].
Therefore, the traditional damage identification process has three main stages: measurement, feature extraction, and classification using various machine learning (ML) and classification methods. Detailed investigations of traditional ML and classification methods have also been conducted to classify the extracted features, transitioning from an intact state to a damaged state. Number of pioneering and advanced application can be represent as artificial neural networks (ANN) [69], Bayesian probabilistic approaches [70], fuzzy logics [71], simple genetic algorithm (GA) [72], multiobjectgive GA [73], and support vector machines (SVM) [74].
However, in real-world settings, where a variety of disturbances and uncertainties, such as temperature variations, may exist, damage diagnosis based on modal properties and damage-sensitive features retrieved from different signal processing approaches may not be effective. Although a number of studies have tried to use algorithms such as curve fitting methods to solve these issues, these methods are not very effective in detecting minor but significant damage. Because of this, these conventional techniques often only perform effectively for straightforward structures or idealized numerical Finite Element Methods (FEM) with low measurement uncertainties and errors [75].
Comparatively, computer vision-based approaches have gained significant interest since they provide explicit, clear visual proof of damage within images [76]. Various image processing techniques have been utilized for crack detection in red, green, and blue (RGB) images [77]. Moreover, different image processing techniques have been employed for the extraction of damage-sensitive features related to different types of structural damage in RGB images [78, 79].
Moreover, computer vision-based approaches have been developed for measuring vibrations [80, 81] and strains [82]. Using these methods, it is possible to measure strain and reliable structural responses, such as displacements and accelerations, without the need for a physical reference point. Vibration-based damage detection uses the measured responses as input.
In spite of this, damage-sensitive features cannot be automatically extracted or

formulated, even when using computer vision approaches. Furthermore, these approaches often detect only one type of damage, and their performance is heavily influenced by lighting conditions. Traditional image processing techniques can struggle to extract damage-sensitive features from blurry, shadowed, or unevenly lit images, which is made worse by the limited classification abilities of traditional machine learning techniques [56].
Consequently, for over a quarter of a century, the field of SHM has struggled to formulate and extract damage-sensitive features that are robust to changes in ambient temperature, noise, and lighting conditions. In addition, there is a growing demand for more efficient and robust machine learning algorithms capable of classifying damage-sensitive features accurately.

## 3.2 Deep Learning Based Structural Health Monitoring

In order to overcome the limitations and difficulties inherent in manually identifying damage-sensitive features and classifying them with traditional machine learning methods, in 2017, Cha et al. [55] proposed a deep learning (DL)-based approach for damage detection utilizing a deep Convolutional Neural Network (CNN). The deep learning method involves the extraction of features from raw input data and passing them along to deeper modules as higher-level representations through a combination of simple yet nonlinear modules. A variety of applications of deep learning have been demonstrated, including speech recognition, object detection, genomics, and many more.
Because of its nature, DL can automatically extract robust multilevel damage-sensitive features from raw input images by training on large labeled datasets. The CNN method in Cha et al. research [55] was developed to detect concrete cracks using a defined size of a sliding window to localize the detected cracks in RGB images. The results were almost satisfactory, with a detection accuracy of 97%. Despite issues in RGB image conditions, such as blurriness, spot-lighting, shadows, etc., the trained CNN still detected cracks, showing its ability to address different environmental and sensing uncertainties and noise.

## 3.3 Computer Vision-based Approaches for Surface Damage Detection

After the revolutionary paper of Cha et al. [55] that represented the feasibility of automatically extracting damage-sensitive features from RGB image inputs using

CNN methods, numerous subsequent studies focused on damage detection using DL algorithms. The accelerated progression of computer vision and DL, along with improvements in camera resolution, computational capabilities, and SHM, contributed to the rapid adoption of vision-based damage detection. Contactless cameras can be easily integrated into UAVs, Unmanned Ground Vehicles (UGVs), or vehicle-mounted systems for data collection. Generally, the development of computer vision-based damage detection using DL algorithms involves three stages. The first is image classification-based damage identification, the second is bounding box level object detection-based damage identification, and the third is pixel-wise segmentation. These stages are discussed in the following Sections respectively. Each approach has been extensively reviewed in the context of various infrastructure types, including bridges, buildings, dams, pavements, tunnels, and sewer systems.

### 3.3.1 Damage Classification

In this context, classification refers to how the DL model categorizes an input image as damaged or intact, or how it identifies certain types of damage. As a result, the entire input image is grouped into one of the desired categories as the output of the DL process.

As an example, as previously mentioned, Cha et al. [55] developed a CNN composed of four convolution layers, two pooling layers, a dropout layer with ReLU activation function, a fully connected layer, and SoftMax to classify intact and crack images. The CNN employed a sliding window concept to screen large input images and localize the damage. The sliding window size in this study was $256 \times 256 \times 3$ pixel resolution. As a result, a total of 40,000 images, with the size of the sliding window, were prepared from 277 images of dimensions $4928 \times 3264 \times 3$ and used for training and validation. The well-trained CNN was then tested on another set of 55 images, with dimensions $5888 \times 3584 \times 3$, achieving an impressive accuracy of 97%.

Furthermore, the CNN's performance was compared with traditional edge detection-based crack detection methods, such as Sobel and Canny edge detection. The CNN consistently demonstrated superior performance, significantly outperforming the traditional methods, even under challenging conditions like shadows, blurriness, and strong spot lighting. Notably, the designed CNN was integrated with autonomous UAVs for crack detection in GPS-denied areas, effectively simulating scenarios beneath a bridge deck or indoors [83].

The concept of image classification in DL has been employed to different civil infrastructure applications, including buildings [84], sewer systems [85], and pavements [86], to detect various types of damages, such as cracks in concrete members, obstacles, joint openings, faults, debris, and silty conditions in sewers, and tile deterioration in buildings.

For instance, Li et al. [87] proposed a ResNet-18 model with a hierarchical SoftMax approach for defect detection in sewer lines, concerning imbalanced data,

which reduced the network's overall performance. The hierarchical method oversaw the learning process at various levels throughout training, with the upper-level task aimed at distinguishing between normal images and those with defects, while the lower-level task assessed the likelihood of defects in the image. The ultimate classification outcomes were established by applying the chain rule of conditional probability.
Hassan et al. [85] developed an AlexNet [88] model for defect classification in CCTV videos obtained from underground CCTV models. Data augmentation techniques, including transformation, flipping, rotation, translation, and deformation, were utilized. The original AlexNet model, initially used for the classification of 1000 natural objects in the ImageNet dataset [53], was modified and fine-tuned with transfer learning specifically for sewer defect detection.
There is also an initiative to enhance the performance of CNNs by utilizing probabilistic methods. For instance, Chen et al. [89] developed the NB-CNN, which uses a CNN and a Naive Bayes data fusion scheme to improve the performance of crack classification. Adam et al. [90] utilized a combined method for precise crack detection in RGB images of concrete structures. They combined a CNN with an SVM classifier and proposed a noise reduction technique to minimize classifier issues.
It is important to note that civil infrastructures are typically located in complex background scenes (CBSs), so the detection of damages can be challenging purely based on the CBSs. Among these studies, no method considered CBSs in their training or testing datasets. Therefore, bounding box level object detection methods were adopted to detect damages more efficiently.

### 3.3.2 Bounding Box Level Damage Detection

Bounding box level object detection methods of DL can resolve the limitations of the fixed sizes of sliding window techniques in damage detection and localization problems. For example, the faster R-CNN [39] provides flexible sizes of bounding boxes to localize the detected damages in input images.
[58] introduced various damage datasets and trained the faster R-CNN with four separate steps. The trained faster R-CNN architecture achieved 89.7% accuracy to detect structural damage in bridges by considering CBS. In addition to vehicle load and extreme weather conditions, bridge structures are subjected to physical changes that can be represented as damage, such as cracks, corrosion, loosening bolts, settlement, deflections, excessive vibration, internal defects, spalling, and delamination.
Numerous studies utilizing bounding box level DL networks using R-CNN [**?**], faster R-CNN [38], single-shot multi-box detector (SSD) [91], DINN [92], and different versions of YOLO series. Studies in this field employed bounding box level damage detection for different types of defects, including delamination and peeled paint [58], ceiling damage [93], steel cracks, steel corrosion, and loosened bolts [94].

Furthermore, Yeum et al. [95] utilized AlexNet for the detection and localization of welded connections of steel bridges using images collected from a UAV. Zhang et al. [96] introduced a novel approach for detecting multiple types of defects in concrete highway bridges by using YOLOv3. Li et al. [91] employed transfer learning from a convolution-based autoencoder to SSD for the training of images of buildings after Hurricane Sandy to detect the damages. The transfer learning approach improves approximately 10% damage detection performance.
In another study, Cheng et al. [97] developed using faster R-CNN for defect detection in sewer lines, utilizing 3000 images from CCTV videos for model training. The study achieved concrete results and suggested that increasing the dataset size, adjusting filter dimensions, and adding convolution layers can improve and enhance the performance.
Overall, damage detection at the bounding box level shows a better performance in terms of localization compared to image classification-based approaches, and it is comparatively less expensive to establish data and hardware compared to pixel-wise segmentation approaches, which will be discussed in the following section. Although this approach is highly used in SHM, it is still insufficient for performing damage analysis, which is the final step of SHM's reliability assessment.

### 3.3.3 Pixel Level Damage Segmentation

Several methods for segmenting detected damages at the pixel level have been developed to quantify and compare the damages. These approaches employed DL techniques to identify defects and damages on a pixel-by-pixel basis. This approach is more precise than just identifying approximate bounding box positions of the damage. The main benefit of pixel-level damage segmentation is its ability to offer a more detailed method for determining a defect's shape, size, and location. This enables researchers to better determine and measure the degree of damage.
The majority of segmentation networks use the encoder-decoder design. This structure aids in extracting damage features and restoring the original spatial dimensions of the input image. In order to do this, it marks the object pixel by pixel, displaying the damage that has been detected pixel by pixel. In this area, numerous thorough research projects have been carried out.
Most of the research in this area has focused on modifying open-source networks that were initially created for purposes unrelated to structural damage segmentation. For example, the underlying architecture for these techniques has been chosen from networks like SegNet [98], UNet [99], FCN [100], Mask R-CNN [41], DeepLapV3+ [101], and PANet [102], which are frequently paired with faster R-CNN [39], ResNet series [35], DenseNet series [103], and VGG series [33]. Dong et al. [104] used the SegNet technique to propose a pixel-wise segmentation network for crack and spalling as an example. In order to detect cracks and spalling problems, the researchers in this study incorporated the focal loss (FL) function into

the FL-SegNet model.
In one successful study, PANet was combined with the A* algorithm for crack width and length calculation [105]. The proposed approach was compared to UNet, Mask R-CNN, and DeepCrack [106], achieving an mIoU of 50.28%, outperforming Mask R-CNN by 2%, DeepCrack by 19%, and UNet by 23%. In another study, zhao et al. [107] employed Mask R-CNN for tunnel crack instance segmentation. Liu et al. [108] developed an architecture based on UNet for the crack segmentation by integrating VGG19, InceptionRexNetv2, and ENetb3. In another study, Ji et al. [109] also leveraged DeepLapV3+ for crack segmentation with the use of 3D reconstruction of point cloud data. Xi et al. [110] in a similar approach, created YDRSNet by integrating DeepLabV3+ and YOLOv5 to solve the problem of real-time gear-fitting measurement.
Although most of these studies mainly focus on crack detection, DL techniques have been utilized to identify other defects, including concrete spalling, seepage, and internal damages of infrastructures. Beckman et al. [111] introduced a DL network for the detection of spalling of concrete using depth camera data for volumetric damage quantification utilizing faster R-CNN, and achieved an average precision of over 90%.
As a result, numerous studies have been conducted on SHM for different purposes. These goals include improving network performance by focusing on enhancing metrics such as mean intersection over union (mIoU), F-score, recall, and precision for the semantic segmentation task, and Average Precision (AP) and Average Recall (AR) for instance segmentation tasks. Existing network-based and hybrid segmentation methods have been extensively researched for damage detection tasks.
DL approaches have been applied to segment concrete and pavement cracks in SHM, but some limitations have been noted. The majority of these methods are based on existing networks developed for different purposes, resulting in many learnable parameters that may make them less suitable for real-time image processing. There are several factors that can be affected by the size of input images, including the computational resources required for image processing and the accuracy of the methods [112].
Increasing the input size may result in more detailed representations of the structure and, therefore, increased accuracy, but it may also increase the computational resources required to process the images and the size of the images. Moreover, the number of data samples and the separation of training, validation, and testing data can also influence the results and generalizability of the methods. Instead of focusing just on clean concrete and pavement surfaces, it is crucial to take into account the CBS that are frequently seen while creating computer vision-based damage segmentation techniques for real-world SHM applications. This is required to guarantee more reliable damage detection techniques that function in a variety of visual contexts.

Table 3.1 represents a brief comparison between three computer vision-based approaches for surface damage detection discussed in the sections 3.3.1, 3.3.2, and 3.3.3.

Table 3.1: CV-based method comparative analysis.

| **Network** | **Advantage** | **Limitation** |
|---|---|---|
| Image classification with less computational cost, and convenient to establish a ground truth dataset | Difficulties in damage localization and quantification | |
| Bounding box level detection | Better damage localization and quantification than image classification approaches, and convenient to establish a ground truth dataset | Still less effective than pixelwise segmentation methods in damage localization and quantification |
| Pixelwise segmentation | High accuracies in damage location and quantification; through an object-specific design of networks, real-time, efficient processing is possible (e.g., SDDNet, STRNet) | More computational cost if existing heavy segmentation networks are used, and tedious labeling of ground truth data |

# Chapter 4

# Methodology

## 4.1 Methodology

In this chapter, the methodology used for completing the research will be discussed. The chapter consists of five parts, including 4.1.1-*Data Acquisition Method*, which discusses the approach for collecting the data from Italian tunnels and discusses detailed information about the equipment used during the data collection. Next section, 4.1.2-*Image Processing*, discusses image processing methods, which describe how original raw images are processed to first overcome the lightning issues and eventually become prepared for the labeling phase. 4.1.3-*The Labeling and Creation of the Dataset* introduced the defects and non-defects part investigated in this research and proposed the methodology on how the images are labeled, and eventually getting how to create the dataset. The first part, 4.1.4-*Neural Network Training*, discussed and introduced the deep learning models discussed in this research and gave a comparison between these different approaches. Lastly, the 4.1.5-*Evaluation Methods* describe the metrics used in this thesis to address the efficacy of the dataset by comparing this with the standard metrics present in this field of study.

### 4.1.1 Data Acquisition Method

#### Overview

For collecting data in this research, the Laser Scanner technique, combined with thermography, is applied as a non-destructive investigation methodology with the following objectives:
1. Precision survey of the geometry of the tunnel
2. Survey of the installations and all the protection devices (wave panels, wire mesh, etc.)
3. Survey of the surface crack pattern Survey of cavities and surface wear

4. Mapping of wet areas and water infiltrations Survey of the deformation pattern The Laser Scanner is a very useful technology for surveying complex and large works, such as a tunnel. Compared to traditional topographic surveying, the competitive advantage of a laser survey is the ability to obtain both geometric and photographic information that is continuous, complete, and metrically rigorous by nature of the object. The large amount of data acquired in a short time allows for the geometric survey of the structure and its features with a remarkable level of detail and completeness. It is also a direct measurement system as it allows for measurements correlated to an instrumental accuracy certified by a calibration certificate, which officially documents the measurement results.

The speed of acquisition, the rapidity in the processing and utilization phase of geometric data, also testify to the flexibility of this surveying technique aimed at the three-dimensional modeling of infrastructures. This also allows for the scheduling of a systematic survey to be repeated periodically on tunnels in order to carry out an important precision comparison between the geometries measured at different times. The possibility of acquiring topographically referenced three-dimensional shapes allows for instrumental monitoring to assess the structure's response over time, keeping any ongoing evolutionary phenomena under control.

