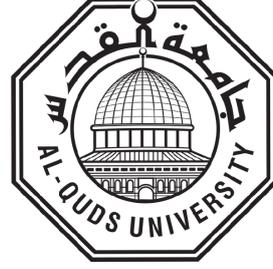


AUTOMATIC DETECTION OF ROAD ANOMALIES USING COMPUTER VISION AND DEEP LEARNING



AUTOMATIC DETECTION OF ROAD ANOMALIES USING COMPUTER VISION AND DEEP LEARNING

By:

Rasha Derar Saffarini

Supervisor:

Dr. Faisal Khamayseh

Co-Supervisor:

Dr. Muath Sabha

Dr. Derar Eleyan

This Dissertation proposal was submitted in Partial Fulfillment of the

Requirements for the PhD

In Engineering of Information Technology

Joint PhD Program- AAUP, PPU and QU

Deanship of Scientific Research and Graduate Studies

Ramallah - Palestine

[December] [2024]



Joint PhD Program- AAUP, PPU and QU

[December] [2024]

AUTOMATIC DETECTION OF ROAD ANOMALIES USING COMPUTER VISION AND DEEP LEARNING

By:

Rasha Derar Saffarini

Rasha Saffarini

Signature of Author

Committee Members

Signature and Date

Dr. Faisal Khamayseh.(Chairman)

Faisal Khamayseh 8/3/2025

Dr. Muath Sabha (Co-Advisor)

Muath Sabha 8/3/2025

Prof. Derar Eleyan (Co-Advisor)

D. Eleyan 8/3/2025

Dr. Alaa Halawani (Member)

Alaa Halawani 13/03/2025

Dr. Alaa M. Alhalees (External Examiner)

A. Alhalees 13/3/2025

Dr. Khader Mohammad (External Examiner)

Khader M. Mohammad
13/3/2025

[December] [2024]

تفويض

أنا الموقع أدناه، أتعهد بمنح الجامعات الشريكة في برنامج الدكتوراه في هندسة تكنولوجيا المعلومات حرية التصرف في نشر محتوى الأطروحة الجامعية، بحيث تعود حقوق الملكية الفكرية للأطروحة الى الجامعات وفق القوانين والأنظمة والتعليمات المتعلقة بالملكية الفكرية وبراءة الاختراع.

المشرف الرئيس	الطالب
فيصل طالب خمائسه	رشا سفاريني
التوقيع والتاريخ <i>Faisal Khamaysch</i> 8/3/2025	الرقم الجامعي والتوقيع 201912959..... <i>Rasha Saffarini</i>

DEDICATION

Declaration

I declare that this thesis entitled "Automatic Detection of Road Anomalies using Computer Vision and Deep Learning" is my own work and has been composed solely by myself and does not contain any work from other researchers and has not been submitted for any other degree or scientific qualification, and I except that all the references are done correctly.

Dedication

Proudly, I dedicate my thesis to my parents, as I always feel their prayers in all aspects of my life. I also dedicate my thesis to my sisters, friends, colleagues who are always willing to provide any support.

ACKNOWLEDGMENTS

In completing this thesis, I want to express my deep gratitude to the many people who supported me during my academic journey.

First and foremost, I am grateful to my parents. My father, who has been by my side since I was young, provided unwavering support to help me reach and achieve this dream. My mother, who was my rock, offered me strength and encouragement during challenging times.

I am also thankful to my second mother, my mother-in-law, who has supported me and made sacrifices on my behalf.

To my life partner, my husband Khalil, your patience, understanding, and unlimited support during my studies were like a lifeline that I resorted to to complete the journey and reach the end of this path.

I must extend all thanks and love to my beloved brother Muhammad for his continuous support and willingness to help me when needed. To my sisters for their ongoing support and love.

My sincere appreciation also goes to my thesis supervisors, Dr. Faisal Khamayseh, Dr. Derar Elyan, and Dr. Moath Sabha, for their guidance, support, and dedication throughout the research and thesis work. Finally, I would like to thank Dr. Yousef Daraghmeh for his support and guidance.

TABLE OF CONTENTS

DEDICATION	i
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF ABBREVIATIONS	x
ABSTRACT	xi
Chapter One: Introduction	1
1.1 Problem Statement	2
1.2 Objectives	4
1.3 Research Questions	4
1.4 Significance of the Study	5
Chapter Two: Literature Review and Background	6
2.1 Road Anomaly	7
2.2 Road Anomaly Detection and Classification Techniques	8
2.2.1 Vision-based Road Anomaly Detection and Classification	8
2.2.2 Vibration-based Road Anomaly Detection and Classification	9
2.2.3 Applications of Road Anomaly Detection and Classification	9
2.3 Computer Vision	10
2.4 Object Detection	11
2.5 Region Proposal	13
2.6 Graph Segmentation and Graph Similarity	14
2.6.1 Graph Segmentation	14

2.6.2	Graph Similarity	15
2.7	GAN and DCGAN	16
2.8	Literature Review	18
2.9	Accelerometer-based Approaches	18
2.10	Image-based Approaches	19
2.11	Research Gap	31
Chapter Three: Methodology		33
3.1	Image Acquisition	33
3.2	Preprocessing	34
3.3	Proposed Detection and Classification Models	35
3.3.1	Dynamic Similar R-CNN model (DSR-CNN)	36
3.3.2	Dynamic Generative R-CNN (DGR-CNN)	44
3.4	Evaluation	54
Chapter Four: Results		55
4.1	Data Collection	56
4.2	Training	57
4.3	Developed Deep Learning Model Results	67
4.3.1	Model 1: DSR-CNN	69
4.3.2	Model 2: DGR-CNN	75
Chapter Five: Discussion		76
5.1	Limitations	81
Chapter Six: Conclusion and Future Work		81
6.1	Conclusion	81
6.2	Future work	83
References		84

LIST OF FIGURES

<u>Figure</u>	<u>Description</u>	<u>Page</u>
2.1	Algorithmic workflow in [6]	Page 22
2.2	Classification of predicted crack examples: True Positive-(a), False Positive-(b), False Negative-(c) obtained from YOLO v2 in [44]	Page 24
2.3	Proposed speed bump detection using convolutional neural network (CNN) archi- tecture in [22]	Page 27
2.4	Proposed speed bump detection using GAN in [50]	Page 29
2.5	Overall image acquisition flow and three-axis accelerations with a smartphone in [39]	Page 30
3.1	The overall methodology	Page 34
3.2	DJI Mavic Air 2 Drone	Page 35
3.3	Image before and after gamma correction	Page 36
3.4	Image before and after applying Gaussian blur	Page 36
3.5	Dynamic Similar R-CNN model	Page 37
3.6	Dynamic Generative R-CNN model	Page 38
3.7	Graph-based segmentation result	Page 39
3.8	Adjacency list of similar graphs	Page 41
3.9	DCGAN architecture	Page 45
3.10	Relu activation function	Page 47
3.11	Leaky Relu activation function	Page 47
3.12	Matrix to flatten	Page 48
3.13	Sigmoid function	Page 49
3.14	Generator architecture	Page 50
3.15	Up sampling process	Page 50
3.16	Tanh Function	Page 51

3.17	Confusion matrix	Page 54
4.1	Cracks	Page 58
4.2	Speedbumps	Page 59
4.3	Manholes	Page 60
4.4	Potholes	Page 61
4.5	Cracks pre-processing using Gamma Correction and Gaussian blur	Page 62
4.6	Speedbumps pre-processing using Gamma Correction and Gaussian blur	Page 63
4.7	Manhole pre-processing using Gamma Correction and Gaussian blur	Page 64
4.8	Pothole pre-processing using Gamma Correction and Gaussian blur	Page 65
4.9	Loss vs number of epochs	Page 67
4.10	Confusion matrix	Page 68
4.11	Potholes Segmentation using graph-based segmentation	Page 71
4.12	Manhole Segmentation using graph-based segmentation	Page 72
4.13	Cracks Segmentation using graph-based segmentation	Page 73
5.1	Statistics for DSR-CNN compared with DGR-CNN for potholes class	Page 78
5.2	Statistics for DSR-CNN compared with DGR-CNN for manholes class	Page 78
5.3	Statistics for DSR-CNN compared with DGR-CNN for speedbumps class	Page 79
5.4	Statistics for DSR-CNN compared with DGR-CNN for cracks class	Page 79

LIST OF TABLES

<u>Table</u>	<u>Description</u>	<u>Page</u>
2.1	Summary of results of [35]	Page 24
3.1	Studied Road anomalies and their main features	Page 43
4.1	Accuracy of the different CNN backbones	Page 66
4.2	Number of region proposals using DGR-CNN vs selective search	Page 70
4.3	F1 score, Recall, Precision, and Accuracy of using DSR-CNN	Page 74
4.4	F1-score, Recall, Precision, and Accuracy of the proposed system with enhancement	Page 76
5.1	DSR-CNN	Page 77
5.2	F1-score, Recall, Precision, and Accuracy of the proposed system with enhancement	Page 80
5.3	DGR-CNN vs. road anomaly classification models	Page 81

LIST OF ABBREVIATIONS

<u>Abbreviation</u>	<u>Description</u>
BGR	Boundary-conscious Graph Reasoning 15
CNN	Convolution Neural Network 1
DCGAN	Deep Convolutional GAN 2
DFN	Deep Fusion Network 10
DGR-CNN	Dynamic Generative R-CNN 35
DSR-CNN	Dynamic Similar R-CNN 35
ESRGAN	Enhanced Super-Resolution Generative Adversarial Networks 19
FCN	Fully Convolutional Network 25
GSGT	Graph Structure Guided Transformer 15
HMA	Hot-Mix Asphalt 26
HOG	Histogram of Oriented Gradients 21
IMU	Inertial Measurement Unit 21
IOU	Intersection Over Union 39
R-CNN	Region-based Convolutional Neural Network 1
RGB	Red Green Blue 28
RNN	Recurrent Neural Network 10
RPN	Region Proposal Network 13
SSD	Single-Shot Detector 21
SVM	Support Vector Machine 7
UAV	Unmanned Aerial Vehicle 23
VGG	Visual Geometry Group 1
YOLO	You Look Only Once 1

ABSTRACT

AUTOMATIC DETECTION OF ROAD ANOMALIES USING COMPUTER VISION AND DEEP LEARNING

By

Rasha Derar Saffarini

Automatic road anomaly detection and recognition systems are essential due to their effect on the comfort and safety of drivers and passengers. Drivers should be aware of their routes' bad road conditions and anomalies to avoid accidents, reduce the possibility of car malfunction or damage, and take the most appropriate route to their destinations. This led to a greater interest in research in automatic detection and recognition of road anomalies. Different techniques have been developed for automatic road anomaly detection and classification. The related studies can be categorized into accelerometer-based techniques and vision-based techniques. Both methods have problems. As for the accelerometer-based methods, the car's vibration causes errors in the accuracy of the detection process. As for the image-based methods, their problems are represented by the lack of road anomalies dataset. In addition, the failure to calibrate the camera installed on the car leads to image distortion and loss of important data.

This research created a new dataset from images of Tulkarm city roads taken using a DJI Mavic Air2 drone. A total of 15,326 images were extracted to create the dataset. The dataset consists of four classes: 4781 images of cracks, 4196 images of potholes, 3475 images of manholes, and 2874 images of speed bumps. All images are in 4K resolution and free from distortion, noise, or blur.

Moreover, two novel deep-learning models were developed automatically to detect and classify all the different types of road anomalies, such as potholes, cracks, speed bumps, and man-

holes. The first model is called Dynamic Similar R-CNN (DSR-CNN). This model employs graph segmentation, graph similarity, and dynamic programming algorithms to reduce the number of proposed regions and increase the detection and classification speed without affecting the accuracy. The results showed that DSR-CNN significantly reduces the number of candidate regions compared to the selective search algorithm employed in R-CNN and fast R-CNN. The developed technique proposed, on average, 8% of the segments proposed by the selective search algorithm. The DSR-CNN model achieved an average speed of 0.1173 seconds per frame with a mean average precision (MAP) of 82.82%.

The second model, called Dynamic Generative R-CNN (DGR-CNN), utilizes graph segmentation, graph similarity, dynamic programming for the region proposal phase to improve detection speed, and DCGAN technique to enhance the proposed regions and thus improve the accuracy of recognition and classification. This model achieved a mean average precision of the proposed method of 94.85% with a speed of 0.262 seconds per frame, which is considered a substantial improvement in detection and classification accuracy compared to the accuracy resulting from other state-of-the-art road anomalies detection and classification research. The increase in accuracy is achieved without significantly compromising the speed of faster R-CNN.

Chapter One: Introduction

Road conditions are important to countries' development and economic and political stability. This keeps the movement from one place to another uninterrupted and decreases the cost of transportation as well.

Additionally, road conditions affect the safety and quality of vehicles on the road, as poor road conditions and increased anomalies increase the possibility of accidents, subsequently increasing injuries and loss of lives [21, 10, 14]. Thus, road conditions should be monitored constantly to detect anomalies as early as possible since delayed detection of road damage increases the possibility of accidents and traffic jams and maximizes maintenance costs.

Road damage is classified into several categories: cracks and other damages. Cracks are divided into alligator cracks and linear cracks. Other damages include speed pumps, manholes, rutting, potholes, separation, white line blur, and crosswalk blur [42]. Manual road damage detection costs time and money, leading to delayed road maintenance. Therefore, detecting this particular road hazard requires the support of appropriate technology [53, 15].

There are two main techniques used to detect road damage: vibration-based detection and computer vision-based detection. Vibration-based techniques involve reading data from a tri-axial accelerometer, analyzing the data, extracting pattern features, and then predicting the type of anomaly. On the other hand, computer vision-based techniques capture images of the roads, prepare and analyze these images to detect anomalies, and then recognize these anomalies. Several machine learning, image processing, and deep learning techniques have been proposed within these two methodologies [47].

Deep learning techniques such as different versions of You Look Only Once (YOLO) [57], Region-based Convolutional Neural Network (R-CNN) detector series [29, 28, 58] with different backbones for the Convolution Neural Network (CNN) (Visual Gomtry Group (VGG)16, ResNet-152, ResNet-50, and others), are commonly used in road anomalies detection for fast and accurate signal, images analysis, anomalies detection, and classification [41, 55].

This research introduces a new image-based deep learning model for automatically detecting and classifying various types of anomalies, such as potholes, manholes, speed bumps, and

cracks. The model utilizes images taken by a drone, graph segmentation, graph similarity, Deep Convolutional GAN (DCGAN), and R-CNN algorithms to detect and classify road anomalies quickly and accurately.

1.1 Problem Statement

Several types of obstacles may be faced by the driver on the road. These anomalies can cause vehicle damage or even driver injury. Furthermore, traffic jams may increase due to these obstacles, such as potholes or large street cracks. These problems can be avoided if the driver knows the location of these anomalies. This will help the driver take suitable action beforehand. Consequently, it is necessary to automatically detect road anomalies, alert the driver of their existence by sending a notification to his smartphone, and localize these anomalies.

Road anomalies are categorized into cracks and other damages. Cracks are divided into linear cracks and alligator cracks, each of which is categorized into different types. Other damages are rutting, speed pumps, potholes, manholes, separation, white line blur, crosswalk blur, etc.

Numerous studies have focused on automatically detecting road anomalies using two primary methodologies: vibration-based and computer vision-based. The vibration-based approach involves data acquisition from the tri-axial accelerometer, followed by analysis, feature extraction, and anomaly prediction based on observed patterns. However, vibration-based detection poses some challenges. Firstly, the driver must drive over the anomaly to detect it, which could damage the car and pose a risk to the driver. Secondly, only the areas where the vehicle touches the road are studied, not the entire road. Lastly, the road may contain some non-anomalous obstacles that create noise in the signal, which could be incorrectly identified as an anomaly. Vision-based techniques were therefore developed to overcome these problems and improve detection accuracy. The vision-based approach automatically captures, analyzes, and classifies road images to detect anomalies.

Road anomalies can be categorized into two main types: cracks and other damages. Cracks

can be further classified into linear and alligator sub-types, each with distinct variations. Other damages include rutting, speed bumps, potholes, manholes, separation, white line blur, cross-walk blur, and more.

Most recent studies only concentrate on one or two types of anomalies and neglect any other obstacles on the road [69, 6, 56, 34], which makes them a partial solution to anomaly detection problems. Some studies detected potholes on the roads, while others detected speed bumps or classified the road as crack/no-crack roads. Most vision-based studies used cameras to capture road images. These cameras are placed in the car [6][8][12]. However, if the camera is not at a specific angle, it cannot take proper images for anomaly detection. This may cause the misclassification of these anomalies and, in turn, reduce detection accuracy. Additionally, the car's vibration may cause distortion and blur in the pictures, which reduces anomaly detection accuracy. Furthermore, capturing images using cameras inside the car will reduce the ability to measure detected anomaly's dimensions such as size, height, depth, etc. Finally, most of the proposed methodologies did not localize the detected anomalies to map them and alert the driver to avoid them or to make the appropriate decision.

In this research, a drone is used as a new image acquisition tool to overcome camera calibration problems that have arisen in previous studies and create a new real road anomaly image dataset to train and test the deep learning models. Additionally, two new fast and automatic deep learning models are developed to detect and classify all anomalies. These models comprise several newly developed algorithms for region proposals, region enhancement, feature extraction, detection, classification, and measurement of all road anomalies. The first model focuses on reducing the total number of proposed regions and thus increasing the speed of detection and classification. The second model focuses on enhancing those proposed regions before classification and subsequently enhancing detection and classification accuracy.

1.2 Objectives

In this research, several aspects of road anomaly detection and classification are studied, including:

1. Create a reliable road anomaly image dataset by capturing videos and images using drones. Drones became widely used, cost-effective, and easy to use for capturing videos and images of the roads due to their ability to fly in many directions with reasonable heights to capture very high-resolution videos and images of the whole road. In addition, drones can capture videos of many roads as fast as possible.
2. Propose new deep learning and computer vision models to automatically detect all the different types of anomalies, namely, potholes, manholes, types of cracks, speed bumps, and others.
3. Increase the detection and classification speed by reducing the number of proposed regions utilizing the following:
 - a novel region proposal model for identifying regions from the image where an anomaly may exist.
 - Utilize dynamic programming techniques to expedite the classification process.
4. Increase the detection and classification accuracy by proposing an image enhancement model for improving the proposed regions before feeding them to the CNN.

1.3 Research Questions

This research will attempt to answer the following questions:

- Can drones improve the quality of images taken to monitor road anomalies?

- How does combining graph segmentation and graph similarity with dynamic programming affect the total number of proposed regions for detecting and classifying road anomalies?
- How do reduced proposed regions affect road anomaly detection and classification, speed, and accuracy?
- In what ways does DCGAN improve the quality of proposed regions for classification, and how does it impact the accuracy and speed of the detection and classification process?
- Can the combined Dynamic Generative R-CNN (DGR-CNN) model improve the accuracy of detecting and classifying road anomalies without significantly compromising speed?

1.4 Significance of the Study

The quality of road surfaces plays a critical role in determining ride comfort and ensuring the safety of drivers and passengers. Poor road conditions, such as potholes, cracks, uneven surfaces, and undetected anomalies, significantly elevate the risk of accidents. These accidents can lead to severe injuries, fatalities, and substantial financial losses. To mitigate these risks, it is essential to implement systems that can continuously and automatically identify road anomalies at the earliest possible stage. Delays in detecting such issues can result in a higher frequency of accidents, increased vehicle damage, and costly repairs for drivers. Additionally, these delays contribute to traffic congestion, wasting valuable time and, in severe cases, leading to loss of life. This research aims to address these challenges by developing an advanced, efficient model capable of detecting road anomalies, classifying them into specific categories, and promptly alerting drivers to their presence. This model is designed to overcome the limitations of existing road anomaly detection methods, particularly in terms of speed and accuracy in identifying and categorizing anomalies. By leveraging innovative techniques, the proposed system achieves remarkable precision and rapid response times, making it highly effective in detecting a wide

Chapter Two: Literature Review and Background

range of road irregularities. Ultimately, this research aims to reduce the likelihood of accidents, thereby saving lives, minimizing repair costs, and reducing time wasted in traffic while enhancing overall road safety.

There have been many studies addressing the issue of potholes, cracks, and other road irregularities. With the advancement of technology, methods for fixing these problems have improved. Traditional methods, including area detection and texture analysis, involve capturing images or video frames through vehicle cameras. These strategies are fairly simple but often lack robustness and accuracy under different lighting and weather conditions. Another technique involves using sensors such as accelerometers and gyroscopes to detect road surface irregularities. These methods are effective in capturing physical vibrations caused by anomalies but require significant sensor calibration and are less effective in distinguishing between different types of irregularities.

With the rise of machine learning, researchers have begun using supervised training models for anomaly detection. Anomalies are detected by training classifiers such as Support Vector Machines (Support Vector Machine (SVM)s) and Random Forests using features extracted from images or sensor data. While these methods have improved detection rates, they heavily depend on manually designed features and labeled datasets.

The advent of deep learning has substantially evolved this field of study. CNNs and their variations have superior performance in characteristic extraction and anomaly detection. These models can mechanically examine hierarchical features from raw information, leading to more accurate detection systems.

Some research has explored hybrid methods that combine traditional image processing, sensor information, and deep learning. These techniques aim to leverage the strengths of each technology to enhance detection accuracy and robustness.

2.1 Road Anomaly

Road anomalies refer to any deviation or variation from normal road conditions. They can include various types of irregularities or defects on the road surface, such as potholes, cracks, bumps, gravel, cobblestone, and damaged concrete. Road anomalies can result from factors such as repeated use, traffic loads, weather conditions, and the age of the road infrastructure.

Monitoring and detecting road anomalies is crucial for maintaining road safety, reducing fuel consumption, and preventing damage to vehicles and pedestrians. Various methods, including vibration-based and vision-based techniques, utilize machine learning and deep learning approaches to detect and categorize road anomalies [40].

2.2 Road Anomaly Detection and Classification Techniques

Different methods are used to detect and categorize road anomalies. These methods can be divided into two main classes:

- Vision-based techniques: These involve using images to identify road anomalies. This can be achieved through image-processing algorithms and/or deep-learning algorithms.
- Vibration-based techniques: These methods utilize different data types for detection and categorization, such as vibration or accelerometer data. Various techniques like threshold-based methods, feature extraction methods, or deep learning algorithms can be employed for this purpose.

2.2.1 Vision-based Road Anomaly Detection and Classification

Vision-based road anomaly detection and classification techniques generally involve cameras that take images of the street surface. The images are then processed using image processing or deep learning techniques to detect road anomalies.

Vision-based road anomaly detection and categorization are crucial for safe autonomous driving. Several papers have proposed methods to address this challenge. Tian et al. [74] developed an unsupervised anomaly detection technique that uses pairwise binary logits of the scene and language anchors to identify anomaly regions in road images. Fei et al [26] proposed a machine vision-based algorithm that identifies undeveloped road areas and categorizes different road conditions using image statistics and vehicle status indicators. Baek and Chung [9] presented an anomaly detection technique using a deep feature vector representation based

on a vision transformer, improving decision-making reliability by identifying the location of anomalies. Shiyang [66] introduced a road vehicle anomaly detection technique based on image classification and analysis that is capable of detecting alterations and abnormalities in road vehicle chassis images.

Image processing algorithms typically filter images to remove noise, improve contrast, and segment them to identify objects. Once the images have been segmented, a set of rules can classify them as anomalies or non-anomalies based on their length, form, and other features.

Deep learning algorithms are a type of machine learning algorithm that is trained on large datasets of road images with and without anomalies. Once the algorithm has been trained, it can be used to classify new images of road surfaces as anomalies or non-anomalies [40].

2.2.2 Vibration-based Road Anomaly Detection and Classification

Vibration-based road anomaly detection and classification strategies typically use sensors to measure the vibration of the vehicle driving over the road surface. The vibration data are then processed using threshold-based techniques, feature extraction strategies, or deep learning algorithms to identify road anomalies. Threshold-based methods normally involve evaluating the vibration information to a threshold level. If the vibration exceeds the threshold, then a set of rules identifies the road anomaly [80] [60] [46].

