REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

BRIDGE SURFACE CRACK DETECTION BASED ON ARTIFICIAL INTELLIGENCE TECHNIQUES

Abbas Abdulameer Hameed HAMEED

Master's Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

İstanbul, 2023

BRIDGE SURFACE CRACK DETECTION BASED ON ARTIFICIAL INTELLIGENCE TECHNIQUES

Abbas Abdulameer Hameed HAMEED

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled BRIDGE SURFACE CRACK DETECTION BASED ON ARTIFICIAL INTELLIGENCE TECHNIQUES prepared by ABBAS ABDULAMEER HAMEED HAMEED and submitted on 27/12/2023 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Oğuz KARAN

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbaş University	
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, İstanbul Arel University	

I hereby declare that this thesis meets all format and submission requirements of a Master's Thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Abbas Abdulameer Hameed HAMEED

Signature

DEDICATION

I want to start by expressing my sincere gratitude to Allah Almighty for giving me the mental clarity, health, and fortitude needed to finish this course of study.

I would like to express my sincere gratitude to everyone who has helped me along the way in school, especially my thesis supervisor Asst. Prof. Dr. Oğuz KARAN provided me with tremendous guidance and assistance.

I owe my parents a debt of gratitude for their unwavering love and support throughout my life, from the humble start to this present milestone. The depth of their contributions at every stage of my life cannot be adequately expressed in words. This thesis is dedicated to my parents as a sign of my sincere gratitude for their steadfast assistance and sacrifices, without which I would still be unable to complete this task.

ABSTRACT

BRIDGE SURFACE CRACK DETECTION BASED ON ARTIFICIAL INTELLIGENCE TECHNIQUES

HAMEED, Abbas Abdulameer Hameed

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Oğuz KARAN

Date: December/2023

Pages: 76

The manual inspection of concrete bridge surfaces for cracks takes a long time and wastes a lot of materials and labor. This study examines this problem by offering a quick and automated machine-learning method for visually inspecting concrete bridge surfaces with unmanned aerial vehicles (UAVs) in large open spaces. UAVs are used to capture high-resolution photographs of concrete bridge surfaces, providing thorough coverage of the inspection area. Utilizing the potent YOLO (You Only Look Once) technique, deep learning technology, a subset of machine learning, enables single-shot detectors. Use is made specifically of the YOLO v8 model, which was very accurate machine learning techniques were used to train. Since this model is based on Convolutional Neural Networks (CNNs) with the CSPDarknet-53 backbone structure, accurate and trustworthy results are guaranteed. The suggested model has outstanding performance in machine learning-based real-time crack detection. The Deep Learning model achieves remarkable precision and correctness in crack detection with an accuracy rating of 98.54%. The deep learning model can accurately detect cracks while limiting false positives, according to the F1 score, a measure of precision and recall balance, which reaches 97%. The robustness and accuracy of the machine learning model are demonstrated by the mean average precision (mAP), a statistic frequently employed in machine learning, which is calculated to be 97.90%. With a mAP50-95 of 83.11%, the model also achieves a high mAP across various Intersections over Union (IoU) thresholds. The recall rate reaches 94.66%, ensuring a high detection rate for

true cracks, while the precision rate measures an astonishing 99.41%, minimizing false alarms. Additionally, the model runs at a phenomenal 95.24 frames per second (fps), which makes it possible to use machine-learning approaches for effective real-time crack detection. This speed suits it for practical applications since it enables prompt decision-making and intervention. Our study suggests a new, effective machine-learning method for instantly detecting cracks in concrete bridge surfaces. Utilizing image collecting from UAVs, With the help of the YOLO algorithm.

Keywords: Crack detection, Machine learning, Deep Learning, Unmanned Aerial Vehicles (UAVs), Convolutional Neural Networks (CNNs), YOLO v8.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
CV	:	Computer Vision
ML	:	Machine Learning
DL	:	Deep Learning
CNN	:	Convolutional Neural Network
UAV	:	Unmanned Aerial Vehicle
YOLO	:	You Only Look Once
GPU	:	Graphics Processing Unit
DCNN	:	Deep Convolutional Neural Network Region-
RCNN	:	Based Convolutional Neural Network Support
SVM	:	Vector Machine

1. INTRODUCTION

1.1 INTRODUCTION

An essential part of guaranteeing bridges' structural reliability and dependability is carrying out a thorough visual inspection to identify and detect surface problems. The terrible and tragic event of the recent collapse of the Morandi bridge serves as stark evidence that skipping this important practice can have far-reaching, disastrous effects [1]. Almost 67,000 of the 607,380 bridges that are now in use in the United States have been determined to have structural flaws, according to readily available data. The authorities in charge of guaranteeing appropriate transportation infrastructure face tremendous difficulty as a result of the approximately 85,000 bridges that have been considered functionally obsolete. According to the National Research Council of Canada, a startling one-third of Canada's highway bridges have various degrees of structural or functional defects, reducing their remaining service life and highlighting the need for quick repair action [2].

Figure 1.1: Morandi Bridge Collapse Incident.

Currently, manual inspection chores are routinely carried out by inspectors, a laborious and time-consuming practice that is extremely prevalent. But recently, there has been a noticeable shift in this pattern, with a rise in the use of unmanned aerial vehicles (UAVs) for inspection tasks [3]. Drones can significantly shorten inspection times while maintaining the

security of remote locations. Transportation authorities in numerous nations have begun to use UAV-based bridge inspection procedures because they are effective, quick, safe, and economical [4]. However, Bridge inspection with human eyes is difficult to obtain accurate information like the length, depth, and width of cracks. This is principally caused by the intrinsic variation in scale and perspective, the unpredictable and arbitrary nature of changes in lighting, and the emergence of overlapping faults that reduce the clarity of the image [5]. There are so much non-human factors, for instance, illumination levels, noises resulted by UAVs, an uneven road surface, and vibration of UAVs. Due to these factors, images acquired by UAVs are usually of low quality.

1.2 BACKGROUND AND MOTIVATION

To guarantee stability and safety, concrete bridge surfaces must undergo routine inspection and maintenance [6]. Traditional manual inspection techniques waste a lot of time, labor, and materials while also frequently failing to find cracks in concrete bridge surfaces. Due to the size, complexity, and occasionally inaccessibility of bridge structures, these methods usually involve human inspectors physically inspecting the bridge surface [7]. Unmanned aerial vehicle (UAV) technological advances are developing quickly, and this presents an opportunity to completely change how concrete bridge surfaces are inspected. the potential applications of UAVs in several engineering domains have received colossal attention, such as monitoring pavements and highways, an inspection of buildings, maintenance of facades, and monitoring bridge defects and cracking. These images can be automatically analyzed to find cracks using computer vision and deep neural networks, offering quicker, more effective, and less expensive alternatives to manual inspection techniques [8]. This study was inspired by the need to address the shortcomings of manual crack detection in concrete bridge surfaces. This research aims to overcome the difficulties associated with conventional inspection methods by proposing a quick and automated approach using UAVs and deep learning technology. The goal is to create a system for crack detection that is accurate and dependable and can quickly locate and analyze cracks in real-time. The proposed crack detection system is built on the YOLO (You Only Look Once) algorithm, specifically the YOLO v8 model. In terms of real-time performance and high accuracy in object detection tasks, YOLO is renowned. The YOLO v8 model is chosen for its accuracy and dependability in detecting cracks in concrete bridge surfaces because it is based on Convolutional Neural

Networks (CNNs) with the CSPDarknet-53 backbone structure. The expected results of this study will have a big impact on how concrete bridge surfaces are maintained and inspected. The suggested method could make bridge inspection procedures more productive, require fewer staff members, and be safer. Resource allocation can be made more efficient by performing crack detection through UAV-based visual inspection, enabling early intervention and maintenance activities. The ultimate goal of this research is to advance infrastructure inspection techniques and guarantee the durability and safety of concrete bridge surfaces.

1.3 PROBLEM STATEMENT

At present, most of the investigation of cracks in engineering structures relies on the traditional manual identification, marking and recording, which is not only inefficient and requires a lot of human resources, but also relies excessively on manual experience. Additionally, a variety of circumstances, some of which include weariness and subjectivity, frequently impede the visual examinations performed by human operators. These reasons can all lead to inaccuracies. Additionally, it is extremely difficult to maintain and repair vital infrastructure effectively due to the lengthy amount of time required to do these manual examinations. such techniques take a long time and frequently lead to late evaluations of cracks close to the unrepairable zone. Modern techniques try to remedy this by substituting manual operations with more advanced, automated crack detection to speed up the process [9]. The inherent subjectivity that accompanies visual inspections of concrete bridge surfaces serves to introduce a multitude of inconsistencies in the process of recognizing and evaluating cracks. over four billion square meters of reinforced concrete bridge surface must be periodically inspected by inspectors, which presents a problem. As a result, there is a growing need for artificial intelligence (AI) to automate infrastructure construction, inspection, and planning [10], [11]. To overcome these difficulties, this study suggests an innovative and effective method for the visual examination of unmanned aerial vehicles (UAVs) in open areas for the automatic detection of cracks in concrete bridge surfaces. Comprehensive coverage of the inspection area can be achieved by using UAVs to capture high-resolution photos of the bridge surfaces. Utilizing the potent YOLO (You Only Look Once) technique, deep learning technology is used to enable single-shot object detector. Convolutional Neural Networks (CNNs) based on the CSPDarknet-53 backbone architecture

are used with the YOLO v8m model, which has been trained for excellent accuracy, to ensure accurate and trustworthy results. This suggested method seeks to improve the crack detection procedure by utilizing the strengths of UAVs and deep learning. It offers real-time detection capabilities, excellent precision, and a reduced risk of false positives. The findings of this study have important ramifications for enhancing the effectiveness, safety, and resource use in the upkeep and inspection of concrete bridge surfaces.

1.4 IMPORTANCE OF CRACK DETECTION IN CONCRETE BRIDGE

Bridges are important transportation infrastructures, and the role of crack detection in evaluating bridge conditions has received increased attention across several disciplines in recent years. To keep concrete bridges structurally sound and reliable, surveys of the bridge's state are crucial [12]. For the sake of preserving public safety and reducing the possibility of catastrophic failures, these bridges' dependability and structural integrity are crucial. The surface is one of the many parts that make up a bridge, and it is essential in supporting the weight and giving both vehicles and pedestrians a smooth surface [13]. With significant effects on structural integrity and public safety, early crack diagnosis is a crucial part of bridge maintenance. When cracks in the bridge's surface go unnoticed and unrepaired, the results could be disastrous. Bridges are essential elements of the transportation infrastructure from a structural standpoint, facilitating the passage of both automobiles and pedestrians. Unchecked cracks that compromise their integrity can result in accidents, disruption of traffic, or, in the worst case, catastrophic breakdowns. These fissures are susceptible to water and corrosive substances that can penetrate the steel reinforcements and rods below. This infiltration can hasten the corrosion process, undermine the internal support system of the bridge, and necessitate lengthy and pricey repairs and maintenance [14].

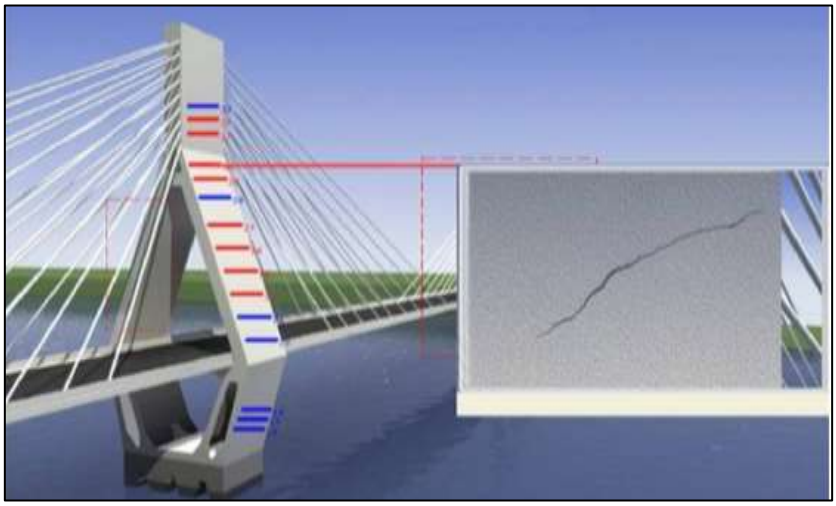


Figure 1.2: Bridge Crack Example.

1.4.1 Importance Of Bridge Health Maintenance for Safety and Longevity

The weighing of vehicles for enforcement first began in the United Kingdom in 1741 when the Turnpike Act was introduced. This act decreed that tolls were to be paid for using the roads according to the weight of the vehicle. The money raised from these tolls was to be used for the maintenance of the roads, similar to how revenue raised from heavy or overweight vehicles is used today (Sanders 2010). Over the past 30 years, bridge construction has advanced quickly due to the ongoing advancement of science and technology [15]. The structural health of bridges must be evaluated through the process of fatigue analysis, which takes load and resistance into account. The actual weight exerted by traffic vehicles, however, frequently deviates greatly from the designed load. Having a precise evaluation process for determining the load is important to precisely calculate the remaining fatigue life of bridges [16]. Gaining knowledge about how bridge components, like girders and surfaces, behave and determining their fatigue life are the goals. Bridge weigh-in-motion (WIM) is a method by which the axle weights of a vehicle traveling at full highway speed can be determined using a bridge instrumented with sensors. Since the sensors are attached to the underside of a bridge, the instrumentation can be installed without disruption to traffic. The vehicle may be guided by curbs to minimize variation in the transverse position of the wheels. The data processing system analyzes the signal from the load cells and takes the vehicle speed into account to accurately calculate wheel or axle

loads. The collected results from this analysis are then used in the fatigue analysis to determine how long the bridge elements will remain fatigue-free [17].

1.4.2 Consequences Of Undetected or Untreated Cracks on Structural Integrity

Structures' dependability and safety are significantly threatened as time goes on by unnoticed or untreated cracks. Every nation's transportation system is severely impacted by bridge failure. In addition to injuries and fatalities, service interruption has seriously detrimental consequences on economic growth [18]. Such mistakes can have severe repercussions, such as the loss of lives, significant property damage, and high financial expenditures [19]. Prioritizing the development and application of efficient techniques for crack detection and repair is essential for ensuring the long-term dependability and safety of structures. Early identification enables prompt action, which stops the cracks from spreading and reduces the risks they bring with them. The structural integrity is restored and further deterioration is stopped through proactive restoration techniques [20]. We may improve our capacity to spot cracks at their earliest stages by investing in research and technology developments, such as sophisticated monitoring systems, non-destructive testing methods, and analysis based on artificial intelligence. Furthermore, the creation of strong repair techniques and materials guarantees that discovered cracks may be successfully repaired, restoring the infrastructure's structural integrity and extending its lifespan. A comprehensive strategy to crack diagnosis and repair is essential for structures to remain structures are to continue being dependable and secure. To reduce the likelihood of failure and the hazards that go along with it, standard processes should incorporate frequent inspections, cutting-edge systems for monitoring, and proactive maintenance techniques. Not only does prioritizing crack detection and repair safeguard public safety, but it also saves priceless infrastructure investments, eases financial pressures, and promotes resilient and sustainable communities.

1.4.3 Need For Efficient and Reliable Crack Detection Methods

Cracks can develop in concrete structures for several reasons, including the environment, getting older, and structural deterioration. These fissures provide considerable safety issues and may lead to structural collapse risks. To guarantee the structural integrity of the building, it is essential to quickly locate and fix these fissures. Crack identification has traditionally relied on manual examination by trained human inspectors, which can be a laborious and

error-prone process [21]. Automated crack detection techniques based on image processing have been developed to get around these restrictions [22]. These techniques analyze images more quickly and objectively by using sophisticated algorithms and computer vision techniques. However, a lot of automated crack detection techniques currently in use prioritize accuracy over computation time, which is also crucial for real-world applications. Crack detection methods that not only deliver accurate results but also lighten the computational load are increasingly needed to meet this challenge [23]. The crack diagnosis method presented in this study is very novel and represents a substantial improvement in the area by utilizing crack detection-based computer vision techniques. This method's major goal is to achieve a careful balance between speed and accuracy, ensuring effective crack detection without sacrificing the results' accuracy. Compared to conventional approaches, this unique approach has several benefits. The first benefit is a quicker examination of large-scale images, allowing for the effective processing of enormous volumes of data in a shorter length of time. This is especially useful for conducting broad bridge network inspections or managing time-sensitive maintenance tasks. The cost-effectiveness of crack diagnosis is increased by using crack detection-based computer vision techniques [24]. By reducing the need for manual examination and arbitrary interpretation of visual data, the technique minimizes human error and boosts overall effectiveness. The expenditures associated with lengthy and needless interventions are decreased as a result of more focused maintenance and repair activities [25]. This research provides a new crack detection-based computer vision system that significantly advances crack detection. By striking a balance between quickness and accuracy.

1.5 GOAL OF THE RESEARCH

This study uses unmanned aerial vehicles (UAVs) and deep learning technologies to suggest a quick and automated method for finding cracks in concrete bridge surface. To improve the safety and maintenance of concrete bridge surfaces, it is intended to overcome the time-consuming nature of human inspections, decrease resource waste, and improve the accuracy and effectiveness of crack detection. This research suggests a novel strategy that makes use of unmanned aerial vehicles (UAVs) for visual examination of concrete bridge surfaces in open areas to overcome these difficulties. UAVs are used to take high-resolution pictures of the bridge surfaces, enabling thorough coverage and in-depth analysis of the inspection area.

Automated crack detection uses deep learning technology, more especially the YOLO (You Only Look Once) algorithm. Accurate and trustworthy results are obtained by using the YOLO v8m model, which is based on Convolutional Neural Networks (CNNs) using the CSPDarknet-53 backbone structure. The proposed method seeks to transform the crack detection procedure by merging UAV-based image gathering with the potent YOLO algorithm. By doing away with manual inspections, this automated method conserves time and cuts down on resource waste. Deep learning technology is used to maximize the effectiveness of bridge surface repair and maintenance while ensuring high accuracy in crack detection. The findings of this study could improve safety, lower costs, and increase the overall reliability of concrete bridge surfaces, which has important ramifications for the construction sector, transportation organizations, and infrastructure management.

1.6 THESIS STRUCTURE

Chapter 1: Introduction This section gives a brief overview of the thesis and summaries of each chapter.

Chapter 2: Theoretical Background This chapter provides an overview of fundamental concepts and examines relevant earlier research about the topic of the thesis. It is important to use deep and machine learning approaches to detect surface cracks in bridge constructions since they play a crucial part in preserving the structural integrity and safety of concrete bridges.

Chapter 3: Methodology The method used in this thesis is theoretically explained in this chapter. It describes the structure of the technique; the framework suggested for its application and subsequent comparison and offers recommendations for possible sub-methods that could be used to carry out the method. The chapter also provides information about how to provide this dataset and describes how to use a dataset that contains these detections for training and evaluating the suggested strategy.

Chapter 4: Result In this chapter, we provide the results obtained from the algorithm training process in detail, followed by a thorough analysis and comparison of these results.

Chapter 5: Discussion and Conclusion With a summary and inferences, the last chapter summarizes the thesis. This chapter includes an in-depth discussion of the findings, an

examination of the conclusions and related issues, and an outline of potential directions for further investigation of the subjects touched upon.

2. LITERATURE REVIEW

2.1 INTRODUCTION

Throughout its existence, structures like bridges are susceptible to a variety of loads, including self-weight, service loads, and environmental pressures. These loadings could cause structural damage and, in extreme circumstances, result in structural failure. It is critical to understand how these various loadings might affect bridges and damage their integrity, perhaps leading to a catastrophic collapse. Therefore, a thorough understanding of the effects of various loadings and the implementation of suitable maintenance and inspection protocols are required to ensure the safety and stability of bridges [26]. Cracked reinforced concrete structures pose the possibility of catastrophic failure, among other terrible consequences. The main cause of this is that cracks allow corrosive materials, like water, to penetrate the concrete, as shown in Figure 2. 1, resulting in structural damage and quick deterioration [27]. Professional inspectors must physically visit the site to conduct manual visual inspections and other empirical methods for evaluating the state of bridges. These inspectors physically stamp cracks to detect them and take pictures of the flaws for photographic proof [28], [29]. However, there are several drawbacks to depending solely on physical bridge inspection to find cracks. It is a laborious, subjective process that heavily relies on the inspector's expertise. Additionally, it offers qualitative analyses and struggles to assess regions that are difficult to reach or efficiently inspect [30]. Additionally, the physical inspection procedure can put inspectors in danger of injury, especially if the bridge is challenging to access. Additionally, the inspector's level of expertise and subjective judgment frequently affect the results' accuracy. As a result, the development of an automated flaw detection system that can mimic onsite inspection by humans is urgently needed. To address the drawbacks of manual inspection, computer vision (CV) has recently been incorporated into structural health monitoring systems [31]. The goal of our research project was to create autonomous crack detection technologies that would greatly enhance the precision and effectiveness of asset management procedures. We used unmanned aerial vehicles (UAVs) in our system to get beyond the drawbacks of human inspection techniques. The exploration and application of cutting-edge image-based approaches for effective and precise bridge crack identification was a key component of our research. Notably, we tapped into the potential of cutting-edge deep learning techniques and deep neural networks,

utilizing their capacities to significantly improve the efficiency and performance of our crack-detecting system.