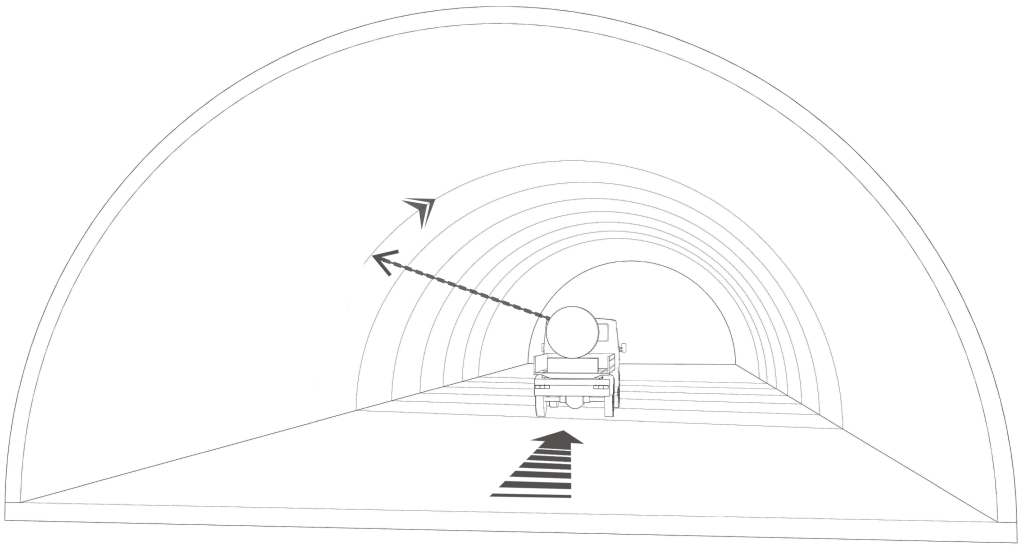


Figure 4.1: Schematic of data acquisition equipment

### Investigation Methodology

The laser scanner is an electro-optical mechanical device that, through the technique of successive scans, allows for the automatic detection of an object in its three dimensions. It uses a structured light beam that does not damage or alter the material consistency of the measured surfaces.

In detail, the tool used, by generating a pair of sinusoidal laser pulses, detects the distance through specific algorithms that measure the measurement via the phase difference between the emitted wave and the received wave. These laser scanners are particularly fast and have a very dense point grid. However, their possibility of phase shift limits the maximum range to 15 meters, which is still sufficient for surveys in tunnels, even with 3 lanes. At the same time, each point is associated with a reflectance value, which depends on the characteristics of the detected materials and is identified through a color parameter called RGB.

Being a light signal, the laser strikes the surface to be detected at the angle dictated by the instrument's point of view, which is why a complete description of the object generally requires multiple scans, which are then unified through a network of appropriately positioned targets, to each of which coordinates referring to the chosen system are assigned through a classic precision topographic survey. However, since laser scanners can also be installed on motorized vehicles, it is also possible to ensure continuity in data acquisition, especially for the shape of tunnels, and to measure a significant number of points on the surface of the object in a relatively short time.

The product of a laser scanner scan is a point cloud of coordinates x, y, z, which can be viewed directly on a computer monitor as a "three-dimensional photograph" made up of millions of points that detail the surface of the detected object, from which dimensional and colorimetric information can be derived.
The laser scanner used in the survey campaign is also equipped with a thermal camera and is therefore able to detect the intensity of radiation emitted by objects in the infrared spectrum and convert it into a temperature. Since the characteristics of the thermal camera are influenced by the type of sensor used, an integrated cooling system and environmental temperature measurement are essential for making the necessary correlations.

In detail, the operating principle of a thermal camera is as follows: the infrared energy emitted by an object is focused by optical components onto an infrared detector, which then sends the information to the electronic sensor for image processing, which can be immediately displayed on an LCD or monitor. Thermography thus transforms an infrared image into a radiometric image on which temperature values can be read. Each pixel of the radiometric image is a temperature measurement obtained through complex algorithms present in the thermal camera. The temperature maps of the exposed surfaces obtained from this type of investigation are highly useful during inspections to identify any material inconsistencies

and areas of moisture concentration, as well as to monitor, through the comparison of measurements taken at subsequent times, the evolution of wetting/drying phenomena on the surfaces.

### Equipment Information and Executive Details

Specifically, the laser scanner used in this research has been specially developed by the German company 'Spacetec' [113] and optimized for surveys in railway and road tunnels. The instrument operates using phase-shift technology, emitting appropriately calibrated sinusoidal waves to ensure maximum surveying accuracy over distances typical of the curvature radii of 2/3 lane tunnels. The scanner head, with continuous rotation, is capable of reaching frequencies of 300 Hz and acquiring up to 10,000 points per section.

The scanning specifications used during the survey are listed below:

- 200 revolutions/second
- 5,000 points/section
- 5 km/h forward speed

For the survey, the laser scanner tool, consisting of a cylindrical device with a rotating head, was mounted on a vehicle in order to obtain a millimeter-level three-dimensional definition of individual scans in a single reference system quickly and efficiently.

The vehicle moves at a speed of about 5 km/h, and both the acquisition of the survey by the laser scanner, which occurs with 200 rotations per second of the instrument's head, and the acquisition of data from the thermal sensor are activated. The measurement operations are carried out until the survey is completed without the need to materialize reference points or identify homologous points among the various scans, thus allowing maximum freedom of movement of the vehicle, while the geo-referencing of the survey itself can take place at a later time, integrating the data returned by the laser scanner with the absolute coordinates collected on specific easily identifiable points of the gallery (e.g., lighting fixtures, signage, etc.).

The width of the instrument's field of view, which reaches almost 360°, allows for scanning the entire vault of the tunnel and also the highway lane, except for the shadow area created by the instrument's mounting plate, which in any case measures a few dozen centimeters.

### Methods of Captured Data Analysis

The output obtained from a laser scanner survey consists of geometric and photographic information that is much more complete and significant than a 'simple' photographic report, as it reproduces the topology of the work with extreme accuracy and in three dimensions.



Figure 4.2: Vehicle equipped with Spacetec laser scanner collecting data in tunnels

The laser beam also has different reflectance values depending on the type of material it encounters, and these different values result in a variation of the chromatic value of the acquired points. Thanks to this property, if a grayscale visualization is set, it provides a perception of the point cloud as if it were mapped with a photorealistic high-resolution black and white texture, making interpretative reading extremely easy and enhanced, as shown in the following images.

The processing of the survey carried out with laser scanning technology allows for the automatic generation of traditional documents such as plans, elevations, and sections from any plane of section and projection, as well as axonometric and perspective views. It is also possible to create photoplans and digital orthophotos by applying photographic documentation directly onto the model with excellent graphic rendering. Finally, it is possible to use the laser scanner survey to perform virtual navigation inside the gallery, focusing, where of interest, on details at a scale specifically calibrated for that purpose.

The ability to query the 3D point cloud at any moment and to navigate and visualize the survey in three dimensions allows the inspector/designer to carry out measurement, investigation, and cataloging operations of elements of interest. The generated model can also be used for all "data mining" operations, that is, for the automatic or semi-automatic extraction of information from vast amounts of data.

This research used the Ultra-High-Resolution RGB images derived from point clouds captured using laser scanners described above. The thermal images and specific point clouds were not included in the main research investigations.



Figure 4.3: Methodology of image capturing

### 4.1.2 Image Processing

The dataset was created from images captured in two road tunnels in Italy, both constructed from concrete and in operation for over 50 years. Their locations provide geographic diversity: one is situated in the Abruzzo region in east-central Italy, while the other is in the Liguria region in the northwest. The Figure 4.4 represents the region of two tunnels under investigation in Italy. As mentioned in the section 4.1.1, the images in this study were acquired using a Mobile Laser Scanner, as laser point clouds are commonly used for data collection in the tunnel's limited lighting conditions. The scanner used was the TS4 model developed by the Spacetec company [113]. This instrument utilizes advanced technology to achieve high-accuracy surveys in 2- or 3-lane tunnels. Its 360-degree laser scanner captures the entire tunnel vault and roadway by measuring distances through phase difference calculations. With a rotating head acquiring up to 10,000 points per full rotation, it collects data while mounted on a vehicle moving at around 5 km/h. The scanner's head rotates at 200 revolutions per second, generating detailed point clouds that represent the tunnel's surface and provide reflectance data for material analysis. While the primary output of this instrument is a point cloud, various outputs can be derived from this, including tunnel panoramic unwrapped images used for this study. These images are generated from a rigorous three-dimensional representation, which ensures precision and reliability in analysis. The images obtained represent the full length of the tunnels, and due to the different lengths of each tunnel, every image has a unique dimension. For example, the images of two tunnels discussed in this paper have $10.000 \times 158.679$ and $10.000 \times 212.414$ pixels.

The images were stored in the .TIFF format as 8-bit gray-scale. Afterward, the panoramic unwrapped view is obtained, and the real-world size corresponding to each pixel is calculated according to the tunnel dimensions described in the tunnel documentation.

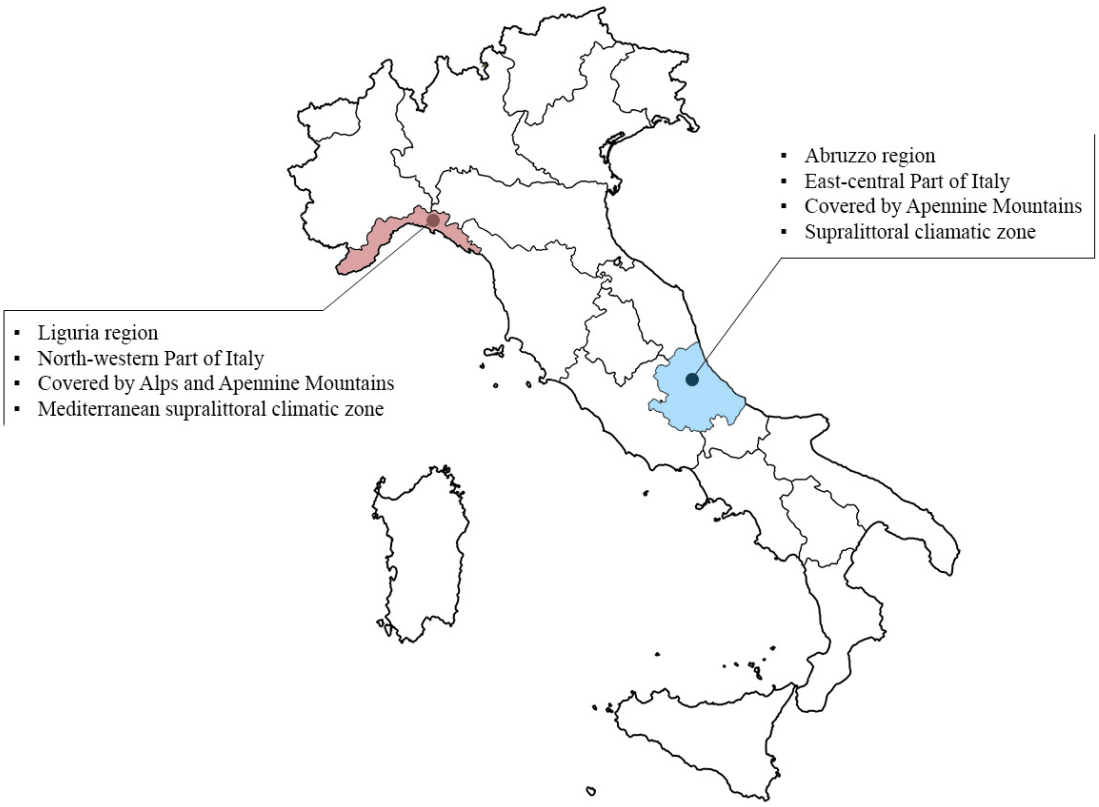
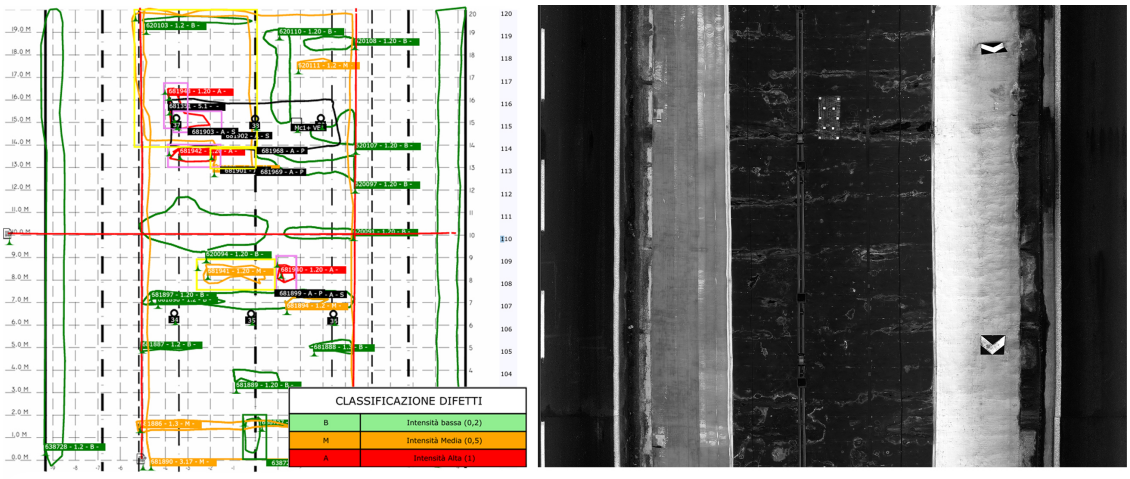



Figure 4.4: Location of the tunnels under investigation in Italy

Each tunnel has a report file defining a description of different defects based on the Italian guidelines - [23] - represented in every 20-meter section, by experienced engineers through visual inspection. Figure 4.5 shows the one section captured image as well as the report file. Accordingly, the tunnel images were divided into 20-meter longitudinal sections along the tunnel direction, and pavement parts on both sides were removed, leaving the tunnel lining for further investigations.

In the images, the crown part of the tunnels and the side walls have a noticeable color difference. The crown part is significantly darker, while the side walls are lighter. To decrease color differences in the images, to facilitate the labeling phase, and also to have better training performance, a gamma correction [114] equal to 0.5 was applied for both tunnels and enhanced the quality of the images as shown in 4.6.




(a) (b)

Figure 4.5: a) Report file of one section annotated by experienced engineers through visual inspection. b) The captured image of the corresponding section.

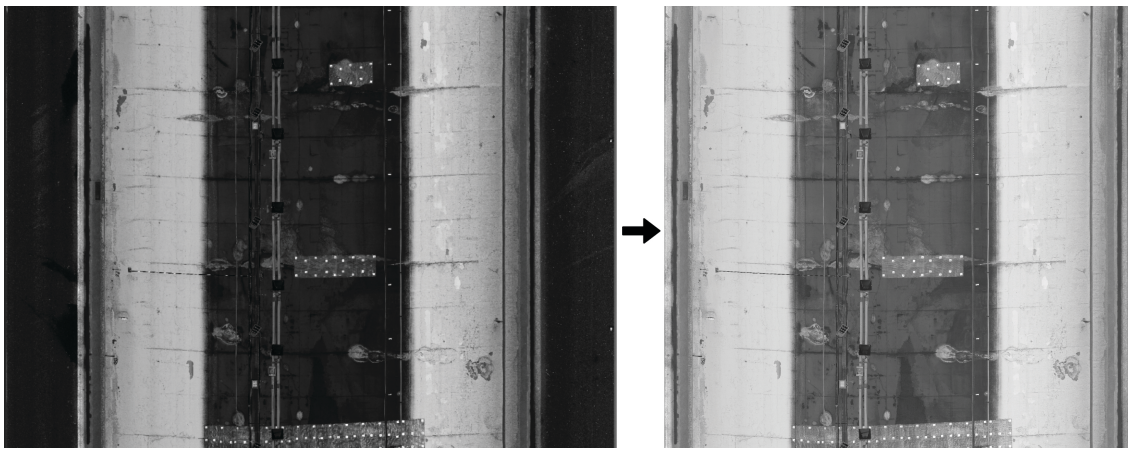


Figure 4.6: comparison of a tunnel section before and after processing. The raw captured image is shown on the left, while the processed version, after applying gamma correction and removing the road section, is on the right.

### 4.1.3 Labeling and Creation of the Dataset

To make the labeling phase more efficient, each 20-meter tunnel section is divided into six equal-sized patches, three columns, and two rows, to increase the efficiency of the labeling phase. The size of these cropped images varies depending on the tunnel; the first tunnel's images were $2155 \times 3539$, while the second tunnel's images were $2144 \times 3821$. The labeling technique is applied to these pictures. The Segment Anything Model (SAM) is the basis for a semi-automatic annotation tool used for the labeling phase [115] [116]. This software greatly improves labeling efficiency

by providing a prompt-based way to create object masks with a small number of points, which can then be manually refined by adjusting the boundaries. The user environment of the ISAT software is shown in Figure 4.7. The photographs were annotated by two specialists; one person did the initial labeling, while the other was in charge of going over the annotations to ensure that the various classifications were consistent and reliable.