2.2.3 Applications of Road Anomaly Detection and Classification

Road anomaly detection and classification may be used for a lot of applications, including:

- **Road safety:** Road anomaly detection and classification can be used to become aware of street anomalies and repair them before they cause accidents.
- **Vehicle maintenance:** Road anomaly detection and classification can alert drivers to road anomalies, allowing them to take steps to avoid damage to their vehicles.

- Traffic control: Road anomaly detection and classification can be used to identify road conditions and anticipate areas prone to congestion.
- Road infrastructure renovation: Road anomaly detection and classification can be used to pick out and restore road anomalies before they become critical and require greater high-priced upkeep.

2.3 Computer Vision

Computer vision has many uses, from autonomous vehicles to surveillance systems, medical imaging, and augmented reality. Deep Neural Networks (s) have revolutionized the field of computer vision, providing cutting-edge performance in a wide range of tasks. [75, 72, 23].

Road anomaly detection systems that are based on computer vision typically employ deep learning techniques, including Deep Fusion Networks (Deep Fusion Network (DFN)s), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (Recurrent Neural Network (RNN)s) [75, 67, 40] which do not necessitate a special highway engineering knowledge due to their ability to process raw data without any transformation or representation [72]. In these systems, inertial data is collected from various sensors, including automotive-mounted smartphones that record accelerometer data, coordinates, velocity, and bearing [46].

The data collected is analyzed and categorized using deep learning algorithms to detect various types of road irregularities, such as potholes, bumps, and other deformations. The effectiveness of these systems is assessed using statistical metrics like accuracy, recall, and precision. Computer vision-based road anomaly detection systems generally use deep learning algorithms to examine sensor data and accurately detect road surface abnormalities. The use of computer vision technology offers several benefits in the detection of road anomalies. It allows for precise and automated identification of road pavement cracks and potholes, which is crucial for early detection and remediation of potential hazards.

This technology is particularly beneficial in low-resource environments when manual inspection of road systems is inadequate and infrequent, leading to increased risks of accidents [23]. By leveraging

deep learning-based solutions and frameworks like You Only Look Once (YOLO) and Faster R-CNN, computer vision can detect and classify road distresses in real-time using vehicle dashboard-mounted smartphone cameras [67]. Additionally, computer vision algorithms are adaptive and efficient in various settings since they can be trained on thousands of photos to precisely locate roadways in diverse conditions [40].

Computer vision-based road anomaly detection systems have some limitations; one limitation is detecting unknown objects or anomalies that are outside of the training distribution, as they rely on learned patterns and features from labeled data, therefore the lack of comprehensive training datasets for anomaly detection poses a challenge [75], and may perform worse in real-world steps compared to "conditioned" steps, as observed in the analysis of the deployment of such a system. Additionally, the accuracy of road anomaly detection depends on the quality and complexity of the training datasets used for system performance analysis [72].

Another limitation is the large amount of data that is needed for these systems; Computer vision-based systems heavily rely on Deep Neural Networks, which require large amounts of labeled training data. Acquiring and labeling such data can be time-consuming and resource-intensive. Moreover, the performance of these systems can be affected by environmental factors such as lighting conditions, weather conditions, and occlusions, which may impact the accuracy of anomaly detection [75] and may fail in detecting anomalies due to potential issues with lighting and weather conditions. Additionally, these systems can be affected by the quality of the feature representation of road conditions, and poor feature representation can lead to lower accuracy in detection [23].

2.4 Object Detection

Object detection in road anomalies is a difficult challenge since road anomalies can have many unique shapes, sizes, and colors. It can also be difficult to exercise under low-mild or foggy conditions. However, recent advances in computer vision and machine learning have made it viable to develop object detection techniques that can accurately detect road anomalies under a variety of conditions. There are two main approaches to object detection in road anomalies:

- Supervised learning: This approach involves training a machine learning module on a labeled dataset of images and videos of road anomalies. The model learns to identify road anomalies by looking for patterns in the data [20].
- Unsupervised learning: This approach does not require any labeled data. Instead, the model learns to identify road anomalies by looking for unusual or unexpected objects in the data [17].

Both supervised and unsupervised deep learning approaches can be used to enhance effective object detection models for road anomalies. However, supervised learning models are generally more accurate, but they require labeled data to train, which is time-consuming.

Several researchers have explored various supervised learning methods for object detection for road anomalies. Yasuno et al. [79] proposed a method that combines auto-encoding reconstruction and isolation-based anomaly detection for road surface monitoring. Alaoui-Belghiti et al. [2] introduced a non-parametric framework using optimal transportation to estimate deviation from an observed distribution, which was effective for anomaly detection in industrial predictive maintenance tasks and anomalous breathing detection. Additionally, Casals, Silvia Gil [16] provided a method for effectively finding the role of road objects using unsupervised type based on distribution density. Furthermore, Bibi et al. [11] discussed the use of deep learning techniques such as the Residual Convolutional Neural Network (ResNet-18) and the Visual Geometry Group (VGG-11) for the automatic detection and classification of road anomalies such as potholes, bumps, and cracks.

According to Qiu et al. [51], supervised object detection cannot be used to detect street anomalies because it requires labeled data, which is expensive and time-consuming to obtain. Additionally, road anomalies are unpredictable and can vary significantly in appearance, making it difficult to study them accurately. Moreover, supervised object detection models are trained to perceive specific objects or classes and will not be able to detect anomalies that do not suit into predefined categories. Instead, unsupervised approaches are suitable for automatically identifying unexpected driving scenarios. Qiu et al. endorse an unsupervised approach with the use of conditional Generative Adversarial Networks (GANs) to detect driving anomalies. Their method trains

a conditional GAN to extract latent functions from different modalities, which include vehicle's CAN-Bus alerts, driver's physiological signals, and distances to nearby pedestrians, vehicles, and bicycles. The framework combines the latent representations from these modalities with an attention model. It is trained with the triplet loss feature to discriminate normal and unusual using segments.

2.5 Region Proposal

Identifying potential abnormal areas in a road image is known as region proposal in road anomalies. This is a crucial stage in detecting street anomalies because it allows the algorithm to focus on the areas of the image that are most likely to have anomalies rather than analyzing the entire image. Several region proposal techniques can be applied to detect road anomalies, and one such strategy is using a selective search algorithm. Selective search works by recursively grouping pixels into regions based on their similarity and then ranking the generated regions based on characteristics such as size, shape, and texture. The candidate regions for anomaly detection are then chosen from the top-ranked regions. By reducing the number of areas to examine, stronger machine learning techniques and appearance models can be utilized for object recognition. Another method of region proposal for street anomaly detection is using a Region Proposal Network (Region Proposal Network (RPN)). RPNs are a specific type of Convolutional Neural Network (CNN) that can be trained to predict the likelihood of an anomaly being present in a given area. They handle the task of detecting all relationships in a scene with multiple objects by using pairs of related regions in images to train a relationship proposer. In conjunction with two-stage object detectors such as Faster R-CNN, RPNs include both a classifier and a regressor to determine the possibility of a proposal containing the target object and to regress the proposal coordinates. Scale and aspect ratio are critical parameters for any image. Once candidate regions have been identified, a classifier can determine whether or not those regions truly contain anomalies. A dataset of labeled road images can be used to train this classifier, with the labels indicating whether or not each image contains anomalies. Region proposal algorithms are a crucial first step in creating accurate and effective road anomaly detection systems. They can

help reduce the computational burden of anomaly detection and improve the accuracy of the overall system by identifying candidate regions that are most likely to contain anomalies. [58].

2.6 Graph Segmentation and Graph Similarity

2.6.1 Graph Segmentation

Graph segmentation involves creating a graph representation of the data. Each node in this graph represents a pixel or other data element, and the edges between the nodes show how similar the corresponding elements are. The goal of graph segmentation is to divide the graph into separate subsets with nodes that are more similar to each other than to nodes in other subsets. There are numerous graph segmentation algorithms, but they all work on the same basic premise: dividing the graph into separate subsets. Among the most popular graph segmentation algorithms are:

- Normalized cuts: This algorithm partitions the graph by minimizing the normalized cut, which is a measure of the similarity between the nodes in different subsets [64].
- Graph cuts: This algorithm partitions the graph by minimizing the cost of cutting the edges between nodes in different subsets [12].
- Random walks: This set of rules walks the graph by simulating random walks on the graph and labeling nodes based on where walks become [31].

The most basic undertaking in image segmentation is precisely defining the spatial extent of a few objects, which the union of multiple areas may represent. According to [13]. Examples of viable strategies are as follows:

1. The photograph is divided into areas, and then the item is composed by means of the union of a number of those areas.
2. The approximate place of the item/boundary is determined, and its spatial volume is described from that place.

Convolutional Neural Networks (CNNs) are adept at capturing local relationships but struggle with understanding global relationships between distant areas. A new approach called Graph Structure Guided Transformer (Graph Structure Guided Transformer (GSGT)) has been proposed to address this issue. GSGT leverages the strengths of transformers in modeling long-range dependencies by transforming the 2-dimensional feature map into a graph structure based on semantic relevance. Additionally, it incorporates a graph embedding attention module to utilize the graph's neighborhood topology and enhance the worldwide context of the transformer. Experimental results in multiple data sets demonstrate that GSGT significantly improves segmentation accuracy and convergence speed [84]. Another technique called Boundary-conscious Graph Reasoning (Boundary-conscious Graph Reasoning (BGR)) focuses on understanding long-range contextual features for semantic segmentation. The BGR module utilizes a boundary score map to direct the graph reasoning process toward boundary regions where difficult-to-segment pixels are typically located. This method has demonstrated effectiveness in semantic segmentation in demanding benchmarks [73].

2.6.2 Graph Similarity

Graph similarity is the measure of how similar two graphs are. There are several different ways to measure graph similarity; some common metrics include:

- Edit distance: The edit distance between two graphs is the number of edge insertions and deletions required to transform one graph into another.
- Graph isomorphism: Two graphs are isomorphic if there exists a one-to-one mapping between their nodes that preserves the edge shape of the graphs.
- Graph kernels: Graph kernels are functions that compute a similarity score among graphs. Graph kernels can be based on numerous specific capabilities of the graphs, such as the wide variety of edges between nodes, the degree of the nodes, and the shortest paths between nodes.

Ok, Seongmin [48] proposed a way to measure the similarity among graphs and study from graph statistics using Graph Neural Networks (GNNs). They introduce a new distance metric known as the Wasserstein Laplacian Score () that captures the structural similarity between graphs. The WLS is based totally on the Weisfeiler-Lehman () test, which is a graph isomorphism test. The author showed that the WLS can correctly measure the similarity among graphs with continuous attributes. The author additionally advocates a GNN model primarily based on the WLS known as WLS-GNN. He evaluated the performance of WLS-GNN on diverse graph category responsibilities and compared it with different GNN models. The consequences show that WLS-GNN achieves excellent overall performance and outperforms other fashions on some datasets. Furthermore, the author displayed the effectiveness of WLS in graph generation responsibilities and that using WLS as a discriminator in a Generative Adversarial Network (GAN) improves the stability and validity of generated molecules.

2.7 GAN and DCGAN

Generative Adversarial Network (GAN) consists of two neural networks: Generator and Discriminator networks that are pitted against each other and are simultaneously trained by an adversarial process. According to Ahirwar, Kailash [1] “A GAN is a Deep Neural Network architecture made up of two networks, a Generator network and a Discriminator network. Through multiple cycles of generation and discrimination, both networks train each other while simultaneously trying to outwit each other.”

GAN has two key networks:

- **Generator:** The generator network uses existing data to create new data, such as generating new images based on existing ones. Its main objective is to produce various types of data like images, video, audio, or text from randomly generated sets of numbers, known as a latent space. When creating a Generator Network, the user needs to specify the network's purpose, which could be image generation, text generation, audio generation, video generation, or more.

- **Discriminator:** The discriminator network tries to differentiate between the real data and the data generated by the generator network. The discriminator network tries to put the incoming data into predefined categories. It can either perform multi-class classification or binary classification. In GANs, binary classification is generally performed.

It's important to note that the generator does not directly access real images. Instead, it learns through interaction with the discriminator. The discriminator has access to both synthetic samples and real-world image stack samples. It receives the error signal indicating whether the image came from the real stack or the generator. This error signal can be used to train the Generator via the Discriminator, allowing it to produce higher-quality forgeries. [19].

Supervised learning with Convolutional Neural Networks (CNNs) has become popular in computer vision applications. However, unsupervised learning with CNNs has received less attention. The purpose of [52] work is to bridge the gap between the success of CNNs in supervised and unsupervised learning. They propose Deep Convolutional Generative Adversarial Networks (DCGANs) as a strong candidate for unsupervised learning. DCGANs are constrained by architectural limitations and demonstrate the ability to learn a hierarchy of representations from object parts to scenes.

The primary distinction between GAN and DCGAN is the use of convolutional layers in DCGAN, which allows for better image data handling and spatial dependency capture. DCGANs have architectural constraints, including stride and fractional-strided convolutions, which aid in mastering hierarchical representations from object components to scenes.

DCGANs have been proven to be a robust candidate for unsupervised learning because they can study significant representations from image datasets and be used for various tasks, demonstrating their utility as general photograph representations. In recent years, researchers have also investigated unsupervised representation learning with DCGANs. Another GAN-based model for learning disentangled representations has been proposed, which includes predicting networks and encoders and measures the correlation amongst encoder outputs. This model exhibits generalizability and robustness, as well as the ability to extract factorial functions from

complex datasets [52].

Furthermore, because the Discriminator learns an unnormalized density function that characterizes the statistics manifold, GANs can provide useful statistics for downstream responsibilities such as feature extraction for classification [82].

2.8 Literature Review

Road anomaly detection research studies are categorized into acceleration-based and vision-based approaches. Information should be collected from roads for anomaly detection. Two data types were collected to be analyzed and used for anomaly detection: accelerometer signals and images or mixed. Different techniques were used for the detection and classification process. In this chapter, state-of-the-art research of each type is presented.

2.9 Accelerometer-based Approaches

Different accelerometer types were used to collect road data as in [68, 62]. In [68], authors proposed a road anomaly detection method using smartphone sensors, in which smartphone accelerometers were used to read data to detect and classify road anomalies. They collected 3-axis acceleration data, latitude, longitude, speed, timestamp, and anomaly-type data from the accelerometer. The collected data are pre-processed to handle invalid and inconsistent data and to join the data rows for each anomaly to reduce the amount of data to be processed. Finally, a tree-based classifier, such as gradient boosting and decision tree, was used to detect and classify anomalies.

[62] implemented a RoADS system to read inertial accelerometer data. Wavelet decomposition analysis was used to process the signals, and finally, a Support Vector Machine (SVM) was used to detect and classify anomalies.

Similarly, [49] suggested a model with a 1-D CNN of 2-hidden layer with a kernel size of 3 for features extraction, pothole detection, and classification and employed an inertial sensor

(accelerometer and gyroscope) to collect data from iOS smartphone put in the car for pothole identification.

Furthermore, [45] proposed a low-cost real-time system to extract information about road conditions by equipping cars with sensors that read acceleration data. The proposed system reads the acceleration signal from the attached sensor, analyzes it using the short-term Fourier transform, and extracts features. Then, various classifiers, such as SVM, were used to distinguish road problems (manholes, potholes, and cracks).

While in [61], raw sensor and location data were collected by developing a mobile app installed on a smart device (mobile or tablet). The collected data were linear accelerometer, rotation vector, and location information. These data were collected from a Linear accelerometer sensor, gyroscope, magnetometer, and other sensors for road anomaly detection and classification after road anomalies were detected and classified using modified threshold—based and machine-learning approaches (K-means clustering).

2.10 Image-based Approaches

Another approach for road anomaly detection and classification is the vision-based approach. Vision-based approaches depend on collecting road images either using a mobile camera or a camera installed on a car as in [69, 6, 56, 34, 21, 76, 63, 24, 39, 4, 44, 25, 30, 35, 22, 81, 8, 59, 18, 77, 5, 32].

Several types of road anomalies were studied, such as potholes, manholes, cracks, and others. Different research focused either on a specific type or on multiple ones. In [69, 6, 56, 63, 34, 21, 59] images of potholes in roads were captured and studied using different tools.

A new approach was proposed in [59] for pothole detection from low-resolution images using Enhanced Super Resolution Generative Adversarial Networks (Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN)) and an object detection network (YOLOv5 and EfficientDet Networks). The method goes through two steps: the first one is the resolution enhancement step, and the second one is the object detection step. To enhance the image's res-

olution from a distance, ESRGAN was used to generate super-resolution images, which were used to train the object detection networks in the second step to detect small, unclear objects. Image combinations from several datasets (around 1300 images) were used to train the model, and 81 images were used for testing. The detection results on high-resolution images showed considerably higher precision and recall than detection using low-resolution images.

[36] represented a robust system for pothole detection using YOLOv8 deep learning model. This approach processes images captured by vehicle-mounted cameras, enabling real-time identification of road anomalies. The system demonstrated superior accuracy and efficiency compared to previous models, making it highly suitable for integration into autonomous vehicles. The researchers employed data augmentation techniques to enhance the model's performance, ensuring reliable detection under various environmental conditions.

In [69], the author proposed a road monitoring algorithm. Using this algorithm, potholes can be identified, detected, tracked, scored, and estimated from images of the road captured by an onboard camera inside the vehicle. This algorithm takes an image of the road, prepares it, digitizes it, applies morphological erosion and dilation to connect white areas, and removes black noise from the binarized image. Finally, they calculated the pothole probability score by computing two scores. First, the difference between the average intensities of pixels inside and outside each contour is calculated. The second computes the $N_{\text{sharp}}/N_{\text{all}}$ ratio. N_{sharp} is the number of pixels around the sharp contour, and N_{all} is the number of all pixels around the contour. A total score is calculated as the contour acceptance score by averaging the pothole-likelihood scores.

In [21], a camera embedded inside the car was used for pothole detection on roads by developing a model using morphological algorithms. The pothole detection algorithm goes through the following steps:

- Road extraction algorithm was used for cropping road segments from the collected images by cropping 20% from each side (50% of its height and 40% of the width).
- Use a Gaussian low pass filter to remove the noise from the original images.

- Binarize the image using Otsu's algorithm to split the foreground from the background.
- Retain the potholes represented by the connected pixels by applying the skeletonization process.

[63] proposed a real-time pothole detection approach where authors experimented with three types of object detectors on different frameworks, such as Single shot Multi-box Detector Single-Shot Detector (SSD) on TensorFlow framework, YOLOv3, and YOLOv4 on Darknet framework. All the processing was done on Google Colaboratory (Colab) on images from a combination of two datasets. The first one is available online; the second one is a combination of images from different sources on the internet and images of Lebanese roads captured using a camera mounted on the windshield of a car. The experiments showed that SSD performs worst and can't be used as a real-time detector. On the other hand, YOLOv4 gave the best results with 81% recall, 85% precision, 85.39% , and a processing speed of 20 frames per second.

A method for 3D reconstruction of potholes using SIFT key points detector and disparity map was used to create a 3D scene of the pothole in [34]. A car was equipped with a stereo apparatus comprising two cameras. The videos of the road were taken from both cameras and then the images were captured from the videos. Each point of the road had a right and left view to be analyzed and construct a 3D view of the roads.

Machine learning algorithms were used in [6, 56] to classify the images based on pothole/non-pothole images. In [6], the Histogram of Oriented Gradients (HOG) algorithm was used for features vector extraction, which entered the Naïve Bayes classifier. The Naïve Bayes assigns the label to an input image based upon the maximum posteriori probability as in figure 2.1

A dataset developed by [37]. were used in [6]. This dataset contains 120 pavement images; 50 were used for training, and 70 were used for testing.

Finally, a wireless portable camera was used to capture images of the potholes in the roads in [56]. In this study, researchers used TensorFlow and OpenCV libraries to detect potholes in images using TensorFlow with a Faster R-CNN inception v2 pre-trained model. Additionally, the sensor device contains a GPS sensor, Inertial Measurement Unit (IMU) sensor, external

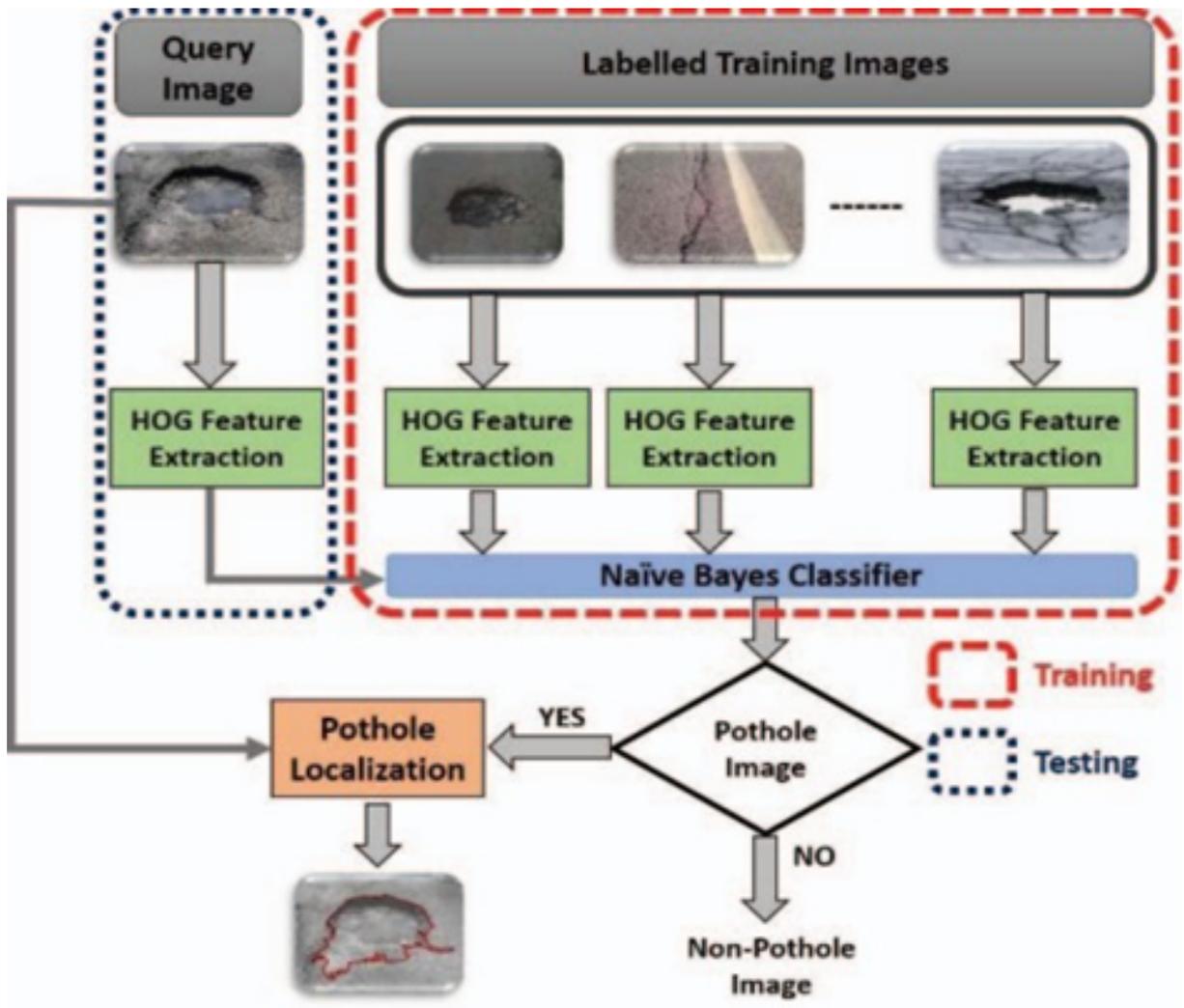


Figure 2.1: Algorithmic workflow in [6]

GPS antenna, and microcontroller to control the sensor, and it is sent to the processing unit to localize the pothole.