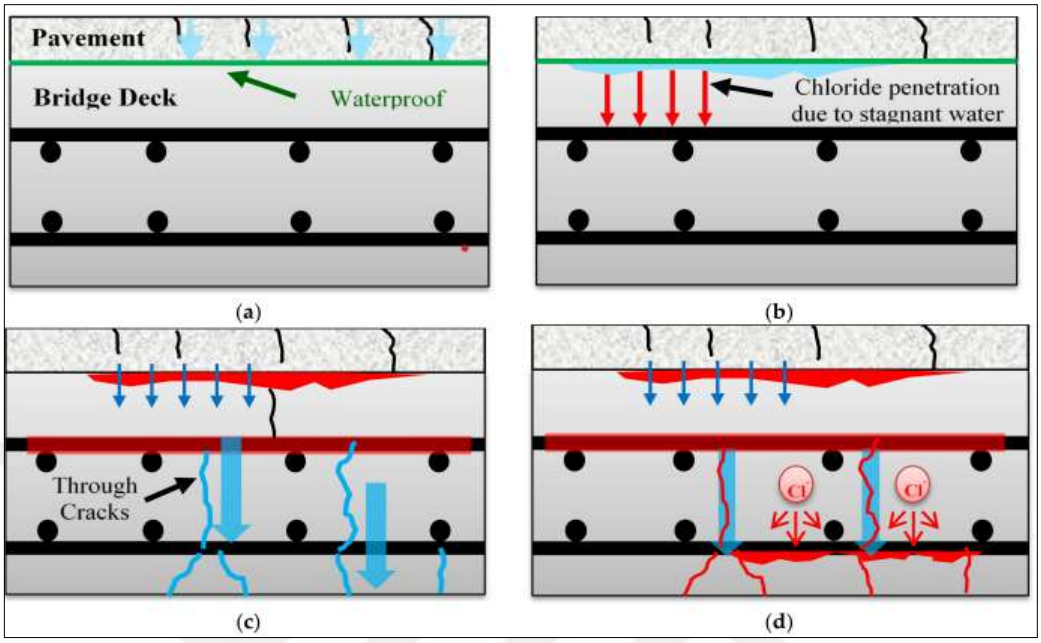


Figure 2.1: Composition and Mechanism of Bridge Deck Deterioration. (a) Rainwater Infiltration Brought on by First Cracks and Harm to the Waterproof Layer (b) Chloride Infiltration Brought on by Salty, Stagnant Water (c) Water Leakage to the Deck Bottom Surface (d) Damage to the Deck Bottom Surface and Corrosion of the Bottom Steel Reinforcement.

2.2 CHALLENGES OF MANUAL CRACK DETECTION IN CONCRETE BRIDGE

When testing the structural integrity of reinforced concrete (RC), cracks must be carefully observed. Researchers can gain important insights on a variety of topics, including failure mechanisms, the deterioration of a structure's stiffness and strength, and other critical information, by examining the distribution of cracks in terms of their directions, patterns, and density. These findings are essential for carrying out failure mode analyses [32] and creating numerical models [33], [34]. In the past, inspectors have spotted cracks by stopping a test to walk up to the concrete surfaces and physically draw lines on the fissures Figure 2. 2 However, this approach can be riskier than it appears. The test structures are harmed and just slightly stable. Specimens are typically tensioned, compressed, bent, or distorted during test suspension for crack observations, with enormous internal forces keeping fissures open.

Even if they appear to be paused, hydraulic actuators that apply forces to specimens are typically still running control loops, gently vibrating, and adjusting tons of applied forces on the specimens. Helmets cannot guarantee human safety if any specimen, experimental equipment, or connected component becomes unpredictably unstable. Image analysis has become a viable substitute for structural experimental measurements due to the quick development of digital imaging technology and the falling cost of hardware. With this method, displacements, strain fields, cracks, and even dynamic motions can be recorded and measured. In the literature now in existence, numerous crack-detecting techniques have been proposed. The majority of these techniques are based on edge and crack detection techniques, which find and quantify pronounced fissures in images that appear as dark shadow lines [35]. Enhancing crack depth prediction [36], finding cracks without picture registration [36], employing artificial neural networks to recognize crack patterns [37], investigating applications to micro-cracks in rocks [38], and attaining efficient sub-pixel width measurement [39] have been the main topics of recent studies.

Figure 2.2: Cracks Manually Marked Using a Pen.

2.2.1 Visual Inspection

Visual testing is very important since it can give skilled people useful information. It makes it possible to recognize visual characteristics linked to craftsmanship, structural functionality, and material degradation. Engineers must be able to accurately differentiate

between several distress indicators, such as cracks, pop-outs, spalling, disintegration, color changes, weathering, staining, surface defects, and a lack of uniformity. A thorough visual inspection can help acquire important information that will help determine the structure's condition and direct the creation of a future testing strategy [40]. While useful, visual inspection is not the only criterion used to evaluate the state of a structure. It can offer some useful information, but its importance shouldn't be overestimated. Although visual signs of workmanship, structural viability, and deterioration of materials can be seen, they do not give a complete picture of the structure's state. It can be difficult and prone to interpretation biases to distinguish between different symptoms of distress, including cracks, pop-outs, spalling, collapse, color change, weathering, staining, surface defects, and lack of homogeneity. Visual inspection can provide a basic indication of the structure's condition, but other testing techniques should be added for a more accurate assessment.

a. Equipment And Tools for Visual Inspection

Engineers doing visual surveys require the proper equipment to facilitate their checks. Measurement tapes, markers, thermometers, anemometers, and other typical accessories are included in this group of tools. Binoculars, scopes, borescopes, endoscopic instruments, or fiber scopes may be useful in locations that are challenging to access. While a magnifying glass or portable microscope allows for a closer view, a crack-width microscope or gauge can be used to measure crack widths. The documenting of faults through photography is made easier by a camera with zoom and microlenses, as well as add-ons like polarizing filters. Variations in concrete color can be distinguished using a portable color chart. To accurately record observations, pertinent drawings including plan ideas, elevations, and details of the structure are also necessary.

2.2.2 Non-Destructive Testing (NDT)

Numerous crack detection techniques have been investigated as a result of research that has focused on the identification of cracks in reinforced concrete. Acoustic techniques, which can be divided into resonance, ultrasound-pulse, surface wave, and acoustic emission methods, have shown the most promise of all of these methods [41]. The impact repeat and ultrasonic pulses (echo or transmission) technologies need special attention since they have the greatest potential for identifying crack depths [42]. NDT can be performed at different

stages of the structure's construction process to ensure the materials and work meet quality standards. A few of the many benefits of NDT are product integrity and reliability assurance, manufacturing.

process control, lower production costs and consistent quality level assurance. Concrete is a material that may be created from several different types of ingredients, and the way it is treated on the construction site determines how it will ultimately perform [43].

a. Hammer Sounding

Impact hammer testing is a regular structure inspection method for detecting surface and internal damages. Inspectors use the sound from impact hammer testing to determine the damaged area. However, manual impact hammer testing cannot meet the reliable accuracy for small damages, such as concrete cracks [44]. Figure 2. 3 illustrates how concrete can be evaluated simply and effectively by tapping or sounding it with a hammer. The concrete surface can be hammered to reveal important details about its state. A loud and resonant ringing sound is produced when a hammer makes contact with the sound and intact concrete. The procedure of the hammer test starts with the plunger being pressed against the material being tested. After the plunger is released, the impact spring releases the mass powered with constant energy. The mass impacts the material being tested then rebounds, providing a result on the window and scale. This ringing sound is a sign indicating the concrete is sound and free from any major flaws. The sound produced by the hammer, however, is more akin to a drum when it strikes concrete that has cracks in it. This drum-like sound serves as a warning sign for concrete fissures or structural problems.

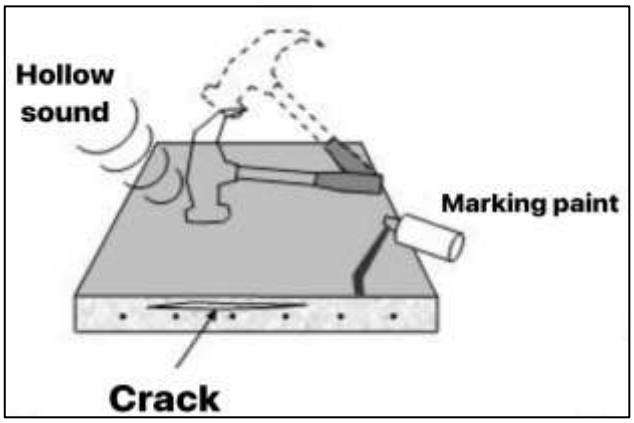
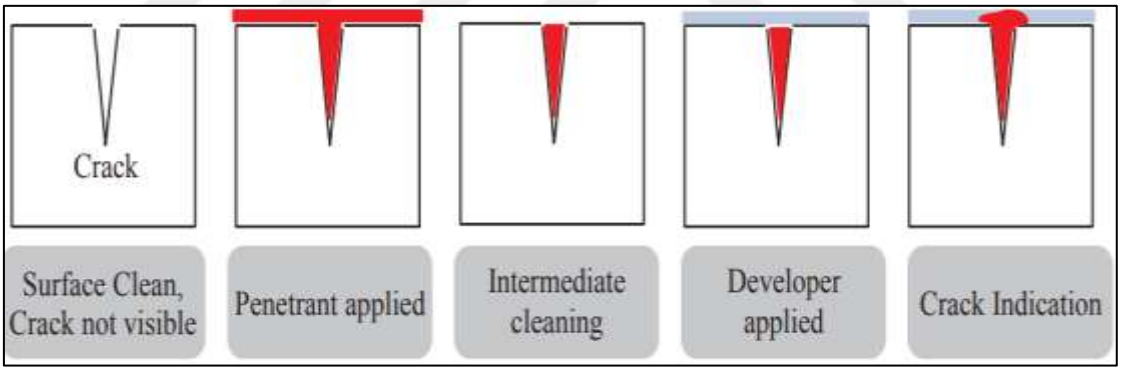


Figure 2.3: Comparing Sound Characteristics in Concrete Evaluation Through Hammer Sounding.

b. Dye Penetrant Testing

Particle testing, another name for dye penetrant testing, is a technique used by inspectors to find and assess small cracks in concrete bridges. The inspector applies a dye penetrant with a striking contrast during this procedure to spot any surface faults or inconsistencies [45]. Although surface preparation is required, basic crack indicators like dye penetrants and magnetic particles can be used [46]. The cleaning, the penetrant, and the developer are all present in the kit that typically contains the penetrant dye along with the other necessary chemicals in three aerosol spray cans. Each can correspond to a particular stage in the process [47]. Additionally, dye penetrant testing (DPT) was carried out to look for stud fatigue cracks. The concrete surface was cleaned before the DPT procedure began, and a penetrant was added, giving it time to work its way into any unnoticed surface cracks. After applying the dye, the surface was washed to remove any extra penetrant, and a developer solution was then applied to bring the penetrant to the surface, making the previously undetectable surface imperfections obvious through visual inspection [48]. In Figure 2. 4, these DPT phases are shown.

a)



b)

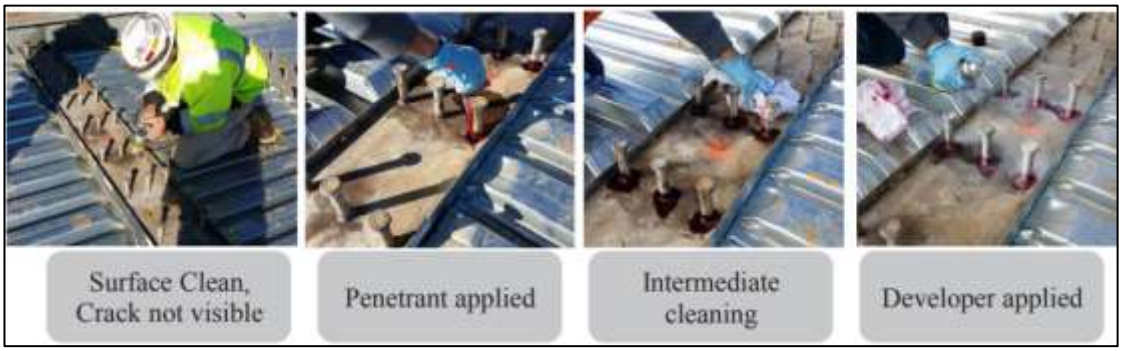


Figure 2.4: Depicts the Dye Penetrant Testing Procedure, including (a) A Breakdown of the Various Processes and (b) An Example of Dye Penetrant Testing Being Used on a Bridge.

2.3 UNMANNED AERIAL VEHICLES (UAVS) IN BRIDGE INSPECTION

According to FHWA and AASHTO regulations, roping rigs, scaffolding, or "snooper trucks" are frequently used to get inspectors of bridge elements, even though doing so could put them at risk. It may take two weeks or longer to set up, check, and take down the equipment for extensive inspections, and additional time will be needed for reporting at the office [49]. Recently, remote sensing technology based on unmanned aerial vehicles (UAVs) has become a viable solution to address these issues and enhance bridge inspection procedures. The performance of UAVs has significantly increased in terms of in-flight stability, mobility, and size due to rapid technological advancements. Some researchers have created specialized UAV systems to photograph bridges while conducting inspections [50], [51], However, these studies did not locate, quantify, or measure damage to evaluate the status of the bridge; instead, they merely used the acquired images to support the inspectors by acting as "eyes in the sky." gives a summary of the suggested system. A UAV platform with visual sensors hovers above the building while conducting air missions to gather data. The data is processed by several technologies after takeoff.

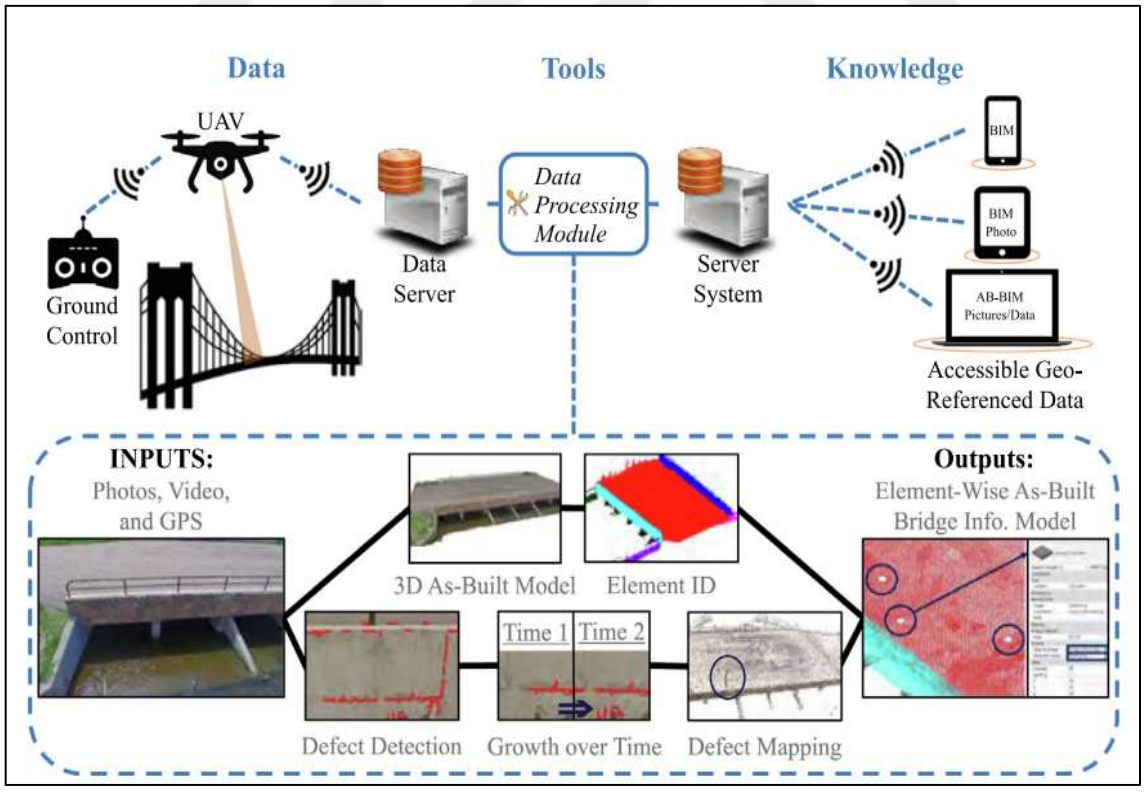


Figure 2.5: Proposed Automated Bridge UAVs Inspection System.

2.4 TRADITIONAL MACHINE LEARNING METHODS

Deep learning is not the only technique used in traditional ML methods, before training the models, a predetermined feature extraction stage is necessary to handle vast amounts of unstructured visual data [52]. The research in this category has been carried out continuously since over 30 years ago [53]. Traditional ML-based methods' two most widely used algorithms are support vector machines (SVM) in [53], [54] and artificial neural network (ANN) [55]. Additionally, other conventional ML methods applied, such as random forests [56] are also used. In a variety of disciplines, including civil engineering, materials science, and infrastructure maintenance, it is critical to understand the fundamental roles of methodologies for crack detection and crack-type classification as well as how to choose the most effective parameters for non-machine learning (ML) crack detection. These methods are essential for ensuring structural and component safety and integrity. The primary flaw in traditional ML methods is that they depend on shallower learning techniques. While higher-level feature learning isn't supported by these techniques, these systems struggle to handle the complex information found in images. An illustration of how light and location may have a significant impact on backdrop design is found in images of bridges.

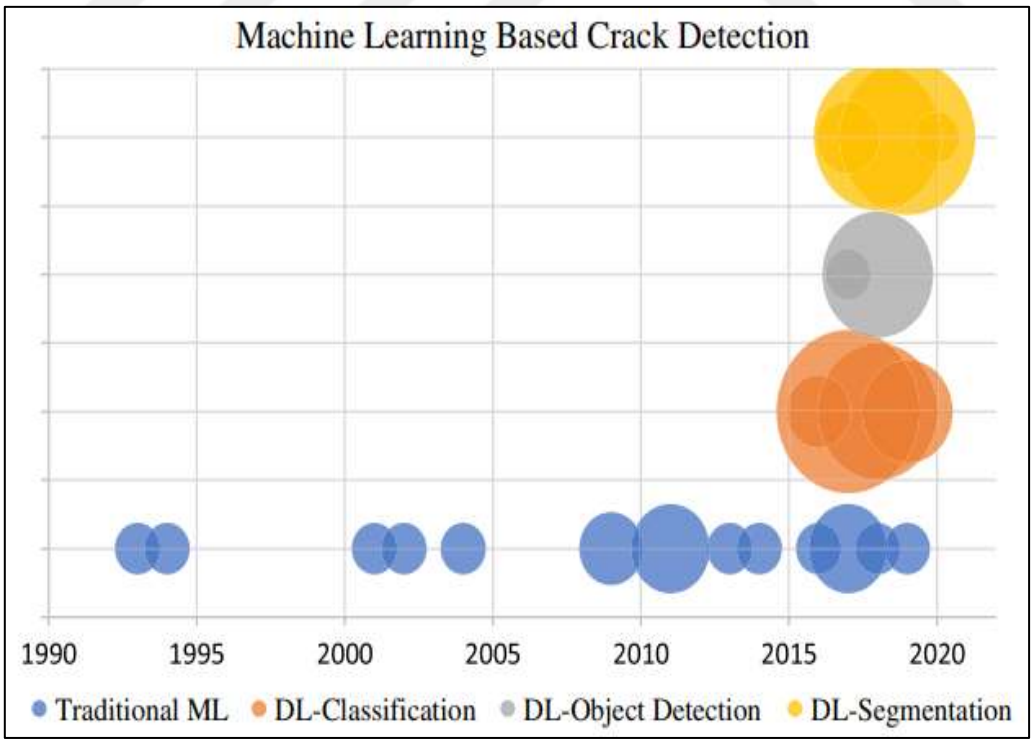


Figure 2.6: ML-Based Crack Detection Research Dissemination.

2.5 DEEP LEARNING METHODS

Deep learning is widely used as a complementary branch of machine learning, and deep learning neural networks have strong feature extraction and autonomous learning capabilities, making them efficient and accurate for practical needs. One of its main benefits is deep learning's efficiency in removing characteristics from large datasets. Convolutional neural networks (CNNs) have been the most extensively used technology among those employed for this purpose, enhancing feature extraction from huge datasets. For tasks like object identification, object categorization, image segmentation, and even face recognition, among other applications, researchers from a variety of fields have embraced CNN [57]. CNNs have demonstrated their ability to reliably identify and validate persons in tasks involving face recognition and object identification [58], such as crack detection. They have also shown astounding proficiency in identifying and locating items in pictures and videos. In other words, the machine learning model built around feature engineering has a substantially worse detection result than the detection model based on deep learning, The effectiveness of crack detection will be significantly increased when combined with the most effective object detection model and used to crack detection [59].