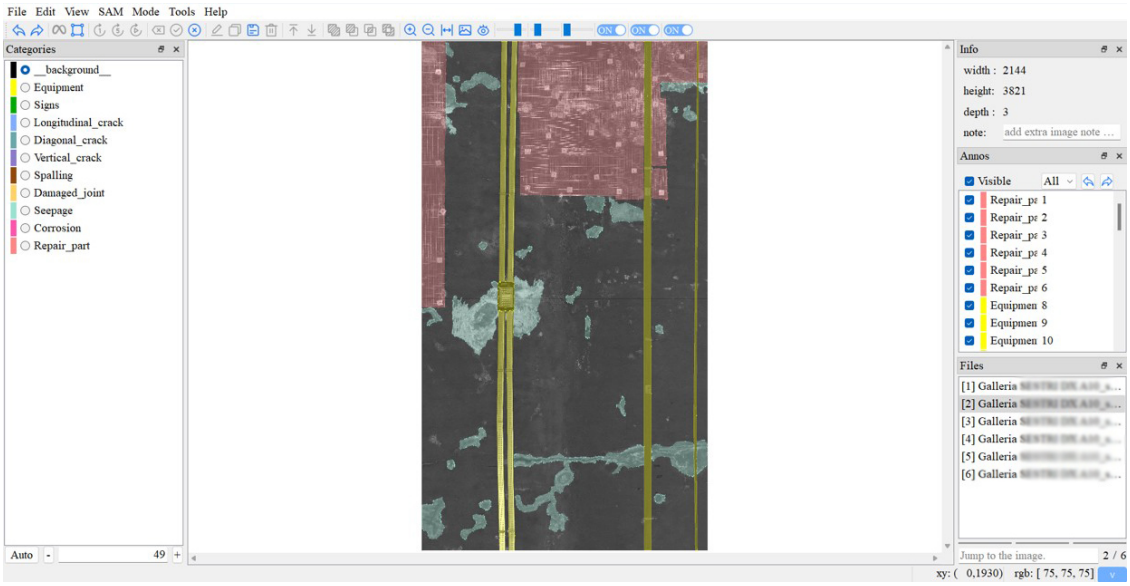



Figure 4.7: The ISAT software user environment used during the labeling phase.

Eight different classes, including defect parts and non-defect parts, are included in the dataset. While non-defect portions deal with tunnel equipment, repaired areas, and traffic signs, defect parts concentrate on the kinds of tunnel problems recommended by Italian regulation. The labeling phase is completed in the first stage by identifying the exact kinds of defects that the guideline introduced. This guideline presents 61 different kinds of defects in 12 categories. Following some testing, the team decided to combine subcategories in the defect section and created five classes for defect labeling because of the poor performance in differentiating the various defect classes. Table 4.1 shows the taxonomy of defects, while 4.8 shows an example of these classes taken from the dataset.

A JSON file created by labeling software connects each image and the corresponding annotation. As a result, a Python script was used to transform the annotated JSON file into a Microsoft COCO dataset [63]. The annotation consists of a dictionary containing two ”main” keys: info, where the image is described, and ”objects”, which contains the list of things that have been categorized.

The dataset’s original photos were $2155 \times 3539$ and $2144 \times 3821$ in size, respectively. A preprocessing step was used to standardize the models’ input size. Each image was cropped into six patches in this step, which were placed in three rows

Table 4.1: Taxonomy of Defects

| **Guidelines Categories** | **Code** | **Defects Description** | **New Class** |
|---|---|---|---|
| Defects caused by water presence | 1.1 | Drippings | Seepage |
| | 1.2 | Water Ingress | |
| | 1.3 | Concretions – Deposits – Encrustations | |
| | 1.4 | Effects of frost - traces of salts | |
| | 1.5 | Efflorescence on mortar or concrete | |
| Defects in the coating materials (concrete) | 1.19 | Cracks and spalling due to reinforcement corrosion | Corrosion |
| Defects related to the structural elements and geometry of the tunnel | 3.1 | Presence of longitudinal cracks along the coating | Cracks |
| | 3.2 | Diagonal cracks | |
| | 3.3 | Vertical cracks | |
| | 3.4 | Shrinkage cracks | |
| | 3.5 | Curvilinear cracks | |
| Defects relating to the structural elements and geometry of the tunnel- Construction Defects | 3.14 | Deterioration of concrete joints | Damaged Joint |
| | 3.15 | Surface defects in concrete | Spalling |

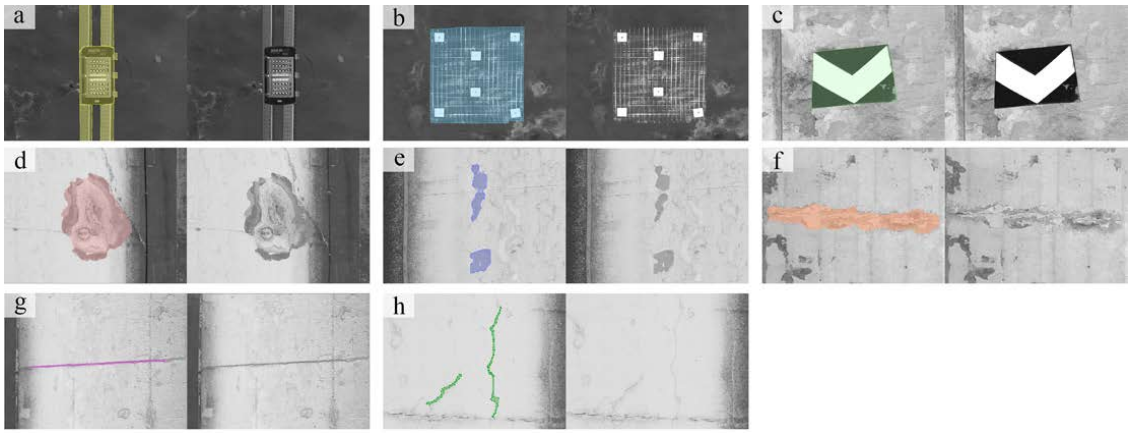



Figure 4.8: Example of categories represent: a) Equipment, b) Repair part, c) Traffic Sign, d) Seepage, e) Spalling, f) Corrosion, g) Damaged joint, h) Crack

and two columns. After that, each of these patches was further cropped centrally to create square images that were $1024 \times 1024$ in size. The COCO annotation files were changed to appropriately reflect the increased image dimensions to account for these changes.

The dataset consists of 1800 photos after being cropped to $1024 \times 1024$. Even though there were 100 photos among them that lacked annotations, they were not included in the collection. To sum up, the dataset includes 1700 photos with 6821 comments in five different categories. The dataset was separated into training and validation sets for the purpose of training the model. Eighty percent of the photos were randomly assigned to the training set, and the remaining twenty percent were assigned to the validation set, following standard practices. Figure 4.9 represents the whole procedure of image processing from the original raw image to the processed labeled one.

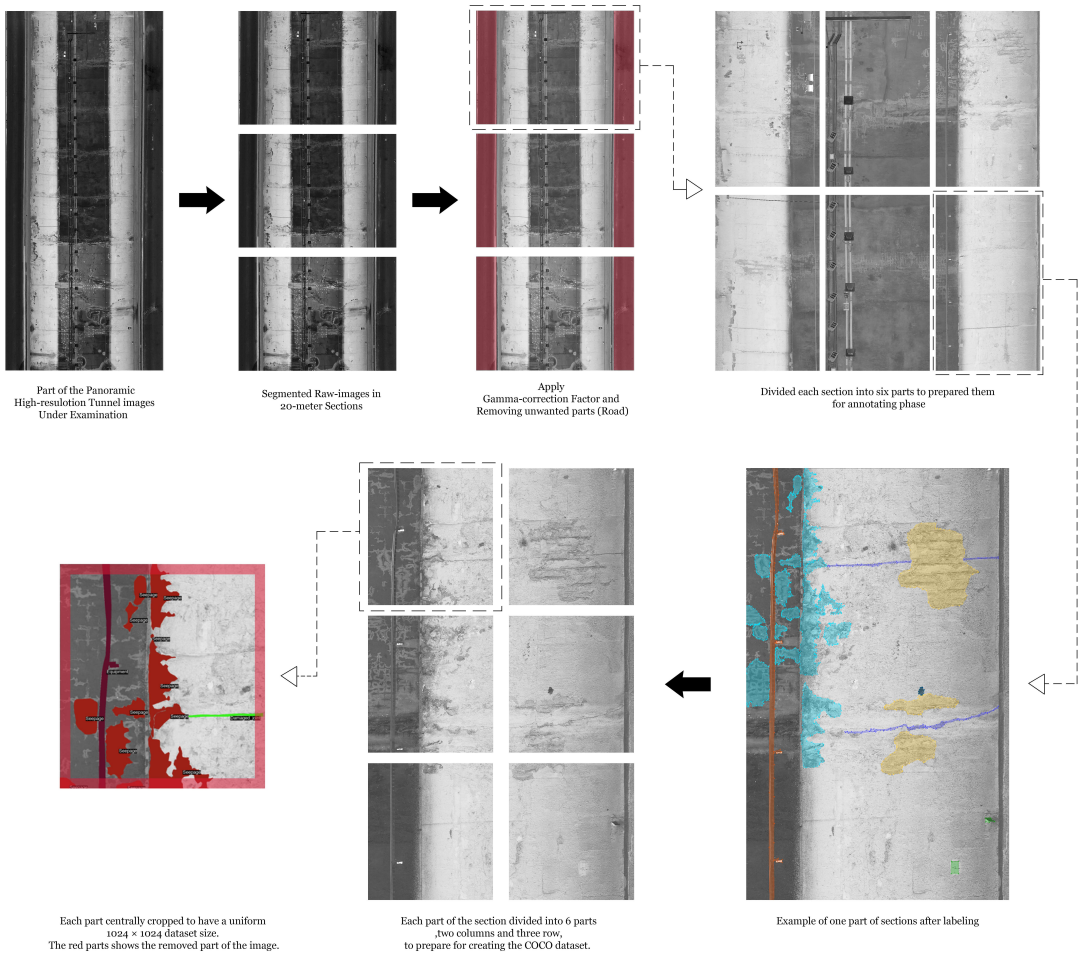



Figure 4.9: Overall process of image processing from a large ultra-high resolution image obtained directly during the inspection to a labeled processed image for the dataset.

### 4.1.4 Neural Network Training and Evaluation

In this research, instance and semantic segmentation tasks were used to benchmark the introduced dataset. The instance segmentation experiments tackled distinguishing eight different defect and non-defect categories, while the semantic segmentation task just focused on investigating between two categories, defect and non-defect, as well as background. The models used in this research are described briefly in 2.2. The table 4.2 shows the information regarding the instance segmentation models utilized to benchmark the dataset, and table 4.3 represents the semantic segmentation models. The model used in this research can be categorized as CNN- and Transformer-based. We benchmark and evaluate our proposed dataset using these

two methods for comparison with different approaches in deep learning algorithms.

Table 4.2: Instance segmentation models under investigation.

| Model type | Method | Backbone | Params (Million) | Flops($1 \times 10^{12}$) |
|---|---|---|---|---|
| CNN-Based | Mask R-CNN | ResNet50 | 44.009 | 0.263 |
| | MS R-CNN | ResNet50 | 60.349 | 0.302 |
| | Cascade Mask R-CNN | ResNet50 | 77.048 | 1.785 |
| | HTC | ResNet50 | 77.456 | 1.719 |
| | PointRend | ResNet50 | 59.846 | 0.214 |
| | SOLOv2 | ResNet50 | 46.593 | 0.249 |
| | CondInst | ResNet50 | 34.164 | 0.350 |
| Trasnformer-Based | Mask R-CNN | SWIN L | 215.253 | 0.938 |
| | Mask R-CNN | ViT B | 111.000 | 0.840 |
| | QueryInst | SWIN L | 344.152 | 0.829 |
| | Mask2Former | SWIN L | 216.156 | 0.991 |

Table 4.3: Semantic segmentation models under investigation.

| Model type | Method | Backbone | Params (Million) | Flops($1 \times 10^{9}$) |
|---|---|---|---|---|
| CNN-Based | BiSeNetV1 | ResNet50 | 56.857 | 0.396 |
| | UperNet | ResNet50 | 64.051 | 0.948 |
| | DeepLabV3+ | ResNet50 | 65.74 | 1.079 |
| | PSPNet | ResNet50 | 46.603 | 0.714 |
| Trasnformer-Based | SegFormer | MIT | 44.604 | 0.239 |
| | Segmenter | ViT B | 101.609 | 0.504 |
| | Mask2Former | SWIN L | 216.156 | 0.991 |

### 4.1.5 Evaluation Metrics

The damage identification task was implemented using instance segmentation methods. As a result, their mainstream assessment will also be presented. The widely accepted COCO (Microsoft Common Objects in Context) official evaluation criteria

are typically followed using the segmentation evaluation method. If the projected category matches the Ground Truth (GT) category and the IoU between the anticipated instance's result and the GT surpasses a predetermined threshold $\tau$, the prediction is deemed accurate. The following is a related evaluation approach.

$$IoU = \frac{TP}{TP + FP + FN} \tag{4.1}$$

where $TP$, $FN$, and $FP$ denote the true positive, false negative, and false positive. A schematic overview of IoU explanation is shown in 4.10

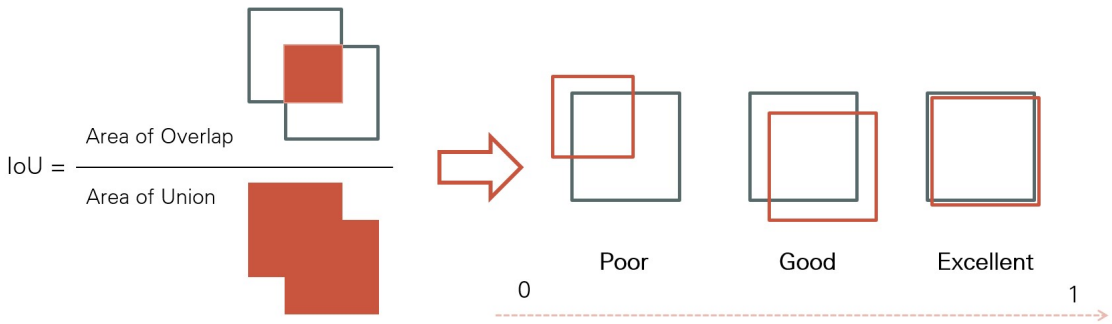



Figure 4.10: Explanation of the Intersection Over Union (IoU)

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4.2}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4.3}$$

$$AR(\tau) = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i(\tau)}{TP_i(\tau) + FN_i(\tau)} \tag{4.4}$$

$$AP = \frac{1}{101} \times \sum_{r \in 0, 0.01, \ldots, 1} p_{interpolation}(r) \tag{4.5}$$

where $TP_i$, $FN_i$, and $FP_i$ denote the true positive, false negative, and false positive instances $i$, and $p_{\text{interpolation}}(r)$ is the precision obtained through interpolation at the given maximum recall level $r$. Here, $i = 1, 2, 3, \ldots, n$, where $n$ represents the total number of instances. The Average Recall (AR) measures the average recall across a dataset and is computed as the mean recall over a set of predefined IoU thresholds $\tau$ or a single IoU level. It quantifies the model's ability to detect or segment objects, considering both true positives and missed detections. The Average Precision (AP) quantifies the area under the precision-recall curve and is computed

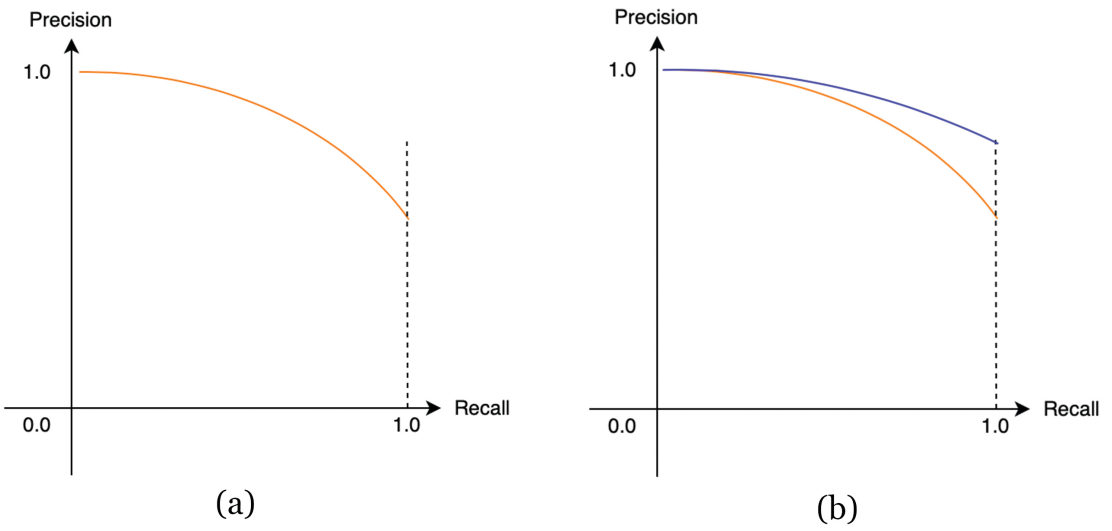



Figure 4.11: Precision-recall curves for AP: (a) the area under the curve (AUC) represents the AP. (b) The larger AUC represents the higher AP. In this curve blue line has a higher AP than the orange line.

as the mean precision over 101 equally spaced recall levels: $[0.0, 0.01, 0.02, \ldots, 1.0]$. The evaluation of instance segmentation encompasses both bounding boxes and masks, with the results presented as $AP_b$ for bounding boxes and $AP_m$ for masks. Overall, we provide AP at IoU thresholds of 0.5, 0.75, and the average over 0.5 to 0.95 with a 0.05 interval. Accordingly, the $AR_m$ represents the average recall for masks and the $AR_b$ shows the average recall for bounding boxes at the average IoU thresholds over 0.5 to 0.95 with a 0.05 interval. Figure 4.11 shows the concept of Average Precision concerning the precision-recall curve. Mathematically, AP is the area under the precision-recall curve, calculated either by integrating over all recall points or using specific interpolation methods.

$$Accuracy = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{4.6}$$

$$F_\beta - Score = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \beta = 1{,}2 \tag{4.7}$$

Detection and segmentation tasks are highly dependent on substantial volumes of labeled image data and considerable computational resources. Historically, one of the primary challenges in utilizing convolutional neural networks (CNNs) was their reliance on extensive labeled datasets and the significant computational costs involved. However, advancements in labeling techniques and the adoption of parallel computation using graphics processing units (GPUs) have largely mitigated these limitations. Despite these developments, there remains a scarcity of well-annotated,

open-source datasets containing information on tunnel defects. Consequently, it is essential to gather a diverse and sufficient collection of images depicting various tunnel defects to facilitate accurate defect segmentation.

