

**DETECTION AND QUANTIFICATION OF PAVEMENT
DEFECTS USING UNMANNED AERIAL VEHICLE
IMAGERY**

**İNSANSIZ HAVA ARACI KULLANILARAK ELDE
EDİLEN GÖRÜNTÜLERDEN YOL YÜZEYİ HASARI
TESPİTİ**

TUĞBA YILDIZLI

ASST. PROF. BURCU GÜLDÜR ERKAL

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements for the Award of

the Degree of Master of Science in Civil Engineering

November 2022

To My Family...

ABSTRACT

DETECTION AND QUANTIFICATION OF PAVEMENT DEFECTS USING UNMANNED AERIAL VEHICLE IMAGERY

Tuğba YILDIZLI

Master of Science, Department of Civil Engineering

Supervisor: Asst. Prof. Burcu GÜLDÜR ERKAL

November 2022, 94 pages

Roads have an important role in the development of the country's economy and social structure. Today, with the expansion of transportation networks, routine and proper maintenance of roads has gained importance. Cracks and other types of damage appear on the surfaces of roads due to many factors. The detection and measurement of these damages have an important place in the condition assessment of the roads. The surface defects not only affect the visual appearance of roads but also accelerates the aging of concrete infrastructure, which affects their normal use, resulting in shorter lifespans. It is also a potential threat to safe driving. Early detection of reduced capacity due to this deterioration is, therefore, a priority since timely and accurate detection of damages is of vital importance. Nowadays, various methods that are used for detecting surface damage on roads can be listed as follows: an operator inspecting the damage by using traditional guidelines, microscopic examination of the crack using special tools, taking images by unmanned aerial vehicles and automatically interpreting the surface damage by analyzing collected images. If we compare damage assessment using unmanned aerial vehicles with

other methods, we can say that it has many advantages such as less risk of accident, low cost, time savings, and fewer logistics requirements. Due to the expanding and increasing road networks, it is indeed a difficult task to conduct an extensive investigation using traditional methods. Therefore, it is possible to use unmanned aerial vehicles with high-resolution cameras, which have been used frequently recently, for surface damage detection of the investigated road. The detection and quantification of damages can be performed using Deep Learning methods from the collected images. With this method, images that accurately reflect the geometry of the damage can be obtained by UAVs. The proposed thesis study aims to automatically detect potholes and cracks on roads, via drones. The results of this study have the potential of contributing the national and international literature on damage detection on roads with the help of UAVs.

Keywords: UAV, Deep Learning, Pavement Defects, YOLO, Faster R-CNN, U-Net

ÖZET

İNSANSIZ HAVA ARACI KULLANILARAK ELDE EDİLEN GÖRÜNTÜLERDEN YOL YÜZEYİ HASARI TESPİTİ

Tuğba YILDIZLI

Yüksek Lisans, İnşaat Mühendisliği Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Burcu GÜLDÜR ERKAL

Kasım 2022, 94 sayfa

Yollar ülkenin ekonomisinin ve sosyal yapısının gelişmesinde önemli bir role sahiptir. Günümüzde ulaşım ağlarının genişlemesiyle birlikte yolların bakımının düzenli ve doğru olarak yapılması önem kazanmıştır. Birçok faktör nedeniyle yolların yüzeylerinde çatlaklar ve diğer hasar çeşitleri ortaya çıkmaktadır. Bu hasarların tespiti ve ölçümü yolların durum değerlendirmesinde önemli bir yere sahiptir. Bu sadece yolların görsel görünümünü etkilemekle kalmaz, aynı zamanda onların normal kullanımını etkileyen beton altyapısının yaşlanmasını hızlandırır ve daha kısa kullanım ömrüyle sonuçlanır. Ayrıca güvenli sürüş için potansiyel bir tehdittir. Bu bozulma nedeniyle azalan kapasitenin erken teşhisi öncelikli bir gerekliliktir. Çünkü bu hasarların zamanında ve doğru olarak tespit edilmesi hayati bir önem taşımaktadır. Günümüzde, yollarda oluşan yüzeysel hasarları tespit etmek amaçlı çeşitli yöntemler şöyle sıralanabilir: bir operatörün yapıyı incelemesi esnasında geleneksel yöntemler kullanılarak insanın hasarı teftiş etmesi, özel aletler kullanarak çatlağın mikroskobik incelenmesi, insansız hava araçları aracılığı ile görüntü alınması ve bu görüntülerin analiziyle yüzeysel hasarın otomatik olarak teşhis edilmesi ve ölçülmesi. İnsansız hava aracı kullanılarak yapılan hasar

tespitini diđer yöntemlerle karşılaştırırsak kaza riskinin daha az olması, düşük maliyet, zaman tasarrufu, daha az lojistik gereksinimi gibi birçok avantajı olduğunu söyleyebiliriz. Genişleyen ve artan yol ağları nedeniyle geleneksel yöntemler kullanarak geniş çaplı inceleme yapmak gerçekten zor bir çalışmadır. Bu yüzden son zamanlarda sıkça kullanılmaya başlanan yüksek çözünürlüklü kameralara sahip insansız hava araçlarını, incelenen yolun yüzeysel hasar tespitinde kullanmak mümkündür. Toplanan görüntülerden derin öğrenme yöntemleri kullanılarak hasar tespiti ve derecelendirilmesi yapılabilir. Bu yöntemle İHA'lar tarafından hasarın geometrisini doğru bir şekilde yansıtan görüntüler elde edilebilir. Önerilen tez çalışması, yollarda oluşan çukurların ve çatlakların, insansız hava araçları yoluyla otomatik olarak tespit edilmesini amaçlamaktadır. Çalışmanın tamamlanması ile ulusal ve uluslararası literatüre İHA yardımıyla yollarda hasar tespiti konusunda katkı sağlanmış olunacaktır.