The study in[?] presents an innovative approach to enhancing urban road safety and maintenance. The authors developed the PDS-Unmanned Aerial Vehicle (UAV) system, which leverages Unmanned Aerial Vehicles (UAVs) equipped with cameras to capture aerial images of road surfaces. These images are processed using a deep learning model based on YOLOv8 to automatically detect and assess potholes. The system achieved impressive performance metrics, with an F1 score of 95%, precision of 98%, and recall of 92%. Additionally, PDS-UAV offers a user interface that overlays detected potholes onto maps, allowing road users to visualize and avoid damaged areas, thereby enhancing safety. The study also involved user interviews and questionnaires to gather feedback, ensuring the system's design aligns with user needs. Future work aims to incorporate additional features and further improve the model's accuracy.

On the other hand, in [44, 30, 35, 54, 33, 25, 5, 32], authors focused on studying cracks presence and severity on roads. Images of cracks were taken using vehicle cameras as in [44, 35, 54, 33, 5].

In [44], 9053 images were captured from 7 local governments across Japan. Of the captured images, 7240 were used for training, and 1813 were used for testing. The captured images consist of different types of cracks to be studied.

YOLOv2 is used for road crack detection in this study. YOLOv2 uses Single CNN with standard layer types, such as the Convolutional layer with 3x3 kernels and the Max Pooling layer with 2x2 kernels. A final convolutional layer with a 1x1 kernel minimizes the data to 13x13x125 format. This 13x13 structure represents the grid size into which the image is divided. All these grid cells predict five bounding frames, each described by 7 data items. x, y, width, height values; confidence score; probability distribution of cracked and non-cracked roads as in figure 2.2

The proposed model achieved an average F1 score of 0.8780 for detecting distress. It showed high accuracy in detecting alligator cracks but did not perform as well in recognizing



Figure 2.2: Classification of predicted crack examples: True Positive-(a), False Positive-(b), False Negative-(c) obtained from YOLO v2 in [44]

two types of transverse cracks, with F1 scores of 0.7137 and 0.6885.

Another method for detecting roadway cracks using deep learning with Transfer Learning was proposed by [35]. The proposed model was built based on several pre-trained network architectures: GoogleNet, AlexNet, and ReseNet101. The network was designed by setting the training parameters, such as the initial value of the learning rate, the initialization of the epochs, and the optimizer's choice. Then, a dataset of 136 images was used to train and test the proposed model, where 60% of the images were used as training data, and the remaining images were used for testing.

The results showed that transfer learning using a pre-trained Google Net gave the best performance in pavement crack detection due to the adaptability at each iteration, the number of layers used, and reduced losses. The crack detection results obtained with the different networks when are shown in table 2.1:

Table 2.1: Summary of results of [35]

Network	Learning rate	Accuracy
AlexNet	10^{-4}	75%
GoogleNet	10^{-3}	100%
AlexNet	10^{-3}	95%

In [?] study, the authors present an upgraded YOLOv7 algorithm designed for road crack de-

tection. YOLO (You Only Look Once) is a well-known deep learning model used for real-time object detection. The researchers improved YOLOv7 by incorporating modifications that optimize its performance for road crack detection. The system was tested on the RDD-2022 dataset, which contains over 20,000 images of different types of road surface cracks, such as alligator, longitudinal, and transverse cracks. The model achieved high detection accuracy while maintaining efficiency, even with the high-resolution images. The proposed system is lightweight, which is beneficial for deployment on mobile devices or vehicles for real-time crack detection. The authors emphasize that the improved YOLOv7 can potentially be used in smart transportation systems to provide real-time maintenance alerts and reduce infrastructure damage.

Faster R-CNN was used in [33] for crack detection in concrete roads. In this research, a dataset of 323 images with a resolution of 4128x2322 was created using a smartphone camera.

A novel semantic segmentation model named crack transformer () was proposed for improving the detection of long and complicated pavement cracks in [32]. This study aims to enhance crack detection, especially for noisy images. The authors developed this model based on an encoder-decoder system. They employed the Swin Transformer as the encoder and a decoder with multi-layer perception (MLP) layers. Three different datasets were used for testing the developed model, namely, public CFD [65], Crack 500 [78] datasets, and a customized dataset of 197 crack images called the CrackSC dataset, which concentrates on heavy shadow and dense cracks. The results showed a considerably enhanced performance in crack detection from noisy images.

Additionally, In [25], authors used Fully Convolutional Network Fully Convolutional Network (FCN) and VGG16 as the backbone of the FCN to classify each pixel into “crack” or “non-crack” classes. The pixel density ratio was also computed. The proposed approach was applied to a public already generated concrete data set of 40000 227x227 crack images.

In [5], a new approach is proposed for detecting pavement cracks. Crack images were captured using a camera, and some enhancements were applied to these images to adjust and reduce the noise in the captured images. After that, the edge detection method is used to detect

the pavement cracks. The proposed method started using a gray-scale transformation function with the reconstruction filter to convert the image into a gray-scale image. Crack gradient information is mapped using the adaptive thresholding method, and depending on the gray-scale feature, the crack edge is extracted. After that, detected edge points were gathered in line with the gradient information. Finally, the complete crack was extracted by filling the detected crack edge.

Deep Convolutional Neural Network (DCNN) trained on the ‘big data’ ImageNet database (VGG16) was used by [30] as a feature vector extractor. The generated vector was used as an input to several classifiers such as Single layer NN, Random Forest, Extremely Randomized Tree, Support Vector Machine, and Logistic Regression for automatic road cracks detection in images of Hot-Mix Asphalt (Hot-Mix Asphalt (HMA)) and Portland Cement Concrete (PCC) surfaced pavement. The method was tested on a subset of 1056 images from the pavement distress images dataset from the Federal Highway Administration (FHWA).

Finally, The detection of pavement cracks was improved in [77, 18] by expanding the small dataset used in the training phase using a Generative Adversarial Network (GAN). In [77], a small pavement crack dataset captured by an unmanned aerial vehicle (UAV) was used as an input to the GAN network to train it to generate new images. The expanded dataset was then used to train the CNN for detection. Detection results were compared before and after the expansion of the dataset, which showed a significant improvement in the detection accuracy from 80.75% to 91.61%. While in [18], GAN was used to generate a big dataset of virtual pavement crack images with features like the original cracks. A three-stage model was proposed for this task. First, use GAN to generate crack images. Second, train the CNN classifier and DeepLab-v3+ on this generated dataset. Finally, real crack images are used to evaluate the proposed model. A significant enhancement is shown in the detection results using their proposed model.

Researchers have also studied speed bumps as a type of road anomaly. As in [22, 81, 8, 3, 50]. [22], first prepared the main setup for the study. Different shapes of roads and speed bumps (marked and unmarked) were designed and built using sheets of black paper and wood

material. They then took pictures of these speed bumps to build a dataset and took about 550 pictures. Images were pre-processed using augmentation techniques to increase the number of images in the dataset and make them suitable for use in CNN. After these steps, the dataset had about 3400 images. After that, a CNN using 3x3 and 5x5 filters was built to extract the feature vector. Feature vectors were sent to a pooling layer to extract feature vectors of defined size. The resulting vector was sent to the fully connected layer to classify the presence of speed bumps. Finally, the distance between the car and the speed bump was calculated depending on the size of the box drawn around the speed bump. The proposed methodology is shown in figure 2.3.

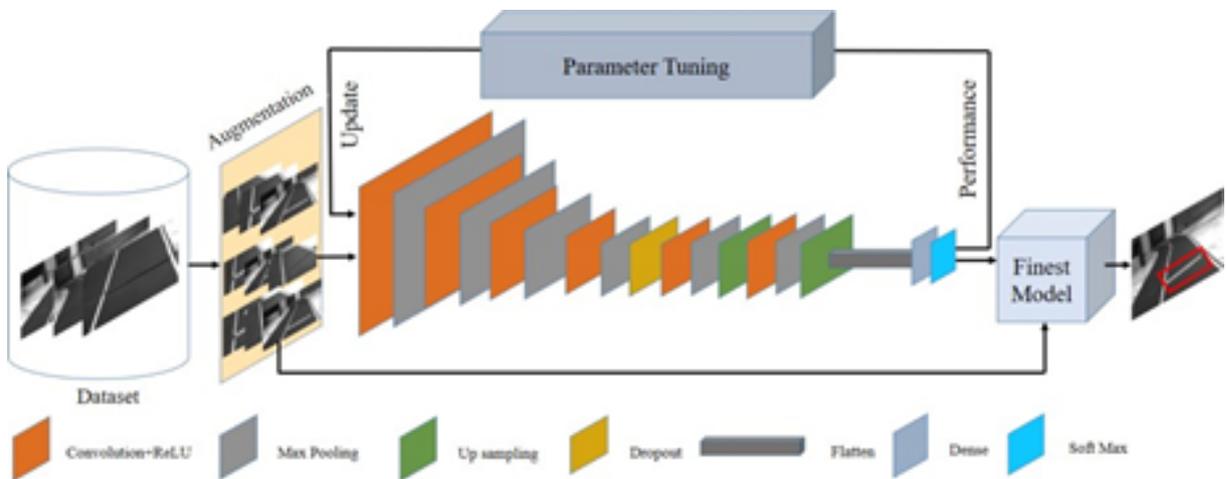


Figure 2.3: Proposed speed bump detection using convolutional neural network (CNN) architecture in [22]

Moreover, a camera and Lidar were used in [81] to detect speed bumps. First, images are converted to grayscale and then pre-processed and binarized. After that, the pattern of the speed bumps is used as an input to the Haar classifier, which is considered the first detector to detect the regions of the image that represent candidates for a speed bump. The resulting regions were filtered using a second detector. Objects with a height higher than the vehicle height were filtered, and then HOG was applied to filtered images to extract features. Finally, the SVM classifier is used to recognize the speed bumps. The camera and the Lidar are used to calculate the height of the speed bump and the distance from it. The average gained accuracy of detecting

speed bumps was 85.2%. Furthermore, their method takes 10ms processing more than the other methods.

A new approach to detect unmarked speed bumps is proposed by [8]. The proposed approach utilized image processing techniques to detect unmarked speed bumps from images of Indian streets captured by a Raspberry Pi camera. Images were captured from the video, and several steps were performed to detect the presence or absence of speed bumps in the images and classify the degree of sharpness of the detected speed bumps. This model pre-processes the image, converts it from Red Green Blue (RGB) to grayscale, and then applies a Gaussian filter to process the lighting and blurring of the image. A Canny edge detection algorithm is then used to identify edges in the image. Finally, a Hough transform is used to identify lines representing speed bumps based on line length. The model was tested on a dataset of 1385 images taken at various times of the day. As a result, the average accuracy rate was 95.5%.

Novel approaches were proposed for upcoming speed bump detection for self-driving cars in [3, 50]. In [3], the SegNet algorithm was used for speed bump detection. SegNet is a DNN for semantic pixel-wise segmentation. The model was applied to road images captured using a monocular camera feed placed in front of the vehicle. [50] used GAN for speed bump segmentation by giving colored images as an input and then generating an output image with a conditioned label as in figure 2.4. Fake images were generated by the generator network, which is so close to the real input images that the discriminator network can't differentiate between real and fake images at the end of training.

Some researchers focused on detecting only one type of road anomaly, while others considered the need to detect more than one type of anomaly. In [76, 24], the authors used Faster R-CNN to detect and classify damaged roads. The method proposed in [76] consists of two steps. The initial input image is pre-processed and enhanced. Then, features are extracted using the ResNet-152 feature extractor. The feature map is then fed as input to the Region Proposal Network as a set of rectangle object proposals containing the scores provided by the network. In the second stage, a Fast R-CNN detector is used for each object proposal, and a fixed-length

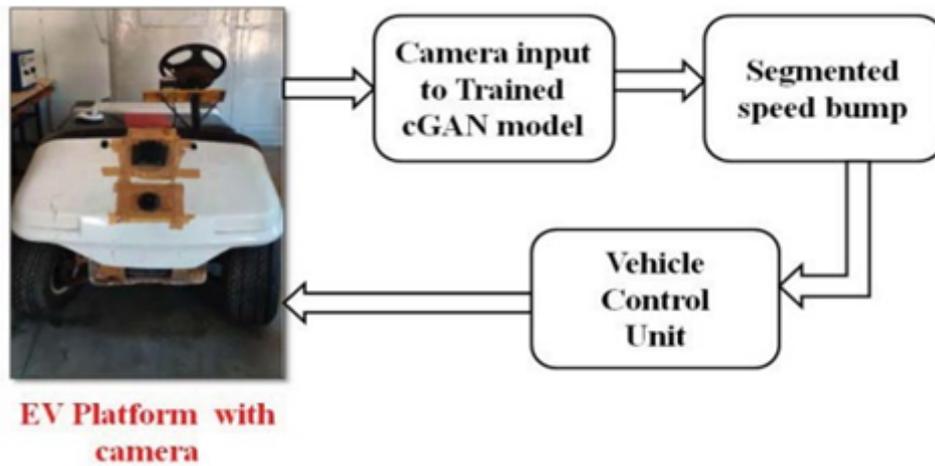


Figure 2.4: Proposed speed bump detection using GAN in [50]

feature vector is extracted from the feature map by a Region of Interest (RoI) pooling layer. Each feature vector is passed through a series of fully connected layers to predict the damaged class labels and draw bounding boxes. Cracks, crosswalks, and manholes were also studied. However, [24] proposed an ensemble model for efficiently detecting and classifying road damage (various types of cracks and potholes) using the YOLOv4 object detector.

The research aim was to detect road anomalies, their location, and their severity level in [39, 70]. These studies collected road images and accelerometer data for this task using a smartphone camera and accelerometer. Images were used to detect anomalies, while the acceleration data corresponding to the detected anomaly were used to estimate anomaly localization and severity level. Location and severity estimation can only be determined if the car exceeds the studied anomaly.

In [39], a hybrid approach is proposed to examine the presence of road anomalies using cameras and accelerometers, as shown in figure 2.5. An image of the road surface was taken with a smartphone camera from inside the car and then input into the FCN model to detect anomalies. When an anomaly is detected, it measures the 3-axis acceleration of the smartphone's accelerometer for 3 seconds. Acceleration values in the Z direction are primarily used to detect and identify road anomalies and determine the severity of the detected anomalies. In general, the greater the magnitude of the road anomaly, the greater the change in Z-axis acceleration can

be expected. They found that the maximum variation in z-axis acceleration is less than 2 m/s² when the road is free of anomalies. However, when the road anomaly was detected, most of the z-axis acceleration variation exceeded 2 m/s². Their study showed that when the detected anomaly is an irregularly shaped pothole, the acceleration changes and is more distributed than otherwise. However, for manholes with a constant circular shape, the change in value was concentrated in a narrow range. The severity of road anomalies can be identified by comparing the image with the maximum Z-axis acceleration variation.

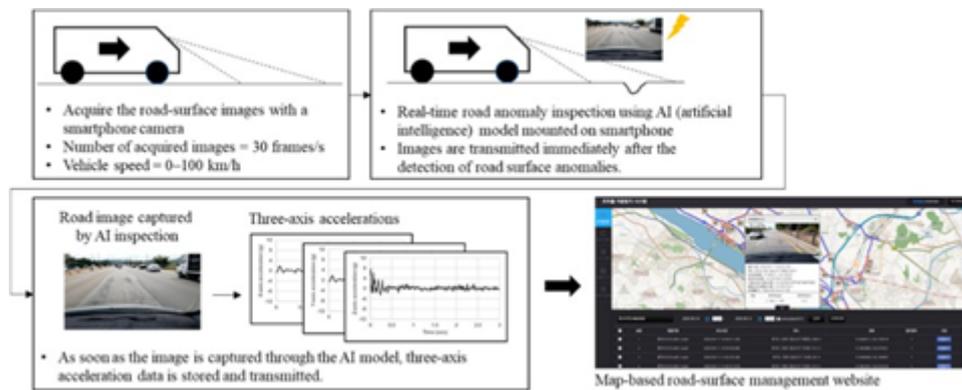


Figure 2.5: Overall image acquisition flow and three-axis accelerations with a smartphone in [39]

Another hybrid model has been proposed in [70] to evaluate paved road conditions, using a smartphone on the car dashboard to simultaneously record a video of the road and the associated acceleration response of a vehicle while driving. The system was developed to analyze accelerometer data. When an anomaly is detected, the corresponding video frame is analyzed and segmented using semantic segmentation to identify the detected anomaly.

[4] develop different CNN models for detecting pits and cracks using two variants of YOLOR, three variants of YOLOv5 (Yl, Ym, Ys), and a Faster RCNN with different backbones such as VGG16, ResNet50, MobileNetv2, Inception v3, and finally a proposed CNN called MVGG16, which is a modification of the original VGG16. The experiment was performed on a dataset created with a smartphone camera. The images were taken in different sunlight conditions in different countries. Experiments showed that Faster R-CNN ResNet50 achieves the best accuracy with 91.9%, while MobileNetV2 has the worst with 63.1%. The results also showed that

the performance of the proposed MVGG16 was improved compared to the original VGG16, but it was not the best.

2.11 Research Gap

Several factors affect the efficacy of any road anomaly detection model: type of anomaly detected, accuracy, speed, and applicability. Some researchers only focused on a specific type of anomaly [69, 6, 56, 63, 34, 21, 59], [44, 30, 35, 54, 33, 25, 5, 32], [22, 81, 8, 3, 50], and others focused on several types [76, 24, 39, 70, 4]. The accuracy of the models mentioned in the Above state-of-the-art literature review ranged between 78% and 99%. Some researchers enhanced the accuracy at the expense of speed [59, 44, 35, 32, 77, 18, 81, 8], and others increased the speed and accuracy [63]. Other applicability factors arise when using accelerometer-based or vision-based detection methods, summarized as follows:

1. For accelerometer-based approaches:
 - (a) vehicles should go over the anomaly to detect it, which may cause a lot of damage or problems to the car and the driver, like in [68, 62, 62, 49, 45, 61].
 - (b) The vehicle could go next to not over the anomaly, so the anomaly will be missed because there will be no variation in the acceleration data to indicate anomaly presence here.
2. For vision-based approaches:
 - (a) Limited datasets they used affect the accuracy obtained, especially when using Deep neural networks for classification. As [6, 35, 33, 22, 59], they used a dataset of only a small number of images rounding between 120 and 3000 images, which are also divided into training and testing images.
 - (b) Some studies did not consider the weather condition and lighting factors and their impact on the images as in [22]; they relied on images taken from inside laboratories

at their best. This will affect the detection accuracy of their systems when applied to real road images captured in different conditions.

- (c) Finally, In the studies that used cameras placed in the car, such as [21, 63, 34, 56, 44, 35, 54, 33, 5, 33, 5, 81], the calibration of the camera's place is a limitation for their work because if the camera is not at a specific angle, it cannot take proper images to be used for anomalies detection and may cause misclassification of these anomalies.

In this research, the contribution is to develop a new model to detect various road anomalies, including potholes, speed bumps, and cracks, in a quick and automated manner. Develop a novel image-based model that improves the detection accuracy and the speed at the same time. We used a drone to capture videos and images of road anomalies to overcome the limitations of using a car's camera. These were used to create a dataset for training the proposed model, while other images were used for testing.

The proposed model will pass through two phases after the pre-processing and preparation of the captured images:

1. A newly proposed region proposal model For detecting candidate regions to be enhanced and then classified using CNN, different techniques are used together to take out the least possible number of suggested regions to speed up the whole process. The proposed technique goes through several steps:
 - (a) Image is segmented using a graph-based segmentation method.
 - (b) Overlapped regions with a certain threshold are merged as one region.
 - (c) Graph-similarity technique is used to find out similar regions (regions with the same features (two potholes, for example)).
 - (d) Depending on the dynamic programming concept, similar regions are stored in an adjacency list with each other.
 - (e) One candidate region from each group entered the CNN for classification, not all of them.

Chapter Three: Methodology

2. Proposed Image enhancement model: DCGAN enhances regions from the first model. An image could have half a pothole, half a speed bump, or any anomaly that is not clear in the image, which may cause misclassification of the anomalies by the CNN in the classification phase, so here comes the role of this step. DCGAN is used to build a new image for the given image, an improved image in which what is missing in the original image is built and completed, and the accuracy of what is not clear in the original image is improved. This will increase the accuracy of the detection process in the classification phase.

In previous research, various methods were used for road anomaly detection and classification, including different models of image-based and vibration-based methods. However, as discussed in previous chapters, these models have several limitations regarding accuracy, speed, and applicability. In this research, A drone is used to capture videos of different roads in different conditions to provide a consistent and clear view of roads compared with vehicle-mounted cameras used in previous research. The video is then converted frame by frame into images. After that, a novel AI model is developed to improve the accuracy and speed compared to other models developed by previous research. This is done by developing a region proposal algorithm that enhances the model's speed. This model is then trained to detect and classify different types of road anomalies instead of focusing on one type of road anomalies. Finally, each image is then studied to detect and classify each object within this image. As depicted in figure 3.1

3.1 Image Acquisition

There are several locations where a camera can be mounted on a moving vehicle to capture road images, such as rear, front, dashboard, and others. However, the images captured from these locations can be distorted and take considerable time to cover city roads. Subsequently, a drone is used in this research to overcome these issues. Since the drone can be programmed to cover city roads during times of low traffic and good lighting conditions, the drone can also capture an overhead view of the road, providing clearer and more accurate images compared with a camera

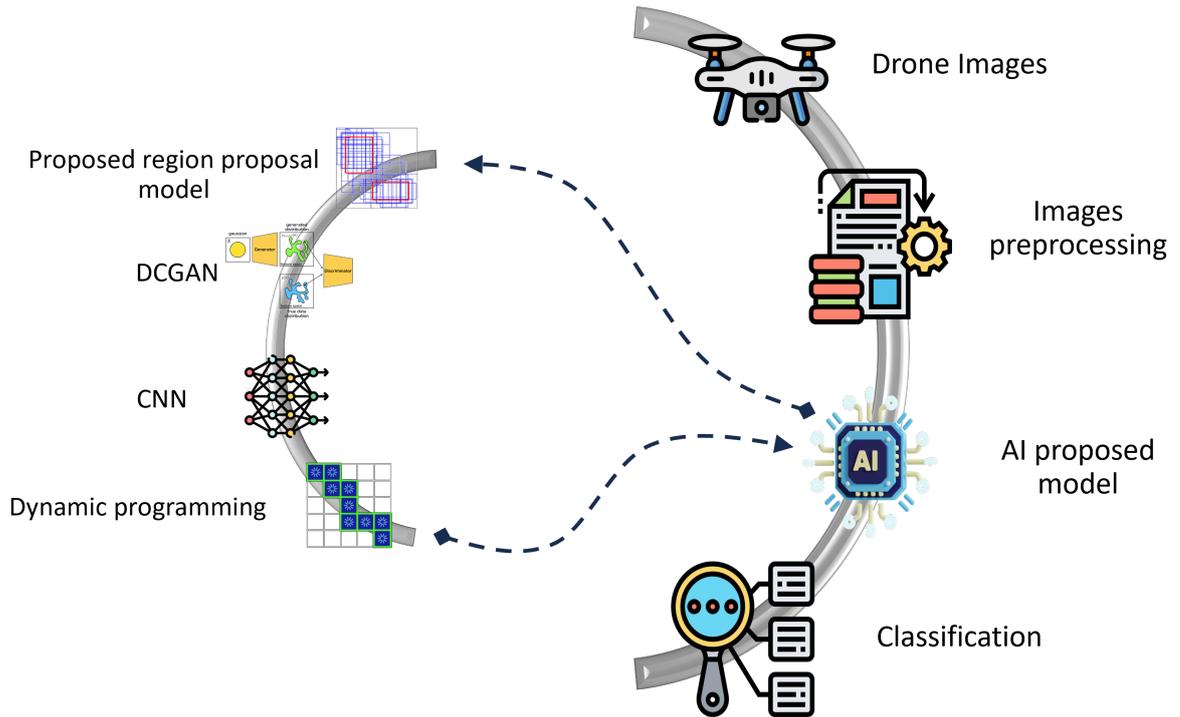


Figure 3.1: The overall methodology

mounted on a car or other moving vehicle. This research uses the DJI MAVIC Air 2 drone, as shown in figure 3.2. The drone has a camera sensor of $\frac{1}{2}$ CMOS with 48MP resolution and a field of vision (FOV) of 84O. This drone can capture videos with 4k resolution (3840x2160) with 60 frames per second. Additionally, the drone is equipped with optical image stabilization and is programmed to resist wind.