# Chapter 5

# Result and Discussion

Detection and segmentation tasks are highly dependent on substantial volumes of labeled image data and considerable computational resources. Historically, one of the primary challenges in utilizing convolutional neural networks (CNNs) was their reliance on extensive labeled datasets and the significant computational costs involved. However, advancements in labeling techniques and the adoption of parallel computation using graphics processing units (GPUs) have largely mitigated these limitations. Despite these developments, there remains a scarcity of well-annotated, open-source datasets containing information on tunnel defects. Consequently, it is essential to gather a diverse and sufficient collection of images depicting various tunnel defects to facilitate accurate defect segmentation.

## 5.1 Parameter Setting and Hardware

All experiments were conducted using NVIDIA Tesla V100 SXM2 Tensor Core GPUs. The software environment consists of MMCV 2.0.1 and PyTorch 2.0.0 with CUDA version 11.8. For each experiment, the official configuration files for all algorithms from MMDetection [117] and MMSegmentation [118] were used. Data augmentation strategies configured in the official setup were applied. For all models, the strategies described below were used. The model input is standardized to $1024 \times 1024$. The model has been trained for 100 epochs with the AdamW optimizer [119]. The learning rate schedule follows a cosine annealing strategy [120]. Training begins with a linear warm-up phase lasting 1000 iterations, after which the learning rate is set to $le-4$ and gradually reduced to $le-7$.

## 5.2 Result of instance segmentation algorithms

For instance segmentation tasks nine state-of-the-art models, including Mask R-CNN [41], Ms RCNN [42], Cascade Mask R-CNN [43], HTC [44], PointRend [46],

SOLOv2 [47], CondInst [48], QueryInst [50],and Mask2Former [51], were used. These models have been chosen based on their architectural features, concerning CNN-based and Transformer-based approaches, to highlight a comparison between these methodologies as well. All models were trained to start from weights pre-trained on the ImageNet-1k dataset [53] except for Swin-Base and Swin-Large, which were pre-trained on ImageNet-22k. The result of the instance segmentation models presented in 5.1. All models generate bounding boxes and continue with creating masks for each instance. Except for the SoloV2 model, which directly obtains instance segmentation results without calculating bounding boxes. The confusion matrix of these models is represented in Figure 5.3. Some examples of model predictions are visualized in Figure 5.1 and Figure 5.2.

Table 5.1: Instance segmentation models result

| Method | Backbone | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ | $AR_b$ | $AR_m$ |
|---|---|---|---|---|---|---|---|---|---|
| Mask R-CNN | ResNet50 | 0.293 | 0.424 | 0.313 | 0.299 | 0.429 | 0.322 | 0.358 | 0.357 |
| MS R-CNN | ResNet50 | 0.295 | 0.413 | 0.307 | 0.301 | 0.422 | 0.322 | 0.361 | 0.360 |
| Cascade M R-CNN | ResNet50 | 0.333 | 0.448 | 0.364 | 0.326 | 0.457 | 0.342 | 0.398 | 0.385 |
| HTC | ResNet50 | 0.347 | 0.468 | 0.380 | 0.339 | 0.469 | 0.363 | 0.417 | 0.406 |
| PointRend | ResNet50 | 0.296 | 0.416 | 0.313 | 0.309 | 0.430 | 0.327 | 0.362 | 0.371 |
| SOLOv2 | ResNet50 | - | - | - | 0.293 | 0.433 | 0.297 | - | 0.358 |
| CondInst | ResNet50 | 0.284 | 0.408 | 0.306 | 0.295 | 0.422 | 0.308 | 0.365 | 0.371 |
| Mask R-CNN | SWIN L | 0.316 | 0.437 | 0.343 | 0.310 | 0.435 | 0.327 | 0.386 | 0.378 |
| Mask R-CNN | ViT B | 0.316 | 0.447 | 0.337 | 0.313 | 0.453 | 0.333 | 0.392 | 0.382 |
| QueryInst | SWIN L | 0.358 | 0.494 | 0.388 | 0.348 | 0.500 | 0.369 | 0.507 | 0.459 |
| Mask2Former | SWIN L | **0.396** | **0.526** | **0.419** | **0.398** | **0.559** | **0.414** | **0.555** | **0.522** |

### 5.2.1 Detailed Analysis

Also, a deeper analysis with various backbones for three models including Mask R-CNN, Cascade Mask R-CNN and Mask2Former have been utilized for a better investigation. The result of all these three models with different backbones is described in the Tables 5.4, 5.5, and 5.6. Confusion matrix of these three mentioned models are represented in Figure 5.4. For the calculation of the confusion matrices, the models with Swin-Large backbone were used. Confusion matrices were calculated in the four different confidence thresholds of 0.1, 0.2, 0.3, and 0.5.

Consequently, the Table 5.7 presented an comparative analysis of the three mentioned models based on each categories. Some examples of visualized prediction

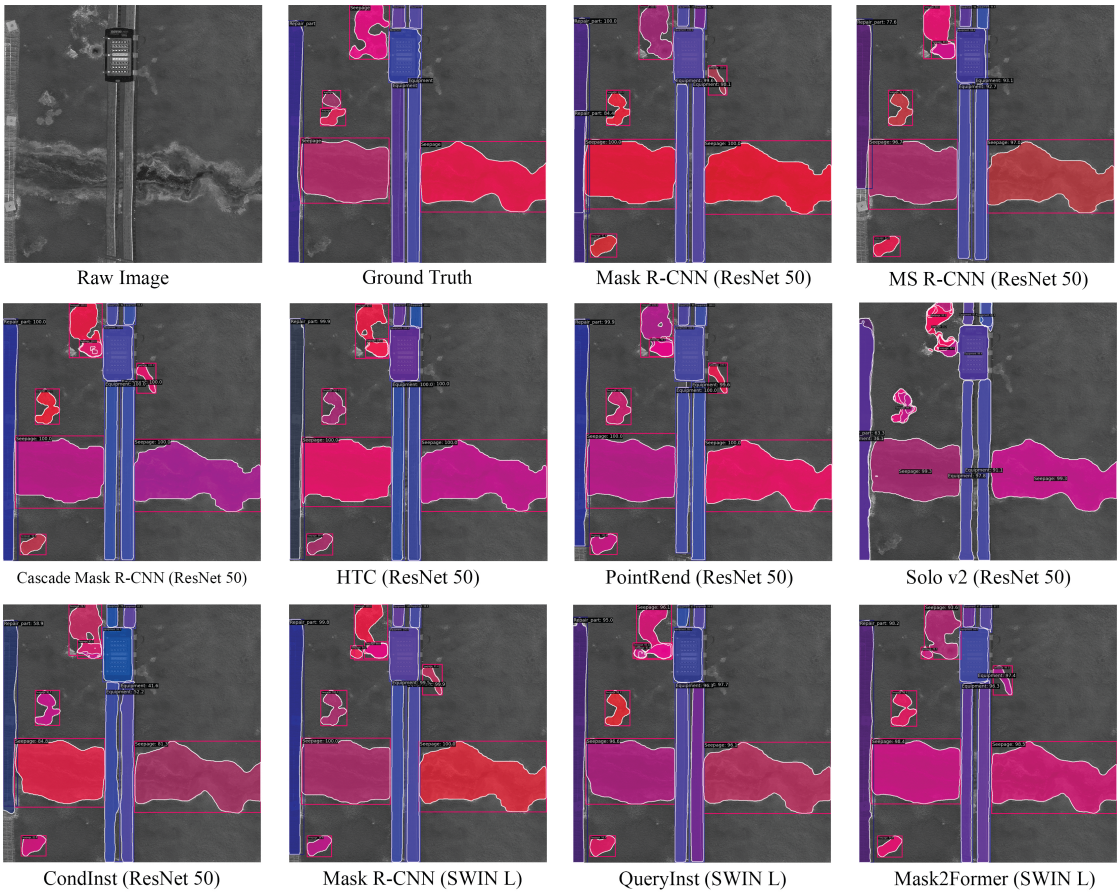



Figure 5.1: Prediction visualization of instance segmentation models with confidence score of 0.3

of the models presented in 5.5. For these visualization confidence thresholds of 0.3 were utilized.

### 5.2.2 Discussion

After a comprehensive comparison, we found that under the conditions of our current dataset, Mask2Former performs the best, followed by QueryInst and HTC. The reason for this is that Mask2Former has a better Transformer-based architecture [51]. Overall, the transformer-based architectures show slightly better performance than CNN-based on our dataset, which can be derived from their novel architecture.

In terms of specific defect categories, it's important to note the following:

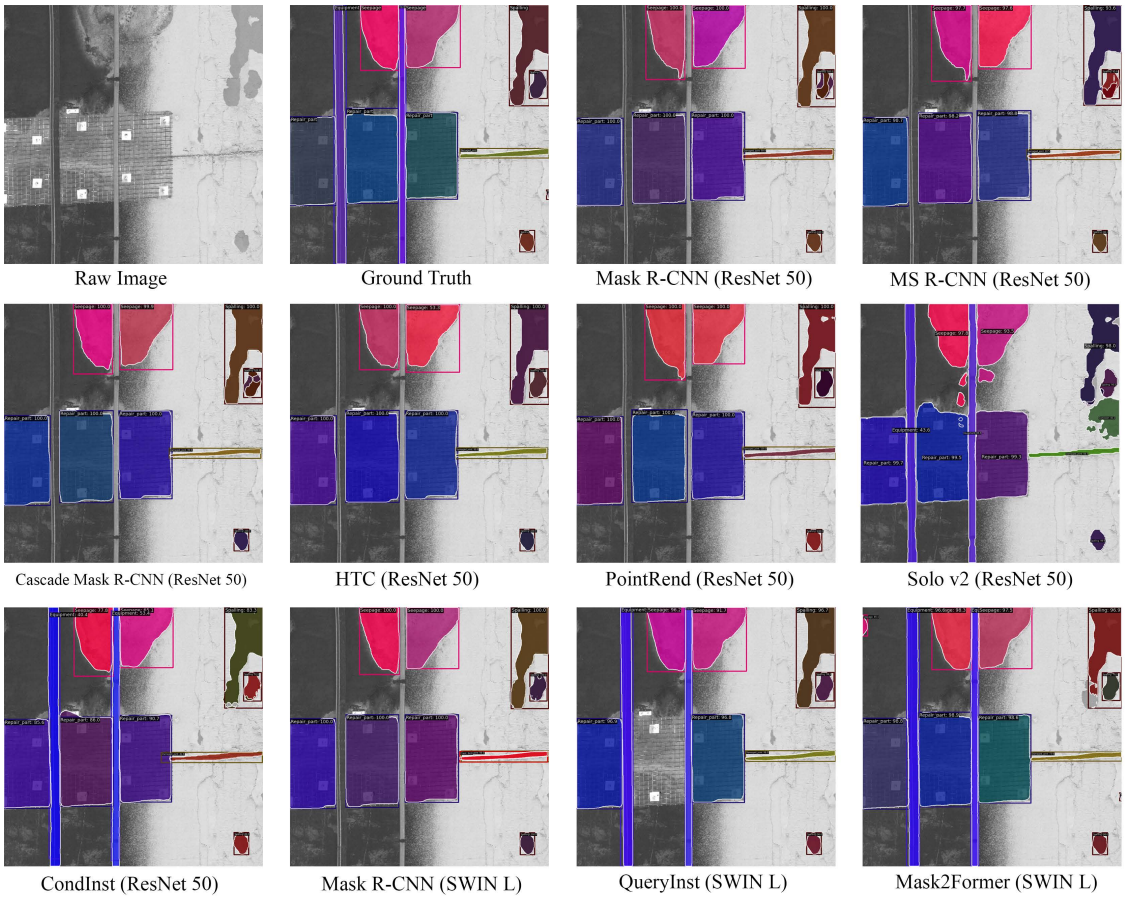



Figure 5.2: Prediction visualization of instance segmentation models with confidence score of 0.3

- **Seepage**: For this defect category, all the models demonstrate high performance, but Mask2Former achieves the highest $AP_{mask}$ of 0.450, indicating its efficiency in identifying this class of defects. Cascade mask R-CNN has $AP_{mask}$= 0.395 and, Mask R-CNN has the lowest performance with $AP_{mask}$= 0.395.

- **Spalling**: Mask2Former also shows the highest performance in this case as well, outperforming other models with $AP_{mask}$ = 0.466, which indicates superior precision in segmenting spalling defects. While Mask R-CNN and Cascade Mask R-CNN show a good performance, they have lower results compared to Mask2Former.

- **Corrosion**: Cascade Mask R-CNN had an $AP_{mask}$ of 0.125, this is a slight improvement over both Mask R-CNN (0.101) and Mask2Former (0.116). This shows that all models struggle with the accurate identification and delineation

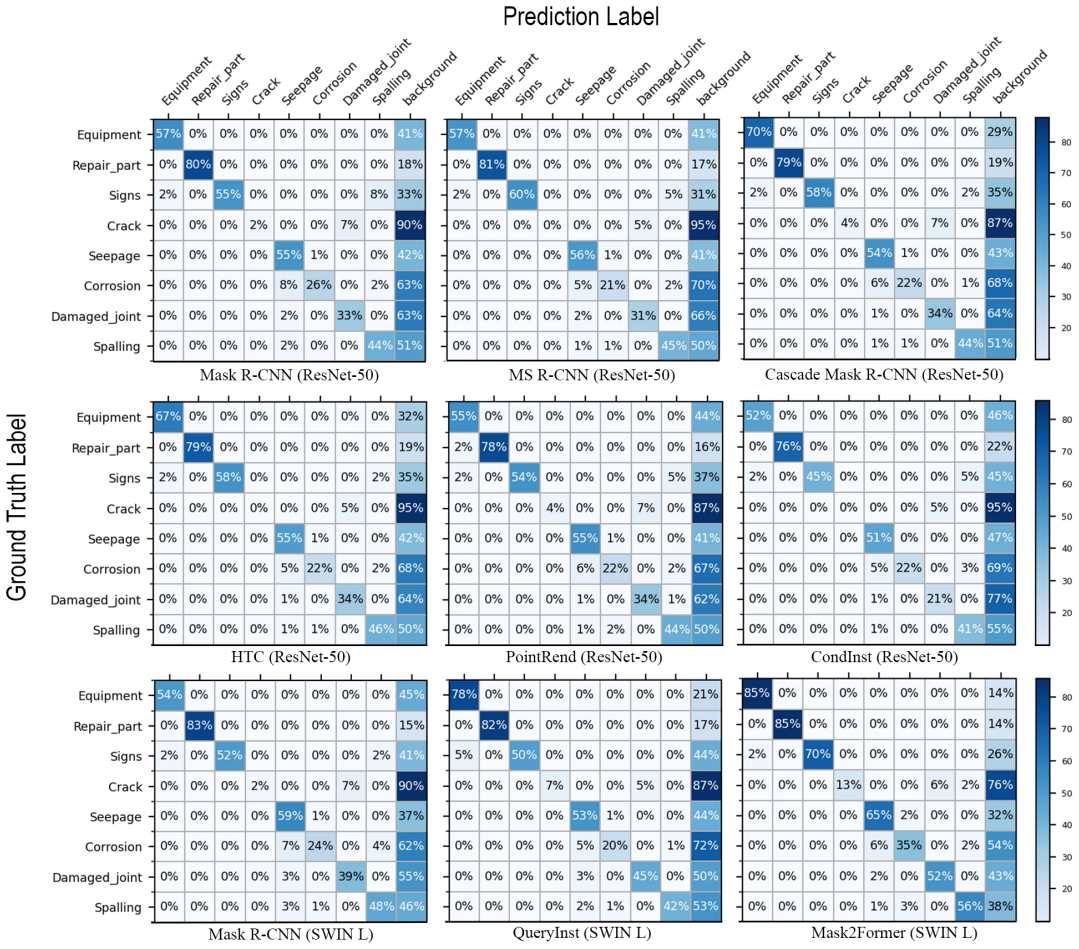



Figure 5.3: Confusion matrix of instance segmentation models with confidence score of 0.3

of corrosion-related instances.