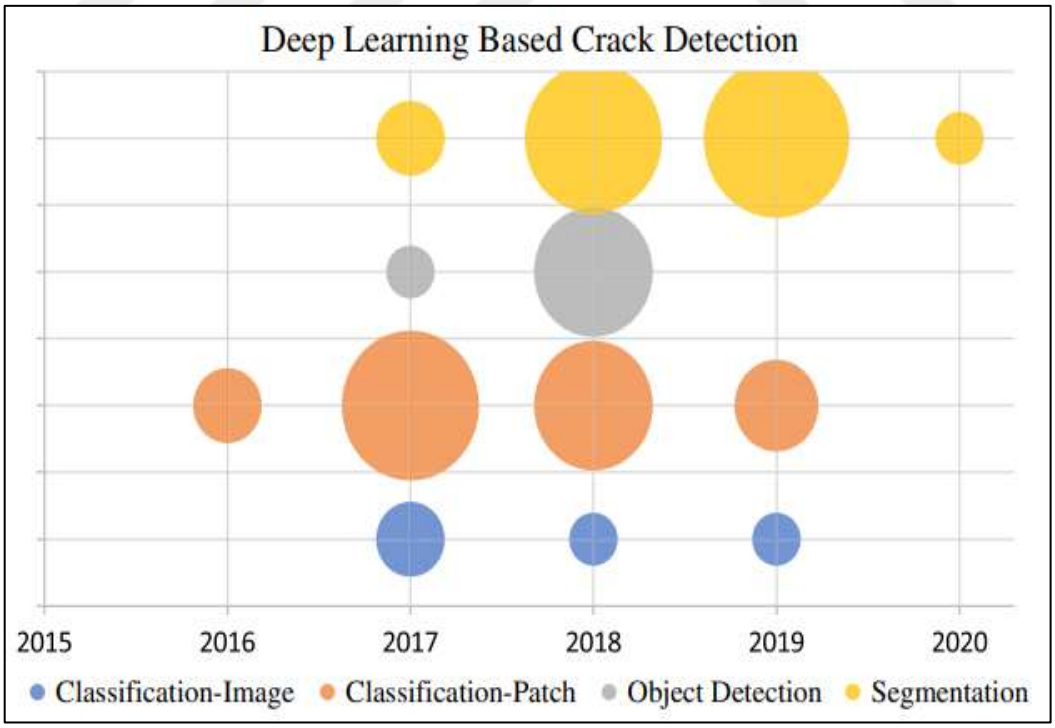


Figure 2.7: DL-Based Crack Detection Research Dissemination.

2.5.1 Object Detection

Deep learning-based object detection can automatically find cracks in real-world images [60]. The technique of locating and identifying things in a picture or a video is called object detection. The system assigns class probabilities and divides the image into bounding boxes. Drawing bounding boxes around each object in an image allows you to identify and categorize each one [61].in Figure 2. 8 create bounding boxes around crack-containing locations. Numerous real-world applications are driven by the important challenge of object recognition in natural settings [62]. The Single Shot Multibox Detector (SSD) is one example of recent work on object detection that used a regression-based methodology [63], Deconvolutional Single Shot Detector (DSSD) [64], and You Only Look Once (YOLO) [65]. Creating well-designed feature descriptors to extract embeddings for regions of interest has been a key component of many effective traditional approaches for object detection. Strong region classifiers and the selection of suitable feature representations in these methods have produced outstanding results. These techniques have been successful in capturing discriminative information necessary for precise object detection by emphasizing the thoughtful design of feature descriptors [66].

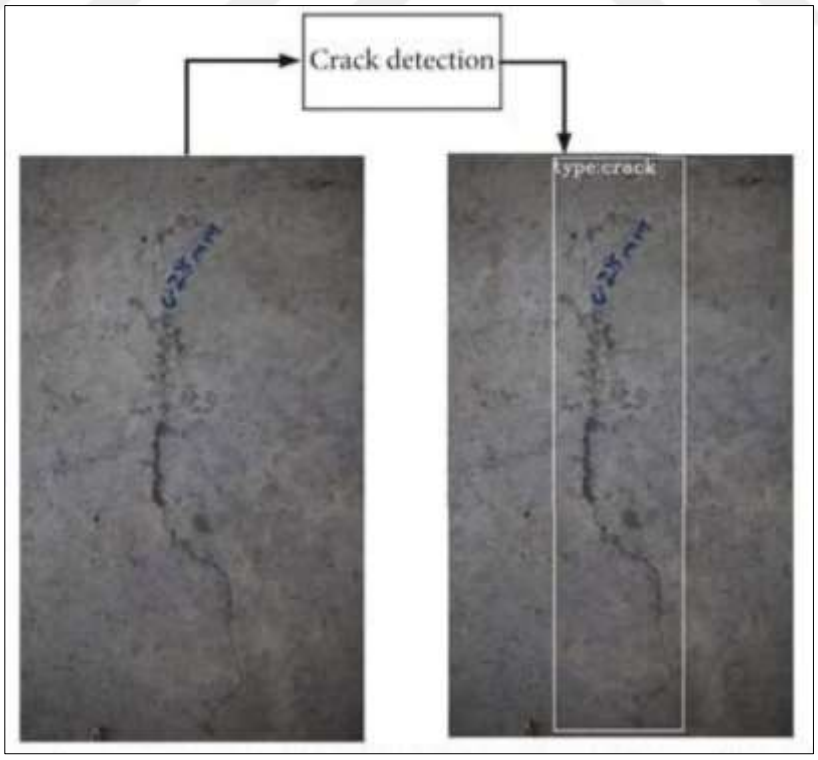


Figure 2.8: Object Detection: The Creation of Bounding Boxes Around Regions with Cracks.

2.5.2 Classification

Determine whether an image patch contains cracks and, if so, classify them using classification, was created in 2016 and is currently in its early stages of use in the classification of cracks [67], [68]. however, in 2017 it rose to the top DL method. Because the task is regarded as simple binary classification, CNN models with fully connected (FC) layers and lightweight models with fewer than 10 layers are typically used. Several academics have carried out image-level classification where the entire data set's image is categorized. other hand Patch-level classification [68][69][70], which divides each image into tiny image patches as shown in Figure 2. 9, is frequently used because it has two benefits. First, by segmenting images into small regions, additional data can be produced. Second, by categorizing each tiny area of the original image, crack localization information can be discovered. Crack-type classification can make use of the outcomes of patch-level classification. Due to patch-level classification's shortcomings in providing comprehensive information about cracks, such as their length, width, and branches, recent studies have given pixel-wise prediction priority. Patch-level classification produces coarse and blocky results, making it inappropriate for estimating crack features. The focus of scientists and engineers has therefore changed to pixel-wise prediction techniques.

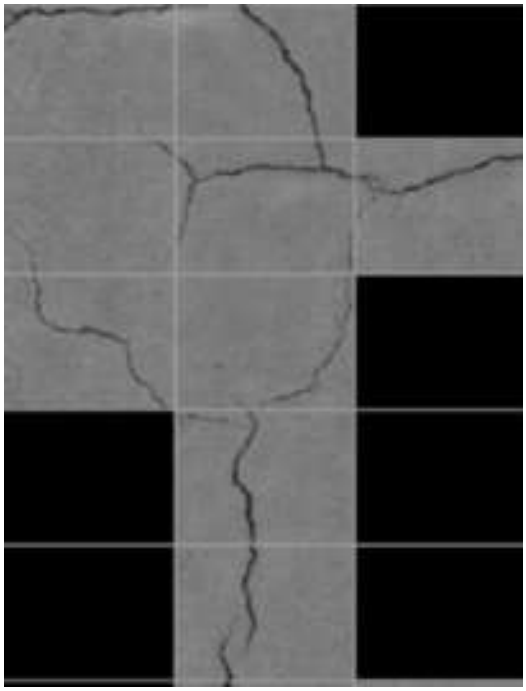


Figure 2.9: Image Patches are Classified into Two Categories: Crack and Noncracking.

2.5.3 Segmentation

In recent years, deep learning methods, particularly Deep Convolutional Neural Networks (DCNNs), have rapidly advanced computer vision technologies. current DCNN-based crack detection methods can be mainly categorized as crack segmentation methods, which identify the presence of cracks at the image (patch) level, object level, and pixel level, respectively [72]. The benefit of providing high-resolution crack detection has caused researchers to become increasingly interested in DCNN-based crack segmentation methods over the past few years [73]. Automated crack detection, which uses pixel-by-pixel crack segmentation, is a key method for determining the degree of structural element damage. This method entails categorizing each pixel of a picture as either having a crack or not, providing crucial information for precisely determining the extent of structural damage. This method provides a precise and thorough insight into the distribution, size, and severity of cracks within the structural part by painstakingly examining every pixel. This detailed information makes it easier to estimate the damage, identify the impacted locations, track changes over time, and make well-informed judgments about upkeep and structural integrity [74]. crack detection at the pixel level a fully convolutional network was suggested by [75] to segment crack pixels on various surfaces.

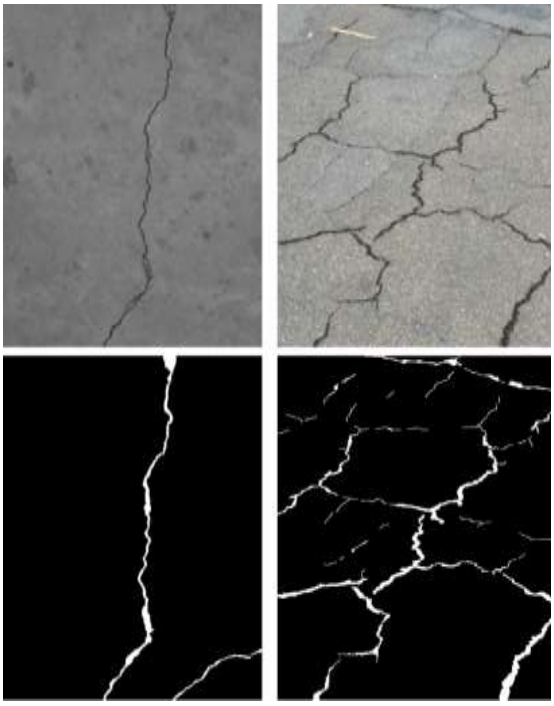


Figure 2.10: Sample Crack Segmentation Results.

2.6 CONVOLUTIONAL NEURAL NETWORK

A particular kind of artificial neural network (ANN) called a convolutional neural network (CNN) is used to process visual data, including images and videos. Without intentional feature engineering, it automatically recognizes and extracts useful features from the incoming data, detecting edges, textures, and shapes by sliding filters or kernels over the data. The network can learn complicated information at different levels of abstraction according to many convolutional layers. Through pooling layers, it is possible to down-sample the feature maps more efficiently while preserving key data and lowering computational complexity. The learned features are processed by fully connected layers for classification or regression tasks. The output of the network is created by combining features using activation functions and mathematical processes. By minimizing a loss function while comparing predictions with actual labels, CNNs train their parameters [71]. Backpropagation iteratively modifies parameters to enhance performance, Backpropagation entails Each location computes the difference between each kernel element and the input feature map element with which it overlaps, and the results are added up to obtain the output at that particular location. CNNs have achieved remarkable success in computer vision, surpassing human-level performance in image classification. They have applications in natural language processing and have revolutionized deep learning, enabling automatic feature learning and advancements in various domains.in Figure 2. 11 showing CNN architecture.

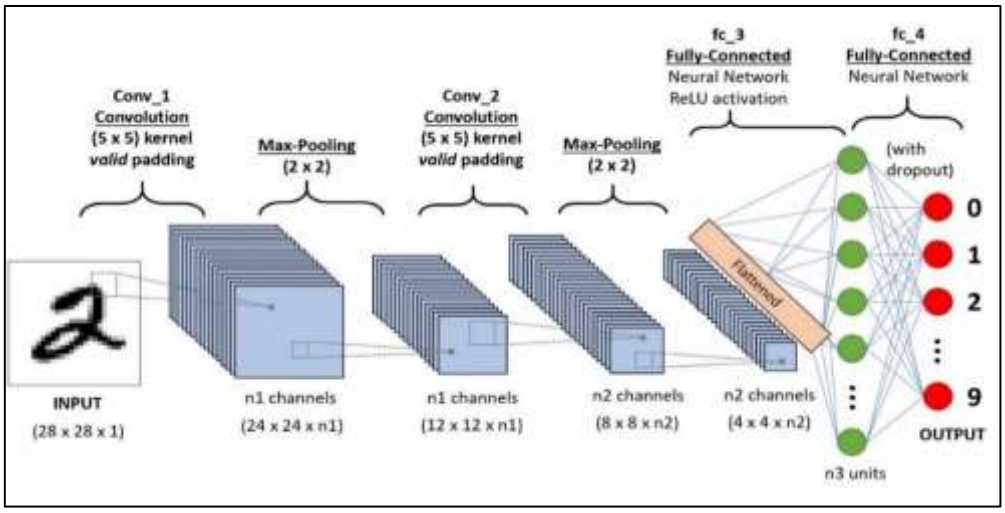


Figure 2.11: CNN Model Architecture.

2.6.1 Input Layer

CNN's input nodes only employ digital images. The three components of these images are [width x height x depth], which corresponds to a vector of image pixels. For instance, the density stands for RGB CNN, or Hue CNN, if the input signal was (width=32, height=32, depth=3), or [32x32x3]. The input image must also be divided by two several times. The three-dimensional matrix must first be reconfigured into a single column to work with the input layer. For instance, if an image has the dimensions 32x32=1024, it must be changed to 1024x1 before being passed into the input. The input dimension for training examples with a value of 'x' will be (1024, x). Figure 2. 12 depicts a CNN's input layer.

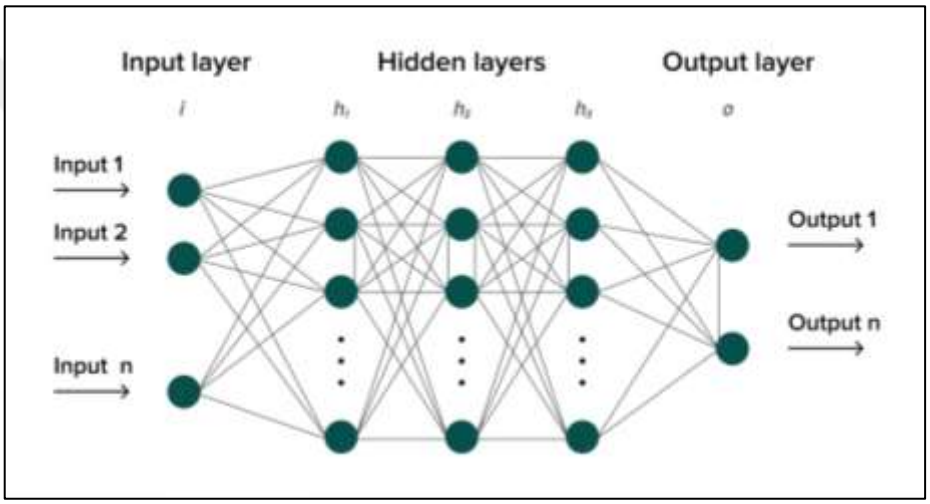


Figure 2.12: Input Layer.

2.6.2 Convolutional Layer

Several filters glide across the convolutional layer for the given input data. The output of this layer is then determined as the sum of an element-by-element multiplication of the filters and the receptive field of the input. The weighted summation is a component of the following layer. In Figure 2. 13, the concentrated area (left matrix), which is represented by the colors blue and red as its center, is multiplied by the filter matrix (middle), The outcome of this multiplication will be kept in the subsequent layer's center of focus' equivalent location. the other components of the convolution result can then be filled and the focus area can be moved [76]. Stride, filter size, and zero padding are the parameters that determine each convolutional operation. The sliding step is determined by the positive integer stride. Stride

1, for instance, instructs us to move the filter from one position to the right each time before computing the output. The receptive eld (filter size) must be constant for all filters employed in the same convolutional process. To control the size of the output feature map, zero padding increases the input matrix's original rows and columns by zero. The data towards the edge of the input matrix is primarily included via zero padding. The convolution result would be less than the input if no zero-padding existed. As a result, having more than one layer of convolutions causes the network size to decrease, which is why convolutional networks can only have a certain number of layers. However, zero padding stops networks from contracting and gives our network architecture endless deep layers. Figure 2. 13 shows the convolutional layer of the CNN model.

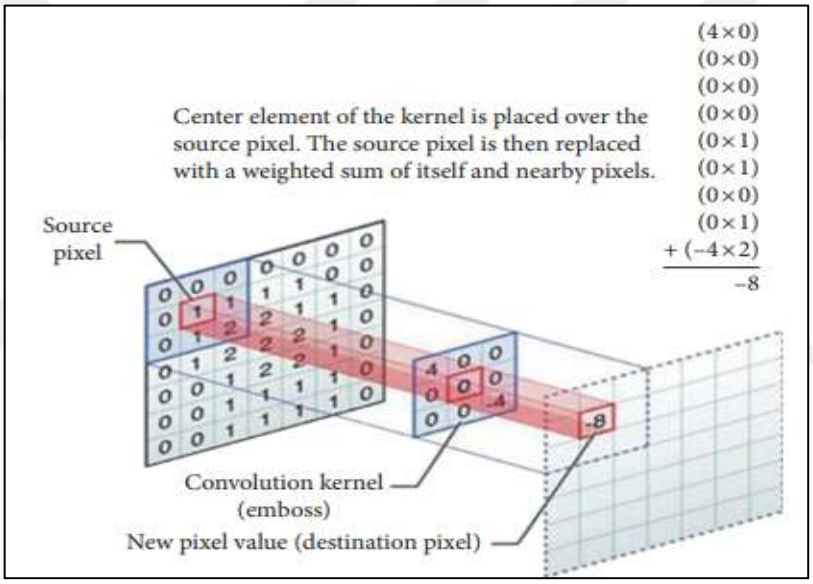


Figure 2.13: The Convolution Layer Glides the Filter Across a Given Input. The Output of a Filter and Receptive Field is the Sum of Each Element by its Element Matrix Multiplication.

2.6.3 Rectified Linear Unit Layer

In Convolutional Neural Networks (CNNs) used for recognizing images and computer vision tasks, the Rectified Linear Unit (ReLU) layer, acting as an activation function, is crucial. ReLU introduces non-linearity by eliminating negative values and keeping only positive values by using the equation f(x) = max(0, x), where x is the input to the layer. Through this action, the network can concentrate on key elements of the input data. ReLU has been widely adopted and used in a variety of deep learning architectures due to its efficiency as an

activation function, capacity to solve the vanishing gradient problem, and ease in learning complex patterns. Figure 2. 14 shows the Rectified Linear Unit (ReLU) layer of the CNN model.

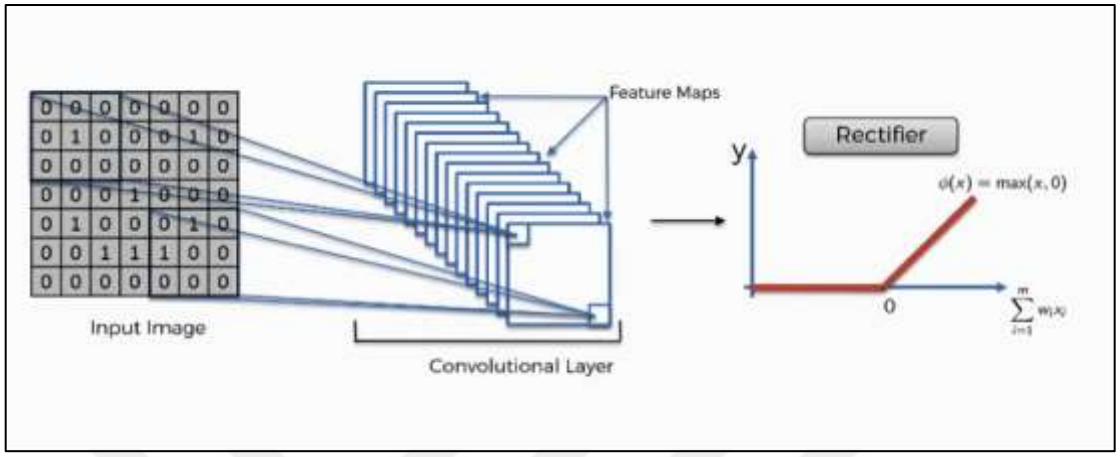


Figure 2.14: Rectified Linear Unit (ReLU) Layer.