3.2 Preprocessing

The videos are captured with 4k x 60fps resolution in .mp4 format. Two key frames are extracted from each second of the captured video to form the image dataset. The videos are captured at two different times: at noon and late afternoon. This is done to consider different lighting conditions of the day. Gamma correction is utilized to unify the image lighting for images taken at different times of the day. For images captured at noon (i.e., good lighting), a gamma of 1.5 is applied to the image. Meanwhile, a gamma of 0.8 is applied to the image



Figure 3.2: DJI Mavic Air 2 Drone

captured later in the afternoon. As shown in figure 3.3

Moreover, Gaussian blur is utilized to remove noise in the image. This filter is applied with a kernel value of 5. As shown in figure 3.4

3.3 Proposed Detection and Classification Models

Two models are developed in this research. The first model, called Dynamic Similar R-CNN (Dynamic Similar R-CNN (DSR-CNN)), is divided into two parts. In the first part, each image is divided into many regions utilizing graph segmentation, graph similarity, and dynamic programming algorithms to find regions most likely to contain an object. These regions are categorized into groups utilizing a graph similarity algorithm. In the second part, these regions are used in the classification process. The second model, the Dynamic Generative R-CNN (Dynamic Generative R-CNN (DGR-CNN)) model, is utilized to enhance the proposed regions



Original image



Gamma correlation with gamma of 1.5

Figure 3.3: Image before and after gamma correction



Gamma correlation



Gaussian blur

Figure 3.4: Image before and after applying Gaussian blur

from model 1 before the classification. Models 1 and 2 are shown in figures 3.5, and 3.6.

3.3.1 Dynamic Similar R-CNN model (DSR-CNN)

To enhance the performance of Fast R-CNN, a new model is proposed to choose candidate regions from the input image to be used as CNN inputs for feature extraction, object detection, and classification. The model utilizes a graph-based segmentation algorithm to divide the image into a set of regions. After that, the graph similarity technique is used to identify similar regions, store each group of similar regions in a list, and then submit one region from each set to the

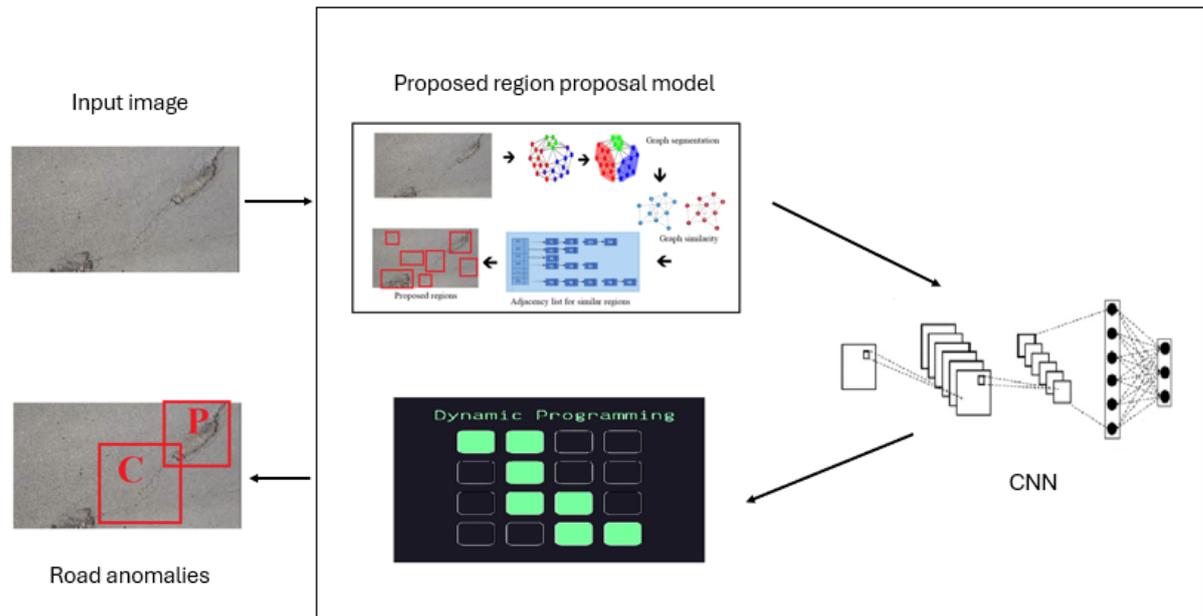


Figure 3.5: Dynamic Similar R-CNN model

CNN model for the feature extraction process.

3.3.1.1 Image segmentation

Image segmentation is the process of dividing an image into a group of areas called regions/segments. This reduces the image's complexity and makes it easy to study, analyze, deal with, and extract the required information. Technically, segmentation labels pixels in an image to distinguish between objects, people, or other significant elements.

A common use of image segmentation is object detection. Instead of processing the entire image, a common practice is using an image segmentation algorithm to find objects of interest in the image. Then, the object detector can operate on a bounding box already defined by the segmentation algorithm. This prevents the detector from processing the entire image, improving accuracy and reducing inference time. One of the commonly used image segmentation algorithms is graph-based image segmentation.

Graph-based segmentation first represents the image as a graph $G = (V, E)$ with vertices $v \in V$ and set of edges $(v_i, v_j) \in E$. Each edge has a weight $w(v_i, v_j)$, indicating the dissimilarity

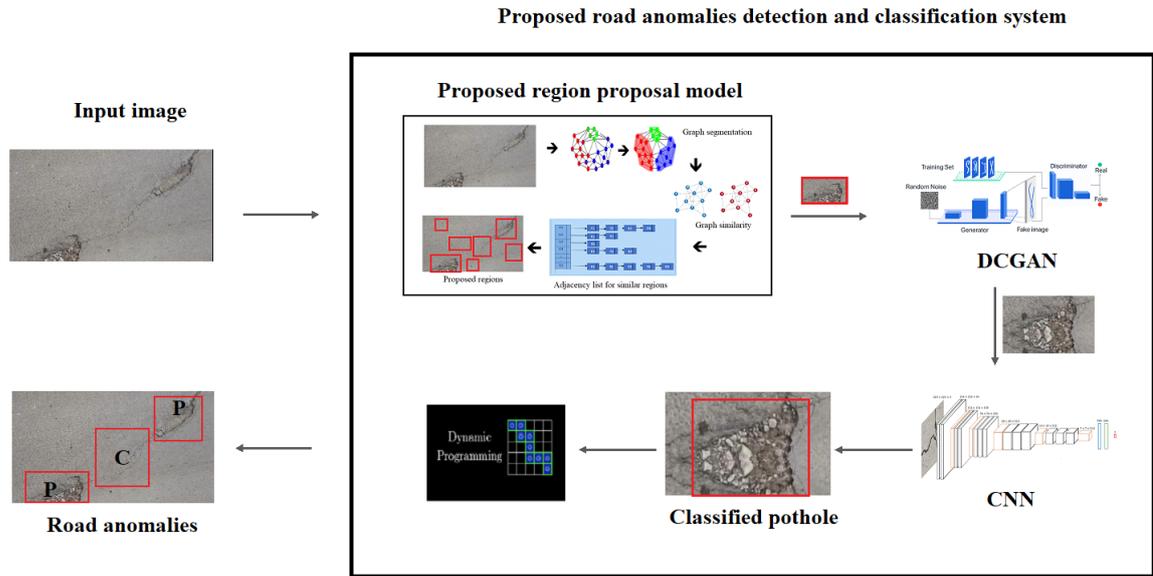


Figure 3.6: Dynamic Generative R-CNN model

of the two connected vertices. In graph-based image segmentation, each pixel is represented by a vertex of V , and the connection between pixels is represented by edges such that each edge (v_i, v_j) represents a connection between pixel i and pixel j . The weights on these edges are a measurement that shows the difference between two adjacent connected pixels in terms of color, intensity, location, or any other parameter. Segmentation process S is segmenting the graph into several connected graphs/segments $C \in S$.

To begin the cutting process, the distances between each pixel and its neighboring pixels are estimated to cut dissimilar pixels (vertices). The entire graph is then divided into many continuous graphs at the end of the process. The edges between nodes in the same graph have relatively low weights, and edges between nodes in different graphs have higher weights. Each resulting graph or region represents a candidate object from the image that can be studied by its features being extracted using any feature extraction techniques [27] in the object detection process. Graph-based segmentation result shown in figure 3.7

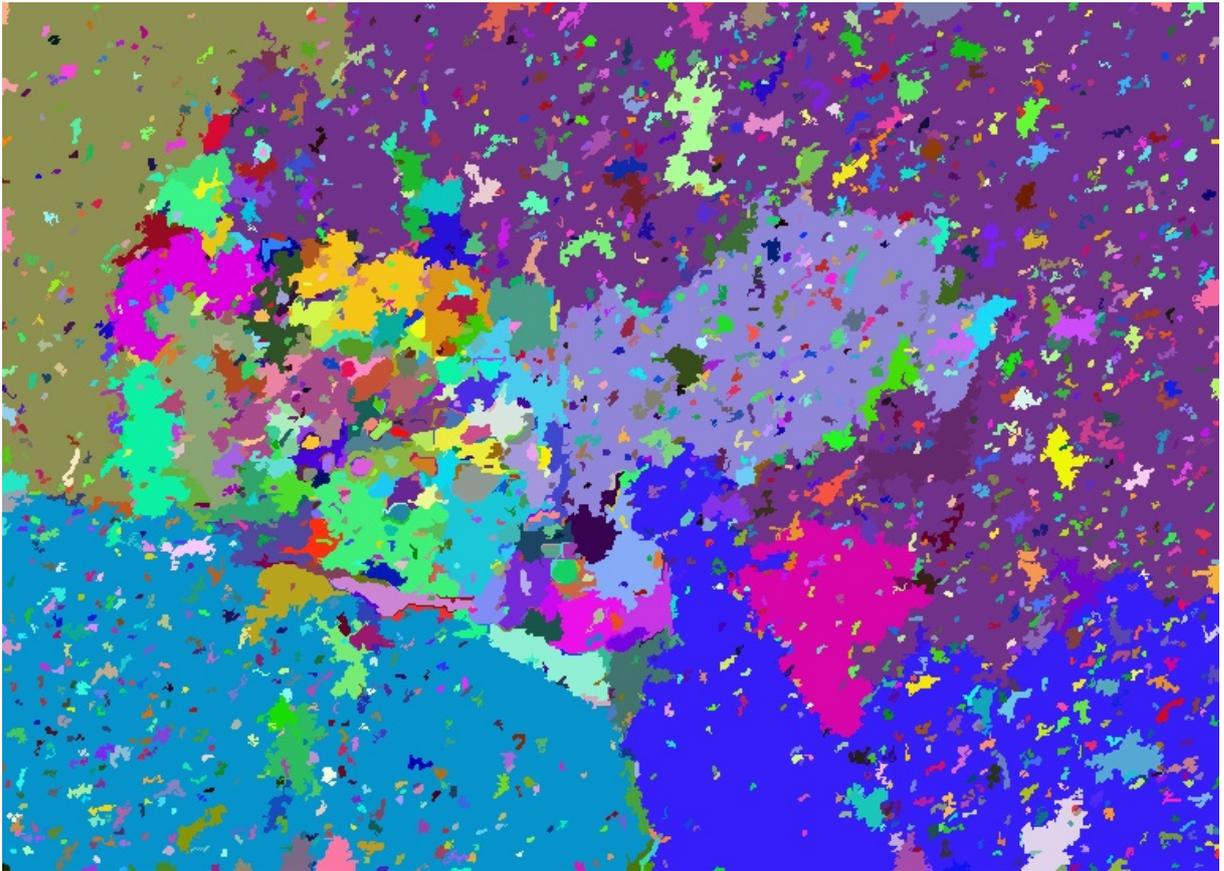


Figure 3.7: Graph-based segmentation result

3.3.1.2 Finding similar regions

Graph-based segmentation of an image produces many potential objects in the form of graphs. However, multiple regions may represent the same thing or contain the same information, which can cause the CNN to analyze the same objects multiple times, leading to increased computations and longer processing times. To address this issue, a method is proposed to avoid repeated analysis of similar graphs by reducing the number of regions to be analyzed by CNN. This is achieved through a two-step approach. Firstly, overlapped regions are combined to form a single region. Secondly, similar regions far apart are identified, reducing the number of candidate regions.

The overlapped regions can be found by calculating the degree of intersection between two regions (Intersection Over Union (IOU)). IOU can be calculated by estimating the area of the

intersection area and dividing it by the area of the union as in equation 3.1:

$$IOU = \frac{intersectionArea}{((area1 + area2) - intersectionArea)} \quad (3.1)$$

Area 1 and Area 2 represent the areas of the overlapped regions (region 1 and Region 2), and the intersection area is the area where the two regions overlap.

Finally, if the overlap exceeds a predetermined threshold of 70%, the two regions are combined and treated as one region. This method combines a large number of regions, greatly reducing the number of candidate graphs.

The second step in reducing the number of proposed regions for CNN is identifying candidate regions that are similar but located far apart from one another. The graph similarity technique will determine the degree of similarity between these graphs since each candidate region (from the graph-based segmentation) is represented by a graph. In this study, the eigenvector similarity method is employed to determine how similar two graphs are to one another[38].

Consider G_1 and G_2 are two graphs. For eigenvector similarity, the Laplacian of the graphs is calculated as in equations 3.2 & 3.3:

$$L_1 = D_1 - A_1 \quad (3.2)$$

$$L_2 = D_2 - A_2 \quad (3.3)$$

Where A_1 and A_2 , D_1 and D_2 , L_1 and L_2 represent the Adjacency matrix, Diagonal matrix of degrees, and Laplacian of G_1 and G_2 respectively.

Compute the eigenvalues of each Laplacian and find the smallest k such that the sum of the top k eigenvalues is 90% of the sum of all eigenvalues. If both graphs have different k values, use the minimum of both. Then, calculate the similarity as in equation 3.4.

$$sim = \sum_{i=1}^k (\lambda_{1i} - \lambda_{2i}) \quad (3.4)$$

Two graphs are considered similar or related to the same object if their similarity exceeds a certain threshold. Various similarity thresholds (50%, 60%, 70%, and 80%) were tested in this study. A threshold of 70% gave the best results.

3.3.1.3 Group similar regions

The above section explained how to find similar regions from an image. Once these regions are identified, they are grouped and stored in a list. Only one region from each group is then fed to the Convolutional Neural Network (CNN) for analysis. This approach reduces the number of regions that need to be entered into the CNN to one-fifth, resulting in a faster analysis process and overall classification.

An adjacency list is a list used to store similar graphs or segments. The list contains nodes that correspond to groups of similar graphs, and each node points to a list of those groups, as shown in figure 3.8, where each G stands for a group of similar graphs and R for a graph or region from the image.

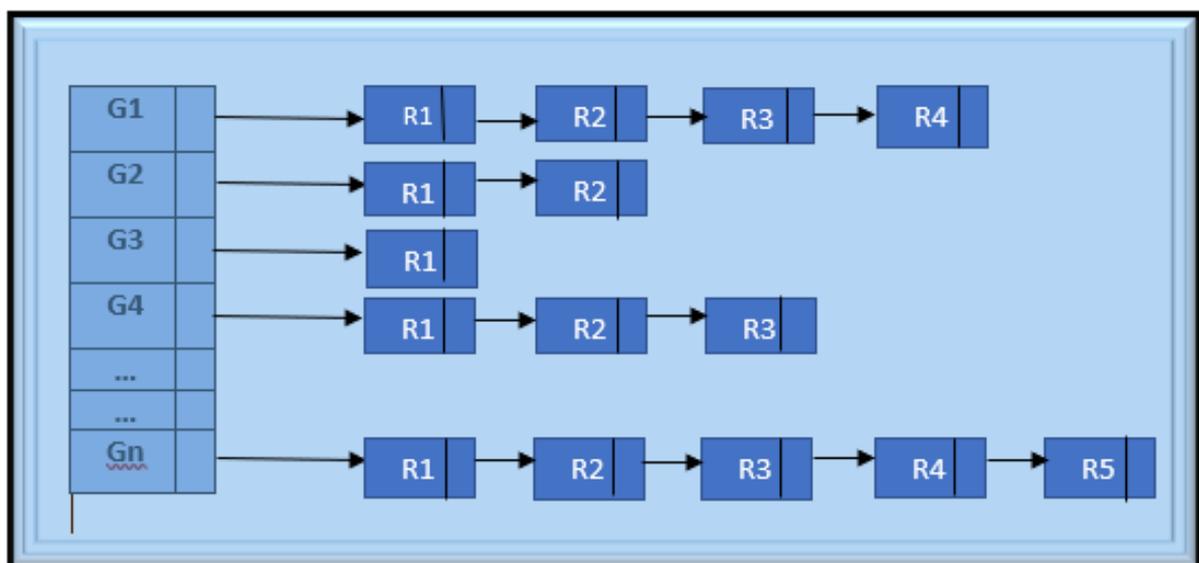


Figure 3.8: Adjacency list of similar graphs

3.3.1.4 Classification step

In the previous steps, the image was segmented into many regions. Most of these regions were eliminated to account for regions more likely to contain an object. These regions were then categorized into groups of similar regions. In this step, the CNN model considers one representative region of each group for classification. However, each representative region is first converted to a vector of 4096 elements. Then, the average of each of the two adjacent elements is calculated to form a vector containing 2048 elements. This is done two more times to get a vector containing 512 elements. This vector size is the default size of the VGG16 convolutional network and does not accept any other dimensions. Finally, this vector is used to train the deep neural network model to get the correct classification for this region. Then, this classification is used for all the regions the classified region represents.

Objects are divided into five categories: speed bumps, manholes, potholes, cracks, and no anomaly, as illustrated in 3.1. Each group has different visual and general characteristics, including shape, color contrast, texture variations, and structural characteristics. Speed bumps are identified by their wide edges, color differences, and elevated structure. The manholes are typically circular or rectangular and are made of concrete or steel. Potholes have irregular edges, shadows, and varying depths. Cracks are darker, show different textures, and create various patterns. Finally, any object that does not fit the previous categories is labeled "No Anomaly." This may include obstacles, normal asphalt conditions, or other instances.

3.3.1.5 Dynamic programming

Dynamic programming is an important technique to improve algorithm performance and minimize calculations. It primarily depends on reusing stored solutions instead of solving them again, which speeds up calculations significantly. This technique is widely used in diverse applications, including image processing, Artificial intelligence, deep learning, and detection and classification problems.

Table 3.1: Studied Road anomalies and their main features

Class	Visual properties	General properties
Speed Bumps	<ul style="list-style-type: none"> 1- Shadows around the edges 2- Wide clear edges 3- Change in texture between road and speed bump 4- Different colors than surroundings 	<ul style="list-style-type: none"> 1- Slightly higher than background
Manholes	<ul style="list-style-type: none"> 1- Circular or rectangular shape with clear edges 2- Different colors the surroundings 3- Different texture between road and Manhole 	<ul style="list-style-type: none"> 1- Made of concrete or steel 2- Location doesn't change
Potholes	<ul style="list-style-type: none"> 1- Lighter or darker color than road 2- Shadows due to depth difference. 3- Irregular edges 	<ul style="list-style-type: none"> 1- Deeper than road 2- Different depths and sizes
Cracks	<ul style="list-style-type: none"> 1- Generally darker in color 2- Different texture than road 3- Different shapes (longitudinal, lateral, irregular) 	<ul style="list-style-type: none"> 1- Appear due surface layer erosion 2- Single or connected in a network
No Anomaly	Any other objects are classified as no anomaly	

Dynamic programming primarily relies on dividing a problem into smaller tasks that may recur during the solution. When this recurrence occurs, the stored solution is used. Several data structures can be used to store these solutions, including a list of lists, a hash table, or a graph structure. Thus, when a solved problem is encountered, the stored solution is retrieved directly, significantly improving the speed and subsequently improving the algorithm's efficiency.

Dynamic programming can be used in digital image processing, detection, and classification problems. The image may contain similar objects, such as identical cars, people wearing the same clothes, buildings with the same architectural style, similar parts of the background, and others.

This research uses dynamic programming after segmenting the image into candidate regions as follows:

1. Similar objects are stored in an adjacency list connecting similar candidate regions. The regions with similar features are stored in the same group in this list. previously done in Section 3.3.1.3.
2. One representative region is taken from each group to be classified. As explained in section 3.3.1.4.
3. Finally, the representative object classification is applied to all the regions in the same group.

This process considerably reduces the calculation time, improving the system's overall efficiency.

3.3.2 Dynamic Generative R-CNN (DGR-CNN)

This model consists of three steps. The first step is finding candidate regions using DSR-CNN, explained in section 3.3.1. The second step is to improve the accuracy of the proposed regions using the DCGAN model. The final step is to classify the objects within these enhanced regions. The second step utilizes the DCGAN model to enhance the accuracy of the proposed regions.

The DCGAN model first converts the image to vector form by applying sequential CONV array filters (i.e., discriminator phase). This vector contains all the features within the image. The generated vector is used to generate a new image. This is done utilizing the convolution transpose algorithm. This algorithm applies the CONV transpose array filter to predict the pixels within the image. This is performed for several epochs to produce a more complete and high-quality image than the original image (i.e., generator phase). The general workflow of the DCGAN model is shown in figure 3.9.