- **Damaged Joints:** In this category, Mask2Former achieves the highest $AP_{mask}$ score, of 0.178. The results reveal that Mask R-CNN and Cascade Mask R-CNN performances are lower compared to other models, indicating an overall difficulty in detecting and segmenting damaged joints.

- **Cracks**: All three algorithms exhibit a poor performance in detecting cracks, the highest $AP_{mask}$ of Mask R-CNN was only 0.006, and Mask2Former and Cascade Mask R-CNN both obtained lower results of 0.005 and 0.008 respectively.

As a result of these findings, Mask2Former generally performed better across the dataset and excelled in segmenting *Seepage* and *Spalling*, but all models had

Table 5.2: Comparative Analysis of CNN-based Instance Segmentation Algorithms on Our Dataset

| Algorithm | Metric | Equipment | Repair Part | Sign | Seepage | Corrosion | Damaged Joint | Spalling | Crack |
|---|---|---|---|---|---|---|---|---|---|
| Mask R-CNN (ResNet-50) | $AP_b$ | 0.431 | 0.619 | 0.395 | 0.336 | 0.119 | 0.104 | 0.337 | 0.004 |
| | $AP_b^{50}$ | 0.594 | 0.810 | 0.567 | 0.493 | 0.216 | 0.268 | 0.429 | 0.015 |
| | $AP_b^{75}$ | 0.485 | 0.684 | 0.437 | 0.360 | 0.109 | 0.050 | 0.376 | 0.000 |
| | $AP_m$ | 0.461 | 0.668 | 0.386 | 0.336 | 0.125 | 0.072 | 0.344 | 0.000 |
| | $AP_m^{50}$ | 0.609 | 0.829 | 0.567 | 0.493 | 0.240 | 0.253 | 0.436 | 0.000 |
| | $AP_m^{75}$ | 0.514 | 0.746 | 0.446 | 0.370 | 0.109 | 0.010 | 0.381 | 0.000 |
| MS R-CNN (ResNet-50) | $AP_b$ | 0.437 | 0.652 | 0.412 | 0.337 | 0.092 | 0.093 | 0.332 | 0.002 |
| | $AP_b^{50}$ | 0.595 | 0.809 | 0.582 | 0.486 | 0.179 | 0.235 | 0.414 | 0.003 |
| | $AP_b^{75}$ | 0.469 | 0.694 | 0.454 | 0.359 | 0.070 | 0.047 | 0.363 | 0.003 |
| | $AP_m$ | 0.468 | 0.673 | 0.413 | 0.341 | 0.105 | 0.068 | 0.339 | 0.000 |
| | $AP_m^{50}$ | 0.608 | 0.819 | 0.603 | 0.488 | 0.221 | 0.230 | 0.410 | 0.000 |
| | $AP_m^{75}$ | 0.513 | 0.728 | 0.475 | 0.368 | 0.097 | 0.013 | 0.379 | 0.000 |
| Cascade Mask R-CNN (ResNet-50) | $AP_b$ | 0.553 | 0.693 | 0.430 | 0.353 | 0.100 | 0.171 | 0.351 | 0.014 |
| | $AP_b^{50}$ | 0.699 | 0.809 | 0.583 | 0.502 | 0.204 | 0.317 | 0.440 | 0.031 |
| | $AP_b^{75}$ | 0.639 | 0.765 | 0.474 | 0.382 | 0.088 | 0.158 | 0.392 | 0.015 |
| | $AP_m$ | 0.574 | 0.714 | 0.395 | 0.353 | 0.106 | 0.113 | 0.348 | 0.007 |
| | $AP_m^{50}$ | 0.713 | 0.828 | 0.583 | 0.508 | 0.211 | 0.346 | 0.433 | 0.031 |
| | $AP_m^{75}$ | 0.657 | 0.786 | 0.395 | 0.384 | 0.104 | 0.019 | 0.394 | 0.000 |
| HTC (ResNet-50) | $AP_b$ | 0.565 | 0.696 | 0.482 | 0.365 | 0.120 | 0.174 | 0.375 | 0.002 |
| | $AP_b^{50}$ | 0.725 | 0.817 | 0.629 | 0.536 | 0.239 | 0.321 | 0.467 | 0.010 |
| | $AP_b^{75}$ | 0.633 | 0.761 | 0.546 | 0.391 | 0.118 | 0.174 | 0.415 | 0.000 |
| | $AP_m$ | 0.586 | 0.720 | 0.426 | 0.364 | 0.124 | 0.114 | 0.376 | 0.002 |
| | $AP_m^{50}$ | 0.739 | 0.837 | 0.629 | 0.539 | 0.225 | 0.298 | 0.473 | 0.010 |
| | $AP_m^{75}$ | 0.660 | 0.804 | 0.456 | 0.395 | 0.125 | 0.041 | 0.425 | 0.000 |
| PointRend (ResNet-50) | $AP_b$ | 0.415 | 0.669 | 0.384 | 0.339 | 0.110 | 0.096 | 0.339 | 0.017 |
| | $AP_b^{50}$ | 0.581 | 0.805 | 0.538 | 0.507 | 0.182 | 0.248 | 0.430 | 0.040 |
| | $AP_b^{75}$ | 0.446 | 0.746 | 0.412 | 0.358 | 0.115 | 0.040 | 0.386 | 0.000 |
| | $AP_m$ | 0.451 | 0.714 | 0.394 | 0.352 | 0.117 | 0.079 | 0.353 | 0.008 |
| | $AP_m^{50}$ | 0.593 | 0.833 | 0.538 | 0.517 | 0.208 | 0.264 | 0.445 | 0.040 |
| | $AP_m^{75}$ | 0.520 | 0.791 | 0.388 | 0.388 | 0.120 | 0.011 | 0.401 | 0.000 |
| SOLOv2 (ResNet-50) | $AP_m$ | 0.580 | 0.658 | 0.344 | 0.332 | 0.090 | 0.073 | 0.269 | 0.000 |
| | $AP_m^{50}$ | 0.777 | 0.817 | 0.524 | 0.492 | 0.189 | 0.257 | 0.412 | 0.000 |
| | $AP_m^{75}$ | 0.673 | 0.686 | 0.293 | 0.368 | 0.088 | 0.004 | 0.268 | 0.000 |
| CondInst (ResNet-50) | $AP_b$ | 0.413 | 0.614 | 0.407 | 0.338 | 0.092 | 0.082 | 0.330 | 0.000 |
| | $AP_b^{50}$ | 0.590 | 0.780 | 0.574 | 0.485 | 0.187 | 0.197 | 0.426 | 0.000 |
| | $AP_b^{75}$ | 0.444 | 0.676 | 0.509 | 0.355 | 0.076 | 0.034 | 0.367 | 0.000 |
| | $AP_m$ | 0.473 | 0.672 | 0.365 | 0.338 | 0.091 | 0.058 | 0.330 | 0.000 |
| | $AP_m^{50}$ | 0.612 | 0.810 | 0.546 | 0.496 | 0.215 | 0.200 | 0.433 | 0.000 |
| | $AP_m^{75}$ | 0.532 | 0.729 | 0.392 | 0.362 | 0.091 | 0.008 | 0.368 | 0.000 |

limited performance in detecting *Cracks*. There are many data sets in literature that focus primarily on *Crack* defects like MCrack1300 [121]. This performance gap can be filled by combining our data set with existing ones.

Figure 5.5 shows some visualized results derived from experiments. A qualitative assessment of the models' performance is possible with these visualizations, which help understand where the algorithms succeed or fail. Mask2Former, for example, correctly identifies large seepage regions, while sometimes underestimating smaller

Table 5.3: Comparative Analysis of Transformer-based Instance Segmentation Algorithms on Our Dataset

| Algorithm | Metric | Equipment | Repair Part | Sign | Seepage | Corrosion | Damaged Joint | Spalling | Crack |
|---|---|---|---|---|---|---|---|---|---|
| Mask R-CNN (SWIN L) | $AP_b$ | 0.417 | 0.647 | 0.425 | 0.399 | 0.098 | 0.151 | 0.371 | 0.017 |
| | $AP_b^{50}$ | 0.545 | 0.832 | 0.541 | 0.553 | 0.188 | 0.339 | 0.473 | 0.021 |
| | $AP_b^{75}$ | 0.465 | 0.718 | 0.481 | 0.440 | 0.110 | 0.098 | 0.411 | 0.021 |
| | $AP_m$ | 0.441 | 0.682 | 0.374 | 0.395 | 0.101 | 0.108 | 0.372 | 0.006 |
| | $AP_m^{50}$ | 0.548 | 0.832 | 0.536 | 0.562 | 0.185 | 0.318 | 0.476 | 0.021 |
| | $AP_m^{75}$ | 0.498 | 0.747 | 0.410 | 0.433 | 0.100 | 0.015 | 0.411 | 0.000 |
| Mask R-CNN (ViT-B) | $AP_b$ | 0.477 | 0.626 | 0.419 | 0.361 | 0.105 | 0.156 | 0.363 | 0.024 |
| | $AP_b^{50}$ | 0.643 | 0.786 | 0.534 | 0.525 | 0.209 | 0.308 | 0.462 | 0.109 |
| | $AP_b^{75}$ | 0.547 | 0.696 | 0.418 | 0.396 | 0.097 | 0.143 | 0.391 | 0.005 |
| | $AP_m$ | 0.514 | 0.657 | 0.371 | 0.355 | 0.115 | 0.112 | 0.369 | 0.011 |
| | $AP_m^{50}$ | 0.656 | 0.825 | 0.529 | 0.536 | 0.238 | 0.324 | 0.482 | 0.033 |
| | $AP_m^{75}$ | 0.582 | 0.718 | 0.412 | 0.401 | 0.119 | 0.026 | 0.401 | 0.000 |
| QueryInst (SWIN L) | $AP_b$ | 0.634 | 0.717 | 0.369 | 0.386 | 0.102 | 0.245 | 0.348 | 0.065 |
| | $AP_b^{50}$ | 0.825 | 0.846 | 0.546 | 0.532 | 0.177 | 0.478 | 0.45 | 0.098 |
| | $AP_b^{75}$ | 0.728 | 0.778 | 0.453 | 0.424 | 0.089 | 0.206 | 0.377 | 0.047 |
| | $AP_m$ | 0.454 | 0.241 | 0.245 | 0.387 | 0.087 | 0.271 | 0.438 | 0.076 |
| | $AP_m^{50}$ | 0.681 | 0.813 | 0.771 | 0.458 | 0.148 | 0.29 | 0.407 | 0.070 |
| | $AP_m^{75}$ | 0.539 | 0.740 | 0.774 | 0.471 | 0.184 | 0.290 | 0.439 | 0.070 |
| Mask2Former (SWIN L) | $AP_b$ | **0.664** | **0.734** | **0.529** | **0.434** | **0.099** | **0.247** | **0.435** | **0.027** |
| | $AP_b^{50}$ | **0.841** | **0.851** | **0.676** | **0.592** | **0.202** | **0.422** | **0.558** | **0.044** |
| | $AP_b^{75}$ | **0.739** | **0.775** | **0.572** | 0.461 | 0.092 | 0.208 | **0.472** | **0.034** |
| | $AP_m$ | **0.686** | **0.774** | **0.505** | **0.450** | 0.116 | **0.178** | **0.466** | 0.005 |
| | $AP_m^{50}$ | **0.888** | **0.908** | **0.680** | **0.634** | **0.271** | **0.476** | **0.603** | **0.016** |
| | $AP_m^{75}$ | **0.792** | **0.840** | **0.525** | **0.489** | **0.107** | **0.052** | **0.504** | **0.001** |

ones. It has decent performance in spalling, but struggles with accuracy when the texture or contrast is significantly different. Compared to Cascade Mask R-CNN and Mask2Former, Mask R-CNN displays a consistent but moderate performance.

## 5.3 Result of semantic segmentation algorithm

To display the compatibility of our database, we conducted semantic segmentation experiments with multiple State-of-The-Art architectures as shown in Table 4.3, considering dividing the dataset into two classes, Defect and Non-defect parts. The defect part includes the seepage, spalling, damaged joint, cracks, and corrosion, while the non-defect part represents the equipment, repair part, and sign classes. The pixel-level evaluations are described in Table 5.8. A comprative analysis of

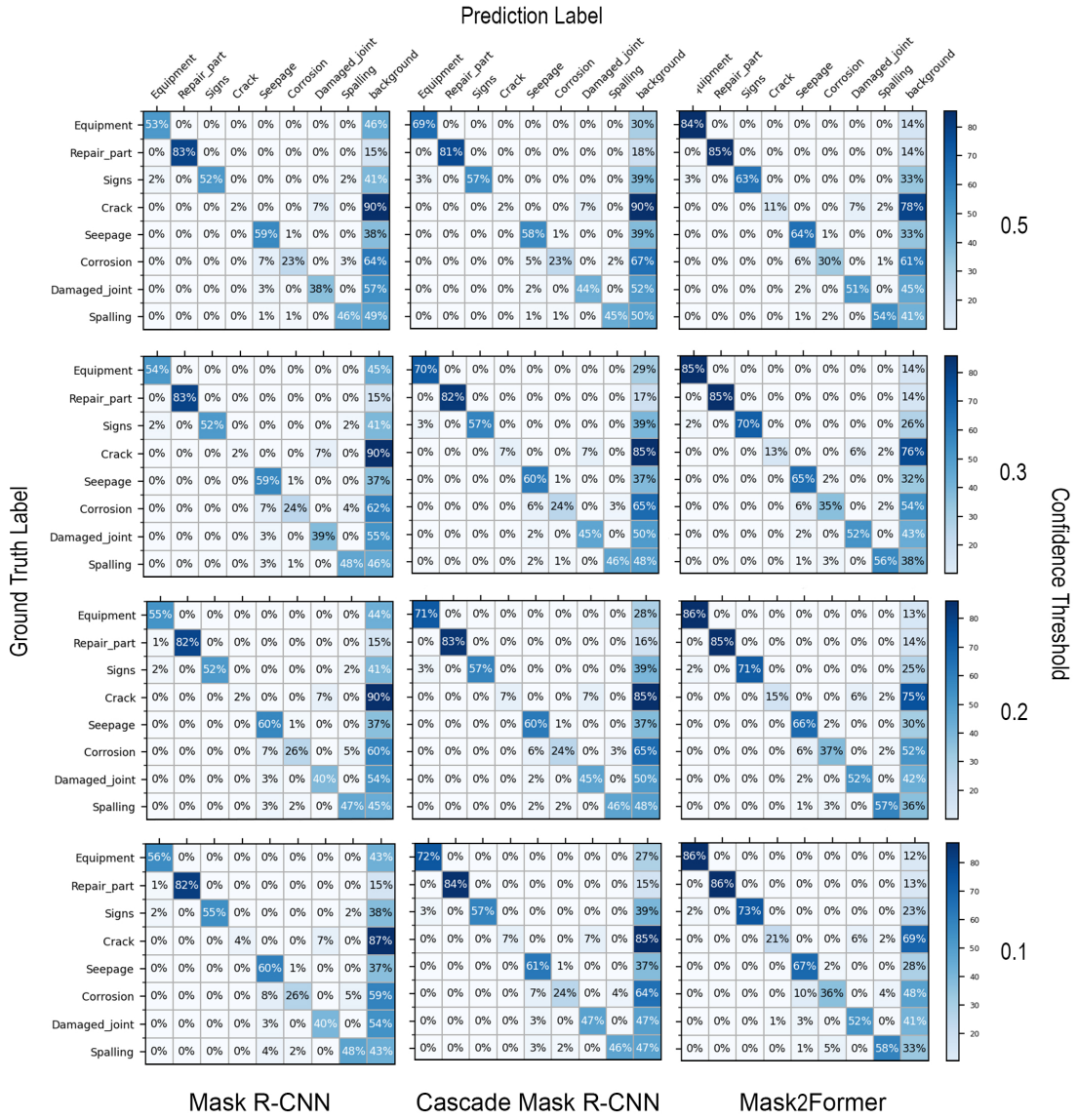



Figure 5.4: Normalized Confusion Matrix of Instance Segmentation models with Swin-Large backbone with multiple confidence Thresholds

semantic segmentation algorithms concering the each class introduced in Table 5.9. Also, some prediction examples are represented in Figure 5.6, and Figure 5.7.