2.6.4 Pooling Layer

Convolutional neural networks (CNNs) pooling layers play a crucial role in feature extraction and image processing. These layers efficiently cut the input feature maps spatial dimensions while retaining crucial data. Pooling layers enable translational invariance by applying the process of pooling independently to various feature map regions, allowing the network to recognize patterns despite their exact location in the input. Max pooling and average pooling are the most widely utilized types of pooling layers. While average pooling computes the average value within each window, presenting a summary of the features available, max pooling selects the largest value inside each pooling window capturing the most significant information as shown in Figure Figure 2. 15. By minimizing the number of parameters and computations, pooling layers also aid in lowering model complexity. Pooling layers are a crucial part of CNN designs because of the reduction's impact on both overfitting prevention and computational efficiency. In a CNN architecture, pooling layers are typically placed in between convolutional layers. The particular problem and the constraints of the architecture determine the pooling method to use and its characteristics, including the pooling window's size and stride.

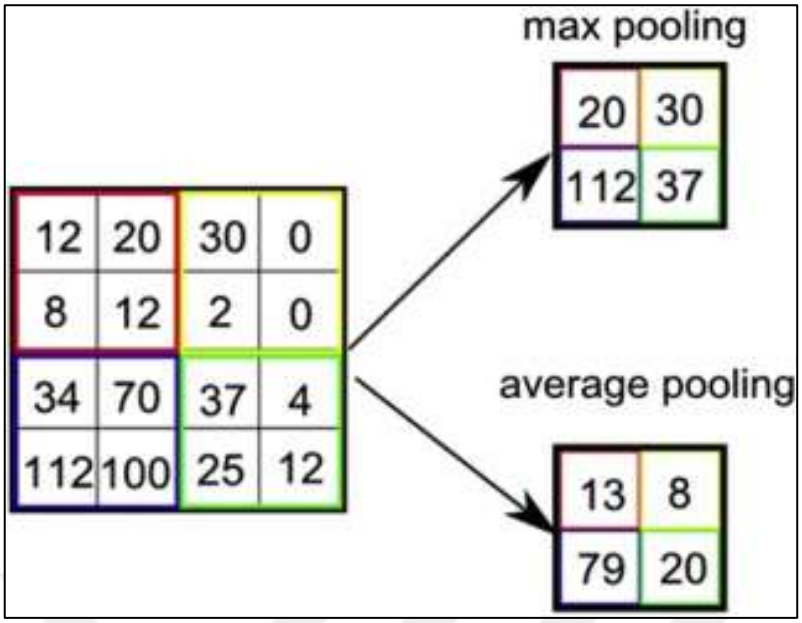


Figure 2.15: Max Pooling and Average Pooling Layers.

2.6.5 Flatten Layer

For image recognition and computer vision tasks, the flattened layer in CNNs is an essential component. its goal is to flatten and create a one-dimensional vector from the multi-dimensional feature maps produced by convolutional layers. The Flatten layer reshapes the spatial dimensions into a linear format so that the following fully linked layers can process the features efficiently. The flattened layer, which comes after the pooling layer and before the fully linked layers, offers high-level abstractions and decision-making in tasks like object detection and image classification. Figure 2. 16 shows the flattened layer of the CNN model.

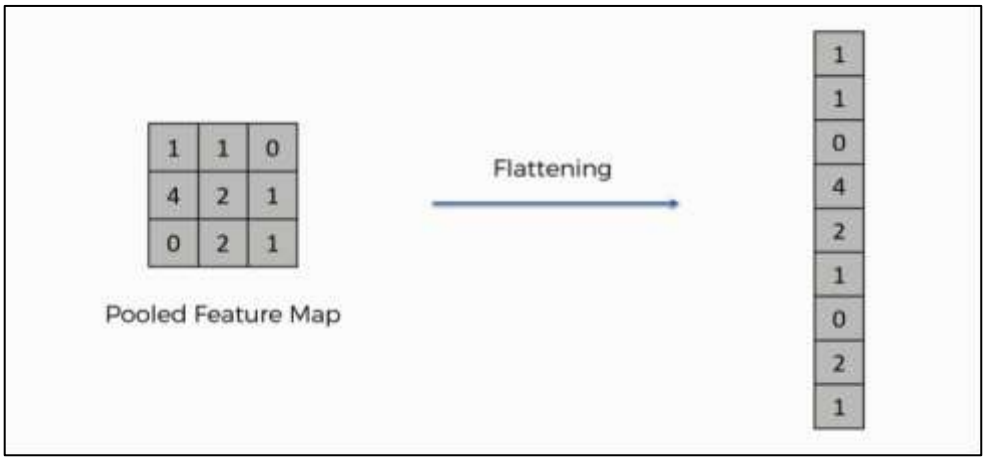


Figure 2.16: Flatten Layer.

2.6.6 Normalization Layer

the normalization layer normalizes the activation outputs of the preceding layer. Activations within a batch or layer are normalized using this technique, which enhances network stability and convergence during training. Typically, techniques like layer normalization are used following the convolutional or fully connected layers. By providing a zero mean and unit variance for the activations, the normalization layer addresses the problem of internal covariate shift. It reduces overfitting, stabilizes training, and enhances generalization in addition to assisting in solving the disappearing or expanding gradient problem. Figure 2. 17 shows the Normalization Layer of the CNN model.

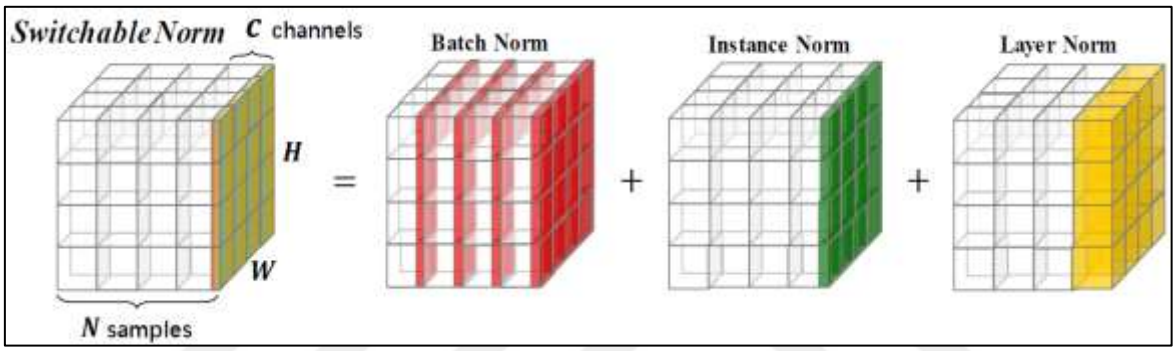


Figure 2.17: Normalization Layer.

2.6.7 Dropout Layer

In neural networks, the process of randomly removing nodes (input and hidden layers) during training is referred to as "dropout". The percentage of nodes that are temporarily eliminated is determined by the dropout probability, abbreviated as p, which in turn results in a modified network design. For instance, given input x: {10, 20, 30, 40, 50} to the fully connected layer the 20% of the nodes would be dropped during forward propagation if p = 0.2 (or keep probability = 0.8), x could become {10, 0, 30, 40, 50} or {10, 20, 0, 40, 50} and so on Similarly. The input and hidden layers both get this dropout operation. By removing nodes, the network learns to rely more on a variety of features and less on particular neurons. This precludes sophisticated co-adaptations that might correct errors made by other units but fail to generalize to data that has not yet been observed, hence reducing overfitting. Figure 2. 18 shows the Dropout Layer of the CNN model.

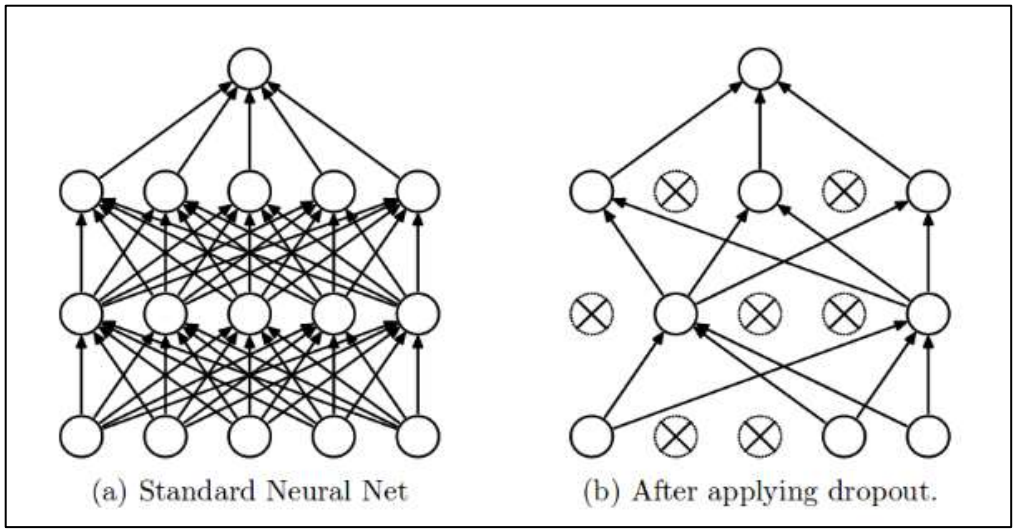


Figure 2.18: Illustrates a Comparison Between a Neural Network with a Dropout and One Without a Dropout.

2.6.8 Fully Connected Layer

By assigning the extracted characteristics to certain output classes, the fully connected layer of a Convolutional Neural Network (CNN) performs a critical role in determining the most appropriate label for the image. This layer efficiently mixes local data, allowing the network to discover complex associations in the data. To do this, the layer flattens the output from the convolutional and pooling layers that came before it, turning it into a feature vector. Each neuron in the fully connected layer receives input from every neuron in the layer above and applies weighted sums. These weights are changed throughout training to improve the performance of the network. Relu functions are also used to add non-linearity, allowing the network to learn complex patterns and make precise predictions. The number of output classes and the number of neurons in the fully connected layer correspond, and this relationship produces probability scores that show how likely it is that the input image belongs to each class. Overall, the fully connected layer combines the received data, assigns them to the relevant classes, includes non-linearity, and finally makes accurate predictions possible for a variety of tasks, Figure 2. 19 shows the Fully Connected Layer of the CNN model.

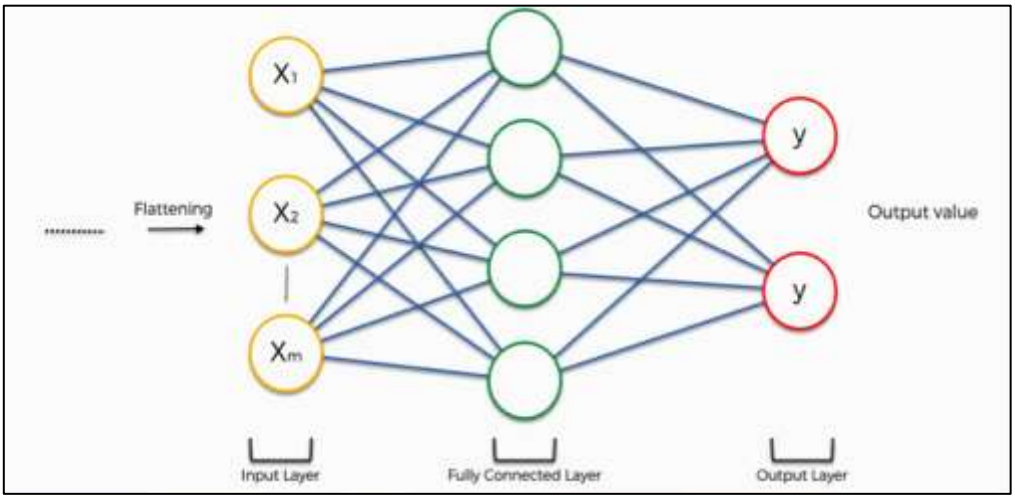


Figure 2.19: Fully Connected Layer.

2.6.9 Output Layer

The final predictions or outputs are generated by the CNN's output layer, which comes after the FC layer. Based on the current goal, it converts the learned feature representations into useful outcomes. The output layer of classification uses activation functions like SoftMax to assign a probability to neurons that represent each class. Neurons in regression relate to the expected variables. The output layer's parameters are changed during training to reduce the discrepancy between expected and actual values. Overall, the output layer applies activation functions, customizes predictions, and enhances network performance.

2.7 CRACK DETECTION

With the growth of deep learning in recent years, computer vision technology has reached the peak of advancement. Breakthroughs have been made in image classification, object detection, and semantic segmentation. The basic purpose of image classification is to identify the objects in each image. The goal of object detection is to find the position and type of items that appear. Semantic segmentation aims to determine which group each pixel point belongs to. The safety risks associated with working at heights have been partially eliminated by the introduction of object detection technology to detect cracks on the surface of bridges. Real-time bridge crack detection improves detection effectiveness while guaranteeing crack identification accuracy, resulting in significant time and labor cost savings.

Traditional object recognition techniques may be split into three stages: feature extraction, feature classification, and region selection, before utilizing deep learning. The sliding window algorithm is first used to select the location of a possible object in the image. Following the discovery of the object's position, feature extraction is carried out with the aid of manually created extractors. Finally, the extracted features are typically categorized using algorithms such as SVM [77]. Deep neural networks with lots of parameters may be able to extract more useful characteristics once they enter the deep learning development phase, which has led to the use of deep learning for object detection. The concept of the anchor was put out by faster R-CNN [78], pushing object detection to its initial peak. Using the YOLO method [79] was suggested to accomplish one-stage detection without an anchor. while enhancing detection performance, Mask R-CNN achieves instance segmentation. 2023 saw the release of YOLO v8, which strengthened the detecting performance over its predecessors. By using the cracks as the object and locating them in the image using a bounding box, the detection algorithm may be used to find cracks on bridge surfaces. In 2020, YOLO v4 improved to detect cracks on bridge surfaces [80]. YOLO and Faster R-CNN are the most effective object detection-based crack detection techniques. Certain researchers used object detection algorithms to find cracks. [81] proposed a polytype crack detection approach based on Faster R-CNN that enhanced the Faster R-CNN's initial structure and employed seven images of different structures for validation. This method allowed for the real-time detection and localization of numerous types of cracks. [82] utilized the SSDLite algorithm for detecting cracks. [83] designed a pavement detection system based on YOLO v2. However, it is unfortunate that these approaches' accuracy has to be improved and isn't as good as it may be. This paper's main goal is to identify bridge surface cracks using deep learning's powerful object detection algorithms. A training and validation data set for bridge surface cracks was first created. Additionally, the YOLO v8 was used to find surface cracks in concrete structural bridges. The YOLO v8 has a complicated structure and many parameters. To balance the speed of bridge surface crack detection with the loss of detection accuracy, an enhanced YOLO v4 bridge crack detection method was developed. Finally, we assessed and tested the suggested procedure and discussed the specifics of the detection.

3. METHODOLOGY

Three different iterations of the YOLO v8 object detection model, known as YOLO v8m, YOLO v8s, and YOLO v8n, were trained in this study using Google Colab Pro+ as the computational platform. This platform had significant resources, such as a sizable quantity of RAM, ample disk space, and powerful GPUs, allowing for effective object detection. While the test operation was carried out using Anaconda, the training and validation operations were carried out on Google Colab Pro+. Simulator speed was enhanced thanks to Google Colab Pro+'s increased access to virtual memory and longer runtime. The platform's ability to execute code in the background was essential since it allowed the code to run continuously for up to 24 hours without requiring the browser to be active. Our analysis utilized high-performance Google Colab Pro+ GPUs, such as the A100, which offer significant power. Additionally, the system boasted 52 GB of RAM and 2 vCPUs to support our computational tasks. A modest increase in these allotted resources would be beneficial to further optimize the simulation, enabling more effective processing and supporting larger datasets. It is significant to mention that the Roboflow platform was used to collect, label, and augment the dataset used for our research. Roboflow gave us the tools and abilities we needed to efficiently clean up and enhance the dataset for training our object detection model. The hardware setups utilized in the current study are shown in Table 3.1.

Table 3.1: Google Colab Pro+ Model Training Hardware Specifications**.**

Specification	**Details**
Platform	Google Colab Pro+
Operating System (OS)	Linux-based
GPU	NVIDIA A100
GPU Runtime	24 hours
RAM	52GB
CPU	2 × vCPU

installed CUDA Deep Neural Network libraries to utilize the GPU. Python's easy-to-understand syntax and broad use in the deep learning industry were major factors in the choice to make it the primary programming language. It is the best option for building

complex algorithms and utilizing different deep-learning libraries because of its simplicity and versatility. We performed a quick but thorough testing phase in Python before moving on to the in-depth analysis, and the program performed flawlessly, proving that the implementation was reliable and error-free. We tapped into the potent capabilities of both the TensorBoard and ClearML platforms to effectively view and analyze the YOLO v8 model's output. These tools allowed us to produce detailed and illustrative graphs, which aided in a deeper comprehension of the model's performance and insights into the relevant object detection tasks. The analysis process was further streamlined by the seamless integration of various technologies, increasing the overall effectiveness and dependability of our research. Figure 3. 1 shows the Proposed Method in this study.

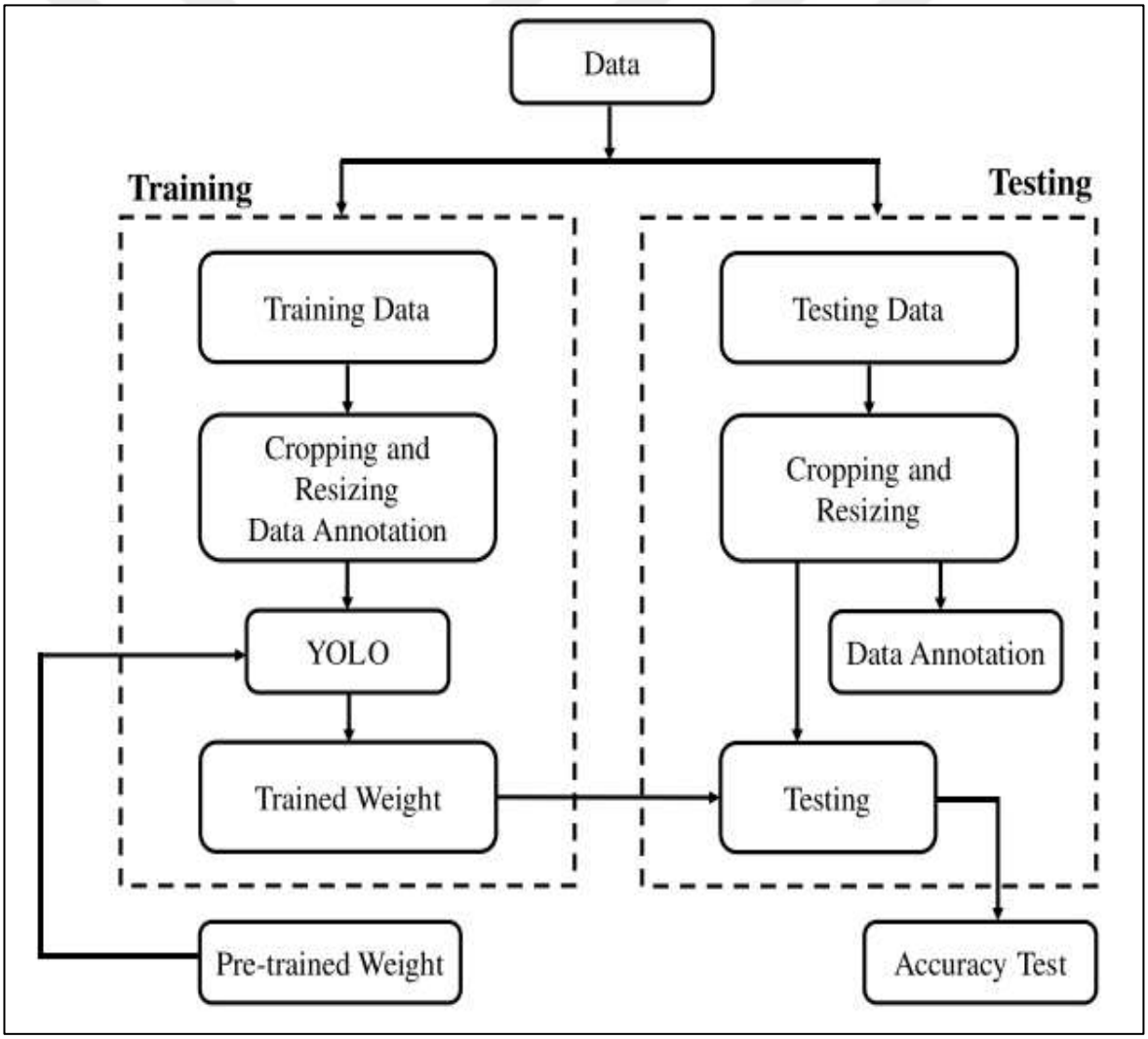


Figure 3.1: Diagram of the Methodology Used in This Study.