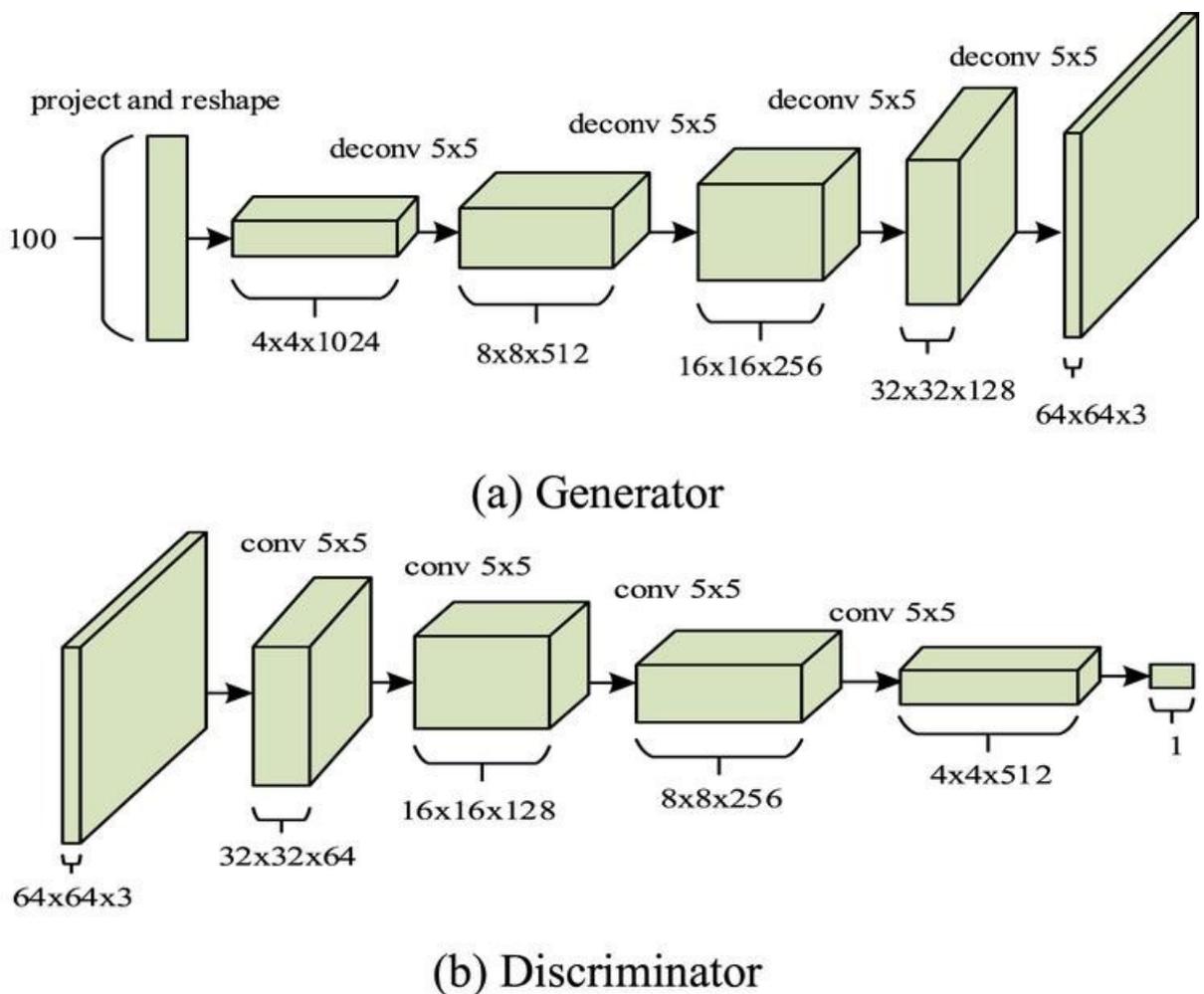


Figure 3.9: DCGAN architecture

3.3.2.1 Discriminator phase

In this phase, the region is first converted to the LAB color space since it has a larger color range than the RGB color space. Then, five sequential CONV array filters are applied to this image. The output of these filters is a vector containing the image's essential features. This vector is entered into an activation function to decide whether this image is real or fake. The image is first reshaped to 128x128x3 size. Then, the first CONV array filter is applied. This filter comprises a window of 8x8 size (64 filters) that moves two pixels per stride. Then batch normalization and the Leaky Relu function are applied to the resulting image. The Leaky Relu activation functions, shown in equations 3.5, are utilized to add non-linearity to the network. The Relu function shown in figure 3.10 sets all inputs below zero to exactly zero. If Relu is applied to the input image, all values below average will be treated as the same (=0). Since image inputs are often standardized below the average intensity of some colors, they will have negative values. Using the Leaky Relu function, shown in figure3.11, pixels below the average will be given the "opportunity" to build features based on significant local combinations of pixels that have values >0 and are not "squashed" by the activation function.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \quad (3.5)$$

Where x is the input to the Leaky Relu function, and α is the slope for negative inputs, usually a small positive value like 0.01.

When training deep neural networks, batch normalization is used to standardize the inputs for each layer of each mini-batch. As a result, the learning process is stabilized, drastically decreasing the number of training epochs needed to train deep networks.

After the batch normalization and the Leaky Relu are applied, it is then entered into the second CONV array filter. This filter is made with a window of 128 filters with a stride of 2 pixels. The resulting image, which is 32x32x128 in size, is normalized using the batch normalization

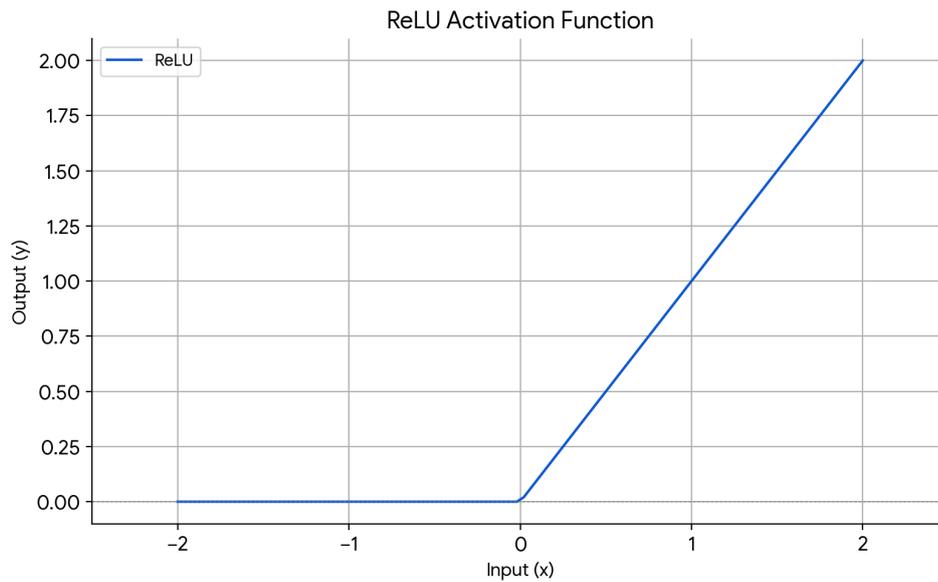


Figure 3.10: Relu activation function

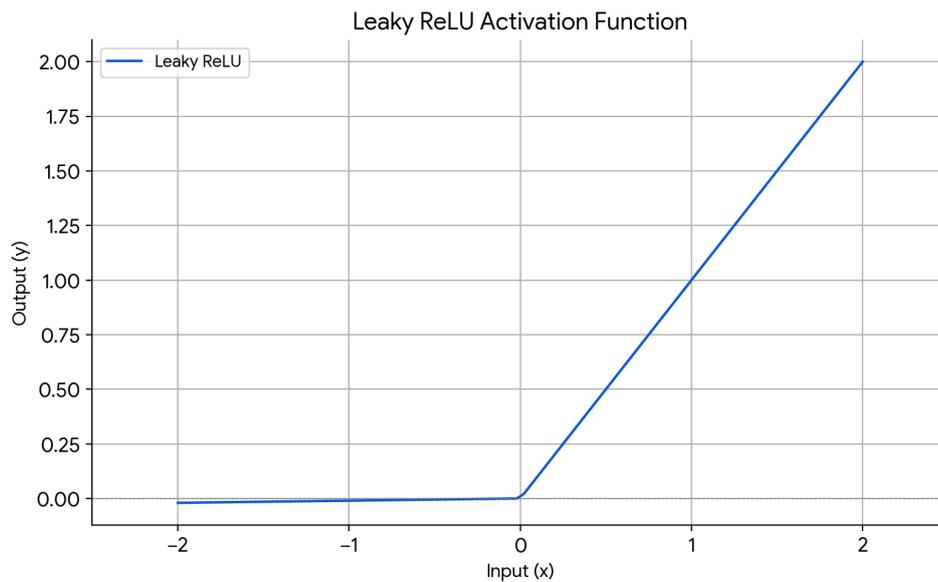


Figure 3.11: Leaky Relu activation function

discussed earlier, and then the Leaky Relu activation function is applied as well.

The third CONV array filter contains a window of 256 filters with a stride of one pixel. Then, batch normalization and leaky Relu functions are applied. The fourth and fifth filters comprise windows of 512 filters and 1024 filters, respectively, with a stride of 2 pixels.

A stride of one pixel is used in the third filter since the image was significantly reduced.

A stride of two pixels will result in a significant loss in the image features. The image, which is 8x8x1024 in size, is flattened to a vector form that contains 65,536 elements. As shown in figure 3.12.

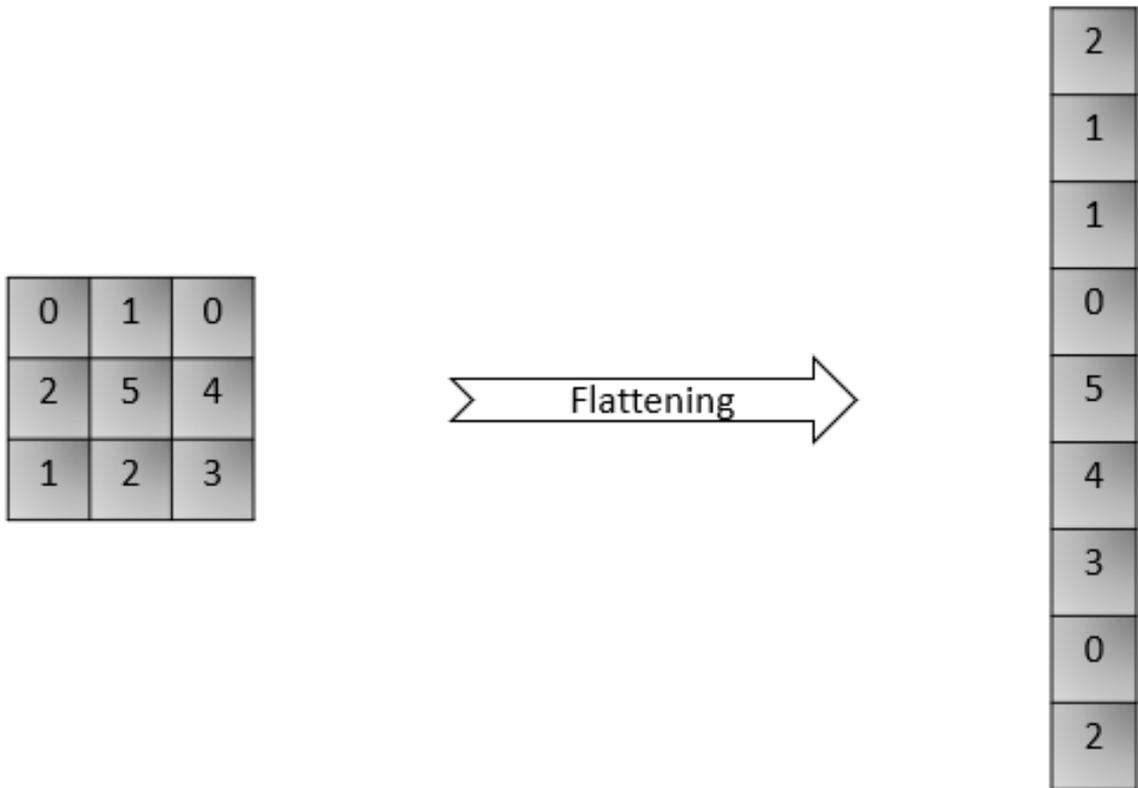


Figure 3.12: Matrix to flatten

This vector is entered into the Sigmoidal function shown in figure 3.13 and equation 3.6 to decide whether the image is real or fake.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$

3.3.2.2 Generator phase

The generator begins with a fully connected layer, followed by a sequence of transpose convolution layers, batch normalization, and the Leaky Relu activation function. This model is

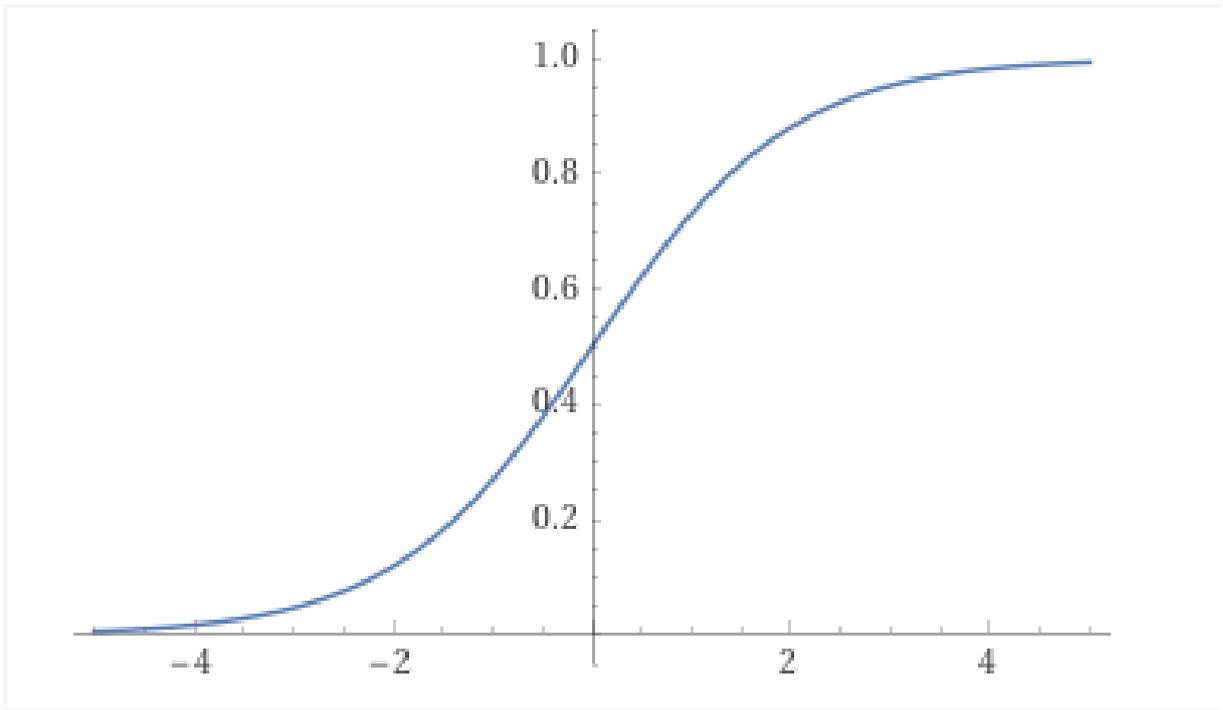


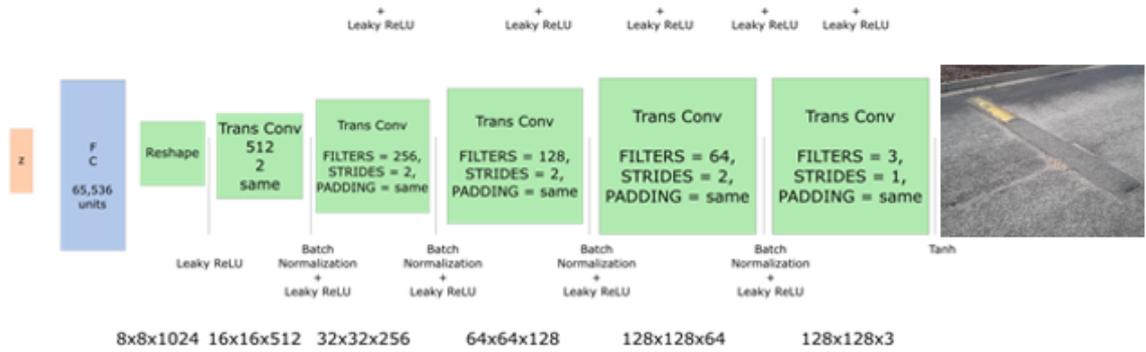
Figure 3.13: Sigmoid function

developed to “generate” a new image from a given vector. Which will contain the features described by the given vector. The image is then sent to the discriminator model to decide whether this image is real or fake. Suppose that the discriminator returns a false value (fake image). In that case, the original vector is modified utilizing the back-propagation technique and entered into the generator to produce a new image to send to the discriminator. This is repeated until a true value (real image) is obtained from the discriminator, as shown in figure 3.14. The generator goes through several steps as follows:

- Vector Generation

The first step is to generate a random vector comprised of 65,536 elements. The vector is then entered into a fully connected convolution layer that contains 65,536 nodes. The vector is then reshaped to a size of 8x8x1024. After that, the reshaped vector goes through several up-sampling, batch normalization, and Leaky Relu steps to get a 128x128x3 size. The batch normalization and the Leaky Relu steps are discussed in section 3.3.2.1.

- Up-sampling



Generator

Figure 3.14: Generator architecture

The upsampling is performed via the CONV-Transpose filter. This filter comprises a kernel array of size (N) and a stride of length (M). The new vector size is given by equation 3.7. The values of the new vector elements are obtained by multiplying the old and the kernel arrays and summing the values in each element location to get the new vector as represented in figure 3.15.

$$NewSize = (oldSize * N) - M \quad (3.7)$$

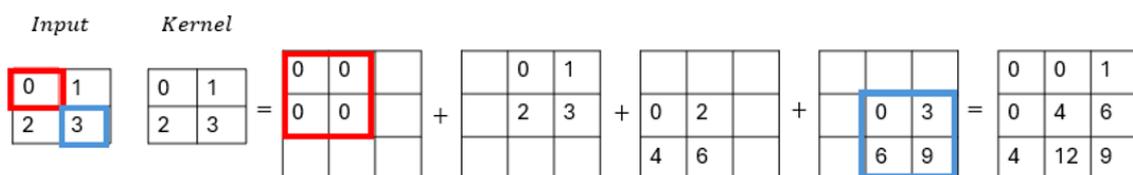


Figure 3.15: Up sampling process

As depicted in figure 3.15, the vector is converted from 8x8x1024 to 16x16x512 size in the first up-sampling step using a kernel array of 512 with a stride of 2 elements. The second up-sampling step uses a kernel array of 256 with a stride of 2 elements to convert the vector into 32x32x256. The remaining steps utilize 128, 64, and 3 kernel arrays. And

a stride of 2, 2, and 1 elements, respectively. The back-propagation technique updates the kernel vector element values in each step. The Leaky Relu function is applied to the first reshaped vector. After that, the Leaky Relu function and batch normalization are applied between any two consecutive CONV-Transpose filters. Finally, the Tanh activation function, shown in figure 3.16, is applied to finally to produce an enhanced image. This function is utilized due to its ability to model non-linear boundaries as it does not saturate for small inputs as sigmoid.

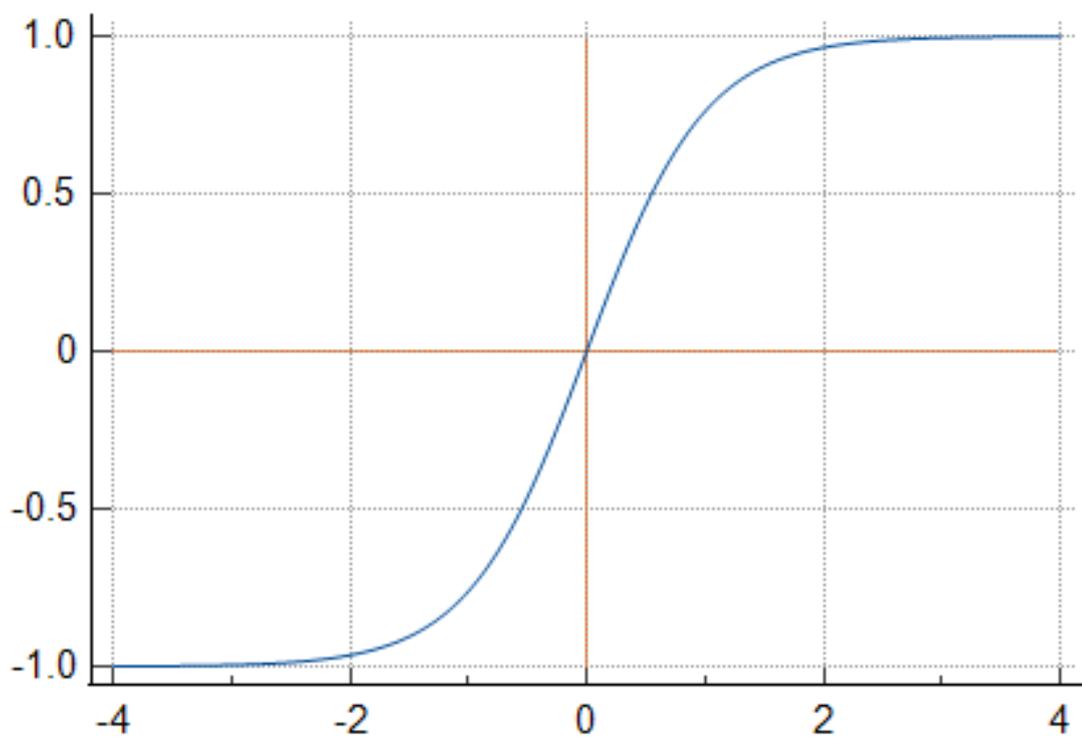


Figure 3.16: Tanh Function

3.3.2.3 DCGAN training

The following steps are repeated in training

- First, the Generator creates some new examples.

- The Discriminator is trained using real data and generated data.
- After the discriminator has been trained, both models are trained together.
- The Discriminator's weights are frozen, but its gradients are used in the Generator model so that the Generator can update its weights.

3.3.2.4 Standard GAN loss function (min-max GAN loss)

The discriminator and generator losses are calculated to train the model to move in the right direction. These losses are computed during the discriminator training and image generation phases, respectively. The overall loss function for a DCGAN is defined as the sum of the loss functions for G and D.

- Generator losses

The generator's goal is to generate real images to deceive the discriminator. Generally, the generator's loss is based on the difference between the generated and actual images. The most commonly used loss function for the generator is binary cross-entropy loss.

The generator attempts to minimize this loss by generating images that are almost identical to the original ones according to the discriminator. The generator loss can be written as in equation 3.8:

$$GL = -\log(D(G(z))) \quad (3.8)$$

Where $G(z)$ is the image generated when random noise z is given to the generator, and $D(G(z))$ is the discriminator's evaluation of the generated image. The generator's goal is to minimize this loss.

- Discriminator losses

The goal of the discriminator is to differentiate real from generated images. The discriminator loss consists of two components: one for the real image and the other for the

generated image. The discriminator aims to maximize the ability to distinguish between the two images. The discriminator loss function is expressed in equation 3.9:

$$DL = -\log(D(x)) - \log(1 - D(G(z))) \quad (3.9)$$

Where $D(x)$ represents the output of the discriminator when it evaluates a real image x , while $D(G(z))$ is the discriminator's output when it evaluates a generated image.

During training, the generator and discriminator are iteratively updated to find the Nash equilibrium. It is important to balance generator and discriminator losses during the training process. If the generator becomes too powerful, it will produce too realistic images, and the discriminator will have trouble distinguishing between real and generated images.

3.3.2.5 Classification step

The image was initially divided into multiple regions, and most of these regions were removed to focus on those more likely to contain an object. Next, the selected regions were grouped based on similarity and improved. The CNN model then considers one representative region from each group for classification. First, each representative region is converted into a 4096-element vector. This vector is then averaged to form a 2048-element vector, and this process is repeated twice more to obtain a 512-element vector. Finally, this vector is used to train the deep neural network model to correctly classify the region, and this classification is then applied to all other regions represented by the classified region.

3.3.2.6 Dynamic programming

In the final stage, we utilized the dynamic programming principle, which consists of two parts. First, we implemented the storage principle discussed in Section 3.3.1.3. This entailed organizing images and similar images into an adjacency list.

The second part involved classifying each group’s representative. Subsequently, we identified all images similar to this representative in the list and assigned them the same label.

3.4 Evaluation

Finally, the performance of the developed models is evaluated by calculating different statistical parameters, namely precision, recall, accuracy, and F1-score. These parameters are calculated from the confusion matrix shown in 4.10

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure 3.17: Confusion matrix

Precision measures how many of the predicted positive cases are correct, as shown in 4.2, indicating the model’s performance in identifying the correct class. However, Recall captures how many positive cases were correctly identified, as shown in 4.3, which highlights the missing positive cases. Accuracy provides an overall measure of correctness, as shown in 4.1. Finally, the F1 score balances precision and recall, making it useful when there is a trade-off between false positives and false negatives as in equation 4.4. By analyzing these metrics together, it can be determined whether a model favors precision over recall, handles class imbalances effectively, or achieves an optimal balance between errors and correct predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.10)$$

Chapter Four: Results

$$Precision = \frac{TP}{TP + FP} \quad (3.11)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.12)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.13)$$

In addition to these statistical measures, detection and classification speed is measured to assess the model's efficiency in processing and identifying objects. A model that achieves high accuracy but operates too slowly may not be practical in real-world applications. Therefore, balancing detection accuracy with computational speed is essential for optimizing performance.