### 5.3.1 Discussion

After a comprehensive comparison of semantic segmentation algorithms,we found that under the conditions of our current dataset,Mask2Former performs the best

Table 5.4: Result of Mask R-CNN across different backbone sizes.

| Backbone | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ | $AR_b$ | $AR_m$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 0.293 | 0.424 | 0.313 | 0.299 | 0.429 | 0.322 | 0.358 | 0.357 |
| ResNet-101 | 0.300 | 0.421 | 0.326 | 0.300 | 0.434 | 0.312 | 0.367 | 0.366 |
| ResNeXt-101-64x4d | 0.301 | 0.410 | 0.328 | 0.291 | 0.402 | 0.316 | 0.359 | 0.349 |
| ResNeXt-101-32x8d | 0.313 | 0.424 | 0.350 | 0.307 | 0.427 | 0.333 | 0.380 | 0.368 |
| Swin-T | **0.326** | **0.468** | **0.351** | **0.324** | **0.453** | **0.337** | **0.396** | **0.387** |
| Swin-S | 0.311 | 0.438 | 0.332 | 0.313 | 0.449 | 0.330 | 0.379 | 0.377 |
| Swin-B* | 0.315 | 0.432 | 0.340 | 0.309 | 0.431 | 0.320 | 0.378 | 0.368 |
| Swin-L* | 0.316 | 0.437 | 0.343 | 0.310 | 0.435 | 0.327 | 0.386 | 0.378 |

* Swin-B and Swin-L backbones are pre-trained on the ImageNet-22k dataset.

Table 5.5: Result of Cascade Mask R-CNN across different backbone sizes.

| Backbone | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ | $AR_b$ | $AR_m$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 0.333 | 0.448 | 0.364 | 0.326 | 0.457 | 0.342 | 0.398 | 0.385 |
| ResNet-101 | 0.330 | 0.443 | 0.357 | 0.319 | 0.441 | 0.337 | 0.390 | 0.375 |
| ResNeXt-101-64x4d | 0.343 | 0.453 | 0.378 | 0.330 | 0.460 | 0.347 | 0.407 | 0.391 |
| ResNeXt-101-32x8d | 0.329 | 0.439 | 0.355 | 0.319 | 0.442 | 0.343 | 0.392 | 0.376 |
| Swin-T | 0.352 | 0.461 | **0.387** | 0.339 | 0.458 | 0.359 | 0.410 | 0.394 |
| Swin-S | 0.351 | 0.474 | 0.383 | 0.335 | 0.474 | 0.351 | 0.414 | 0.394 |
| Swin-B* | 0.351 | **0.475** | 0.379 | 0.341 | 0.468 | 0.354 | 0.415 | **0.400** |
| Swin-L* | **0.355** | 0.468 | 0.381 | **0.346** | **0.478** | **0.360** | **0.416** | 0.399 |

* Swin-B and Swin-L backbones are pre-trained on the ImageNet-22k dataset.

considering the IoU metric,followed by Segformer and Segmenter. The reason for this is because Mask2Former has a better Transformer based architecture [51]. Overall, the transformer-based architectures shows slightly better performance than CNN-based on our dataset, it can be derived from their novel architecture.

In terms of specific categories, it's important to note the following:

- **Defect**: For this category, all the models demonstrate reliable performance, but Mask2Former achieves the highest $IoU$ of 70.89, indicating its efficiency in identifying class of defects. SegFormer has second best performance with $IoU$= 67.11 and, BISeNetV1 has the lowest performance with $IoU$= 44.98.

- **Non-defect**: Mask2Former also shows the highest performance in this case as

Table 5.6: Result of Mask2Former across different backbone sizes.

| Backbone | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ | $AR_b$ | $AR_m$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 0.388 | 0.527 | 0.402 | 0.386 | 0.551 | 0.399 | 0.527 | 0.500 |
| ResNet-101 | 0.388 | 0.516 | 0.408 | 0.378 | 0.543 | 0.396 | 0.533 | 0.492 |
| Swin-T | 0.388 | **0.526** | 0.408 | 0.390 | 0.551 | 0.407 | 0.554 | 0.518 |
| Swin-S | 0.391 | 0.523 | 0.401 | **0.400** | 0.557 | 0.412 | 0.544 | 0.511 |
| Swin-B* | 0.381 | 0.504 | 0.398 | 0.383 | 0.537 | 0.401 | 0.516 | 0.491 |
| Swin-L* | **0.396** | **0.526** | **0.419** | 0.398 | **0.559** | **0.414** | **0.555** | **0.522** |

* Swin-B and Swin-L backbones are pre-trained on the ImageNet-22k dataset.

Table 5.7: Comparative Analysis of Instance Segmentation Algorithms with Swin-Large Backbone on Our Dataset

| **Algorithm** | **Metric** | **Equipment** | **Repair Part** | **Sign** | **Seepage** | **Corrosion** | **Damaged Joint** | **Spalling** | **Crack** |
|---|---|---|---|---|---|---|---|---|---|
| Mask R-CNN | $AP_b$ | 0.417 | 0.647 | 0.425 | 0.399 | 0.098 | 0.151 | 0.371 | 0.017 |
| | $AP_b^{50}$ | 0.545 | 0.832 | 0.541 | 0.553 | 0.188 | 0.339 | 0.473 | 0.021 |
| | $AP_b^{75}$ | 0.465 | 0.718 | 0.481 | 0.440 | 0.110 | 0.098 | 0.411 | 0.021 |
| | $AP_m$ | 0.441 | 0.682 | 0.374 | 0.395 | 0.101 | 0.108 | 0.372 | 0.006 |
| | $AP_m^{50}$ | 0.548 | 0.832 | 0.536 | 0.562 | 0.185 | 0.318 | 0.476 | 0.021 |
| | $AP_m^{75}$ | 0.498 | 0.747 | 0.410 | 0.433 | 0.100 | 0.015 | 0.411 | 0.000 |
| Cascade Mask R-CNN | $AP_b$ | 0.563 | 0.672 | 0.431 | 0.425 | **0.113** | 0.240 | 0.374 | 0.021 |
| | $AP_b^{50}$ | 0.703 | 0.841 | 0.539 | 0.559 | 0.182 | **0.429** | 0.453 | 0.036 |
| | $AP_b^{75}$ | 0.626 | 0.716 | 0.438 | **0.467** | **0.129** | **0.248** | 0.407 | 0.015 |
| | $AP_m$ | 0.578 | 0.700 | 0.396 | 0.419 | **0.125** | 0.159 | 0.380 | **0.008** |
| | $AP_m^{50}$ | 0.711 | 0.841 | 0.567 | 0.571 | 0.222 | 0.416 | 0.462 | **0.036** |
| | $AP_m^{75}$ | 0.656 | 0.758 | 0.422 | 0.467 | **0.138** | 0.021 | 0.417 | 0.000 |
| Mask2Former | $AP_b$ | **0.664** | **0.734** | **0.529** | **0.434** | **0.099** | **0.247** | **0.435** | **0.027** |
| | $AP_b^{50}$ | **0.841** | **0.851** | **0.676** | **0.592** | **0.202** | **0.422** | **0.558** | **0.044** |
| | $AP_b^{75}$ | **0.739** | **0.775** | **0.572** | 0.461 | 0.092 | 0.208 | **0.472** | **0.034** |
| | $AP_m$ | **0.686** | **0.774** | **0.505** | **0.450** | 0.116 | **0.178** | **0.466** | 0.005 |
| | $AP_m^{50}$ | **0.888** | **0.908** | **0.680** | **0.634** | **0.271** | **0.476** | **0.603** | **0.016** |
| | $AP_m^{75}$ | **0.792** | **0.840** | **0.525** | **0.489** | **0.107** | **0.052** | **0.504** | **0.001** |

well, outperforming other models with $IoU = 91.15$, which indicates superior precision in segmenting non-defect classes. While other models show a overall good performance, they have lower results compared to Mask2Former.

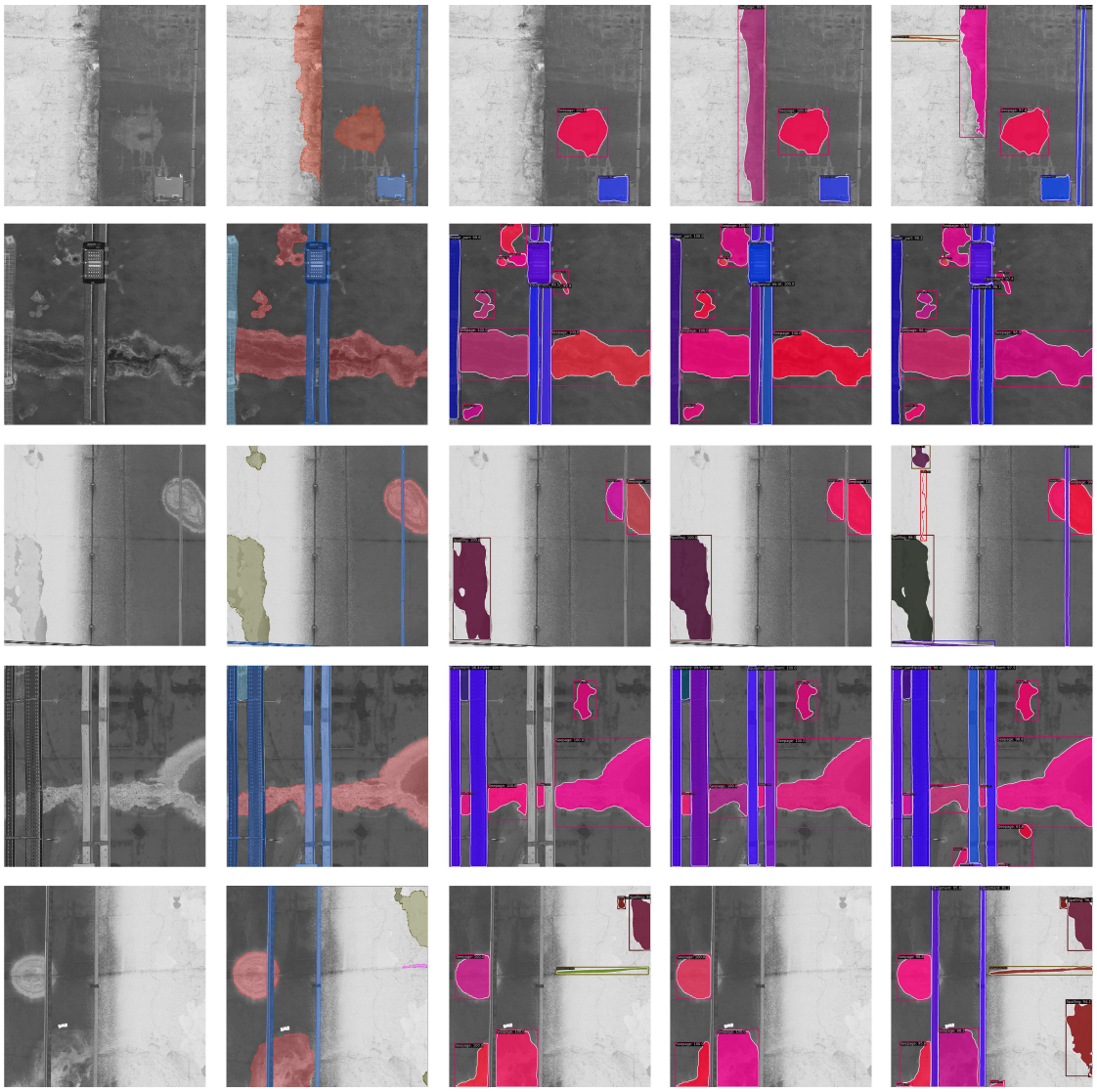



Figure 5.5: Some example of the instance segmentation models prediction with Swin Large backbone with confidence score of 0.3

Table 5.8: Semantic segmentation models under investigation.

| Method | Backbone | *IoU* | *Accurcay* | *Precision* | *Recall* | $F_1 - score$ |
|---|---|---|---|---|---|---|
| BiSeNetV1 | ResNet50 | 73.300 | 79.860 | 87.210 | 79.860 | 82.800 |
| UperNet | ResNet50 | 82.470 | **94.690** | 91.300 | 88.400 | 89.760 |
| DeepLabV3+ | ResNet50 | 83.260 | 89.470 | 91.270 | 89.470 | 90.320 |
| PSPNet | ResNet50 | 83.210 | 89.260 | 91.530 | 89.260 | 90.300 |
| SegFormer | MIT b3 | 84.230 | 89.960 | 92.160 | 89.960 | 90.990 |
| Segmenter | ViT B | 83.510 | 89.790 | 91.390 | 89.790 | 80.32 |
| Mask2Former | SWIN L | **85.650** | 91.470 | **92.37** | **91.470** | **91.910** |

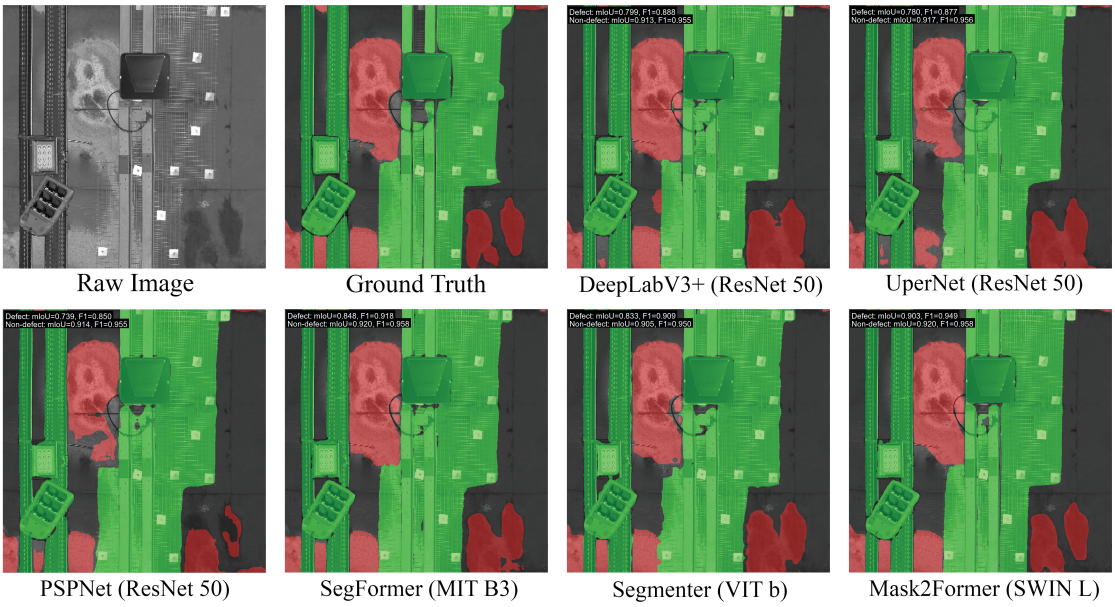



Figure 5.6: Example of prediction of the Semantic Segmentation test on our dataset. Green instances represent the Non-defect class while the red ones show the Defects.

Table 5.9: Comparative Analysis of Semantic Segmentation Algorithms on Our Dataset

| **Algorithm** | **Categories** | *IoU* | *Accurcay* | *Precision* | *Recall* | $F_1 - score$ |
|---|---|---|---|---|---|---|
| BiSeNetV1 | Defect | 44.98 | 52.12 | 76.64 | 52.12 | 62.05 |
| | Non-Defect | 84.15 | 89.22 | 93.67 | 89.22 | 91.39 |
| | Background | 90.91 | 97.45 | 93.12 | 97.45 | 95.24 |
| UperNet | Defect | 63.56 | 73.63 | 82.29 | 73.63 | 77.72 |
| | Non-Defect | 90.08 | 94.03 | 95.54 | 94.03 | 94.78 |
| | Background | 93.77 | 97.53 | 96.05 | 97.53 | 96.78 |
| DeeplabV3+ | Defect | 65.52 | 75.87 | 82.76 | 75.87 | 79.17 |
| | Non-Defect | 90.23 | 95.11 | 94.62 | 95.11 | 94.86 |
| | Background | 94.05 | 97.44 | 96.43 | 97.44 | 96.93 |
| PSPNet | Defect | 65.72 | 74.89 | 84.29 | 74.89 | 79.31 |
| | Non-Defect | 89.8 | 95.28 | 93.98 | 95.28 | 94.62 |
| | Background | 94.11 | 97.62 | 96.32 | 97.62 | 96.97 |
| SegFormer | Defect | 67.94 | 76.89 | 85.37 | 76.89 | 80.91 |
| | Non-Defect | 90.24 | 95.22 | 94.53 | 95.22 | 94.87 |
| | Background | 94.50 | 97.78 | 96.58 | 97.78 | 97.17 |
| Segmenter | Defect | 67.11 | 77.23 | 83.66 | 77.23 | 80.32 |
| | Non-Defect | 89.17 | 94.64 | 93.91 | 94.64 | 94.28 |
| | Background | 94.26 | 97.5 | 96.59 | 97.5 | 97.04 |
| Mask2Former | Defect | 70.89 | 81.45 | 84.54 | 81.45 | 82.97 |
| | Non-Defect | 91.15 | 95.33 | 95.41 | 95.33 | 95.37 |
| | Background | 94.92 | 97.63 | 97.16 | 97.63 | 97.39 |

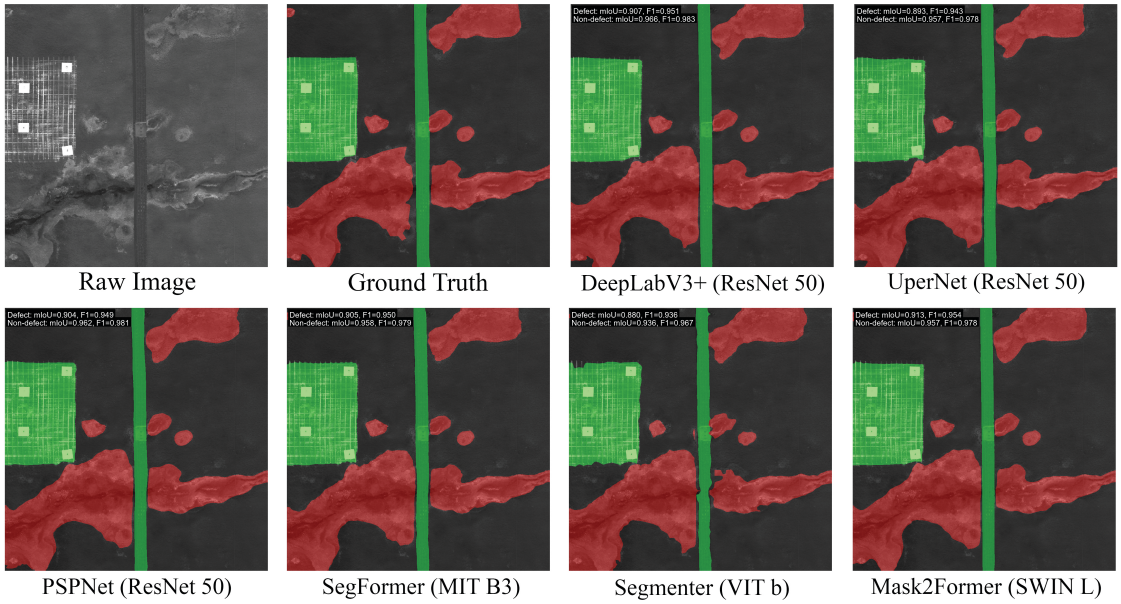



Figure 5.7: Example of prediction of the Semantic Segmentation test on our dataset. Green instances represent the Non-defect class while the red ones show the Defects.