3.1 DATASET

By creating a private dataset designed for this assignment, this master's thesis addressed the problem of bridge surface crack detection. A substantial data-gathering effort was made in the field because there were so few publicly available datasets. Each digital image in the dataset has a dimension of 640 x 640 pixels, making up roughly 2754 total. These photos, which were taken with a digital camera, show a variety of places, but they mostly highlight tiny and medium-sized bridges made of reinforced concrete. The collection includes horizontal, vertical, and diagonal cracks, which make up nearly all frequent types of concrete cracks. Additionally, the dataset includes both dim and high natural light settings. In addition, the concrete where cracks exist has both smooth and rough surfaces. The website makesense.ai was used to allow specific dataset annotation. To guarantee the availability of ground truth data for model training and evaluation, it offered a user-friendly framework for labeling and annotating the gathered photos. The dataset was further expanded using the Roboflow platform to increase its diversity and improve the generalization skills of the model. The dataset was divided into three subsets: a training set that contained 70% of the data, a validation set that contained 20% of the data, and a test set that contained the remaining 10% to enable reliable model training and thorough evaluation. This partitioning made it possible to train models efficiently, optimize hyperparameters, and evaluate model performance objectively. This research adds to solving the lack of publically available datasets for bridge surface crack detection by gathering this distinctive dataset through field imaging utilizing UAVs and rigorously annotating and supplementing the images. The dataset is an invaluable resource for future research and enables the creation and assessment of precise and effective deep-learning models for this crucial task.

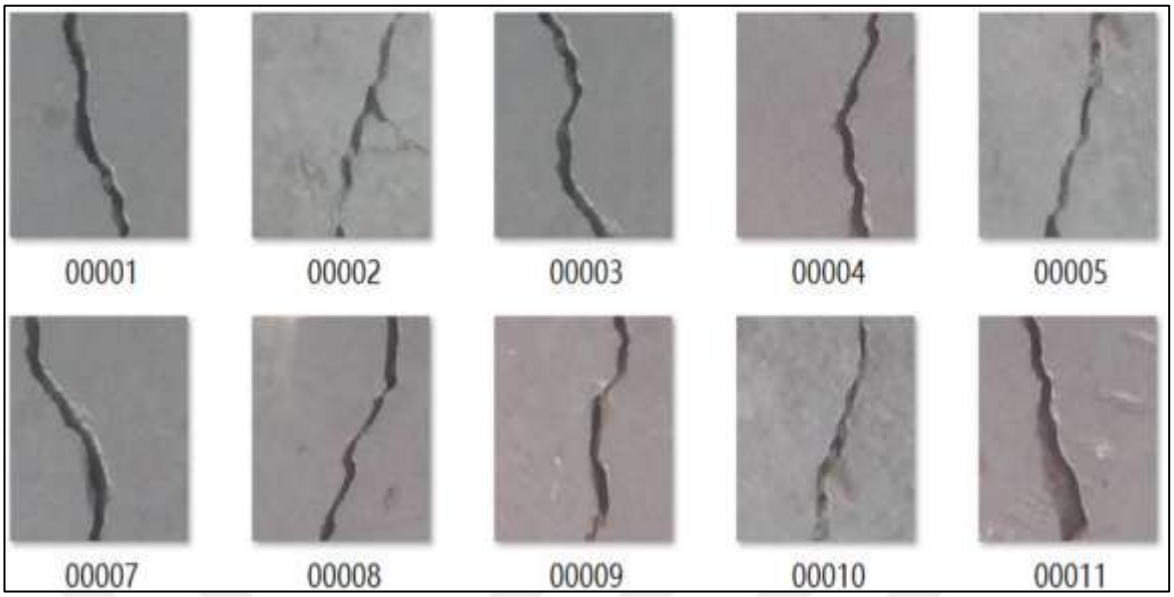


Figure 3.2: Dataset Crack Sample.

3.1.1 Dataset Labeling

As shown in Figure 3. 3 we used MakeSense, a popular tool in the field of deep-learning computer vision. in this study, this tool makes it easier to accurately label the position and type of target objects or classes in images. The steps for using MakeSense to keep the details about target objects safe are as follows:

a. Importing the folder containing the image dataset is the first step.
b. The second step entails finding the cracked object and placing it in a box (the ground truth box).
c. Give the rectangle box labels.
d. To save the information (in the YOLO format required by the YOLO V8 model), choose the right file format before storing the crack information.

The lines in the crack information files each contain five numerical values that describe a target object in the image in depth. The target object's label is represented by the first number, while the coordinates of the center of the rectangle labeling box are represented by the second and third numbers. These coordinates are calculated about the origin in the image's upper left corner. The relative width and height of the ground truth box are represented by the fourth and fifth numbers, respectively.

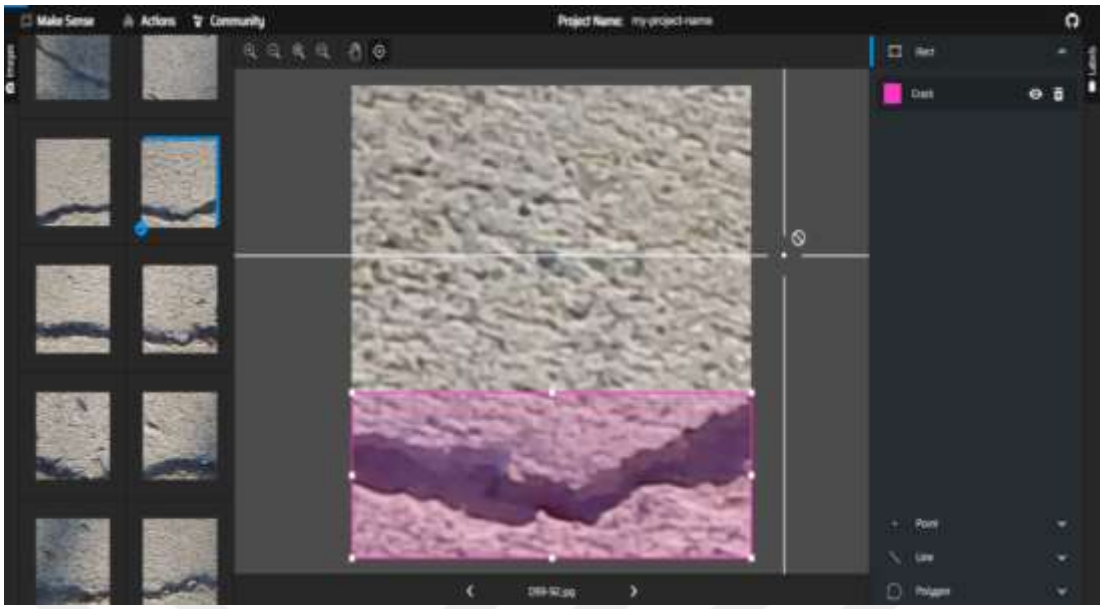


Figure 3.3: Bridge Crack Image Label.

3.1.2 Dataset Splitting

In this work, we used a dataset of 2,754 images representing a significant collection of bridge crack images, as shown in Figure 3. 4, We used the Roboflow platform for dataset management to enable reliable evaluation and model training. The dataset was separated into three subsets: a training set, a validation set, and a testing set. This was done specifically as an essential step. Around 1,928 photos comprised the training set, which comprised roughly 70% of the total dataset. This subset was deliberately selected to offer a sufficiently broad and varied sample for successfully training our models. By utilizing this subset, we hoped to capture the distinctive characteristics and variances found in images of bridge cracks, allowing the models to learn reliable and versatile features. We set aside nearly 20% of the dataset, or about 551 images, for the validation set. During the training process, we used this subset to evaluate the effectiveness of our models and adjust their hyperparameters. We sought to identify the ideal model configurations that performed better on bridge crack images through meticulous validation. We set aside 10% of the dataset, or roughly 275 images, for testing to evaluate the final performance and gauge the generalizability of our models. This subgroup was used as a separate benchmark to assess the model's performance in detecting and classifying bridge cracks in previously unknown conditions. We evaluated the model's overall effectiveness and confirmed its practical applicability using the testing

set. It is important to note that before the dataset-splitting process, each image within the dataset was carefully annotated. This made sure that the labeled data remained accurate in the training, validation, and testing subsets, enabling precise model performance evaluation and comparison.

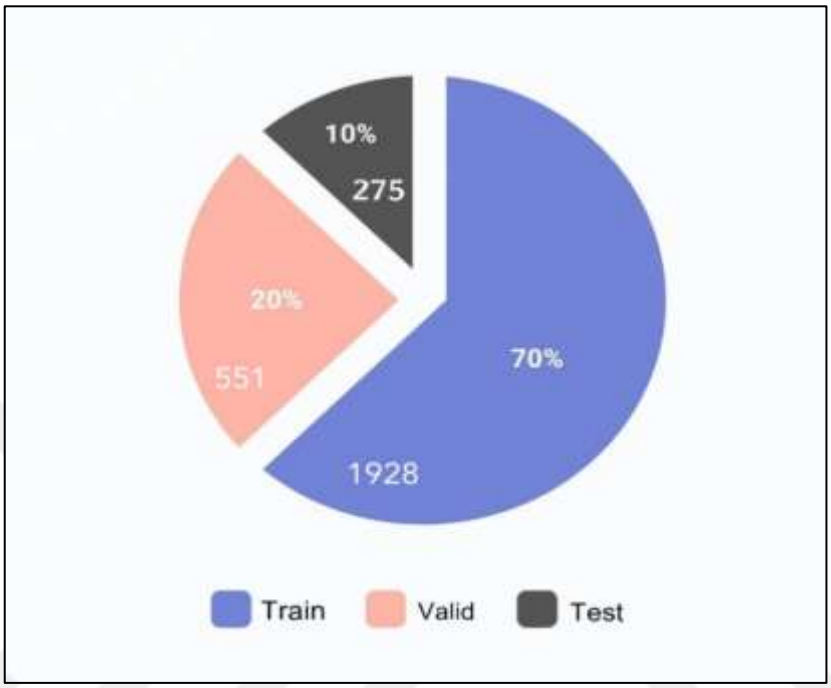


Figure 3.4: Dataset Splitting into Training, Validation, and Testing Sets.

3.1.3 Dataset Augmentation

The procedure of expanding the dataset in this study includes creating modified versions of the current data using a variety of data augmentation techniques. Flip, Rotate, Crop, Rotate, Shear, Grayscale, Hue, Saturation, Brightness, Blur, Noise, and Mosaic were some of the augmentation techniques used as shown in Figure 3. 5. This strategy worked well since it reduced the need for further data collection and labeling, which cut down on the costs involved. Additionally, data augmentation improved the precision of predictions provided by deep learning models. The models were better able to handle a variety of scenarios and reduced the chance of overfitting, which occurs when a model gets overly specialized to the training data and struggles to generalize to new examples. The use of these strategies for enhancing the data was crucial for improving the dataset, enhancing model performance, and enhancing the overall resilience of the deep learning models used in the study.

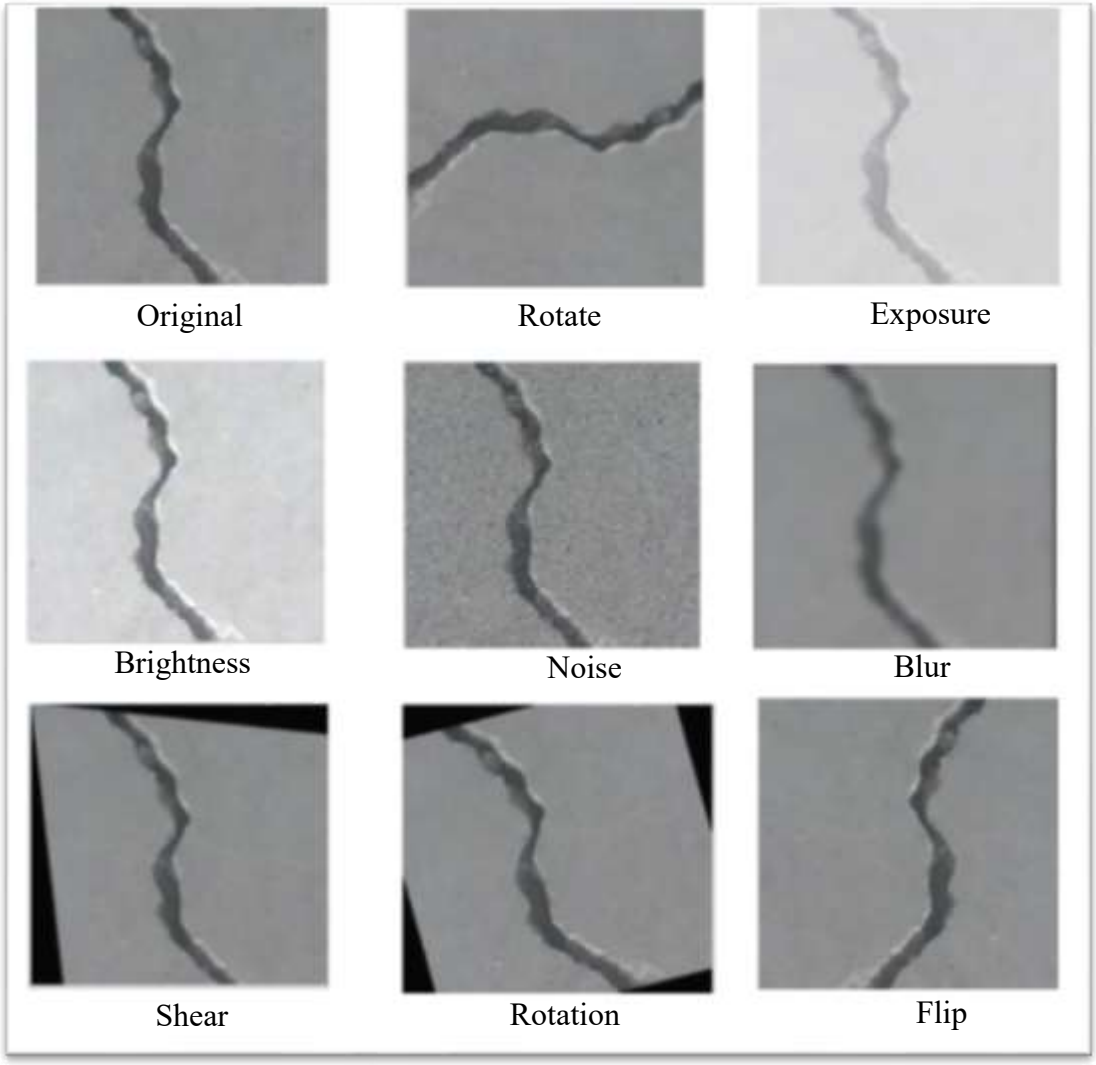


Figure 3.5: Dataset Augmentation.

3.2 YOU ONLY LOOK ONCE

A well-known real-time object recognition algorithm called YOLO (You Only Look Once) transformed computer vision applications. YOLO, which was created by Joseph Redmon and his colleagues, is notable for its ability to quickly and accurately identify objects in photos. We shall explore YOLO's inner workings, its salient characteristics, and its influence on the field of computer vision in this part. A single neural network is the foundation of YOLO, which predicts bounding boxes and class probabilities for various objects in a picture simultaneously. This novel method contrasts with computationally intensive proposal-based object detection approaches like CNN (Convolutional Neural Networks), which require

numerous passes through the neural network. The idea behind YOLO is to divide an image into a grid and identify items within each grid cell by estimating bounding boxes and class probabilities. The model can handle object detection more effectively because of this separation. No of how many items are present, each grid cell must forecast a specific number of bounding boxes. An object's bounding box and class probability are predicted by a certain grid cell if the object's center falls within that cell. The loss function used by YOLO is special since it mixes classification and localization errors. With the help of this loss function, the model is guaranteed to maximize both object classification accuracy and the precision of bounding box predictions. Because of this, YOLO has outstanding generalization abilities and can find different object categories in a variety of settings. The ground-breaking single-stage object identification system YOLO (You Only Look Once) has remarkable real-time processing rates and can identify objects in photos in milliseconds. Because of its effectiveness, YOLO is a good fit for several applications, including video analysis, security systems, and autonomous cars. The YOLO architecture, however, has trouble accurately recognizing smaller objects or items that are close to one another. Additionally, when dealing with objects of various sizes in the same image, multi-scale detection might be challenging. Double-stage algorithms, on the other hand, work in two steps, first producing region recommendations and then honing them for object detection. Double-stage algorithms, while potentially more accurate and capable of handling complex scenarios, are more computationally intensive and take longer to process data than YOLO. The decision between single-stage and double-stage algorithms depends on the needs of the individual applications as well as the desired speed vs. accuracy trade-off.

3.3 BRIDGE CRACK DETECTION BASED ON YOLO V8

Using the one-stage detection method, YOLO v8, a member of the YOLO family, detects objects by combining all the components of the object detection pipeline into a single neural network. With this method, the detection process can be completed without the need for additional phases or modules, speeding the process and increasing overall effectiveness. YOLO v8 streamlines and increases the efficiency of the object detection process by merging all relevant components into a single network. In this study, we carried out object detection training utilizing the YOLO v8m, YOLO v8s, and YOLO v8n models, especially in detecting bridge cracks within a dataset taken by unmanned aerial vehicles (UAVs). The CSPDarknet-

53 backbone structure was used in the creation of the YOLO v8 model, which is based on Convolutional Neural Networks (CNNs) and serves as the basis for feature extraction and representation. Figure 3. 6 shows the YOLO v8 crack detector's overall schematic mechanism. By utilizing CNNs to locate and identify things within images, this model takes a thorough approach to object detection. The YOLO v8 model offers an effective and economical method for identifying bridge cracks in UAV-captured pictures by combining several elements, such as feature extraction, region proposal, and classification, into a single framework. YOLO v8 models may now be used to reliably detect and analyze bridge cracks because of the training process' excellent accuracy and desired outputs.

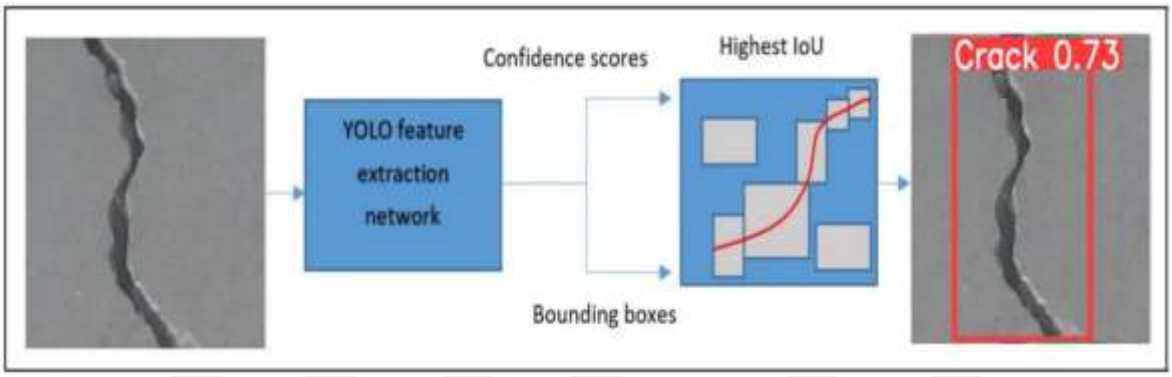


Figure 3.6: Schematic Representation of the YOLO V8 Object Detector with Bounding Box Visualization.