These metrics, speed and accuracy, are calculated for each class in the dataset. The average performance of each model is computed and compared against one another alongside other leading two-shot object detectors like R-CNN [29], Fast R-CNN [28], Faster R-CNN [58], and other state-of-the-art road anomaly detection and classification algorithms. Babu et al. [7] used tiny-YOLOv4 to detect potholes and cracks. Mahenge et al. [43] proposed a deep-learning model for crack detection utilizing the GAN and CNN algorithms. Zhao et al. [83] developed a model (MED-YOLOv8s) to detect cracks and potholes using YOLOv8s and Mobilenetv3 as the backbone of the detection algorithm. Finally, Swain et al. [71] developed a model to detect potholes using CNN with VGG16 as a backbone and SRGAN.

The developed models are compared with two-shot object detectors rather than one-shot models such as YOLO and its variations. Although speed is an important consideration, our application does not require real-time detection and classification, making YOLO's ultra-fast nature less critical. Additionally, YOLO-based models tend to struggle when detecting small and closely packed objects, which is a significant challenge in road anomaly detection and classification.

In this research, novel models are developed to detect and classify different road anomalies.

First, videos are recorded using a DJI Mavic Air 2 drone. The images are then extracted from these videos. Then, the images are pre-processed to remove light disparities and blur. These images are then used to train the proposed models, which have three steps. In the model step, graph similarity and graph segmentation techniques are utilized to divide the image into segments most likely to contain an object and categorize these segments into groups. Then, in the second model, the DCGAN algorithm is applied to enhance these segments and fill incomplete segments. Finally, CNN classifier and dynamic programming techniques are employed to classify the object in each representative segment and increase the model's speed in both models.

This chapter is divided into three main sections: data collection, Model training, and Proposed model results. The first section discusses data collection, preparation, and pre-processing. The second section discusses the training of the developed model in terms of the optimum CNN backbone and the optimum number of training epochs. Finally, the third section discusses the results of the developed model.

Two scenarios are simulated: the first model is without the use of the DCGAN algorithm (Dynamic Similar R-CNN (DSR-CNN)), and the other model is simulated utilizing the DCGAN algorithm (Dynamic Generative R-CNN (DGR-CNN)).

4.1 Data Collection

As discussed in section 2.11, images used in previous research, captured using the camera on the vehicle, were affected by the conditions of the street itself, which caused vibrations and instability in the movement of the car. That leads to taking distorted, inaccurate, and unclear images or missing important details. Additionally, the image taken by the camera inside the vehicle does not depict a complete view of the street, which causes a loss of some parts of the image that may contain anomalies, which affects the detection and classification results. This research uses a drone instead of a vehicle camera to overcome these problems. A new dataset is created. Roads scattered in Tulkarm city with excellent to very poor conditions are studied.

Videos are captured using DJI Mavic Air 2 drones with a resolution of 3840x2160 (4K). 1500 videos covered 81km of road length with a total runtime of around 5112 minutes recorded. The videos were taken at various times of the day, morning and evening, and under different lighting conditions. The drone camera's stabilization system ensures the videos are free from distortion or vibrations, even in windy weather. The videos are filmed at a height of approximately three meters, providing a complete view of the street and capturing all details and anomalies present. This helped to enhance the accuracy of the detection and classification process.

From these videos, 15,326 images were extracted to create the dataset. The images are manually classified into four classes: 4781 images of cracks, 4196 images of potholes, 3475 images of manholes, and 2874 images of speed bumps. These images are in the same resolution as the videos as shown in figure 4.1, 4.2, 4.3, and 4.4

These images are then pre-processed to enhance lighting and remove noise using gamma correction and Gaussian blur filters. A gamma of 1.5 was used for evening images and 0.8 for morning images. A Gaussian blur filter with a kernel size of 5 was applied. A sample of the enhanced images is shown in figure 4.5, 4.6, 4.7, and 4.8. These enhanced images are used in the following steps of the model.

4.2 Training

The processed images are divided into three categories: testing, training, and validation. Seventy-five percent of the images are used for training and validation and twenty-five percent for testing. This section demonstrates the results of training the developed models. Two parameters are optimized during the training process: the CNN CNN's backbone algorithm and the number of training epochs for the DSR-CNN, DGR-CNN, DCGAN discriminator, and DCGAN generator. Google platform, specifically Google Colab, provided high server resources to process and train the developed model.

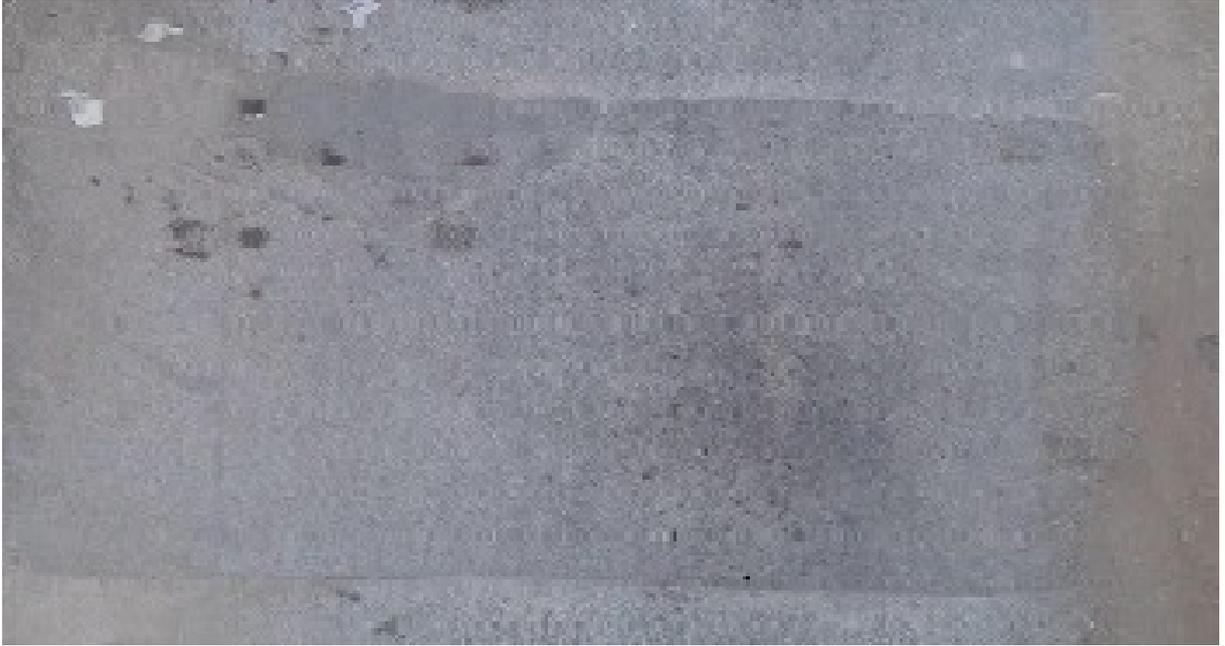


(a)



(b)

Figure 4.1: Cracks



(a)



(b)

Figure 4.2: Speedbumps



(a)



(b)

Figure 4.3: Manholes



(a)



(b)

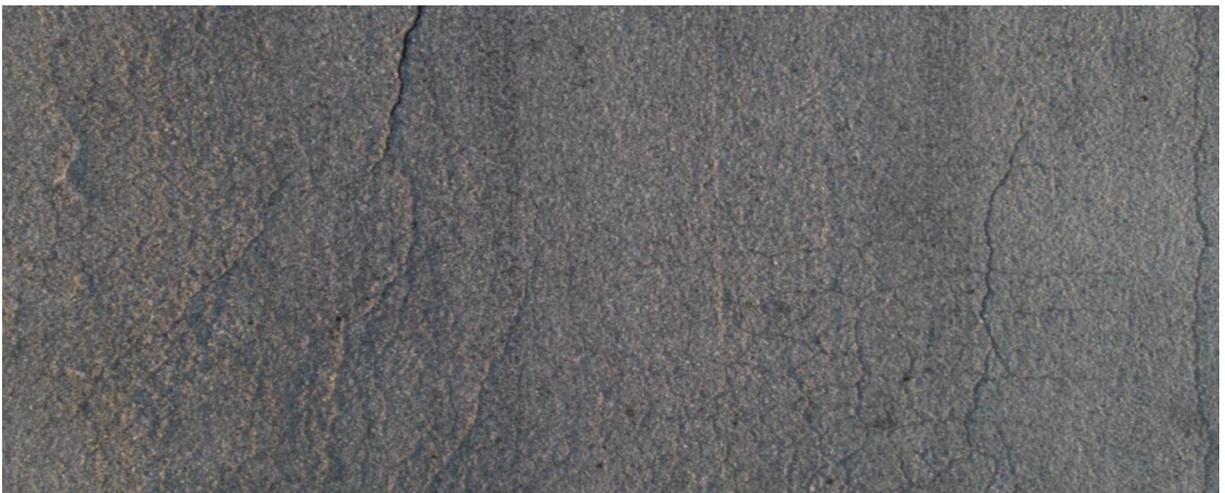
Figure 4.4: Potholes



(a) Original image



(b) Gamma correction



(c) Gaussian blur

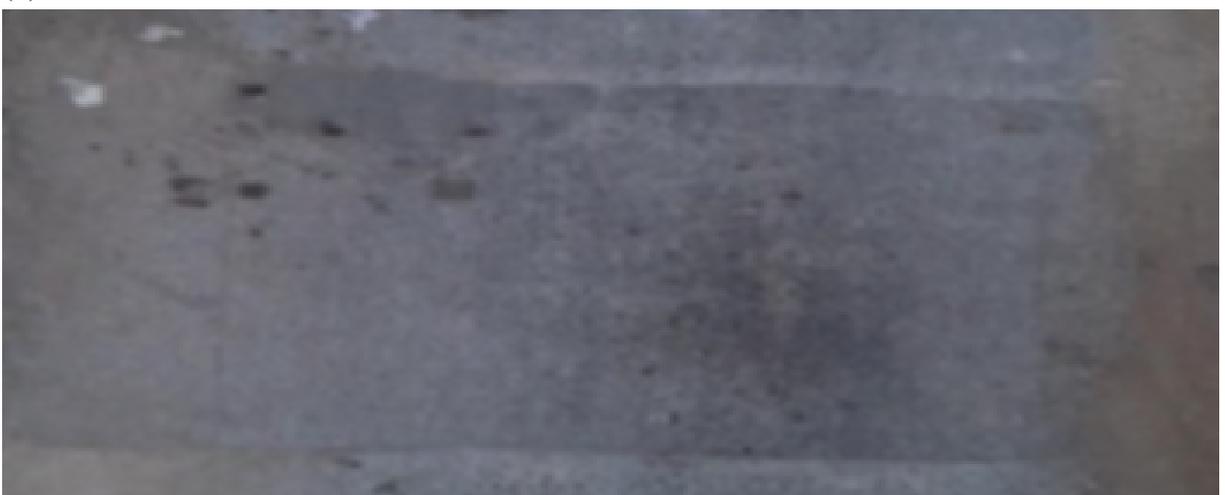
Figure 4.5: Cracks pre-processing using Gamma Correction and Gaussian blur



(a) Original image



(b) Gamma correction



(c) Gaussian blur

Figure 4.6: Speedbumps pre-processing using Gamma Correction and Gaussian blur



(a) Original image



(b) Gamma correction



(c) Gaussian blur

Figure 4.7: Manhole pre-processing using Gamma Correction and Gaussian blur



(a) Original image



(b) Gamma correction



(c) Gaussian blur

Figure 4.8: Pothole pre-processing using Gamma Correction and Gaussian blur

Different backbones for CNN: Several models can be used as a backbone for the CNN model, namely VGG16, RESNET50, and InceptionV3. The accuracy of these backbones is shown in table 4.1. As shown in Table 4.1, the VGG16 algorithm demonstrates the highest accuracy of 94.85 among other algorithms considered in this study, which gained 92.61 and 90.92, respectively. Therefore, the VGG16 algorithm is chosen as the backbone for the developed model.

Table 4.1: Accuracy of the different CNN backbones

Model	Accuracy
Vgg16	94.85%
Resnet50	92.611%
InceptionV3	90.921%

Number of training epochs: The number of epochs required for training is optimized for the DGR-CNN, DSR-CNN, DCGAN Discriminator, and DCGAN Generator models.

As depicted in 4.9, the total loss in all models decreased significantly during the first 400 epochs. After that, the models stabilized, and the loss reduction rate decreased.

The loss in the DGR-CNN decreased from 21.1 at the 400th epoch to 10 at the 700th epoch. After that, the loss in the model decreased by only 1.9 at 1200 epochs. This indicates that the model has reached the optimum number of training epochs, and no further improvements can be made with continued training. This decline in loss reduction was observed between the 1200 and 1300 epochs when the loss decreased from 8.1 to 7.9.

The stabilization started for the DSR-CNN, DCGAN Discriminator, and DCGAN Generator models at the 800th epoch with a loss of 18.7, 17, and 19, respectively. The models reached the optimum number of training epochs at 1200, where the loss reached 17, 14.7, and 15.4. After that, the models' losses did not decrease significantly, indicating that the 1200th epoch is the stabilization point.

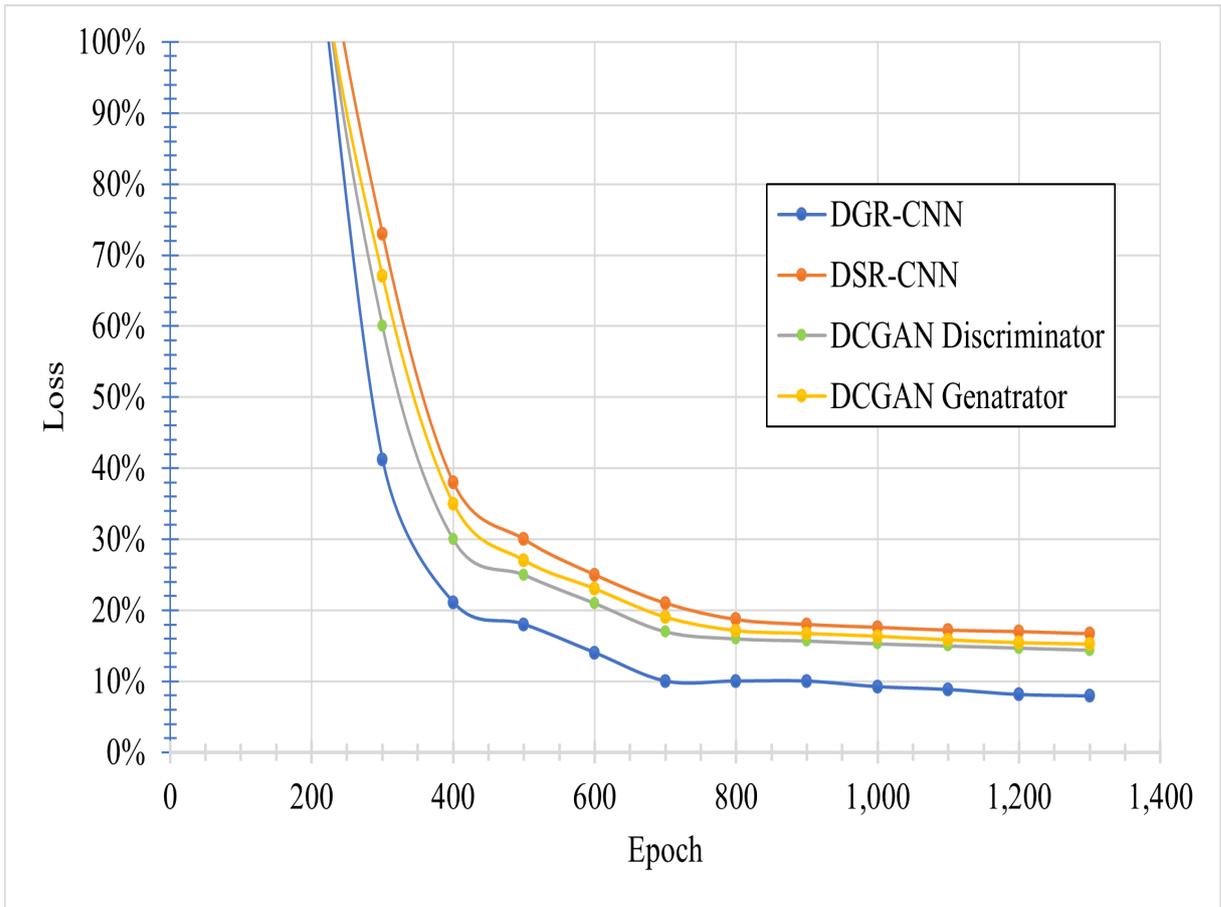


Figure 4.9: Loss vs number of epochs

4.3 Developed Deep Learning Model Results

As discussed above, the model consists of three steps. The first model divides each image into segments and groups them into similar segments. After that, a representative sample from each group is entered into the CNN for classification. The second model enhances the first model using the DCGAN technique. Finally, the enhanced sample is entered into a CNN model for classification, and all segments in the same group are assigned the same classification. The results were obtained for the two models: the DSR-CNN and DGR-CNN.

- **Model 1: Dynamic Similar R-CNN (DSR-CNN)** This model applies Graph segmentation with graph similarity and dynamic programming to store each image with its counterparts.

- **Model 2: Dynamic Generative R-CNN (DGR-CNN)** This model enhances Model 1 to improve recognition and classification accuracy by enhancing the proposed using the DCGAN algorithm.

Google platform, specifically Google Colab, provided high server resources to process and train our model. The VGG-16 classifier is used as a backbone for the CNN model in both configurations. Four parameters are adopted to measure the model’s effectiveness: accuracy, recall, precision, and F1-score. These measures are calculated from the confusion matrix shown in figure 4.10 as shown in equations 4.1, 4.2, 4.3, and 4.4. This matrix divides the results into four categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure 4.10: Confusion matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

4.3.1 Model 1: DSR-CNN

In this section, the results of the DSR-CNN model are illustrated.

The process goes through two phases:

1. Proposed region proposal phase

This phase is performed to extract the informative sections of the input image, which will be analyzed and studied later. This phase is accomplished in several steps:

- divide the image using a Graph segmentation algorithm.
- Using graph similarity to find similar regions among the regions from step one.
- Grouping similar regions in adjacency lists to analyze only one of each group using dynamic programming techniques.

2. Segments classification

The final stage of the developed model classifies the proposed segments into five categories: holes, manholes, speed bumps, cracks, and No Anomaly.

4.3.1.1 Proposed region proposal phase

This step determines the number of segments to be used for classification. The larger the number of segments, the more time the CNN model needs to classify these segments. The selective search algorithm traditionally proposes around 2000 regions be used as input to the CNN model. This research proposes a novel region proposal model to overcome that problem. The developed model utilizes graph segmentation and graph similarity algorithms to propose regions for the CNN model. In this step, each image is segmented using the graph segmentation algorithm as shown in figure 4.11, 4.12, and 4.13. Overlapped regions are merged, and similar graphs are

grouped using graph similarity techniques. The dynamic programming technique is then used to store each group and select only one representative segment from each group for the next steps of the developed model. This process will considerably reduce the number of regions that CNN will process.

As depicted in table 4.2, in the manhole class, the developed model proposed 150 regions compared to 1910 regions proposed by the selective search algorithm. Almost the same ratio is observed in the other remaining classes. 120 to 1450, 140 to 1750, and 135 to 1801 in the class that contains potholes, speed bumps, and cracks, respectively.

Table 4.2: Number of region proposals using DGR-CNN vs selective search

Region Proposal Method	Class			
	Manholes	Potholes	Speed bumps	Cracks
Dynamic Generative R-CNN	150	120	140	135
Selective Search	1910	1490	1750	1801

The developed technique proposed, on average, 8% of the segments proposed by the selective search algorithm. This will significantly increase the model speed without affecting the model's accuracy, as will be discussed in the later sections.

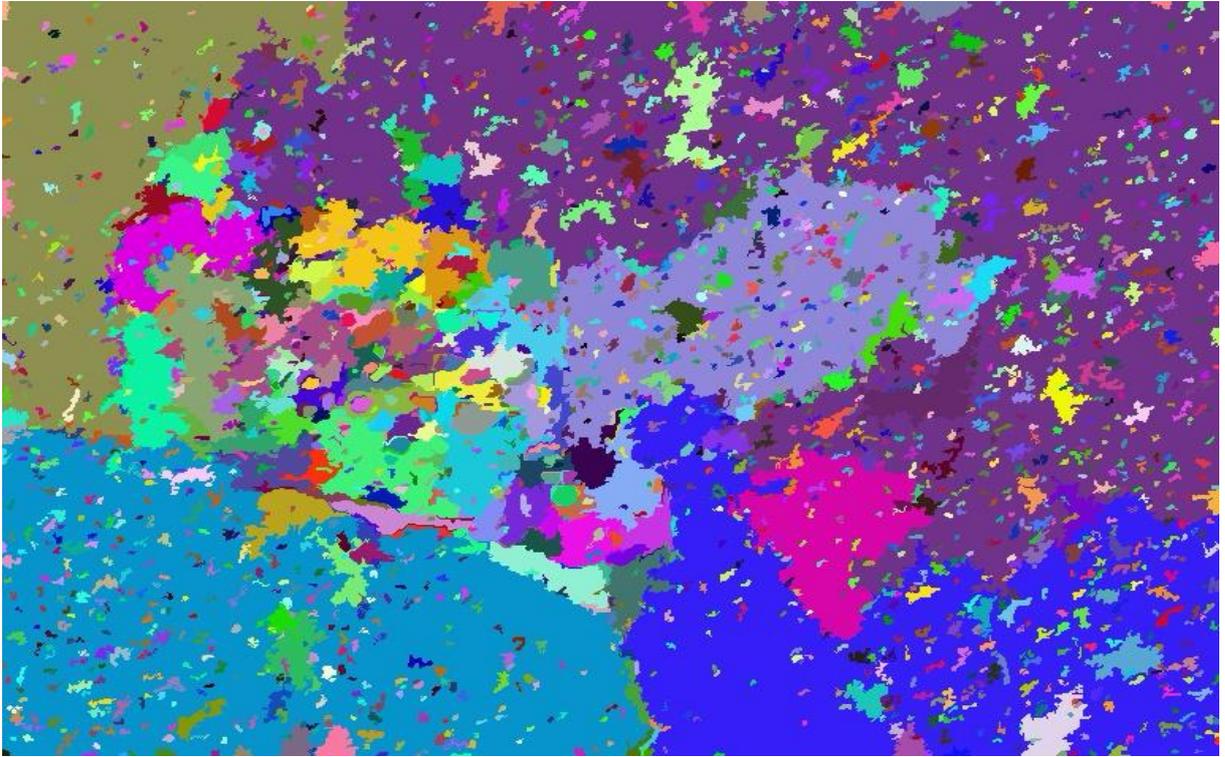
4.3.1.2 Segments classification

The final stage in the developed model is to classify the proposed segments into one of the following classes: potholes, manholes, speed bumps, and cracks. In this model (DSR-CNN), the proposed segment is sent directly to the CNN model for classification. The results of this model are presented in table 4.3

As shown in table 4.3, the class representing "Manholes" achieved the highest accuracy, which is expected because it has a clear and relatively consistent shape, resulting in stable de-