# Chapter 6

# Damage Report

The widely used Gradio [122] is used to build the front-end interactive interface as shown in the Figure 6.1. Users have the ability to view boundary map legends, change between different visualization modes, export/download COCO format annotation files and PDF reports, batch upload images, and modify unified confidence thresholds (as previously discussed, with a dedicated threshold of 0.05 for "crack" and a unified threshold of 0.2 applied to all other categories). The original image and four chosen visualization analysis maps—binary, border, probability, and segmentation maps—are shown simultaneously in the visualization area. The dialogue space for the statistical findings of natural language queries is located beneath the visualization. The question area and the statistical findings that were returned are on the left, and the statistical charts are on the right.

PDF report generation is based on FPDF2 [123] and PyPDF2 [124]. First, we produced a template.pdf file with the required legends and empty tables for data entry. The boundary maps are then subjected to secondary processing, which involves drawing a coordinate system with the top of the vertical axis set to 20m and the bottom to 0m. The coordinate grid is drawn using this as a guide. According to the format used in the manual report displayed in 6.1, we set the image's center to 0 and its right side to be positive for the horizontal axis. The template.pdf file then contains the processed image embedded within it. Lastly, the relevant table is printed with the required text content.

For inference at three scales (50% overlap), roughly 270 cropped photos are needed for every 20m section. Each UHR image requires about 40 seconds of inference on an NVIDIA A100 Tensor Core GPU. The entire process, including pre-processing and post-processing, takes less than a minute. Our method provides consistent, objective, and repeatable results by automating damage identification for a 300m tunnel (15 sections of 20m) in about 15 minutes, in contrast to hand labeling approaches that are subjective and time-consuming. Figure 6.2 and Figure 6.3 are examples of two pages from separate tunnels. Overall, both tunnels exhibit exceptionally good segmentation performance, clearly defining damage areas and

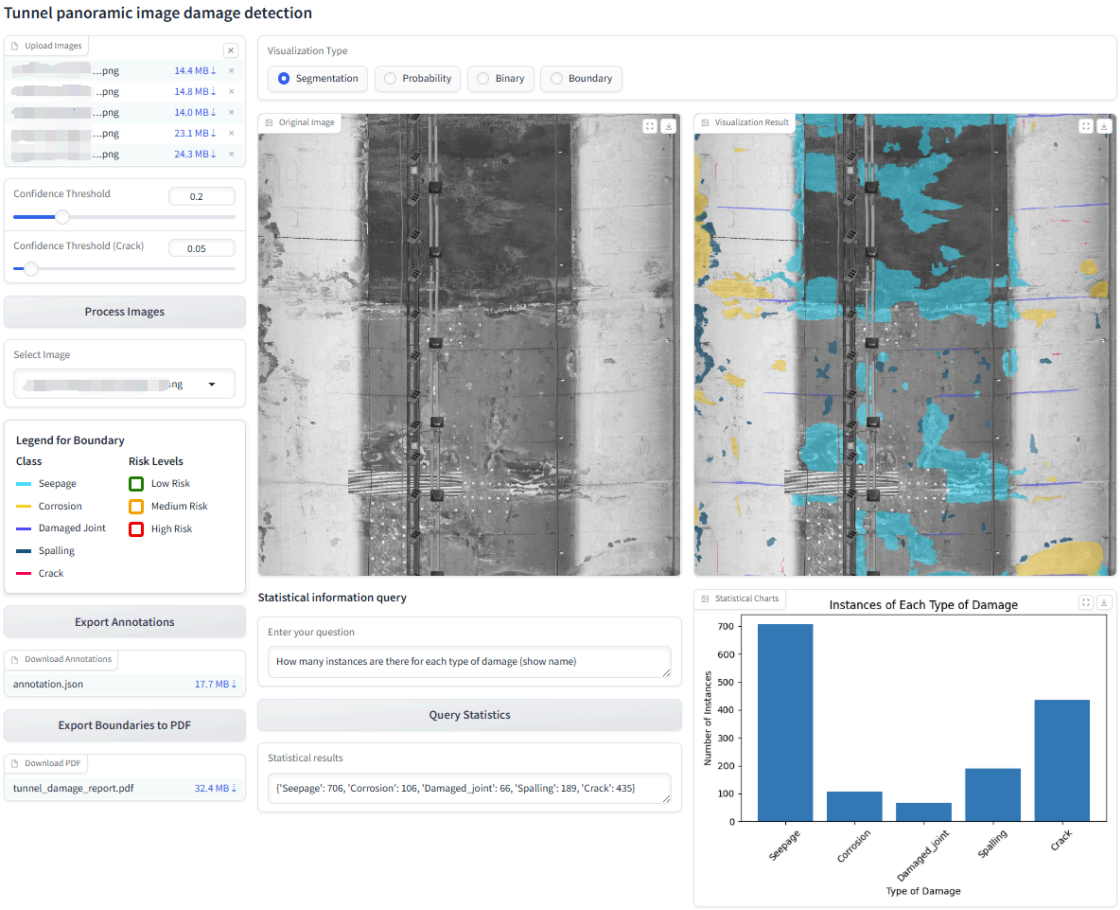



Figure 6.1: Interactive interface for tunnel panoramic image damage detection.

successfully differentiating between various damage kinds. Additionally, a large number of possible cracks were recorded, which is highly advantageous.

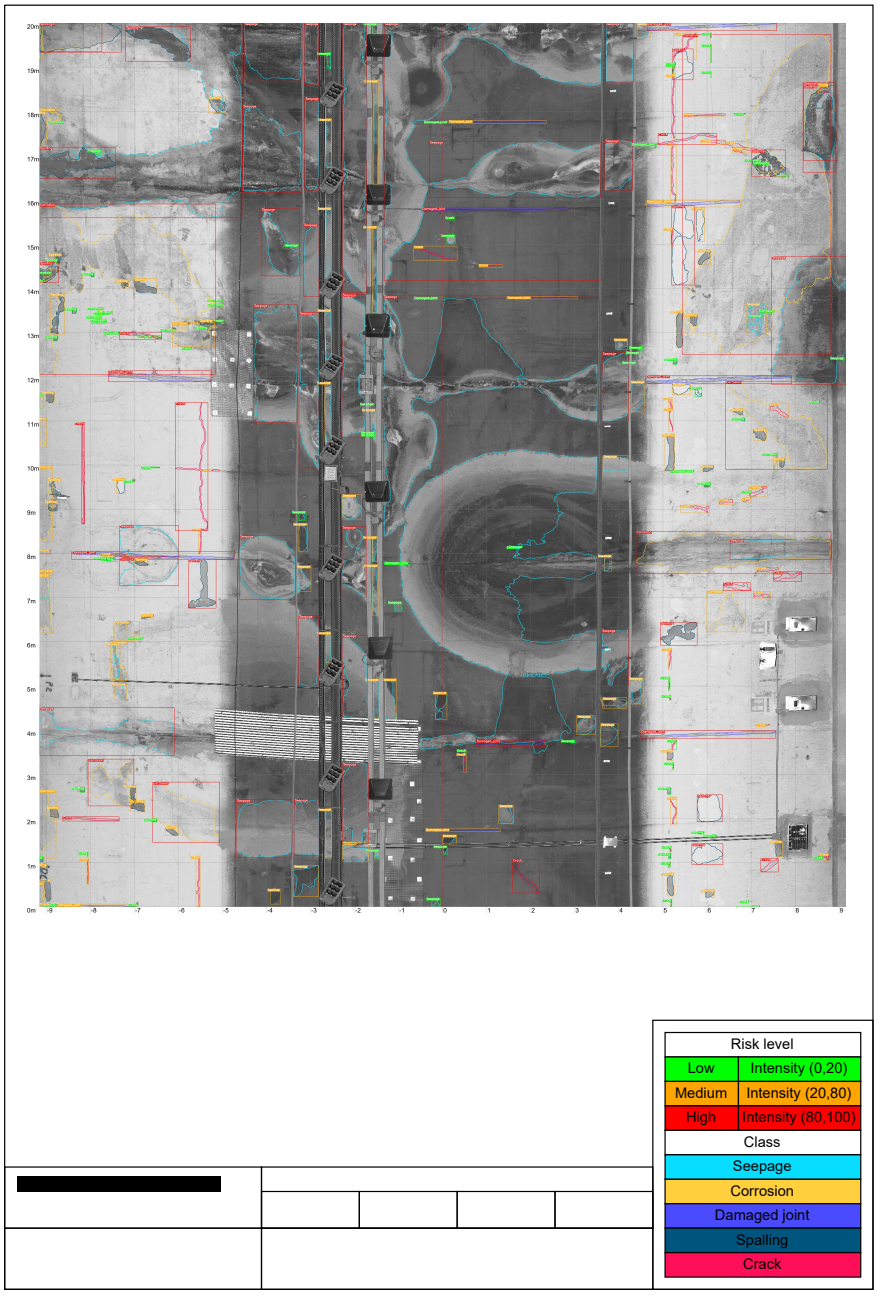



Figure 6.2: First example of automatically generated damage report from two different tunnels based on 20m tunnel local panoramic images.

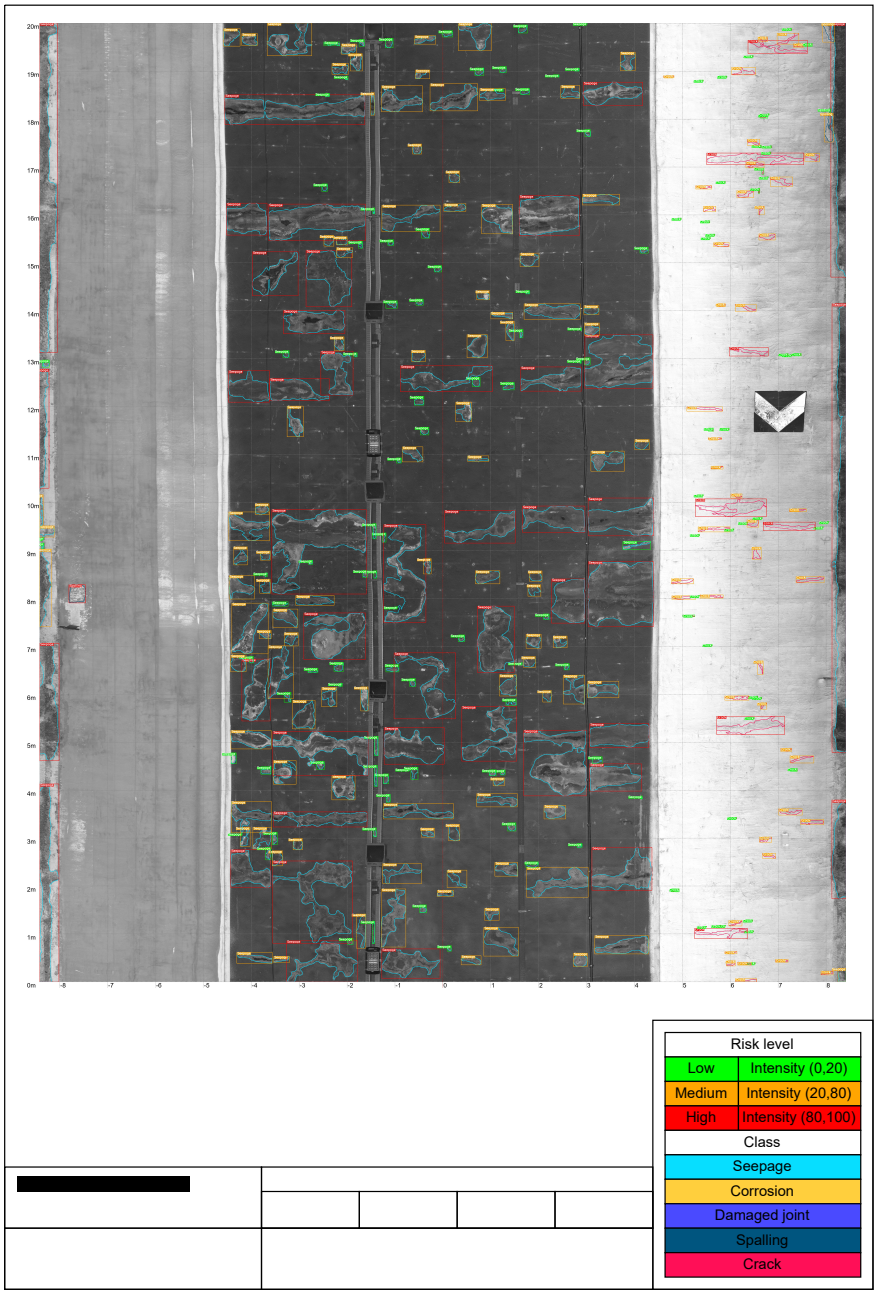



Figure 6.3: Second example of automatically generated damage report from two different tunnels based on 20m tunnel local panoramic images.

# Chapter 7

# Conclusion

In this paper, we present a novel method to process high-resolution panoramic images from concrete road tunnels to develop a database suitable for computer vision models to enhance the defect detection process concerning structural health monitoring criteria to satisfy the new Italian regulation concerns regarding to guide maintenance decision-making, aiming to minimize the risk of dangerous situations and prevent the need for urgent interventions. Subsequently, a damage report interface was introduced to show the compatibility of our proposed dataset in real-world scenarios. The database consists of eight different categories of concrete defects and tunnel equipment, based on Italian regulations, including seepage, spalling, crack, damaged joint, and corrosion for defect classes and equipment, repair part, and traffic sign. Lately, the database has been trained and evaluated with instance segmentation and semantic algorithms to verify the efficacy and performance.
For instance segmentation experiments the Mask2Former has shown the overall best performance with $AP_{mask}$ equal to 0.396. Regarding the categories, seepage and spalling have a satisfactory result, while the damaged joint and corrosion have shown slightly moderate performance. For the cracks, the performance was not satisfactory. It may have been caused due to the lack of subsequent data on this category in the database. In semantic segmentation experiments, we convert our categories into two parts, defects and non-defect parts. Also in this term, Mask2Former shows best performance with overall $mIoU$ equal to 85.650.
There are some limitations in our research. Firstly, dataset imbalance across different classes affects the model's performance and efficacy. Additionally, more state-of-the-art architectures could be explored to better evaluate the dataset and enhance performance analysis.
In future work, we aim to address class imbalance by introducing more instances in underrepresented classes. Furthermore, we plan to incorporate diverse architectures to gain deeper insights into dataset performance. Lastly, integrating multimodal data sources, such as thermal and RGB-D cameras, as well as data-driven Ground-Penetrating Radar (GPR) methodologies, could further enhance overall

model performance.

# Bibliography


[1] E. Parliament and Council, "Regulation (eu) 2024/1679 of the european parliament and of the council of 13 june 2024 on machinery," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1679, June 2024, accessed: 2025-04-21.