The YOLO v8 design includes several key components to make object detection employment easier. The Backbone, one of these components, is essential because it employs a series of convolutional layers to pull out relevant characteristics from the input image. Convolution techniques are used in this procedure to analyze the spatial information of the image and find interesting patterns and structures. The YOLO v8 model successfully collects and depicts critical aspects that are necessary for precise object detection by utilizing the Backbone. The model gets a greater grasp of the input data through this feature extraction process, enabling reliable item detection and localization. The YOLO v8 architecture's SPPF (Spatial Pyramid Pooling Fast) layer is essential for processing features at various scales. Regardless of the size of the input feature maps, this layer is intended to gather context information and encode it into fixed-length representations. The input feature map is divided into numerous regions of various sizes to run the SPP layer. Then, each zone is pooled separately to provide an output with a fixed size. With the help of this pooling process, the

model can gather background data at several scales, both local and global. The SPP layer gives the model the ability to manage items of various sizes and adapt to various spatial situations by incorporating several scales. The SPP layer can be thought of as a particular form of feature pyramid in which the combined outputs from various scales are pooled. This enables the successive convolution layers to process data at various levels of abstraction, collecting both high-level semantic information and fine-grained minutiae. The SPPF layer's multi-scale data collection enhances the model's capacity to precisely detect and localize objects of various sizes within an image. Upsample layers are essential in the YOLO v8 architecture for boosting feature map resolution. These layers are used to upsample or extend, the feature maps' spatial dimensions, representing higher resolution. Typically, methods like closest neighbor interpolation or bilinear interpolation are used to execute the upsampling procedure. The procedure increases the amount of information available for analysis by duplicating and extending the data found in the feature maps. Upsample layers allow the model to collect finer details and better maintain the spatial information of the objects by raising the resolution of the feature maps. This is crucial for finding and locating tiny or intricate items that may have been downscaled during the feature extraction process' preliminary phases. The model's ability to correctly localize objects is aided by the higher-resolution feature maps, increasing the effectiveness of object detection. Additionally, the Upsample layers make it easier to combine low-resolution characteristics from earlier layers with high-resolution features. The model may use both fine-grained details and high-level semantic information thanks to the combination of multi-scale features, which enables a more thorough understanding of the objects in the image. The YOLO v8 architecture's C2f module combines high-level features with contextual data to improve object detection precision. The module seeks to enhance the model's comprehension of the items in the image and enable more accurate detection by merging these two forms of information. The high-level features, which record abstract and semantic information, are combined with contextual data to operate the C2f module. Additional spatial and contextual cues are provided by this contextual information, which can help with the precise localization and detection of an object. The C2f module improves the model's capacity to take into account the connections and dependencies between objects and their surrounding context by integrating contextual information. The detection procedure can be made more thorough and informed thanks to this integration. The C2f module in YOLO v8 enhances the overall detection accuracy by

utilizing both the semantic understanding and the contextual cues present in the image through the fusion of high-level characteristics and contextual data. This integration enables a more reliable and accurate detection capacity, which helps to improve the performance of object detection. The Detection module of the YOLO v8 architecture effectively maps high-dimensional features to output bounding boxes and object classes by using convolutional and linear layers. The module does the critical work of converting the extracted characteristics into accurate predictions about the positions and types of objects in the input image by making use of these layers. This procedure uses several calculations and transformations to accurately and effectively find objects of interest. Without sacrificing high detection accuracy, the overall architecture's design seeks to find a compromise between speed and efficiency, Figure 3. 7 shows the YOLO v8 model structure. To achieve this, it is necessary to carefully engineer the various parts and modules to maximize computational efficiency while assuring reliable and accurate object detection. This balance allows the architecture to produce a powerful solution that can quickly process images while retaining a high level of object detection accuracy. Rectangles are used to represent the layers in the diagram, and each rectangle is labeled with the appropriate layer type (e.g., Conv, Upsample) and any relevant parameters (e.g., kernel size, number of channels). The arrows that connect the rectangles represent how data moves between the layers. The direction of the arrows shows the network's progression of data from one layer to the next. This graphic representation clearly shows the architecture's data flow and the sequential manner in which computations are carried out.

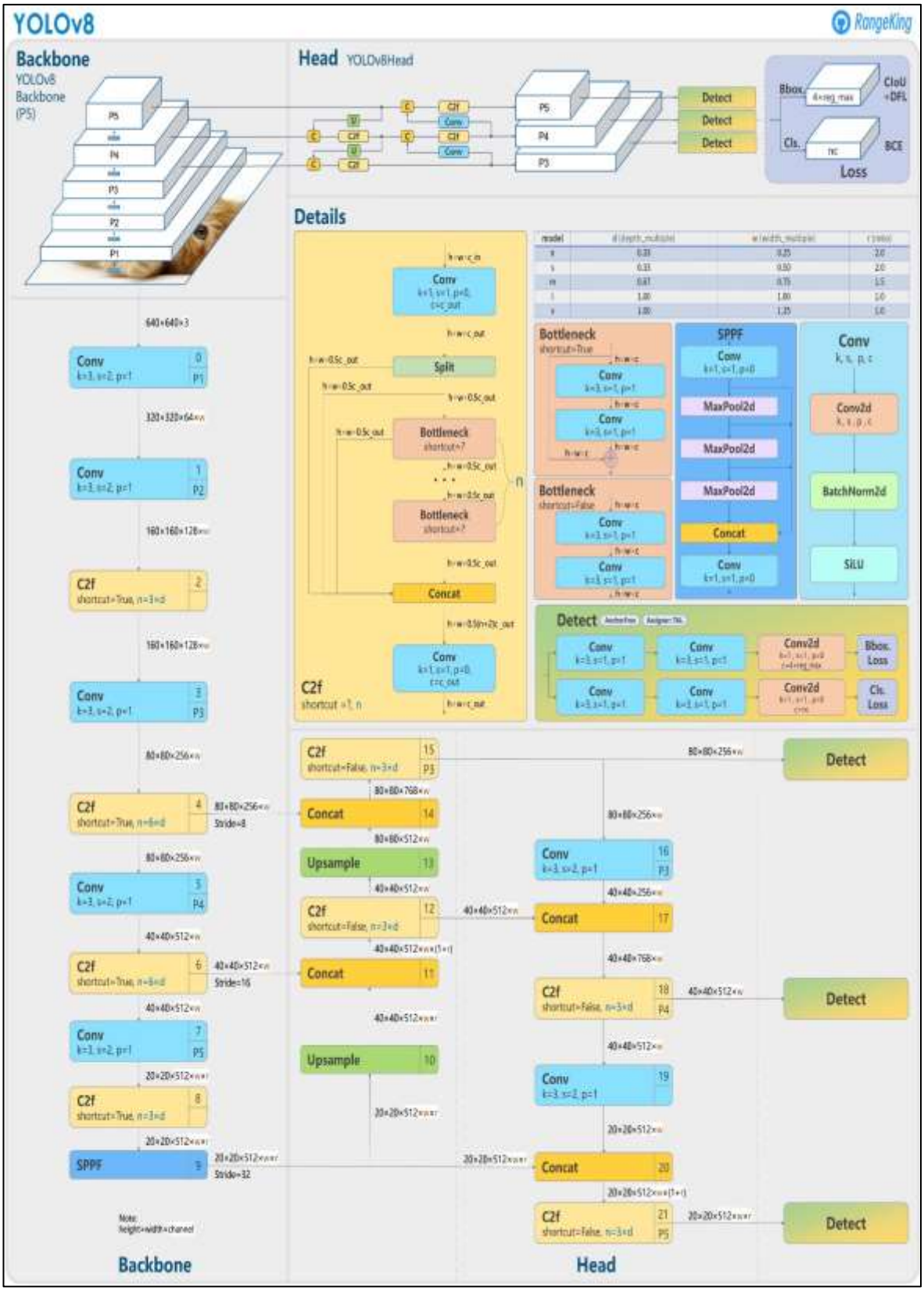
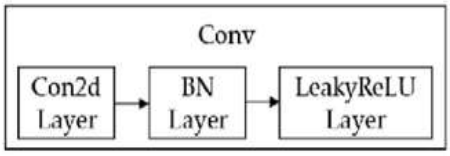


Figure 3.7: Structure of the YOLO v8 Model.

3.4 DARKNET-53 BACKBONE NETWORK

Deep convolutional neural network Darknet-53 is excellent at extracting features for computer vision tasks. It is renowned for its lightweight construction, which makes it suited for locations with limited resources. It has 53 convolutional layers and got its name from being created using the Darknet open-source deep learning framework. Darknet-53 excels at extracting complex patterns from photos despite its lightweight architecture, making it the perfect choice for contexts with limited resources. It's well known for acting as the foundation for the YOLO object detection series, which also includes YOLO v5 and YOLO v8. As a backbone network, Darknet-53 convolutionally transforms incoming images, gradually learning hierarchical features from unprocessed pixel values. Then, for a variety of computer vision applications, such as object identification, image segmentation, and classification. Darknet-53 is still essential for allowing real-time performance and good object identification accuracy in possibly newer versions like YOLO v8. Figure 3. 8 shows Darknet-53 Architecture.

Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv Conv Residual	32 1 × 1/1 64 3 × 3/1	Conv Conv Residual × 1	208 × 208 208 × 208 208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv Conv Residual	64 1 × 1/1 128 3 × 3/1	Conv Conv Residual × 2	104 × 104 104 × 104 104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv Conv Residual	128 1 × 1/1 256 3 × 3/1	Conv Conv Residual × 8	52 × 52 52 × 52 52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv Conv Residual	256 1 × 1/1 512 3 × 3/1	Conv Conv Residual × 8	26 × 26 26 × 26 26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv Conv Residual	512 1 × 1/1 1024 3 × 3/1	Conv Conv Residual × 4	13 × 13 13 × 13 13 × 13



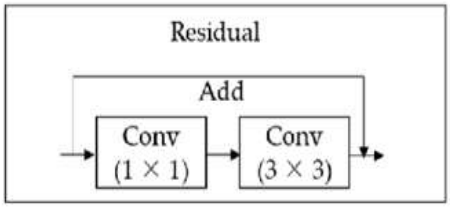


Figure 3.8: Darknet-53 Architecture.

3.5 TRANSFER LEARNING AND FINE-TUNING

Transfer learning and fine-tuning are essential for improving the model's performance when utilizing YOLO for crack detection. As a real-time object detection algorithm, YOLO is frequently used for a variety of computer vision tasks. However, a large dataset and significant computer power would be needed to train a YOLO model from scratch for crack detection. Utilizing YOLO models that have already been trained and have learned broad object identification features from extensive datasets like ImageNet, transfer learning offers a practical approach. The YOLO model can already detect many objects, including cracks in varied surroundings, by using the knowledge from the pre-trained model. The early layers of the pre-trained YOLO model, which are in charge of feature extraction, remain frozen during transfer learning for crack detection to retain the learned representations. The capacity to recognize frequent visual patterns found in cracks and other objects is provided by these feature extraction layers. A fresh dataset including photographs of cracks is used to replace and improve the classification and detection layers, which are specialized for the crack detection task. The model can be fine-tuned to specialize in crack detection while gaining from the broad knowledge acquired from ImageNet by modifying the weights of the classification and detection layers with a lower learning rate. Transfer learning and YOLO fine-tuning allow us to effectively harness the strength of trained models while drastically reducing the amount of data and computing work needed for crack detection. The model gains proficiency in accurately identifying cracks as a result of its ability to draw on its prior experience in object detection and adapt to the unique characteristics of cracks in varied contexts. Therefore, this method works well for real-world crack detection applications, especially when there is a dearth of labeled crack data.

4. RESULT

The objective of the study was to detect bridge cracks using a dataset of 2,754 images. Our goal was to achieve high accuracy and fast crack detection using the powerful YOLO v8 algorithm and its three variants (YOLO v8m, YOLO v8s, and YOLO v8n). The accuracy of the YOLO v8m model was remarkable at 98.54%, while that of the YOLO v8s and YOLO v8n models was 97.81% and 97.45%, respectively. We also evaluated the detection speed of each model, with YOLO v8m operating at 31.45 fps, YOLO v8s at 64.10 fps, and YOLO v8n operating at an amazing 95.24 fps. These findings highlight the efficiency of our method for precisely identifying cracks while taking into account the trade-off between accuracy and processing time. The potential usefulness and effectiveness of our suggested method for actual applications in bridge inspection and maintenance are demonstrated by the combination of high accuracy and quick processing time.

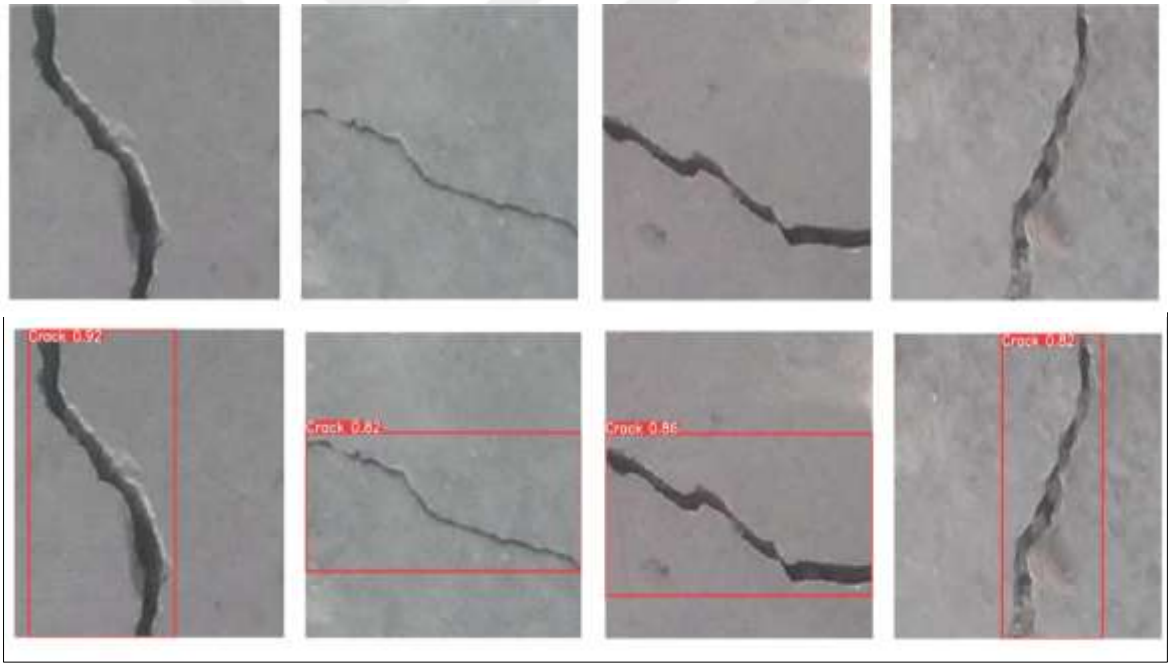


Figure 4.1: Crack Detection Result.

4.1 MODEL TRAINING AND TESTING

Labeled photos were used in both the training and validation sets during the object detection training procedure. The model's performance was assessed using a second validation set of 551 images after it had been trained on 1928 images, and the hyperparameters were modified to get the model to its best state. The trained model was then put to the test on 275 original

photos to determine how accurate and effective it was. On the three separate YOLO v8 variants YOLO v8n, YOLO v8s, and YOLO v8m training was conducted across 300 epochs. During training, each variant received about 18075 iterations.

$$Total\ Iterations = \frac{Number\ of\ Images}{Batch\ Size} * Number\ of\ Epochs \tag{4.1}$$

The quantity of parameters and computing complexity varies between these variations. There are 3.2M parameters in YOLO v8n, 11.2M parameters in YOLO v8s, and 25.9M parameters in YOLO v8m. Additionally, the GFLOPs measure the computational load; YOLO v8n, YOLO v8s, and YOLO v8m each need 8.2, 28.6, and 79.1 GFLOPs, respectively YOLO v8 training Table 4. 1.

Table 4.1: YOLO V8 Training Parameters**.**

Parameter	**Value**
Epochs	300
Batch Size	32
Optimizer	SGD
Learning Rate	0.01
Decay	0.0005
Workers	8
Patience	50

The investigation of anchor-free detections is an area of particular interest. Anchor-free detection, in contrast to conventional anchor-based approaches, predicts an object's center directly rather than an offset from predetermined anchor boxes. To find objects with particular scales and aspect ratios, anchor-based approaches employ predetermined boxes of varying sizes called anchor boxes. During detection, these anchor boxes are tiled throughout the image. The network generates probabilities and properties, such as background details, Intersection over Union (IoU) scores, and offsets for each tiled box, in the anchor-free approach, The IOU's calculation is shown in. The anchor boxes are then adjusted using these characteristics. Multiple anchor boxes are not utilized in anchor-free detection since border box predictions are made only on object centers. This research aims to present a comprehensive analysis and evaluation of the performance and efficiency of different YOLO

v8 variants, considering the use of anchor-free detection. The results of this study contribute to the advancement of object detection methodologies, particularly in the context of efficiency and accuracy. The findings hold potential implications for various computer vision applications and can pave the way for further research in this domain.

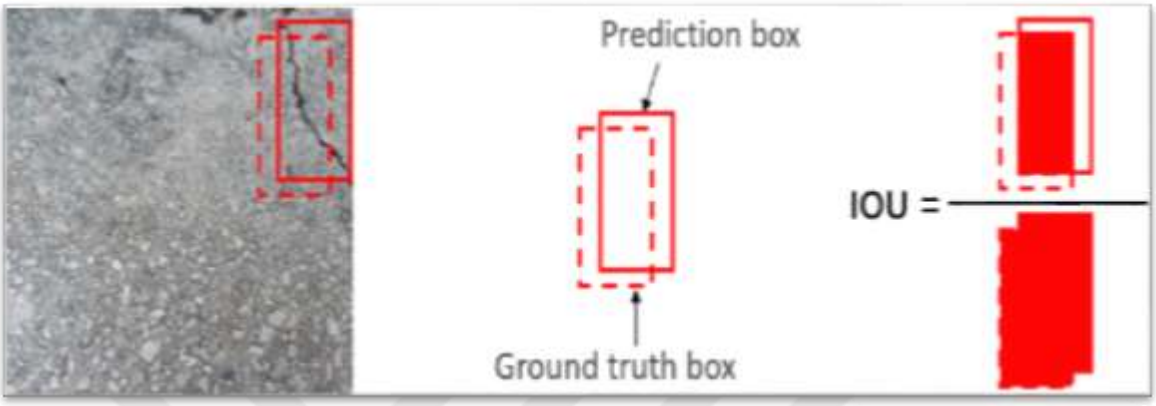


Figure 4.2: IOU Calculation Process.

4.2 PERFORMANCE EVALUATION METRICS

During this phase of the experiment, six evaluation indices: Accuracy, Recall, Precision, F1-Score, mean Average Precision (mAP), and mAP50-95, were used to conduct an extensive evaluation of the network's performance. These indicators were used to provide a comprehensive and exact evaluation of the model's performance. For analysis, the crack test findings were divided into four different states. Following is a definition of these states:

a. True Positives (TP): This statistic shows how many crack samples in the dataset were successfully detected.
b. True Negatives (TN): This metric measures the proportion of non-cracked samples that were correctly identified.
c. False Positives (FP): The number of classification errors where non-cracked samples were mistakenly categorized as crack samples.
d. False Negatives (FN): This term describes the number of misclassifications in which samples with cracks were incorrectly classified as samples without cracks.

A confusion matrix is included in Table 4. 2 to more clearly demonstrate these states. As a measure of the total learning performance for the complete collection of test images,

accuracy is the proportion of properly detected images among all the test images. The following is the calculation method:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

Nevertheless, accuracy might not offer an accurate assessment of the mistake when it comes to false detection. In other words, it becomes difficult to determine with accuracy how accurate false examples are. The evaluation measures of Recall and Precision are provided to overcome this restriction. These results provide a more thorough evaluation of the model's effectiveness in detecting false positives and false negatives. Recall takes into account the original sample. It displays how many correctly detected positive instances there are in the samples. The following is the calculation method:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.3}$$

Precision takes the result of the prediction into consideration. It shows the proportion of samples that actually are positive among those that are projected to be positive. The following is the calculation method:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.4}$$

Recall and Precision are metrics that, in turn, describe how well the dataset's positive and negative samples were learned from. However, there may be a conflicting relationship between these two measurements. For instance, a significant amount of data may be incorrectly labeled as positive samples when the network has an inadequate learning effect on negative samples. This results in a significantly lower Precision rating while producing a greater Recall value. On the other hand, a heavy focus on negative samples can result in a situation where the Precision value is higher but the Recall value is lower. Recall and Precision are thus trade-offs, and their application should be carefully assessed in light of the model's unique learning behavior on positive and negative data. We used the F1 score measure, which effectively includes both Recall and Precision measurements, to conduct a thorough analysis of the network's learning performance on the training data. The F1 score provides an improved understanding of the network's learning efficiency since it takes into

account both the precision with which positive samples are distinguished from false positives and the recall with which positive samples are properly identified. We seek to provide a more thorough and nuanced evaluation of the network's overall learning skills on the training data by using the F1 score. The following is the calculation method:

$$\mathrm{F1} - \ \mathrm{score} = \frac{2Recall \times Precision}{Recall + Precision} \tag{4.5}$$

mAP (mean Average Precision) is an important statistic used to assess the precision of crack detection in YOLO v8. It considers both precision and recall at different confidence thresholds, providing a comprehensive measure of the model's efficiency in identifying cracks. The mAP50-95 evaluation measure uses an IoU threshold range of 0.50 to 0.95 to evaluate the model's performance in detecting items with varying degrees of spatial overlap between predicted and actual bounding boxes.

Table 4.2: Matrix of Confusion for the Four Detection Results.

Ground truth	**Crack (True)**	**Non-Crack (False)**
Crack (True)	TP (True positive)	FN (False negative)
Non-Crack (False)	FP (False positive)	TN (True negative)

The confusion matrix quantifies the model's accurate and inaccurate classifications, enabling a thorough evaluation of the model's performance. Researchers and professionals can examine the model's accuracy, recall, precision, and other performance metrics by assessing the TP, TN, FP, and FN values, which helps in the fine-tuning and optimization of the object detection model in YOLO v8.

4.3 PERFORMANCE OF YOLO V8 MODELS

The results of the 300 training epochs for the YOLO v8n, YOLO v8s, and YOLO v8m models on the bridge cracking dataset are shown in Figure 4. 3, which also shows the mAP50 curves for each of the three YOLO v8 models.