(a) Original pothole

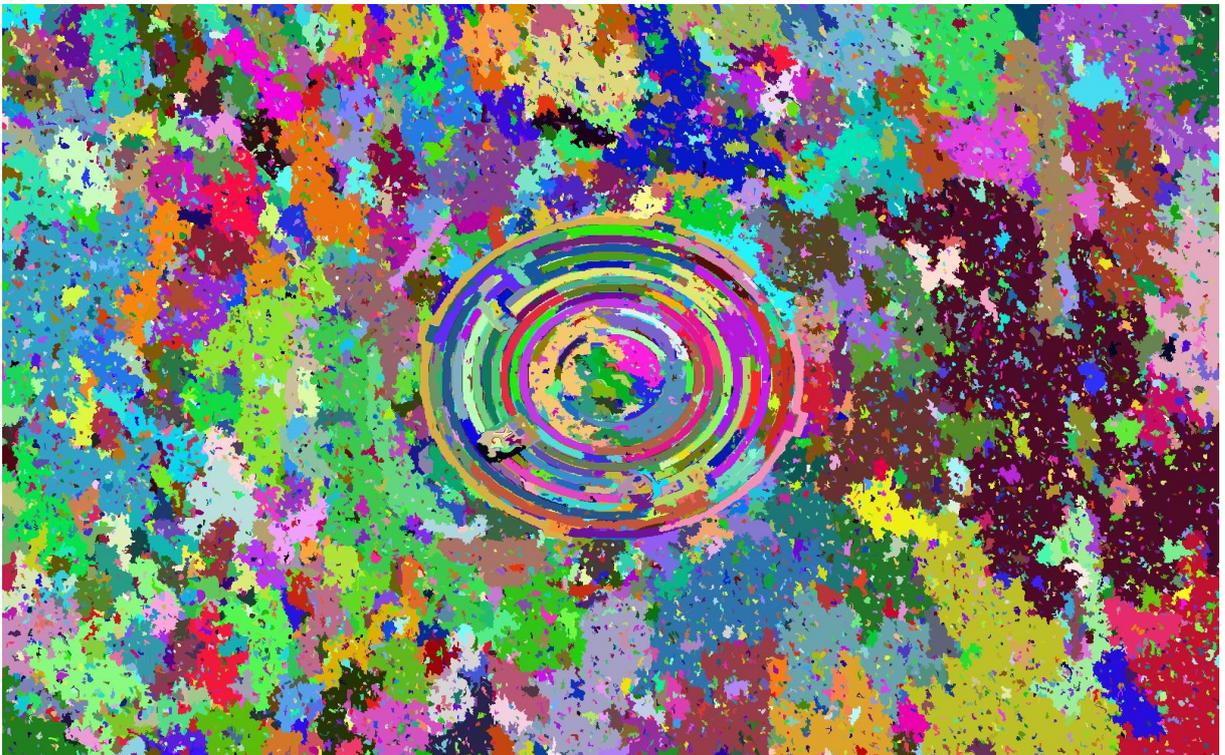


(b) Segmented pothole

Figure 4.11: Potholes Segmentation using graph-based segmentation



(a) Original manhole

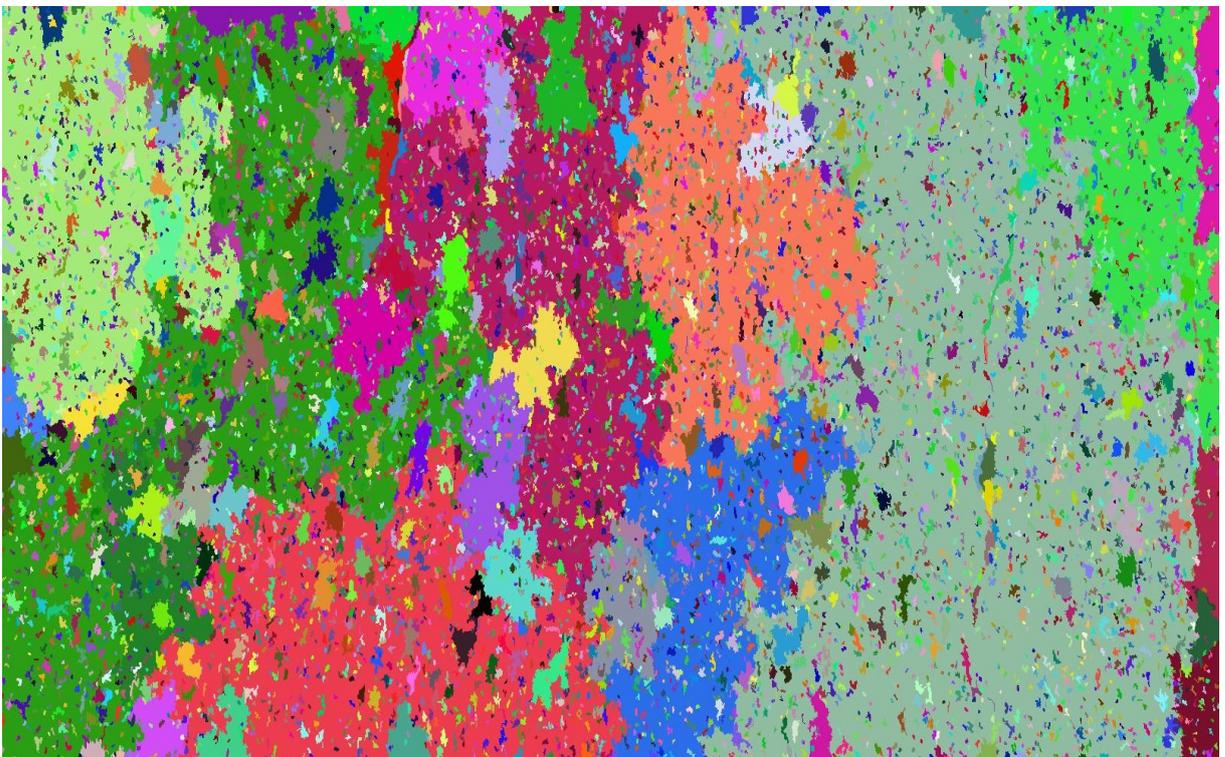


(b) Segmented manhole

Figure 4.12: Manhole Segmentation using graph-based segmentation



(a) Original cracks



(b) Segmented cracks

Figure 4.13: Cracks Segmentation using graph-based segmentation

Table 4.3: F1 score, Recall, Precision, and Accuracy of using DSR-CNN

Class	F1-Score	Recall	Precision	Accuracy
Pothole	82.77%	82.31%	83.24%	84.23%
Manhole	83.32%	83.32%	83.14%	84.52%
Speedbumps	82.19%	82.21%	82.17%	83.34%
Cracks	82.52%	82.34%	82.71%	83.63%
No anomaly	82.6	82.57	82.82	83.65
Average	82.68%	82.55%	82.82%	83.93%

tection. The accuracy for this class was 84.52%. On the other hand, the class representing "speed bumps" had the lowest accuracy. Speed bumps come in various shapes and sizes, and their depth in the images can vary significantly. Detecting the specific features of speed bumps can be challenging, leading to a lower accuracy than other anomalies. Precision can be viewed as a quality measure, and recall as a measure of quantity. Therefore, we see from the previous table that high precision often means in most of the results of the table, especially in the class represented by potholes, so that our model displays results that are more relevant than irrelevant results, meaning that it has the correct classification. Also, the high recall in most table results means that our model displays most of the relevant results (whether irrelevant results are returned), especially in the holes class. Furthermore, most of the Precision, Recall, and F1-score results are higher than 82%.

Finally, the model had the best results in the manhole and pothole classes because they are the most distinctive because the manhole is the largest in shape and size, and therefore, the model was able to distinguish it greatly. As for potholes, it is distinct from other classes, meaning that they have a specific style in their description, so the results were from In terms of accuracy in the class of manholes, it was 84. 52%, while the class of potholes was 84.23%. As for the classes with speed bumps and cracks, it was the least accurate because they do not have

a specific shape or size, and just as the features for speed bumps are close to the street's feature, These results were still distinctive and very good, as the cracks reached 83.63% and the speed bumps reached 83.34%.

4.3.2 Model 2: DGR-CNN

The second model enhances the first model and consists of three stages. In this scenario, DCGAN is dedicated to improving recognition and classification accuracy by enhancing the proposed regions.

The model goes through four phases:

1. Proposed region proposal phase

This phase is performed to extract the informative sections of the input image, which will be analyzed and studied later. This phase is accomplished in several steps:

- Segmenting the image using a Graph segmentation algorithm.
- Finding similar regions among the regions from step one using Graph similarity.
- Grouping similar regions in adjacency lists to analyze only one of each group using Dynamic programming techniques.

2. Segments enhancement

This may involve enhancing and/or completing the segment using the DCGAN algorithm.

3. Segments classification

The proposed enhanced segments were classified using CNN into one of the following classes: potholes, manholes, speed bumps, and cracks.

4. Dynamic programming

Finally, after classification is finished, the same classification is assigned to all images linked to the image entered into CNN for classification; this is based on the principle of dynamic programming, which is storing and retrieving similar storage data.

Chapter Five: Discussion

The same classes (speed bumps, cracks, manholes, and potholes) are detected by this model. And The same accuracy metrics are calculated, namely, Accuracy, Precision, and Recall, and the F1 score, as shown in table 4.4

Table 4.4: F1-score, Recall, Precision, and Accuracy of the proposed system with enhancements

Class	F1-Score	Recall	Precision	Accuracy
Pothole	94.05%	93.8%	94.3%	95.53%
Manhole	94.65%	94.1%	95.2%	95.83%
Speedbumps	92.65%	92.2%	93.1%	93.72%
cracks	93.8%	93.8%	93.8%	94.31%
No anomaly	93.8%	93.5%	94.1%	94.86%
Average	93.79%	93.48%	94.10%	94.85%

Based on the previous table, the Manhole and the Pothole classes achieved the highest accuracy with 95.83% and 95.53%, which is expected as they have a distinct shape. The classes with the lowest accuracy are the speedbumps and cracks classes, as they come in various shapes, sizes, and depths within the image, making them very difficult to distinguish. Additionally, all precision and recall results exceed 93%. These results emphasize how the DGR-CNN model outperforms the DSR-CNN model.

In this chapter, the results of the developed models (DGR-CNN and DSR-CNN) will be compared with each other and with other leading algorithms in general object detection and road anomaly detection fields. These results were obtained for all models for the dataset developed for this research and using CoLab servers.

DGR-CNN vs. DSR-CNN : Table 5.1 depicts both models' average accuracy statistics and speed. The DSR-CNN model achieves an average accuracy of 83.93 % compared with 94.85%

from DGR-CNN, which increases the accuracy by almost 10% while only decreasing the speed by 0.15s.

The DSR-CNN model processes 5.21 frames per second, while the DGR-CNN model processes 3.82 frames per second.

Table 5.1: DSR-CNN

Model	F1-score	Recall	Precision	Accuracy	Speed
DSR-CNN	82.68%	82.55%	82.82%	83.93%	0.117s
DGR-CNN	93.79%	93.48%	94.10%	94.85%	0.262s

Figures 5.1, 5.2, 5.3, and 5.4 depicts the accuracy statistics (precision, recall, and F1 score) for both models in each anomaly class: potholes, manholes, Speed bumps, and cracks.

In all classes, a significant increase in all statistics is observed. The DGR-CNN model improved precision, recall, accuracy, and F1 score with the same ratio. Therefore, using DCGAN in the model significantly increases the classification accuracy without significantly affecting the speed.

Overall, the DGR-CNN model outperforms the DSR-CNN model across all road anomaly classes.

DGR-CNN, DSR-CNN vs. general object detection models: Table 5.2 shows the speed and MAP of the developed models (DGR-CNN and DR-CNN) compared with other leading algorithms in this field.

As shown in tabl 5.2, the DSR-CNN significantly outperforms both R-CNN, and fast R-CNN in terms of speed while achieving similar accuracy. However, the DGR-CNN model significantly outperforms both the fast R-CNN and R-CNN in terms of speed and accuracy as well.

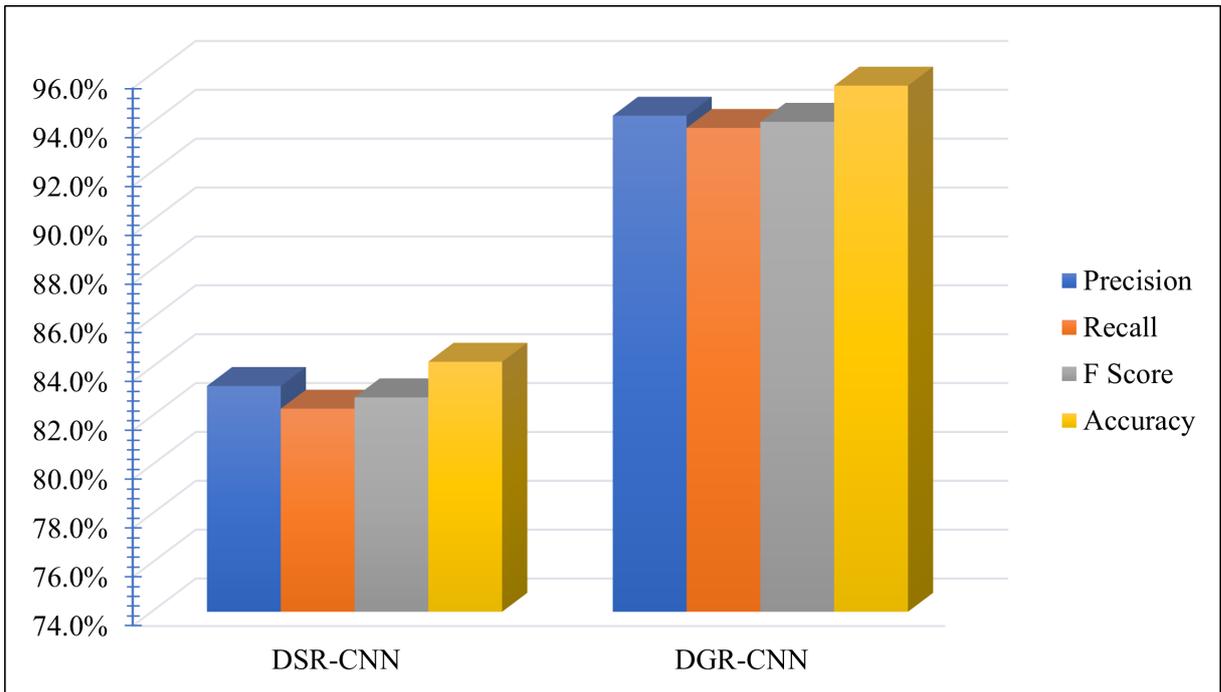


Figure 5.1: Statistics for DSR-CNN compared with DGR-CNN for potholes class

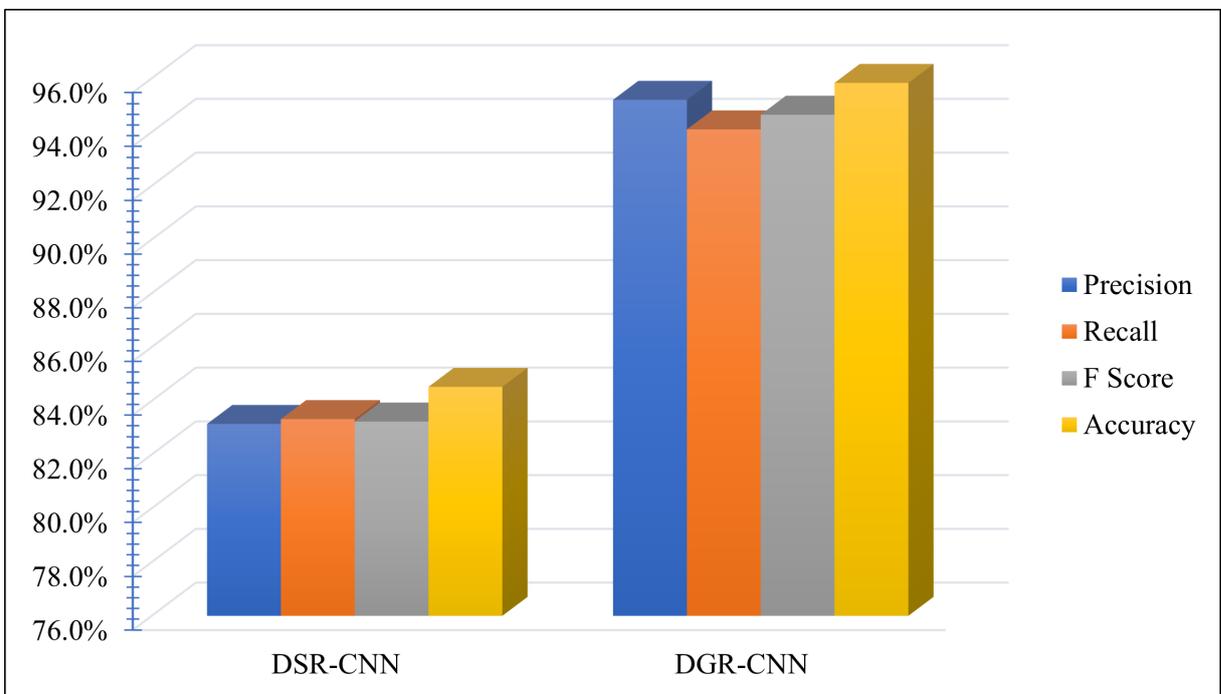


Figure 5.2: Statistics for DSR-CNN compared with DGR-CNN for manholes class

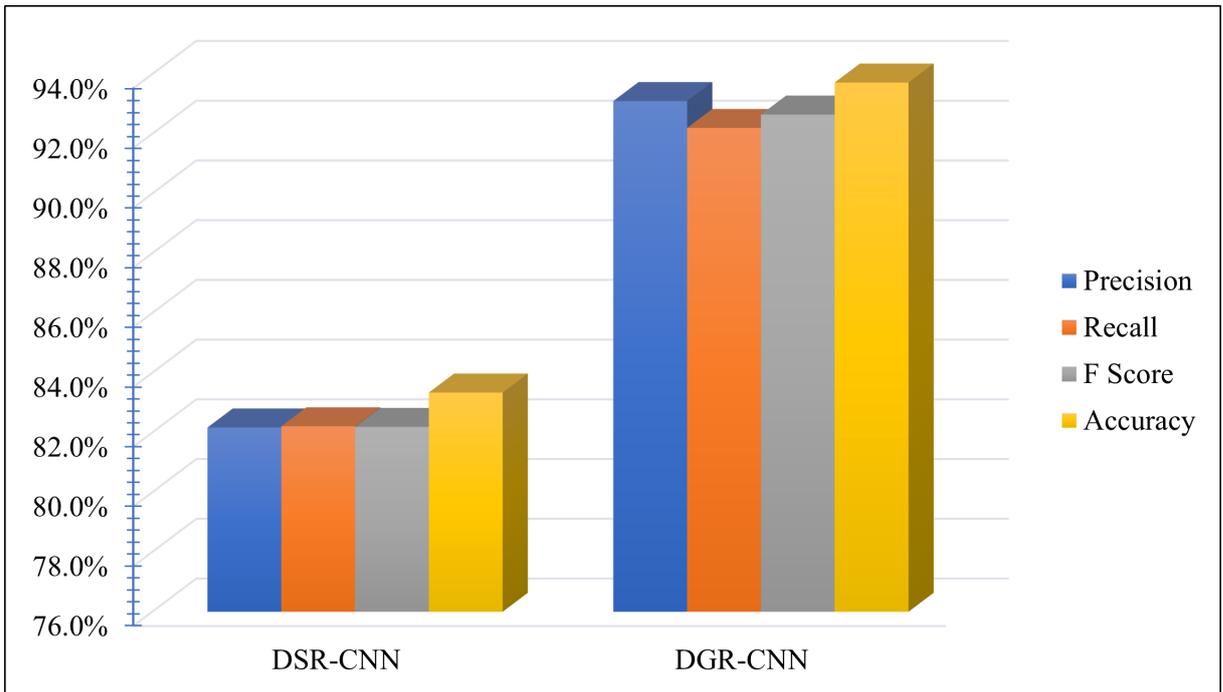


Figure 5.3: Statistics for DSR-CNN compared with DGR-CNN for speedbumps class

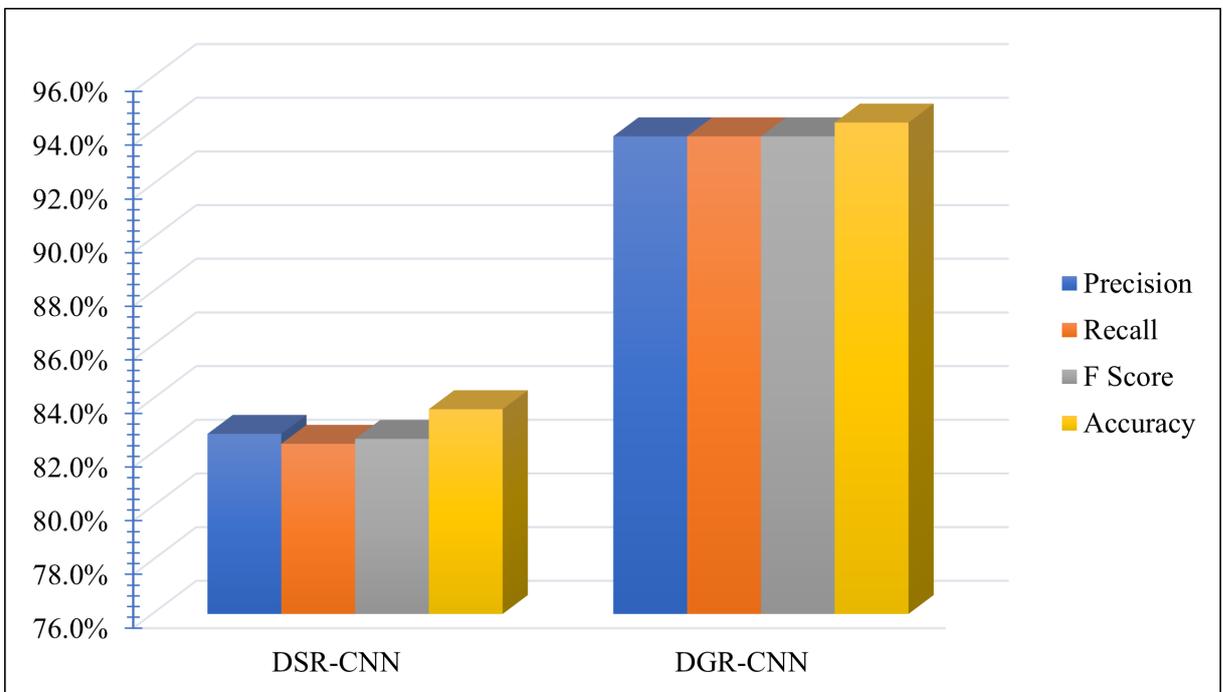


Figure 5.4: Statistics for DSR-CNN compared with DGR-CNN for cracks class

Table 5.2: F1-score, Recall, Precision, and Accuracy of the proposed system with enhancements

speed	testing / per image	MAP
R-CNN [29]	47s	81.61%
FAST R-CNN [28]	2.3s	82.16%
FASTER R-CNN [58]	0.2s	83.29%
DSR-CNN	0.1173	82.82%
DGR-CNN	0.2642	94.1%

Compared with the Faster R-CNN model, the DSR-CNN model achieves similar performance with an accuracy of 83.29% for the Faster R-CNN compared with 82.82% for the DSR-CNN and a speed of 0.1173s compared to 0.2s, respectively. While the DGR-CNN achieves a remarkable accuracy of 94.1 while only slightly reducing the speed. 0.2642s compared with 0.2s.

Overall, the DGR-CNN model demonstrated a substantial improvement over the other models.

DGR-CNN vs. road anomaly classification models Table 5.3 shows the performance of the DGR-CNN model compared with other recent models developed specifically for road anomaly detection and classification. Babu et al. [7] used tiny-YOLOv4 to detect potholes and cracks. Mahenge et al. [43] proposed a deep-learning model for crack detection utilizing the GAN and CNN algorithms. Zhao et al. [83] developed a model (MED-YOLOv8s) to detect cracks and potholes using YOLOv8s and Mobilenetv3 as the backbone of the detection algorithm. Finally, Swain et al. [71] developed a model to detect potholes using CNN with VGG16 as a backbone and SRGAN.

The DGR-CNN model significantly outperforms all the models in all statistical measures.