[2] Ministro delle infrastrutture e dei trasporti, "Relazione concernente lo stato di attuazione degli interventi relativi all'adeguamento delle gallerie stradali della rete transeuropea," https://www.senato.it/service/PDF/PDFServer/DF/426541.pdf, 2023, accessed: 2024-03-15.

[3] H. J. Liu, P. E. Love, J. Zhao, C. Lemckert, and K. Muldoon-Smith, "Transport infrastructure asset resilience: Managing government capabilities," *Transportation Research Part D: Transport and Environment*, vol. 100, p. 103072, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920921003692

[4] s. M. Piryonesi and T. El-Diraby, "Role of data analytics in infrastructure asset management: Overcoming data size and quality problems," *Journal of Transportation Engineering, Part B: Pavements*, vol. 146, p. 04020022, 06 2020.

[5] American Society of Civil Engineers, "2025 report card for america's infrastructure," 2025, accessed: 2025-04-24. [Online]. Available: https://infrastructurereportcard.org/wp-content/uploads/2025/03/Full-Report-2025-Natl-IRC-WEB.pdf

[6] A. Tarazona, M. Coelho, S. Fernandes, and J. Matos, "Transport infrastructure risk management in the context of climate change: policies, challenges, and opportunities," *Transportation Research Procedia*, vol. 72, pp. 1404–1411, 2023, tRA Lisbon 2022 Conference Proceedings Transport Research Arena (TRA Lisbon 2022),14th-17th November 2022, Lisboa, Portugal. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235214652300902X

[7] IPCC, *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, C. W. Team, H. Lee, and J. Romero, Eds. Geneva, Switzerland: IPCC, 2023.

[8] A. Aktan, I. Bartoli, and S. Karaman, "Technology leveraging for infrastructure asset management: Challenges and opportunities," *Frontiers in Built Environment*, vol. 5, p. 61, 05 2019.

[9] M. Piryonesi, T. El-Diraby, and S. Kinawy, "A comprehensive review of approaches used by ontario municipalities to develop road asset management plans," 05 2016.

[10] Federal Highway Administration, "Transportation asset management guide—a focus on implementation: Executive summary," Washington, DC, 2013, fHWA Publication No. FHWA-HIF-13-047. [Online]. Available: https://www.fhwa.dot.gov/asset/pubs/hif13047.pdf

[11] European Environment Agency, "Economic losses from weather- and climate-related extremes in europe," https://www.eea.europa.eu/en/analysis/indicators/economic-losses-from-climate-related, 2024, accessed: 2025-04-24.

[12] K. C. Sinha, S. Labi, and B. R. D. Agbelie, "Transportation infrastructure asset management in the new millennium: continuing issues, and emerging challenges and opportunities," *Transportmetrica A Transport Science*, vol. 13, no. 7, pp. 591–606, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2324993522000987

[13] H. J. Liu, P. E. Love, J. Zhao, C. Lemckert, and K. Muldoon-Smith, "Transport infrastructure asset resilience: Managing government capabilities," *Transportation Research Part D: Transport and Environment*, vol. 100, p. 103072, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920921003692

[14] J. J. Bittner and H. Rosen, "Transportation asset management overview," *Public Works Management & Policy*, vol. 8, pp. 151 – 155, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:110776650

[15] F. Brighenti, V. F. Caspani, G. Costa, P. F. Giordano, M. P. Limongelli, and D. Zonta, "Bridge management systems: A review on current practice in a digitizing world," *Engineering Structures*, vol. 321, p. 118971, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141029624015335

[16] Z. Mirzaei, "Overview of existing bridge management systems - report by the iabmas bridge management committee," 06 2014.

[17] W. Zhang, I. Somerville, G. Paneiro, X. Nong, M. Chwala, and W. Yang, "Design and construction of tunnels and tunnelling: Understanding the importance of geological conditions, landslide susceptibility and risk assessment," *Geological journal.*, vol. 59, no. 9, 2024.

[18] L. Huang, J. Ma, M. Lei, L. Liu, Y. Lin, and Z. Zhang, "Soil-water inrush induced shield tunnel lining damage and its stabilization: A case study," *Tunnelling and Underground Space Technology*, vol. 97, p. 103290, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0886779819301324

[19] J. Wang, S. Zhang, H. Guo, Y. Tian, S. Liu, C. Du, and J. Wu, "Stereoscopic monitoring of transportation infrastructure," *Automation in Construction*, vol. 164, p. 105472, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580524002085

[20] T. Blumenfeld, M. Stöckner, L. Liu, R. Hajdin, M. König, and K. Gavin, "Concepts for the integration of data from asset management systems into bim," *Transportation Research Procedia*, vol. 72, pp. 3738–3745, 2023, tRA Lisbon 2022 Conference Proceedings Transport Research Arena (TRA Lisbon 2022),14th-17th November 2022, Lisboa, Portugal. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146523008426

[21] M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi, "A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future," *IEEE Access*, vol. 11, pp. 17 471–17 484, 2023.

[22] M. M. Rosso, G. Marasco, S. Aiello, A. Aloisio, B. Chiaia, and G. C. Marano, "Convolutional networks and transformers for intelligent road tunnel investigations," *Computers Structures*, vol. 275, p. 106918, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579492200178X

[23] Ministero delle Infrastrutture e della Mobilità Sostenibili, "Linee guida per la classificazione e gestione del rischio, la valutazione della sicurezza ed il monitoraggio delle gallerie esistenti (guidelines for risk classification and management, safety assessment, and monitoring of existing tunnels)," Consiglio Superiore dei Lavori Pubblici, Rome, Italy, Tech. Rep. Parere n. 29/2022, 2022, espresso dall'Assemblea Generale in data 08.04.2022.

[24] Y.-J. Cha and Z. Wang, "Unsupervised novelty detection–based structural damage localization using a density peaks-based fast clustering algorithm," *Structural Health Monitoring*, vol. 17, no. 2, pp. 313–324, 2018.

[25] Z. Wang and Y.-J. Cha, "Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage," *Structural Health Monitoring*, vol. 20, no. 1, pp. 406–425, 2021.

[26] M. H. Rafiei and H. Adeli, "A novel unsupervised deep learning model for global and local health condition assessment of structures," *Engineering Structures*, vol. 156, pp. 598–607, 2018.

[27] Z. Wang and Y.-J. Cha, "Unsupervised machine and deep learning methods for structural damage detection: a comparative study," *Engineering Reports*, vol. 7, no. 1, p. e12551, 2025.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[33] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 2015, pp. 730–734.

[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[38] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[42] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE/CVF conference on computer vision and*

*pattern recognition*, 2019, pp. 6409–6418.

[43] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021.

[44] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4974–4983.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[46] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9799–9808.

[47] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural information processing systems*, vol. 33, pp. 17 721–17 732, 2020.

[48] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 282–298.

[49] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[50] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, "Instances as queries," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6910–6919.

[51] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1280–1289.

[52] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," pp. 248–255, 2009.

[54] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[55] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[56] Y.-J. Cha, R. Ali, J. Lewis, and O. Büyükztürk, "Deep learning-based structural health monitoring," *Automation in Construction*, vol. 161, p. 105328, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580524000645

[57] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, pp. 154–171, 2013.

[58] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[59] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 441–450.

[60] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[63] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[64] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.

[65] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[66] J. Lifshitz and A. Rotem, "Determination of reinforcement unbonding of composites by a vibration technique," *Journal of Composite Materials*, vol. 3, no. 3, pp. 412–423, 1969. [Online]. Available: https://doi.org/10.1177/002199836900300305

[67] J. K. Vandiver, "Detection of structural failure on fixed platforms by

measurement of dynamic response," *Journal of Petroleum Technology*, vol. 29, no. 03, pp. 305–310, 03 1977. [Online]. Available: https://doi.org/10.2118/5679-PA

[68] M. Yuen, "A numerical study of the eigenparameters of a damaged cantilever," *Journal of Sound and Vibration*, vol. 103, no. 3, pp. 301–310, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022460X85904237

[69] L. Niu and L. Ye, "Use of neural networks in damage detection of strutures," in *2009 International Conference on Electronic Computer Technology*, 2009, pp. 258–261.

[70] G. V. Garcia, N. Stubbs, and K. Butler, "Relative performance evaluation of pattern recognition models for nondestructive damage detection," in *Smart Structures and Materials 1996: Smart Systems for Bridges, Structures, and Highways*, vol. 2719. SPIE, 1996, pp. 25–35.

[71] J. T. Yao and H. Natke, "Damage detection and reliability evaluation of existing structures," *Structural Safety*, vol. 15, no. 1-2, pp. 3–16, 1994.

[72] F. L. K. Wan, "Genetic algorithms, their applications and models in nonlinear systems identification," Ph.D. dissertation, University of British Columbia, 1991.

[73] Y.-J. Cha and O. Buyukozturk, "Structural damage detection using modal strain energy and hybrid multiobjective optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 5, pp. 347–358, 2015.

[74] S. Rama Krishna, J. Sathish, M. Tarun, V. Sruthi Jones, S. Raghu Vamsi, and S. Janu Sree, "A support vector machine-based intelligent system for real-time structural health monitoring of port tower cranes," *Journal of Failure Analysis and Prevention*, vol. 24, no. 6, pp. 2543–2554, 2024.

[75] C. R. Farrar, W. Baker, T. Bell, K. Cone, T. Darling, T. Duffey, A. Eklund, and A. Migliori, "Dynamic characterization and damage detection in the i-40 bridge over the rio grande," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 1994.

[76] G. Stephen, J. Brownjohn, and C. Taylor, "Measurements of static and dynamic displacement from visual monitoring of the humber bridge," *Engineering Structures*, vol. 15, no. 3, pp. 197–208, 1993.

[77] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of computing in civil engineering*, vol. 17, no. 4, pp. 255–263, 2003.

[78] Y.-J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the hough transform and support vector machines," *Automation in Construction*, vol. 71, pp. 181–188, 2016.

[79] S. Patsias and W. Staszewskiy, "Damage detection using optical measurements and wavelets," *Structural Health Monitoring*, vol. 1, no. 1, pp. 5–22, 2002.

[80] J. G. Chen, N. Wadhwa, Y.-J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk, "Modal identification of simple structures with high-speed video using motion magnification," *Journal of Sound and Vibration*, vol. 345, pp. 58–71, 2015.

[81] D. Feng, M. Q. Feng, E. Ozer, and Y. Fukuda, "A vision-based sensor for noncontact structural displacement measurement," *Sensors*, vol. 15, no. 7, pp. 16 557–16 575, 2015.

[82] Z. Wang, H. Kieu, H. Nguyen, and M. Le, "Digital image correlation in experimental mechanics and image registration in computer vision: Similarities, differences and complements," *Optics and Lasers in Engineering*, vol. 65, pp. 18–27, 2015.

[83] D. Kang and Y.-J. Cha, "Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 10, pp. 885–902, 2018.

[84] V. Hoskere, Y. Narazaki, T. Hoang, and B. Spencer Jr, "Vision-based structural inspection using multiscale deep convolutional neural networks," *arXiv preprint arXiv:1805.01055*, 2018.

[85] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, "Underground sewer pipe condition assessment based on convolutional neural networks," *Automation in Construction*, vol. 106, p. 102849, 2019.

[86] A. S. Rao, T. Nguyen, M. Palaniswami, and T. Ngo, "Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure," *Structural Health Monitoring*, vol. 20, no. 4, pp. 2124–2142, 2021.

[87] D. Li, A. Cong, and S. Guo, "Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification," *Automation in Construction*, vol. 101, pp. 199–208, 2019.

[88] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[89] F.-C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.

[90] E. E. B. Adam and A. Sathesh, "Construction of accurate crack identification on concrete structure using hybrid deep learning approach," *Journal of Innovative Image Processing (JIIP)*, vol. 3, no. 02, pp. 85–99, 2021.

[91] Y. Li, W. Hu, H. Dong, and X. Zhang, "Building damage detection from post-event aerial imagery using single shot multibox detector," *Applied Sciences*, vol. 9, no. 6, p. 1128, 2019.

[92] R. Ali, J. Zeng, and Y.-J. Cha, "Deep learning-based crack detection in a

concrete tunnel structure using multispectral dynamic imaging," in *Smart Structures and NDE for Industry 4.0, Smart Cities, and Energy Systems*, vol. 11382. SPIE, 2020, pp. 12–19.

[93] A. Semwal, R. E. Mohan, L. M. J. Melvin, P. Palanisamy, C. Baskar, L. Yi, S. Pookkuttath, and B. Ramalingam, "False ceiling deterioration detection and mapping using a deep learning framework and the teleoperated reconfigurable 'falcon'robot," *Sensors*, vol. 22, no. 1, p. 262, 2021.

[94] R. Ali, D. Kang, G. Suh, and Y.-J. Cha, "Real-time multiple damage mapping using autonomous uav and deep faster region-based neural networks for gps-denied structures," *Automation in Construction*, vol. 130, p. 103831, 2021.

[95] C. M. Yeum, J. Choi, and S. J. Dyke, "Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure," *Structural Health Monitoring*, vol. 18, no. 3, pp. 675–689, 2019.

[96] C. Zhang, C.-c. Chang, and M. Jamshidi, "Concrete bridge surface damage detection using a single-stage detector," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 4, pp. 389–409, 2020.

[97] J. C. Cheng and M. Wang, "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques," *Automation in Construction*, vol. 95, pp. 155–171, 2018.

[98] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[99] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[100] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[101] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[102] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[103] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[104] Y. Dong, J. Wang, Z. Wang, X. Zhang, Y. Gao, Q. Sui, and P. Jiang, "A deep-learning-based multiple defect detection method for tunnel lining damages," *IEEE Access*, vol. 7, pp. 182 643–182 657, 2019.

[105] S. Zhao, D. Zhang, Y. Xue, M. Zhou, and H. Huang, "A deep learning-based approach for refined crack evaluation from shield tunnel lining images," *Automation in Construction*, vol. 132, p. 103934, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092658052100385X

[106] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.

[107] S. Zhao, D. M. Zhang, and H. W. Huang, "Deep learning–based image instance segmentation for moisture marks of shield tunnel lining," *Tunnelling and Underground Space Technology*, vol. 95, p. 103156, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0886779819301452

[108] F. Liu and L. Wang, "Unet-based model for crack detection integrating visual explanations," *Construction and Building Materials*, vol. 322, p. 126265, 2022.

[109] A. Ji, A. W. Z. Chew, X. Xue, and L. Zhang, "An encoder-decoder deep learning method for multi-class object segmentation from 3d tunnel point clouds," *Automation in Construction*, vol. 137, p. 104187, 2022.

[110] D. Xi, Y. Qin, and S. Wang, "Ydrsnet: An integrated yolov5-deeplabv3+ real-time segmentation network for gear pitting measurement," *Journal of Intelligent Manufacturing*, vol. 34, no. 4, pp. 1585–1599, 2023.

[111] G. H. Beckman, D. Polyzois, and Y.-J. Cha, "Deep learning-based automatic volumetric damage quantification using depth camera," *Automation in Construction*, vol. 99, pp. 114–124, 2019.

[112] P. Balasubramanian, V. Kaushik, S. Y. Altamimi, M. Amabili, and M. Alteneiji, "Comparison of neural networks based on accuracy and robustness in identifying impact location for structural health monitoring applications," *Structural Health Monitoring*, vol. 22, no. 1, pp. 417–432, 2023.

[113] SpaceTec, "Ts4 product page," 2024, accessed: 31 March 2025. [Online]. Available: https://www.spacetec.de/en/products/ts4/

[114] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[115] S. Ji and H. Zhang, "ISAT with Segment Anything: An Interactive Semi-Automatic Annotation Tool," 2023, updated on 2023-06-03. [Online]. Available: https://github.com/yatengLG/ISAT_with_segment_anything

[116] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[117] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng,

Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[118] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.

[119] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[120] ——, "Sgdr: Stochastic gradient descent with warm restarts," 2017, Conference paper, cited by: 2118. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081410026&partnerID=40&md5=a79a89fa87e1a475a6a99fb2be4450ea

[121] Z. Ye, L. Lovell, A. Faramarzi, and J. Ninić, "Sam-based instance segmentation models for the automation of structural damage detection," *Advanced Engineering Informatics*, vol. 62, p. 102826, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034624004749

[122] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, "Gradio: Hassle-free sharing and testing of ml models in the wild," *arXiv preprint arXiv:1906.02569*, 2019.

[123] py-pdf organization, *fpdf2: A Python library for generating PDF documents*, py-pdf organization, 2021. [Online]. Available: https://github.com/py-pdf/fpdf2

[124] M. Fenniak, M. Stamy, pubpub zz, M. Thoma, M. Peveler, exiled kingcc, and P. Contributors, *The PyPDF2 library*, PyPDF2 Developers, 2022, see https://pypdf2.readthedocs.io/en/latest/meta/CONTRIBUTORS.html for all contributors. [Online]. Available: https://pypdf2.readthedocs.io/en/3.x/