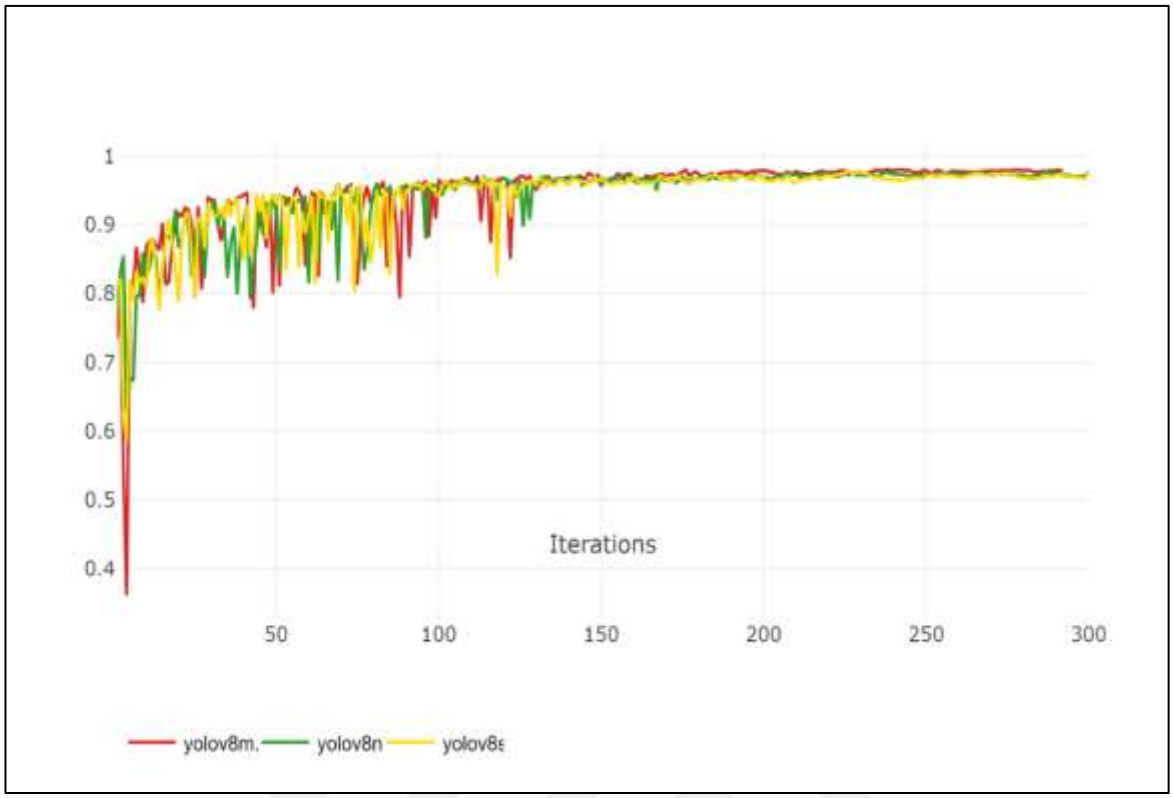


Figure 4.3: Comparison of the YOLO v8 Nano, Small, and Medium Models at mAP50.

The results of the 300 training epochs for the YOLO v8n, YOLO v8s, and YOLO v8m models on the bridge cracking dataset are shown in the Figures below, which also show the precision and recall curves for each of the three YOLO v8 models.

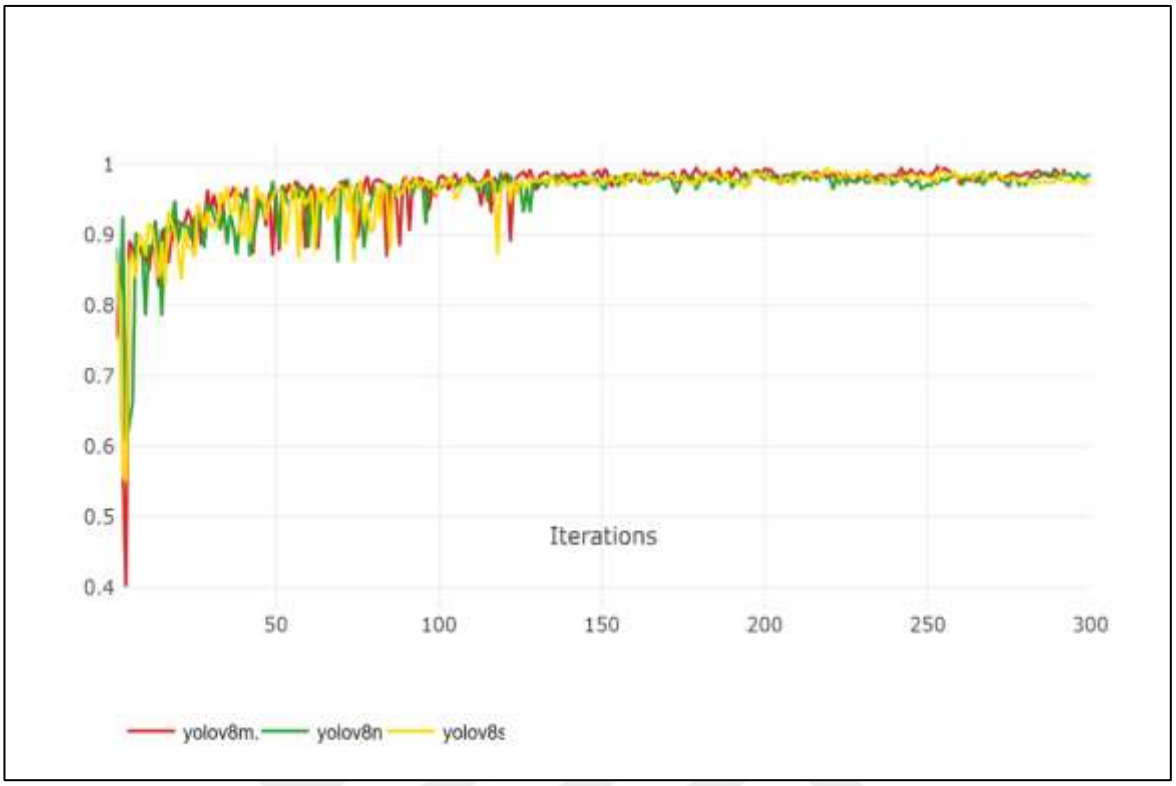


Figure 4.4: Precision Curves in the YOLO v8n, YOLO v8s, and YOLO v8m Models.

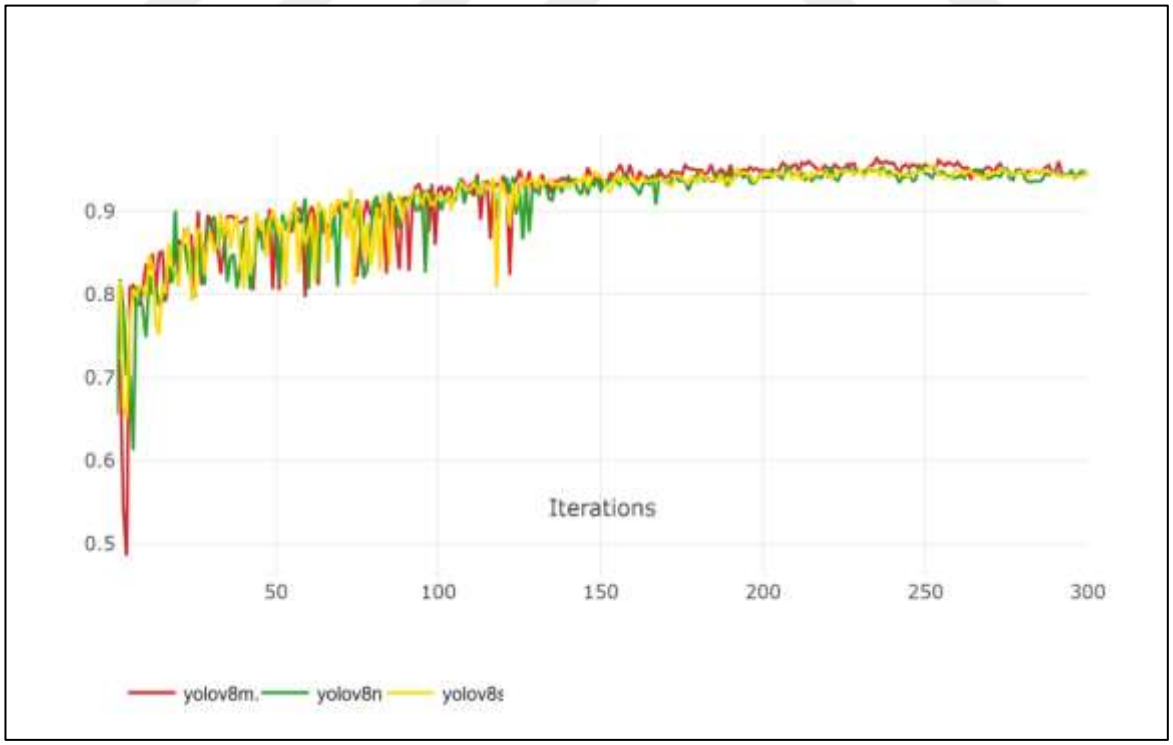


Figure 4.5: Recall Curves in the YOLO v8n, YOLO v8s, and YOLO v8m Models.

On the bridge cracking dataset, the results of 300 training epochs for the YOLO v8n, YOLO v8s, and YOLO v8m models are shown in Figure 4. 6 along with the mAP50-95 curves for each model. The performance of the models is evaluated over a variety of Intersection over Union (IoU) thresholds, specifically from 0.50 to 0.95, using the mAP50-95 metric, which is of utmost significance.

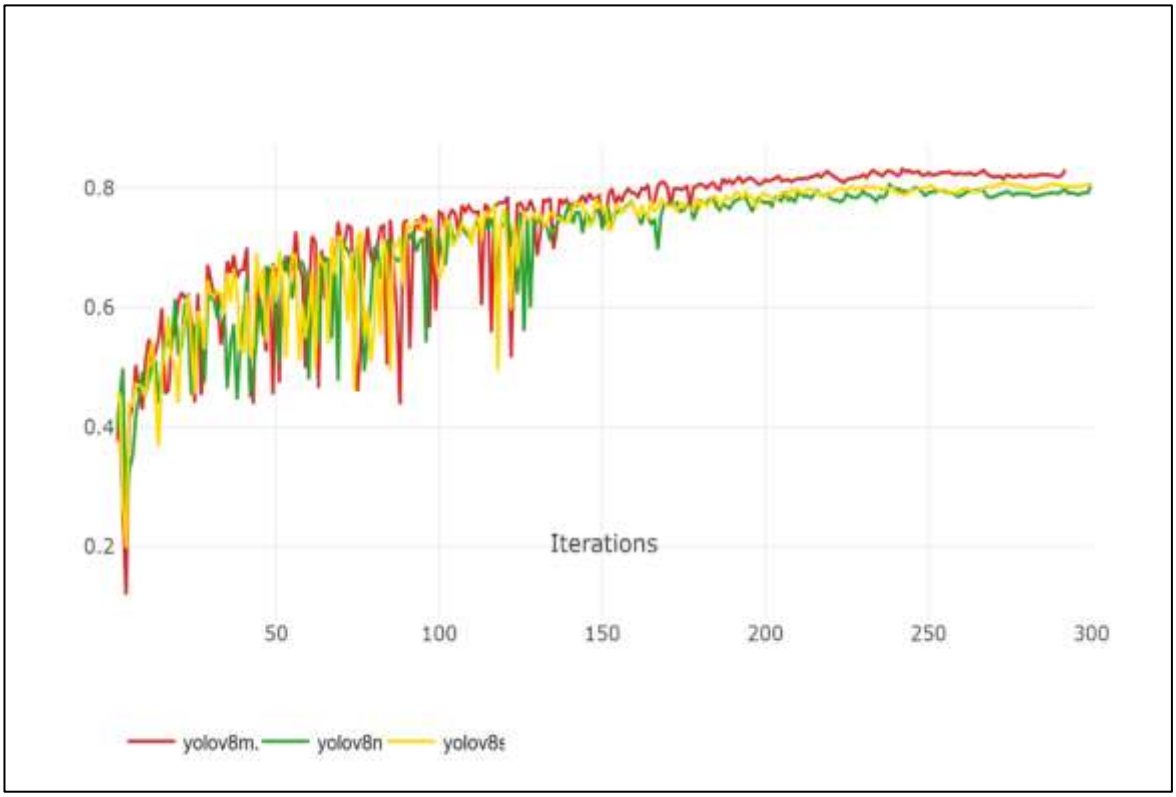


Figure 4.6: mAP50-95 Curves in the YOLO v8n, YOLO v8s, and YOLO v8m Models.

We included an early stopping mechanism with a parameter value of 60 to prevent overfitting throughout the training process, which lasted 300 epochs. During these 300 epochs, the training loss function was tracked continually, measuring the difference between the model's predictions and the actual data. The early termination parameter made sure that the training process would cease after 60 consecutive epochs if the training loss did not improve. In Figure 4. 6 shows the training loss function for YOLO v8 models.

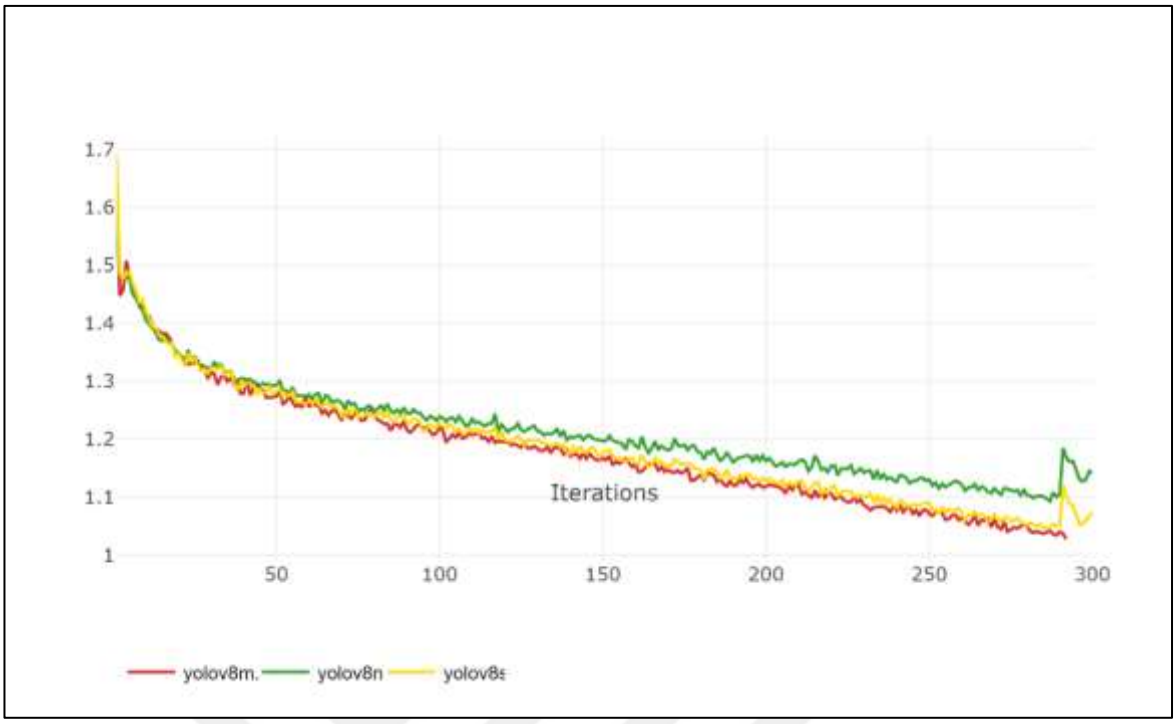


Figure 4.7: Training Loss Curves in the YOLO v8n, YOLO v8s, and YOLO v8m Models.

The presented curves show how different YOLO v8 algorithm models performed over time. The rising curves illustrate the models' development, making the metric an essential tool for evaluating the effectiveness and advancement of the YOLO v8 algorithm.

Table 4.3: Comparison of YOLO V8 Models Results.

Model	Precision	Recall	mAP50	mAP50-95	Loss	Parameters	FPS
YOLO v8n	98.5%	94.5%	97.7%	80.6%	1.04%	3.2 M	95.24
YOLO v8s	97.8%	95%	97.6%	80.8%	1.00%	11.2 M	64.10
YOLO v8m	99.4%	94.7%	97.9%	83.1%	0.98%	25.9 M	31.45

The results are shown in Table 4. 3 show that the YOLO v8 model algorithm has displayed impressive performance in terms of accuracy and computational speed. However, there is still room for improvement in terms of its capacities. To get even better outcomes, ongoing research can concentrate on improving the model architecture, optimizing hyperparameters, and investigating cutting-edge training methods. Through such efforts, the algorithm's

overall performance will be improved and its place as a compelling option for real-time object detection jobs, providing a great balance between accuracy and computing efficiency, will be cemented.

Table 4.4: The Comparison of the Proposed Work with Existing Literature.

Article	Technique	Accuracy
[84]	Recurrent Neural Network (RNN)	92.7%
[85]	Support Vector Machine (SVM)	87.00%
Proposed	You Only Look Once (YOLO)	98.54%

It is crucial to thoroughly compare our planned research with already published studies in the sector. By doing this, we can evaluate our research's innovation and contributions while also comprehending how it compares to the state-of-the-art. This comparison enables us to pinpoint potential enhancements and distinctive features of our methodology, ensuring that our work contributes to the body of current knowledge and fills any gaps or shortcomings in earlier studies. Such a comparison examination also confirms the significance and applicability of our proposed work in the larger scientific setting.

5. DISCUSSION AND CONCLUSION

5.1 DISCUSSION

The results of this study show how successful and efficient the suggested machine-learning method is for detecting cracks in concrete bridges. The 98.54% accuracy rate attained demonstrates the YOLO v8 algorithms' capacity to properly locate cracks in the gathered high-resolution UAV photos. The model's success in limiting false positives and precisely identifying real cracks is further supported by the high F1 score of 97%. These findings demonstrate the YOLO algorithm and other deep learning techniques' potential to revolutionize the inspection of concrete bridges. Manual crack identification in concrete bridges has long been a problem, wasting resources and creating inefficiencies. Recurrent neural networks and support vector machines, among other methods, have been used in several research studies to solve this issue. For instance, a study using RNN attained a precision of 92.7% [84], while a study using SVM attained a precision of 87.00% [85]. These techniques have shown potential in automating crack identification, but compared to the suggested YOLO-based strategy, they might be slower and less accurate in real-time. Faster R-CNN [81], typically performs better than YOLO in terms of accuracy, especially for small objects and complicated scenes. This is due to its two-stage architecture, which enables better handling of diverse item sizes and more accurate localization. YOLO is a single-stage object identification system that divides the input image into a grid to make it easier to anticipate bounding boxes and associated class probabilities directly within each grid cell. By using a unipartite approach, YOLO has a significant speed advantage over the Faster R-CNN architecture. The model's ability to recognize objects in real time is demonstrated by its ability to operate at a speed of 95.24 frames per second (fps). This speed makes the technique ideal for extensive examinations as well as helping with quick decision-making. So, YOLO is especially well suited for real-time applications that require quick processing. The advanced CSPDarknet-53 backbone architecture increases the model's complexity while also improving accuracy. The amount of time and resources needed for training could be impacted by complexity. The quality and diversity of the training dataset are crucial to YOLO's effectiveness, as it is for many other deep learning techniques. Insufficient or biased training data may produce unsatisfactory results. Even though the proposed technique is

excellent at real-time detection, outside factors like severe weather or dim lighting could impair its performance.

5.2 FUTURE RECOMMANDATION

There are still a lot of perspectives that can later be improved. Several studies aim to enhance the models through the amalgamation of several data modalities, such as heat or infrared photography, with the high-resolution visual data obtained by UAVs. When dealing with poor illumination or weather circumstances, this effort toward multimodal data integration holds the possibility of improving crack detection's precision and robustness. Tests for enhanced robustness by testing the suggested model thoroughly in a range of environmental settings, such as bad weather, variable lighting situations, and seasonal fluctuations. This will give information about how the model performs in actual scenarios. deployment in genuine environments via Working together with organizations or businesses that manage infrastructure to implement and test the model during actual bridge inspections. Getting comments from professionals will yield insightful information for more. Automated anomaly identification increases the model's ability to identify additional abnormalities or flaws, such as spalling, corrosion, or structural deformations, in addition to cracks. This might offer a thorough remedy for the entire bridge health check. By enlarging the dataset to include a wider range of concrete bridges with varied levels of deterioration and structural complexity, the dataset can be made more diverse. This can ensure that the model can be used in many styles of concrete bridges.