Chapter Six: Conclusion and Future Work

Although speed is an important consideration, our application does not require real-time detection and classification, making YOLO’s ultra-fast nature less critical. Additionally, YOLO-based models tend to struggle when detecting small and closely packed objects, which is a significant challenge in road anomaly detection and classification.

Table 5.3: DGR-CNN vs. road anomaly classification models

Model	Accuracy	Precision	Recall	F1-Score	speed
DGR-CNN	94.85%	90.61%	91.35%	90.98%	0.2646
Babu et al. [7]	86.44%	85.30%	84.72%	85.01%	0.0975
Mahenge et al. [43]	84.02%	84.88%	83.30%	84.08%	0.3170
Zhao et al. [83]	80.42%	79.29%	80.71%	79.99%	0.0809
Swain et al. [71]	78.42%	77.29%	76.71%	77.00%	0.4443

5.1 Limitations

A large dataset of road images captured by drones has been used in this research. However, obtaining drone-captured images is challenging due to the difficulties arising from the Israeli occupation. Additionally, environmental factors such as adverse weather conditions, traffic, and obstacles may hinder the capturing of clear images. In addition, the DGR-CNN model requires high-performance computer vision due to the large size of the images. Finally, GPUs should be utilized instead of CPUs to handle the computational demands of our neural networks.

6.1 Conclusion

Automatic detection and recognition systems for road anomalies are crucial due to their impact on the safety and comfort of drivers and passengers. Drivers need to be informed about poor road conditions and the presence of anomalies along their routes to prevent accidents, minimize

the risk of vehicle issues, and choose the most suitable route to their destinations. This has resulted in a growing interest in research focused on automatically identifying and classifying road anomalies. Various methods have been developed to detect and categorize road anomalies automatically. The associated research falls into two main categories: accelerometer-based techniques and vision-based techniques. Deep learning and mathematical approaches have been employed in both types of techniques.

This study introduced drones as a new image acquisition tool to capture road anomalies and overcome the distortion and shaking of car-mounted cameras. A new dataset was created using images of Tulkarm City captured using a DJI Mavic Air 2 drone. A total of 15,326 images were extracted to create the dataset. The dataset consists of four classes: 4781 images of cracks, 4196 images of potholes, 3475 images of manholes, and 2874 images of speed bumps. All images have a resolution of 4k and are free from distortion, noise, or blur.

Additionally, two deep-learning models were developed to automatically detect and categorize various types of road anomalies, including potholes, cracks, speed bumps, and manholes. The first model is named "Dynamic Similar R-CNN" (DSR-CNN). This dynamic programming model utilizes graph segmentation, graph similarity, and dynamic programming techniques to decrease the number of proposed regions and increase the detection and classification speed. The DSR-CNN significantly reduces the number of candidate regions compared to the selective search algorithm used in R-CNN and fast R-CNN. On average, the developed technique suggests 8% of the segments the selective search algorithm proposed. This model achieved a speed of 0.117 seconds per frame while maintaining a MAP of 82.8%. This model performs similarly to other recently developed models while considerably reducing the detection and classification time.

Moreover, the second model, Dynamic Generative R-CNN (DGR-CNN), enhances the DSR-CNN and is developed to enhance the proposed regions before classification utilizing the generative feature of the DCGAN algorithm. The DGR-CNN significantly increases the overall detection and classification accuracy and speed compared to other leading models in this area.

The mean average precision of the proposed method was 94.1% with a speed of 0.26 seconds per frame, indicating a substantial improvement in the accuracy and speed of detection and classification compared to other state-of-the-art research in road anomaly detection and classification. This increase in accuracy is accompanied by maintaining the classification speed improvement from the first model.

6.2 Future work

Despite the outstanding performance of the DGR-CNN, there is still room for improvement, including;

- Improving the accuracy of neural networks by modifying the weights using meta-heuristics algorithms.
- Developing the system to be able to measure the dimensions of the abnormality and not just identify it.
- Connect the system to Google Maps to show anomalies on the map to alert drivers so they can take appropriate action at the right time.
- Compare the system performance on images taken by a car-mounted camera
- develop a model to automatically determine the best frame size

References

- [1] Kailash Ahirwar. *Generative adversarial networks projects: Build next-generation generative models using TensorFlow and Keras*. Packt Publishing Ltd, 2019.
- [2] Amina Alaoui-Belghiti, Sylvain Chevallier, Eric Monacelli, Guillaume Bao, and Eric Azabou. Semi-supervised optimal transport methods for detecting anomalies. In *underlineICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2997–3001. IEEE, 2020.
- [3] J Arunpriyan, VV Variyar, KP Soman, and S Adarsh. Real-time speed bump detection using image segmentation for autonomous vehicles. In *underlineInternational Conference on Intelligent Computing, Information and Control Systems*, pages 308–315. Springer, 2019.
- [4] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Alexander Mraz, Takehiro Kashiya, and Yoshihide Sekimoto. Deep learning-based road damage detection and classification for multiple countries. *underlineAutomation in Construction*, 132:103935, 2021.
- [5] Tariq N Ataiwe. Using image processing for automatic detection of pavement surface distress. *underlineAl-Salam Journal for Engineering and Technology*, 2(1):46–52, 2023.
- [6] Kanza Azhar, Fiza Murtaza, Muhammad Haroon Yousaf, and Hafiz Adnan Habib. Computer vision based detection and localization of potholes in asphalt pavement images. In *underline2016 IEEE Canadian conference on electrical and computer engineering (CCECE)*, pages 1–5. IEEE, 2016.
- [7] G Ramesh Babu, D Lavanya, B Chandra Sekhar, Ch Vignesh, and J Nikhil Veera. Pothole and crack detection using deep learning: Advancements in road surface anomaly recognition. *underlineMachine Intelligence Research*, 2022.

- [8] Nelson Babu, Deva Priya, and Srihari. Real-time detection of unmarked speed bump for indian roads. *European Journal of Molecular & Clinical Medicine*, 7(5):2020, 2021.
- [9] Ji-Won Baek and Kyungyong Chung. Explainable anomaly detection using vision transformer based svdd. *Computers, Materials & Continua*, 74(3), 2023.
- [10] H Bello-Salau, AJ Onumanyi, AT Salawudeen, MB Mu'azu, and AM Oyinbo. An examination of different vision based approaches for road anomaly detection. In *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComput-Conf)*, pages 1–6. IEEE, 2019.
- [11] Rozi Bibi, Yousaf Saeed, Asim Zeb, Taher M Ghazal, Taj Rahman, Raed A Said, Sagheer Abbas, Munir Ahmad, and Muhammad Adnan Khan. Edge ai-based automated detection and classification of road anomalies in vanet using deep learning. *Computational intelligence and neuroscience*, 2021(1):6262194, 2021.
- [12] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.
- [13] K Santle Camilus and VK Govindan. A review on graph based segmentation. *International Journal of Image, Graphics and Signal Processing*, 4(5):1, 2012.
- [14] M Ricardo Carlos. A machine learning approach for smartphone-based sensing of roads and driving style. *arXiv preprint arXiv:1908.10187*, 2019.
- [15] Manuel Ricardo Carlos, Mario Ezra Aragón, Luis C González, Hugo Jair Escalante, and Fernando Martínez. Evaluation of detection approaches for road anomalies based on accelerometer readings—addressing who’s who. *IEEE Transactions on Intelligent Transportation Systems*, 19(10):3334–3343, 2018.
- [16] Silvia Gil Casals. System and method for locating the position of a road object by unsupervised machine learning, January 11 2022. US Patent 11,221,230.

- [17] M Emre Celebi and Kemal Aydin. *Unsupervised learning algorithms*, volume 9. Springer, 2016.
- [18] Gongfa Chen, Shuai Teng, Mansheng Lin, Xiaomei Yang, and Xiaoli Sun. Crack detection based on generative adversarial networks and deep learning. *KSCE Journal of Civil Engineering*, 26(4):1803–1816, 2022.
- [19] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.
- [20] Pdraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.
- [21] Dalia Danilescu, Alexandru Lodin, Lacrimioara Grama, and Corneliu Rusu. Road anomalies detection using basic morphological algorithms. *Carpathian Journal of Electronic and Computer Engineering*, 8(2):15, 2015.
- [22] Deepak Kumar Dewangan and Satya Prakash Sahu. Deep learning-based speed bump detection model for intelligent vehicle system using raspberry pi. *IEEE sensors journal*, 21(3):3570–3578, 2020.
- [23] Jihad Dib, Konstantinos Sirlantzis, and Gareth Howells. A review on negative road anomaly detection methods. *IEEE Access*, 8:57298–57316, 2020.
- [24] Keval Doshi and Yasin Yilmaz. Road damage detection using deep ensemble learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5540–5544. IEEE, 2020.
- [25] Cao Vu Dung et al. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99:52–58, 2019.

- [26] XIE Fei, Jian Zhang, Chao Wang, Qiuzheng Liu, Ri Hong, and Liu Yanchen. Unstructured road region detection and road classification algorithm based on machine vision. Technical report, SAE Technical Paper, 2023.
- [27] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004.
- [28] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [29] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [30] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and building materials*, 157:322–330, 2017.
- [31] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.
- [32] Feng Guo, Yu Qian, Jian Liu, and Huayang Yu. Pavement crack detection based on transformer network. *Automation in Construction*, 145:104646, 2023.
- [33] Kemal Haciefendioğlu and Hasan Basri Başağa. Concrete road crack detection using deep learning-based faster r-cnn method. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 46(2):1621–1633, 2022.
- [34] Muhammad Uzair Ul Haq, Moez Ashfaq, Senthan Mathavan, Khurram Kamal, and Adeel Ahmed. Stereo-based 3d reconstruction of potholes by a hybrid, dense matching scheme. *IEEE Sensors Journal*, 19(10):3807–3817, 2019.

- [35] S Jana, S Thangam, Anem Kishore, Venkata Sai Kumar, and Saddapalli Vandana. Transfer learning based deep convolutional neural network model for pavement crack detection from images. *International Journal of Nonlinear Analysis and Applications*, 13(1):1209–1223, 2022.
- [36] Malhar Khan, Muhammad Amir Raza, Ghulam Abbas, Salwa Othmen, Amr Yousef, and Touqeer Ahmed Jumani. Pothole detection for autonomous vehicles using deep learning: A robust and efficient solution. *Frontiers in Built Environment*, 2023.
- [37] Christian Koch and Ioannis Brilakis. Pothole detection in asphalt pavement images. *Advanced Engineering Informatics*, 25(3):507–515, 2011.
- [38] Danai Koutra, Ankur Parikh, Aaditya Ramdas, and Jing Xiang. Algorithms for graph similarity and subgraph matching. In *Proc. Ecol. inference conf*, volume 17. Citeseer, 2011.
- [39] Taehee Lee, Chanjun Chun, and Seung-Ki Ryu. Detection of road-surface anomalies using a smartphone camera and accelerometer. *Sensors*, 21(2):561, 2021.
- [40] Dawei Luo, Jianbo Lu, and Gang Guo. Road anomaly detection through deep learning approaches. *IEEE Access*, 8:117390–117404, 2020.
- [41] Nachuan Ma, Jiahe Fan, Wenshuo Wang, Jin Wu, Yu Jiang, Lihua Xie, and Rui Fan. Computer vision for road imaging and pothole detection: A state-of-the-art review of systems and algorithms. *Xiv preprint arXiv:2204.13590*, 2022.
- [42] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiya, and Hiroshi Omata. Road damage detection using deep neural networks with images captured through a smartphone. *Xiv preprint arXiv:1801.09454*, 2018.
- [43] Shadrack Fred Mahenge, Stephen Wambura, and Licheng Jiao. Rcnngan: an enhanced deep learning approach towards detection of road cracks. In *Proceedings of the 2022 6th International Conference on Compute and Data Analysis*, pages 91–99, 2022.

- [44] Vishal Mandal, Lan Uong, and Yaw Adu-Gyamfi. Automated road crack detection using deep convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5212–5215. IEEE, 2018.
- [45] Alessio Martinelli, Monica Meocci, Marco Dolfi, Valentina Branzi, Simone Morosi, Fabrizio Argenti, Lorenzo Berzi, and Tommaso Consumi. Road surface anomaly assessment using low-cost accelerometers: A machine learning approach. *Sensors*, 22(10):3788, 2022.
- [46] Erick Axel Martinez-Ríos, Martin Rogelio Bustamante-Bello, and Luis Alejandro Arce-Sáenz. A review of road surface anomaly detection and classification systems based on vibration-based techniques. *Applied Sciences*, 12(19):9413, 2022.
- [47] Hafiz Suliman Munawar, Ahmed WA Hammad, Assed Haddad, Carlos Alberto Pereira Soares, and S Travis Waller. Image-based crack detection methods: A review. *Infrastructures*, 6(8):115, 2021.
- [48] Seongmin Ok. A graph similarity for deep learning. *Advances in Neural Information Processing Systems*, 33:1–12, 2020.
- [49] Anup Kumar Pandey, Rahat Iqbal, Tomasz Maniak, Charalampos Karyotis, Stephen Akuma, and Vasile Palade. Convolution neural networks for pothole detection of critical road infrastructure. *Computers and Electrical Engineering*, 99:107725, 2022.
- [50] Sandip Omprakash Patil, VV Sajith Variyar, and KP Soman. Speed bump segmentation an application of conditional generative adversarial network for self-driving vehicles. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pages 935–939. IEEE, 2020.
- [51] Yuning Qiu, Teruhisa Misu, and Carlos Busso. Unsupervised scalable multimodal driving anomaly detection. *IEEE Transactions on Intelligent Vehicles*, 2022.

- [52] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *underlinearXiv preprint arXiv:1511.06434*, 2015.
- [53] Antonella Ragnoli, Maria Rosaria De Blasiis, and Alessandro Di Benedetto. Pavement distress detection methods: A review. *underlineInfrastructures*, 3(4):58, 2018.
- [54] Aravinda S Rao, Tuan Nguyen, Marimuthu Palaniswami, and Tuan Ngo. Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. *underlineStructural Health Monitoring*, 20(4):2124–2142, 2021.
- [55] Roopak Rastogi, Uttam Kumar, Archit Kashyap, Shubham Jindal, and Saurabh Pahwa. A comparative evaluation of the deep learning algorithms for pothole detection. In *underline2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE, 2020.
- [56] Alfandino Rasyid, Mochammad Rifki Ulil Albaab, Muhammad Fajrul Falah, Yohanes Yohanie Fridelin Panduman, Alviansyah Arman Yusuf, Dwi Kurnia Basuki, Anang Tjahjono, Rizqi Putri Nourma Budiarti, Sritrusta Sukaridhoto, Firman Yudianto, et al. Pothole visual detection using machine learning method integrated with internet of thing video streaming platform. In *underline2019 International Electronics Symposium (IES)*, pages 672–675. IEEE, 2019.
- [57] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *underlineProceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *underlineAdvances in neural information processing systems*, 28, 2015.

- [59] Habeeb Salaudeen and Erbuğ Çelebi. Pothole detection using image enhancement gan and object detection network. *underlineElectronics*, 11(12):1882, 2022.
- [60] S Sattar, S Li, and M Chapman. Probabilistic-based crowdsourcing technique for road surface anomaly detection. *underlineThe International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:281–286, 2022.
- [61] Shahram Sattar, Songnian Li, and Michael Chapman. Developing a near real-time road surface anomaly detection approach for road surface monitoring. *underlineMeasurement*, 185:109990, 2021.
- [62] Fatjon Seraj, Berend Jan van der Zwaag, Arta Dilo, Tamara Luarasi, and Paul Havinga. Roads: A road pavement monitoring system for anomaly detection using smart phones. In *underlineBig data analytics in the social and ubiquitous context*, pages 128–146. Springer, 2015.
- [63] Anas Al Shaghouri, Rami Alkhatib, and Samir Berjaoui. Real-time pothole detection using deep learning. *underlinearXiv preprint arXiv:2107.06356*, 2021.
- [64] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *underlineIEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [65] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *underlineIEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.
- [66] Long Shiyang. Road vehicle anomaly detection method based on image classification and image comparison, 2021. This is a patent.
- [67] Nuno Silva, Vaibhav Shah, João Soares, and Helena Rodrigues. Road anomalies detection system evaluation. *underlineSensors*, 18(7):1984, 2018.

- [68] Nuno Silva, João Soares, Vaibhav Shah, Maribel Yasmina Santos, and Helena Rodrigues. Anomaly detection in roads with a data mining approach. *Procedia computer science*, 121:415–422, 2017.
- [69] Thitirat Siriborvornratanakul. An automatic road distress visual inspection system using an onboard in-car camera. *Advances in Multimedia*, 2018, 2018.
- [70] William Sprague and Ehsan Rezazadeh Azar. Integrating acceleration signal processing and image segmentation for condition assessment of asphalt roads. *Canadian Journal of Civil Engineering*, 49(6):1095–1107, 2022.
- [71] Satyabrata Swain and Asis Kumar Tripathy. Automatic detection of potholes using vgg-16 pre-trained network and convolutional neural network. *Heliyon*, 2024.
- [72] Yu-chin Tai, Cheng-wei Chan, and Jane Yung-jen Hsu. Automatic road anomaly detection using smart mobile device. In *conference on technologies and applications of artificial intelligence, Hsinchu, Taiwan*, 2010.
- [73] Haoteng Tang, Haozhe Jia, Weidong Cai, Heng Huang, Yong Xia, and Liang Zhan. Boundary-aware graph reasoning for semantic segmentation. *Xiv preprint arXiv:2108.03791*, 2021.
- [74] Beiwen Tian, Mingdao Liu, Huan-ang Gao, Pengfei Li, Hao Zhao, and Guyue Zhou. Unsupervised road anomaly detection with language anchors. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 7778–7785. IEEE, 2023.
- [75] Tomas Vojir, Tomáš Šipka, Rahaf Aljundi, Nikolay Chumerin, Daniel Olmeda Reino, and Jiri Matas. Road anomaly detection by partial image reconstruction with segmentation coupling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15651–15660, 2021.

- [76] Wenzhe Wang, Bin Wu, Sixiong Yang, and Zhixiang Wang. Road damage detection and classification with faster r-cnn. In *2018 IEEE international conference on big data (Big data)*, pages 5220–5223. IEEE, 2018.
- [77] Boqiang Xu and Chao Liu. Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples. *Measurement*, 196:111219, 2022.
- [78] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.
- [79] Takato Yasuno, Junichiro Fujii, Riku Ogata, and Masahiro Okano. Vae-iforest: Auto-encoding reconstruction and isolation-based anomalies detecting fallen objects on road surface. *arXiv preprint arXiv:2203.01193*, 2022.
- [80] Jongmin Yu, Jiaqi Jiang, Sebastiano Fichera, Paolo Paoletti, Lisa Layzell, Devansh Mehta, and Shan Luo. Road surface defect detection—from image-based to non-image-based: A survey. *arXiv preprint arXiv:2402.04297*, 2024.
- [81] Hyeong-Seok Yun, Tae-Hyeong Kim, and Tae-Hyoung Park. Speed-bump detection for autonomous vehicles by lidar and camera. *Journal of Electrical Engineering & Technology*, 14(5):2155–2162, 2019.
- [82] Shuangfei Zhai, Walter Talbott, Carlos Guestrin, and Joshua Susskind. Adversarial fisher vectors for unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [83] Minghu Zhao, Yaoheng Su, Jiuxin Wang, Xinru Liu, Kaihang Wang, Zishen Liu, Man Liu, and Zhou Guo. Med-yolov8s: a new real-time road crack, pothole, and patch detection model. *Journal of Real-Time Image Processing*, 21(2):26, 2024.

- [84] Johan Kok Zhi Kang, Suwei Yang, Suriya Venkatesan, Sien Yi Tan, Feng Cheng, and Bingsheng He. Dynamic graph segmentation for deep graph neural networks. In *underlineProceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4601–4611, 2022.

الكشف التلقائي عن عيوب الطريق باستخدام الرؤية الحاسوبية والتعلم العميق.

إعداد: رشا ضرار علي سفاريني

الملخص

تعتبر أنظمة الكشف والتعرف التلقائي على شذوذ الطريق ضرورية بسبب تأثيرها على راحة وسلامة السائقين والركاب. يجب أن يكون السائقون على دراية بحالة الطريق السيئة والشذوذ في مساراتهم لتجنب الحوادث وتقليل احتمالية تعطل السيارة أو تلفها واتخاذ الطريق الأكثر ملاءمة لوجهاتهم. أدى هذا إلى زيادة الاهتمام بالبحث في الكشف التلقائي عن شذوذ الطريق والتعرف عليه. تم تطوير تقنيات مختلفة للكشف التلقائي عن شذوذ الطريق وتصنيفه. يمكن تصنيف الدراسات ذات الصلة إلى تقنيات تعتمد على مقياس التسارع وتقنيات تعتمد على الرؤية. كلتا الطريقتين بها مشاكل. فبالنسبة للطرق المعتمدة على مقياس التسارع، فإن اهتزاز السيارة يسبب أخطاء في دقة عملية الكشف. أما بالنسبة للطرق المعتمدة على الصور، فإن مشاكلها تتمثل في عدم وجود قاعدة بيانات كافية ذات علاقة بشذوذ الطريق. بالإضافة إلى أن الفشل في معايرة الكاميرا المثبتة على السيارة يؤدي إلى تشويه الصورة وفقدان البيانات المهمة.

أنشأ هذا البحث قاعدة بيانات جديدة من صور مأخوذة لطرق مدينة طولكرم الملتقطة باستخدام طائرة بدون طيار DJI Mavic Air2. تم استخراج ما مجموعه 15326 صورة لإنشاء قاعدة البيانات. تتكون قاعدة البيانات من أربع فئات: 4781 صورة للشقوق، و4196 صورة للحفر، و3475 صورة لمنافذ الصرف الصحي، و2874 صورة لمطبات السرعة. جميع الصور بدقة 4K وخالية من التشويه أو الضوضاء أو عدم الوضوح. علاوة على ذلك، تم تطوير نموذجين جديدين للتعلم العميق تلقائياً للكشف عن وتصنيف جميع أنواع التشوهات المختلفة في الطرق، مثل الحفر والشقوق ومطبات السرعة ومنافذ الصرف الصحي. النموذج الأول يسمى Dynamic Similar R-CNN (DSR-CNN) يستخدم هذا النموذج تجزئة الرسم البياني وتشابه الرسم البياني وخوارزميات البرمجة الديناميكية لتقليل عدد المناطق المقترحة وزيادة سرعة الكشف والتصنيف دون التأثير على الدقة. أظهرت النتائج أن-DSR-CNN يقلل بشكل كبير من عدد المناطق المرشحة مقارنة بخوارزمية البحث الانتقائي المستخدمة في R-CNN و fast R-CNN. وقد اقترحت التقنية المطورة، في المتوسط، 8% من المقاطع المقترحة بواسطة خوارزمية البحث الانتقائي. وحقق نموذج-DSR-CNN سرعة متوسطة قدرها 0.1173 ثانية لكل إطار مع متوسط دقة متوسطة. 82.82% (MAP)

يستخدم النموذج الثاني، المسمى Dynamic Generative R-CNN (DGR-CNN)، تقسيم الرسم البياني، وتشابه الرسم البياني، والبرمجة الديناميكية لمرحلة اقتراح المنطقة لتحسين سرعة الكشف، وتقنية DCGAN لتحسين المناطق المقترحة وبالتالي تحسين دقة التعرف والتصنيف. حقق هذا النموذج متوسط دقة متوسطة للطريقة المقترحة بنسبة 94.85% بسرعة 0.262 ثانية لكل إطار، وهو ما يعتبر تحسناً كبيراً في دقة الكشف والتصنيف مقارنة بالدقة الناتجة عن أبحاث أخرى حديثة في مجال اكتشاف وتصنيف تشوهات الطرق. وتم تحقيق الزيادة في الدقة دون المساس بشكل كبير بسرعة. faster R-CNN