5.3 CONCLUSION

Bridge surface crack detection is essential for monitoring the health of a bridge and is essential to automating infrastructure and civil engineering projects. Manual visual inspection of bridges may require a lot of labor, and working in some locations during bad weather can be risky. This research proposed an object detection technique to find cracks in bridge surfaces about that. In this master's thesis, a thorough and creative solution to the time-consuming and resource-intensive manual crack detection method in concrete bridge surfaces is described. A major improvement has been made in automating the detection process, ensuring accuracy, efficiency, and improved resource utilization. This has been accomplished by utilizing the capabilities of unmanned aerial vehicles (UAVs) and modern

machine learning algorithms. Traditional ground-based approaches have limits, but the addition of high-resolution imagery acquired by UAVs enables an extensive and complete evaluation of concrete bridge surfaces. With an accuracy rate of 98.54%, the use of deep learning technologies, in particular the YOLO (You Only Look Once) algorithm with the YOLO v8 models, has made it possible to attain extraordinary accuracy levels. This result highlights the potential for machine learning to significantly improve the accuracy and precision of crack detection, hence lowering the chance of missing important structural concerns. High metrics were consistently found across all of the evaluation parameters when the performance of the suggested machine-learning model was evaluated. The model can effectively identify cracks while reducing false positives, as seen by the F1 score of 97%, which displays a remarkable balance between precision and recall. Additionally, the mean average precision (mAP) of 97.90% demonstrates the model's accuracy and reliability. An important accomplishment is the YOLO v8 model's operational speed of 95.42 frames per second (fps), which makes real-time crack detection feasible and useful. The entire effectiveness and safety of bridge maintenance and inspection operations are boosted by this speed, which enables prompt decision-making and action. This paper highlights the potential effects of the particular automated crack detection approach it proposes and validates for the discipline of civil engineering and infrastructure management. The solution provides a way to significantly increase operational effectiveness, improve safety, and use resources wisely. Traditional inspection methods could change as a result of the combination of UAVs and machine learning, which would streamline procedures, lessen human error, and enable data-driven decision-making.

REFERENCES

[1] G. M. Calvi *et al.*, "Once upon a time in Italy: The tale of the Morandi Bridge," *Structural Engineering International*, vol. 29, no. 2, pp. 198–217, Dec 2019, doi: 10.1080/10168664.2018.1558033.

[2] L. Hebbache, D. Amirkhani, M. S. Allili, N. Hammouche, and J.-F. Lapointe, "Leveraging saliency in single-stage multi-label concrete defect detection using unmanned aerial vehicle imagery," *Remote Sens (Basel)*, vol. 15, no. 5, p. 1218, Feb 2023, doi: 10.3390/rs15051218.

[3] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung, "Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle," *Sensors*, vol. 18, no. 6, p. 1881, Jun 2018, doi: 10.3390/s18061881.

[4] M. Mandirola, C. Casarotti, S. Peloso, I. Lanese, E. Brunesi, and I. Senaldi, "Use of UAS for damage inspection and assessment of bridge infrastructures," *International Journal of Disaster Risk Reduction*, vol. 72, p. 102824, Apr 2022, doi: 10.1016/j.ijdrr.2022.102824.

[5] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 8085–8094, Sept 2022, doi: 10.1109/JSTARS.2022.3206399.

[6] S. Abdelkhalek and T. Zayed, "Comprehensive inspection system for concrete bridge deck application: Current situation and future needs," *Journal of Performance of Constructed Facilities*, vol. 34, no. 5, p. 03120001, Jun 2020, doi: 10.1061/(ASCE)CF.1943-5509.000148.

[7] K. H. Kim, M. S. Nam, H. H. Hwang, and K. Y. Ann, "Prediction of Remaining Life for Bridge Decks Considering Deterioration Factors and Propose of Prioritization Process for Bridge Deck Maintenance," *Sustainability*, vol. 12, no. 24, Dec 2020, doi: 10.3390/su122410625.

[8] Z. Ameli, Y. Aremanda, W. A. Friess, and E. N. Landis, “Impact of UAV Hardware Options on Bridge Inspection Mission Capabilities,” *Drones*, vol. 6, no. 3, Feb 2022, doi: 10.3390/drones6030064.

[9] V. P. Golding, Z. Gharineiat, H. S. Munawar, and F. Ullah, “Crack detection in concrete structures using Deep Learning,” *Sustainability*, vol. 14, no. 13, p. 8117, Jul 2022, doi: 10.3390/su14138117.

[10] S. Dorafshan and H. Azari, “Evaluation of bridge deck s with overlays using impact echo, a deep learning approach,” Autom Constr, vol. 113, p. 103133, May 2020, doi: 10.1016/j.autcon.2020.103133.

[11] S. Dorafshan and H. Azari, “Deep learning models for bridge deck evaluation using impact echo,” *Constr Build Mater*, vol. 263, p. 120109, Dec 2020, doi: 10.1016/j.conbuildmat.2020.120109.

[12] W. Jiang, M. Liu, Y. Peng, L. Wu, and Y. Wang, “HDCB-Net: A Neural Network With the Hybrid Dilated Convolution for Pixel-Level Crack Detection on Concrete Bridges,” *IEEE Trans Industr Inform*, vol. 17, no. 8, pp. 5485–5494, Oct 2020, doi: 10.1109/TII.2020.3033170.

[13] Y.-B. Yang, J. P. Yang, Y. Wu, and B. Zhang, “Brief on the Works Conducted by Yang and and Co-workers,” *Vehicle scanning method for bridges*. John Wiley & Sons, USA, 2019.

[14] P. Prasanna *et al.*, “Automated crack detection on concrete bridges,” *IEEE Transactions on automation science and engineering*, vol. 13, no. 2, pp. 591–599, Oct 2014, doi: 10.1109/TASE.2014.2354314.

[15] Z. He, W. Li, H. Salehi, H. Zhang, H. Zhou, and P. Jiao, “Integrated structural health monitoring in bridge engineering,” *Autom Constr*, vol. 136, p. 104168, Apr 2022, doi: 10.1016/j.autcon.2022.104168.

[16] A. C. Estes, “Bridge maintenance, safety, management, and life-cycle optimization.” Taylor & Francis, Jun 2011, doi: 10.1080/15732479.2011.555376.

[17] A. Moghadam, M. AlHamaydeh, and R. Sarlo, “Bridge-weigh-in-motion approach for simultaneous multiple vehicles on concrete-box-girder bridges,” *Autom Constr*, vol. 137, p. 104179, May 2022, doi: 10.1016/j.autcon.2022.104179.

[18] G. Zhang, Y. Liu, J. Liu, S. Lan, and J. Yang, “Causes and statistical characteristics of bridge failures: A review,” *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 9, no. 3, pp. 388–406, Jun 2022, doi: 10.1016/j.jtte.2021.12.003.

[19] J. R. Davidson, “Reliability and structural integrity,” in *Anniv. Meeting of the Soc. of Eng. Sci.*, Apr 1973.

[20] A. Khan, S. Gupta, and S. K. Gupta, “Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques,” *International journal of disaster risk reduction*, vol. 47, p. 101642, Aug 2020, doi: 10.1016/j.ijdrr.2020.101642.

[21] L. E. Mansuri and D. A. Patel, “Artificial intelligence-based automatic visual inspection system for built heritage,” *Smart and Sustainable Built Environment*, vol. 11, no. 3, pp. 622–646, Nov 2022, doi: 10.1108/SASBE-09-2020-0139.

[22] J. J. Kim, A.-R. Kim, and S.-W. Lee, “Artificial neural network-based automated crack detection and analysis for the inspection of concrete structures,” *Applied Sciences*, vol. 10, no. 22, p. 8105, Nov 2020, doi: 10.3390/app10228105.

[23] T. Yamaguchi, S. Nakamura, and S. Hashimoto, “An efficient crack detection method using percolation-based image processing,” in *2008 3rd IEEE Conference on Industrial Electronics and Applications*, IEEE, Singapore, 2008, pp. 1875–1880.

[24] M. Pal, P. Palevičius, M. Landauskas, U. Orinaitė, I. Timofejeva, and M. Ragulskis, “An overview of challenges associated with automatic detection of concrete cracks in the presence of shadows,” *Applied Sciences*, vol. 11, no. 23, p. 11396, Dec 2021, doi: 10.3390/app112311396.

[25] N. J. Bertola and E. Brühwiler, “Risk-based methodology to assess bridge condition based on visual inspection,” *Structure and Infrastructure Engineering*, vol. 19, no. 4, pp. 575–588, Dec 2023, doi: 10.1080/15732479.2021.1959621.

[26] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, “Comparative analysis of image binarization methods for crack identification in concrete structures,” *Cem Concr Res*, vol. 99, pp. 53–61, Sept 2017, doi: 10.1016/j.cemconres.2017.04.018.

[27] P. D. Krauss and E. A. Rogalla, "Transverse cracking," *in newly constructed bridge decks,* no. Project 12-37 FY’92, DC, United States, 1996.

[28] D. Dias-da-Costa, J. Valença, E. Júlio, and H. Araújo, “Crack propagation monitoring using an image deformation approach,” *Struct Control Health Monit*, vol. 24, no. 10, p. e1973, Oct 2017, doi: 10.1002/stc.1973.

[29] Y.-S. Yang, C.-M. Yang, and C.-W. Huang, “Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis,” *Advances in Engineering Software*, vol. 83, pp. 99–108, May 2015, doi: 10.1016/j.advengsoft.2015.02.005.

[30] J. Deng, Y. Lu, and V. C. Lee, “Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 4, pp. 373–388, Apr 2020, doi: 10.1111/mice.12497.

[31] M. H. Rafiei and H. Adeli, “A novel machine learning-based algorithm to detect damage in high-rise building structures,” *The Structural Design of Tall and Special Buildings*, vol. 26, no. 18, p. e1400, Dec 2017, doi: 10.1002/tal.1400.

[32] G. C. Manos, M. Theofanous, and K. Katakalos, “Numerical simulation of the shear behaviour of reinforced concrete rectangular beam specimens with or without FRP-strip shear reinforcement,” *Advances in Engineering Software*, vol. 67, pp. 47–56, Jan 2014, doi: 10.1016/j.advengsoft.2013.08.001.

[33] C. Wu *et al.*, “Collapse of a nonductile concrete frame: Shaking table tests,” *Earthq Eng Struct Dyn*, vol. 38, no. 2, pp. 205–224, Feb 2009, doi: 10.1002/eqe.853.

[34] Y.-S. Yang, C.-M. Yang, and T.-J. Hsieh, “GPU parallelization of an object-oriented nonlinear dynamic structural analysis platform,” *Simul Model Pract Theory*, vol. 40, pp. 112–121, Jan 2014, doi: 10.1016/j.simpat.2013.09.004.

[35] 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, Sep 2003, doi: 10.1061/(ASCE)0887-3801(2003)17:4(255).

[36] R. S. Adhikari, O. Moselhi, and A. Bagchi, “Image-based retrieval of concrete crack properties for bridge inspection,” *Autom Constr*, vol. 39, pp. 180–194, Apr 2014, doi: 10.1016/j.autcon.2013.06.011.

[37] B. Y. Lee, Y. Y. Kim, S.-T. Yi, and J.-K. Kim, “Automated image processing technique for detecting and analysing concrete surface cracks,” *Structure and Infrastructure Engineering*, vol. 9, no. 6, pp. 567–577, Dec 2013, doi: 10.1080/15732479.2011.593891.

[38] A. Arena, C. Delle Piane, and J. Sarout, “A new computational approach to cracks quantification from 2D image analysis: Application to micro-cracks description in rocks,” *Comput Geosci*, vol. 66, pp. 106–120, May 2014, doi: 10.1016/j.cageo.2014.01.007.

[39] H.-N. Nguyen, T.-Y. Kam, and P.-Y. Cheng, “An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack,” *J Signal Process Syst*, vol. 77, pp. 221–240, Dec 2014, doi: 10.1007/s11265-013-0813-8.

[40] T. C. S. No, “Guidebook on non-destructive testing of concrete structures,” *Training Course Series*, 2002.

[41] M. Brigante and M. A. Sumbatyan, “Acoustic methods for the nondestructive testing of concrete: A review of foreign publications in the experimental field,” *Russian Journal of Nondestructive Testing*, vol. 49, pp. 100–111, May 2013, doi: 10.1134/S1061830913020034.

[42] H. Wiggenhauser, C. Köpp, J. Timofeev, and H. Azari, “Controlled creating of cracks in concrete for non-destructive testing,” *J Nondestr Eval*, vol. 37, pp. 1–9, Aug 2018, doi: 10.1007/s10921-018-0517-x.

[43] P. Sharma, A. Kumar, S. Singh, and S. Prof. Mandeep, “Crack Detection of Ferromagnetic Materials through Non-Destructive Testing Methodology,” *International Journal Of Modern Engineering Research (IJMER)*, 2015, doi: 10.6084/M9.FIGSHARE.1290951.V1.

[44] M. N. Alhebrawi, H. Huang, and Z. Wu, “Artificial intelligence enhanced automatic identification for concrete cracks using acoustic impact hammer testing,” *J Civ Struct Health Monit*, vol. 13, no. 2–3, pp. 469–484, 2023.

[45] F. Jalinoos, “NDE Showcase for Bridge Inspectors,” *Public roads*, vol. 73, no. 2, pp. 10–13, Oct 2009.

[46] G. Zhai, Y. Narazaki, S. Wang, S. A. V Shajihan, and B. F. Spencer Jr, “Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks,” *Smart Struct Syst*, vol. 29, no. 1, pp. 237–250, Sept 2022.

[47] S. S. Khedmatgozar Dolati, N. Caluk, A. Mehrabi, and S. S. Khedmatgozar Dolati, “Non-destructive testing applications for steel bridges,” *Applied Sciences*, vol. 11, no. 20, p. 9757, Oct 2021, doi: 10.3390/app11209757.

[48] B. Ovuoba and G. S. Prinz, “Investigation of residual fatigue life in shear studs of existing composite bridge girders following decades of traffic loading,” *Eng Struct*, vol. 161, pp. 134–145, Apr 2018, doi: 10.1016/j.engstruct.2018.02.018.

[49] T. W. Ryan, J. E. Mann, Z. M. Chill, and B. T. Ott, “Bridge inspector’s reference manual (BIRM),” *Publication No. FHWA NHI*, pp. 12–49, VA United States, Mar. 2023.

[50] D. T. Gillins, C. Parrish, M. N. Gillins, and C. Simpson, “Eyes in the sky,” *Bridge inspections with unmanned aerial vehicles,* United States, Feb. 2018.

[51] I. Hernandez, T. Fields, and J. Kevern, “Overcoming the challenges of using unmanned aircraft for bridge inspections,” in *AIAA Atmospheric Flight Mechanics Conference*, USA, 2016, p. 3396.

[52] Y.-A. Hsieh and Y. J. Tsai, “Machine learning for crack detection: Review and model performance comparison,” *Journal of Computing in Civil Engineering*, vol. 34, no. 5, p. 04020038, Jul 2020, doi: 10.1061/(ASCE)CP.1943-5487.000091.

[53] M. S. Kaseko and S. G. Ritchie, “A neural network-based methodology for pavement crack detection and classification,” *Transp Res Part C Emerg Technol*, vol. 1, no. 4, pp. 275–291, Dec 1993, doi: 10.1016/0968-090X(93)90002-W.

[54] Y. Fujita, K. Shimada, M. Ichihara, and Y. Hamamoto, “A method based on machine learning using hand-crafted features for crack detection from asphalt pavement surface images,” in *Thirteenth International Conference on Quality Control by Artificial Vision 2017*, SPIE, Tokyo, 2017, pp. 117–124.

[55] N.-D. Hoang, “An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction,” *Advances in Civil Engineering*, vol. 2018, Jun 2018, doi: 10.1155/2018/7419058.

[56] S. J. Schmugge *et al.*, “Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Lake Placid, NY, USA, 2016, pp. 1–7.

[57] M. Sahu and R. Dash, “A survey on deep learning: convolution neural network (CNN),” in *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2*, Springer, BBSR, India, 2021, pp. 317–325.

[58] R. Kaur and S. Singh, “A comprehensive review of object detection with deep learning,” *Digit Signal Process*, p. 103812, Jan 2022, doi: 10.1016/j.dsp.2022.103812.

[59] G. X. Hu, B. L. Hu, Z. Yang, L. Huang, and P. Li, “Pavement crack detection method based on deep learning models,” *Wirel Commun Mob Comput*, vol. 2021, pp. 1–13, May 2021, doi: 10.1155/2021/5573590.

[60] H. Li, C. Wang, J. Wang, and Y. Zhang, “Crack detection based on deep learning: a method for evaluating the object detection networks considering the random fractal of

crack," *Struct Health Monit*, vol. 22, no. 4, pp. 2547–2564, Feb 2023, doi: 10.1177/14759217221123485.

[61] C. Anusha and P. S. Avadhani, "Object detection using deep learning," *Int J Comput Appl*, vol. 182, no. 32, pp. 18–22, Dec 2018.

[62] E. Hassan, Y. Khalil, and I. Ahmad, "Learning feature fusion in deep learning-based object detector," *Journal of Engineering*, vol. 2020, pp. 1–11, May 2020, doi: 10.1155/2020/7286187.

[63] W. Liu *et al.*, "Ssd: Single shot multibox detector," *in European conference on computer vision. Springer*, 2016.

[64] R. Araki, T. Onishi, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Mt-dssd: Deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Paris, France, 2020, pp. 10487–10493.

[65] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo algorithm developments," *Procedia Comput Sci*, vol. 199, pp. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.

[66] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul 2020, doi: 10.1016/j.neucom.2020.01.085.

[67] S. J. Schmugge *et al.*, "Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Lake Placid, NY, USA, 2016, pp. 1–7.

[68] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)*, IEEE, Phoenix, AZ, USA, 2016, pp. 3708–3712, doi: 10.1109/ICIP.2016.7533052.

[69] S. Park, S. Bang, H. Kim, and H. Kim, “Patch-based crack detection in black box images using convolutional neural networks,” *Journal of Computing in Civil Engineering*, vol. 33, no. 3, p. 04019017, Feb 2019, doi: 10.1061/(ASCE)CP.1943-5487.0000831.

[70] S. Li and X. Zhao, “Image-based concrete crack detection using convolutional neural network and exhaustive search technique,” *Advances in civil engineering*, vol. 2019, Apr 2019.

[71] K. Chen, A. Yadav, A. Khan, Y. Meng, and K. Zhu, “Improved crack detection and recognition based on convolutional neural network,” *Modelling and simulation in engineering*, vol. 2019, pp. 1–8, Oct 2019.

[72] S. Zhou and W. Song, “Concrete roadway crack segmentation using encoder-decoder networks with range images,” *Autom Constr*, vol. 120, p. 103403, Dec 2020, doi: 10.1016/j.autcon.2020.103403.

[73] R. Ali, J. H. Chuah, M. S. A. Talip, N. Mokhtar, and M. A. Shoaib, “Structural crack detection using deep convolutional neural networks,” *Autom Constr*, vol. 133, p. 103989, Jan 2022, doi: 10.1016/j.autcon.2021.103989.

[74] H. M. Madani and K. M. Dolatshahi, “Strength and stiffness estimation of damaged reinforced concrete shear walls using crack patterns,” *Struct Control Health Monit*, vol. 27, no. 4, p. e2494, Feb 2020.

[75] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, “Automatic pixel-level crack detection and measurement using fully convolutional network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1090–1109, Aug 2018.

[76] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, AYT, Turkey, Ieee, 2017, pp. 1–6.

[77] S. M. Optimization, “A fast algorithm for training support vector machines,” *CiteSeerX*, vol. 10, no. 1.43, p. 4376, 1998.

[78] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv Neural Inf Process Syst*, vol. 28, 2015.

[79] 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*, Las Vegas, NV, USA, 2016, pp. 779–788.

[80] J. Zhang, S. Qian, and C. Tan, “Automated bridge crack detection method based on lightweight vision models,” *Complex & Intelligent Systems*, vol. 9, no. 2, pp. 1639–1652, 2023, doi: 10.1007/s40747-022-00876-6.

[81] G. Suh and Y.-J. Cha, “Deep faster R-CNN-based automated detection and localization of multiple types of damage*,” in Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, SPIE, Denver, Colo, United States, 2018, pp. 197–204.

[82] S. Jiang and J. Zhang, “Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 6, pp. 549–564, 2020.

[83] V. Mandal, L. Uong, and Y. Adu-Gyamfi, “Automated Road Crack Detection Using Deep Convolutional Neural Networks,” in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, 2018, pp. 5212–5215. doi: 10.1109/BigData.2018.8622327.

[84] S. Moon, S. Chung, and S. Chi, "Bridge damage recognition from inspection reports using NER based on recurrent neural network with active learning," *Journal of Performance of Constructed Facilities*, vol. 34, no. 6, p. 04020119, 2020.

[85] S. Chanda *et al.*, “Automatic bridge crack detection–a texture analysis-based approach,” in *Artificial Neural Networks in Pattern Recognition: 6th IAPR TC 3 International Workshop, ANNPR 2014,* Montreal, QC, Canada, October 6-8, 2014. Proceedings 6, Springer, 2014, pp. 193–203.