Anahtar Kelimeler: Drone, Derin Öğrenme, Yol Hasarları, YOLO, Faster R-CNN, U-Net

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest appreciation to my esteemed thesis advisor, Asst. Prof. Burcu GÜLDÜR ERKAL, for her continuous support and guidance during my thesis period. What I learned from her was the most important factor in determining my direction. Your motivation for research and your successes in this field will be an example to me for many years to come...

I want to express my deepest appreciation to Prof. Dr. Berna UNUTMAZ, and Assoc. Prof. M. Abdullah SANDIKKAYA, Assoc. Prof. Alper ALDEMİR, Assoc. Prof. Zeynep TUNA DEĞER as jury members, for their constructive comments. It has been an honor to have them on my jury committee.

Words cannot express my gratitude to my family, for their immeasurable support and unconditional love throughout my education and life. I never forget the promise that I gave to my father, and I will keep it. My most precious people are my mother, my sister, and my niece. I know that you will always be there for me...

Special thanks to Ulaş Berkan ÖZTÜRK for his selfless support and for always encouraging me to keep fighting when life strikes back, even in the moments during this work. Also, I have to thank my companion, my cat Milo, for raising my mood and giving me energy every moment.

I could not have undertaken this journey without Şükriye AÇAR and Onur GÖKÇE. They have stood by me all the way for more than a decade and became my family in Ankara. I am looking forward to working with Şükriye at the same university and living together again in Antalya. May our paths never part. I am also grateful to Tuğçe TEKİN due to our heartfelt friendship. She showed me that kilometers mean nothing, and she made me feel that they are always with me.

I am thankful to my friends Zeynep MANDACIOĞLU, Tuğba İLERİ, Sercan ÇALIŞ and Selin YILMAZ for always being there when I needed courage and help since our undergraduate years.

I would like to thank my friend Ali Cemal DELİDUMAN for his considerable support. I am also grateful to Fatih DEMİRTAŞ for his assistance in many cases.

I would also like to thank my officemate Damla Nur ÇELİK for her tremendous support, friendship and help during the last three years. Special thanks are also due to Asst. Prof. Rüya KILIÇ DEMİRCAN and Asst. Prof. Murat PINARLIK from whom I have learned a lot in my academic and professional life. My appreciation is not only their being mentors in academic and professional life, but also for making me feel like a sister and a brother.

Thanks to my valuable colleagues and professors at Gazi University Faculty of Technology. I would like to express my gratitude to my colleagues Dr. Pınar Sezin ÖZTÜRK KARDOĞAN, Dr. Anıl ÖZDEMİR, Ahmet ERDAĞ, Coşkun ÇAKMAK, Özgün BOZDOĞAN and Mert ATAKAN. I enjoyed and learned a lot from every moment we shared.

This study was supported by The Scientific Research and Technical Council of Turkey (TUBİTAK), Project No: 120M765.

Lastly, I gratefully would like to thank TÜBİTAK for supporting me with the 2210-A Scholarship Program during my graduate studies.

Endless thanks...

Tuğba YILDIZLI

November 2022, Ankara

TABLE OF CONTENTS

ABSTRACT.....	i
ÖZET	iii
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS.....	vii
LIST OF FIGURES	ix
LIST OF TABLES.....	xi
SYMBOLS AND ABBREVIATIONS.....	xii
1. INTRODUCTION	1
1.1. Objectives	2
2. LITERATURE REVIEW	3
2.1. Image Processing Techniques.....	3
2.2. Machine Learning Techniques.....	6
2.3. Deep Learning Techniques	8
3. DATASETS	14
3.1 Available Datasets	15
3.2. Dataset Preparation	19
3.2.1. Labeling	19
3.2.2. Data Augmentation	23
3.3. Data Collection	26
3.3.1. Processing the Collected Images	31
4. METHODOLOGY	33
4.1. Deep Learning.....	33
4.2. Convolutional Neural Networks	34
4.3. Object Detection	36
4.3.1. Faster R-CNN	38
4.3.2. YOLOv5	39
4.3.3. YOLOv7	40

4.4. Object Segmentation	41
4.4.1. U-Net.....	42
4.5. Quantification of Pavement Defects	43
4.6. Environmental Setup	46
4.7. Metrics.....	47
4.7.1. Hyperparameters	48
4.7.2. Evaluation Metrics	50
5. RESULTS.....	53
5.1. Training Results	53
5.1.1. Faster R-CNN Defect Detection	54
5.1.2. Yolov5 Defect Detection.....	55
5.1.3. Yolov7 Defect Detection.....	59
5.1.4. U-Net Crack Segmentation	64
5.2. Test Results	67
5.2.1. Defect Detection.....	67
5.2.2. Segmentation of Defects	70
5.3. Performance Evaluation	74
5.4. Discussion/Comparison.....	75
6. CONCLUSION	77
7. REFERENCES.....	79
Appendix	86
Appendix 1 -Criteria for Road Damage Assessment	86
CURRICULUM VITAE	Hata! Yer işareti tanımlanmamış.

LIST OF FIGURES

Figure 1. The flowchart of image processing techniques for crack detection (Munawar et al., 2021)	6
Figure 2. Image classification and localization for single object, object detection and segmentation for multiple objects (Jaiswal et al., 2020).....	8
Figure 3. Architecture of LeNet-5, the first famous CNN architecture. (LeCun et al., 1998)	9
Figure 4. CrackNet model architecture (Zhang et al., 2017)	10
Figure 5. The common distress types for flexible and rigid pavements	14
Figure 6. Samples of the images in the dataset presented by Majidifard et al., (2020)..	18
Figure 7. Samples of the images in the dataset introduced by Passos et al., (2020).....	18
Figure 8. An example of a labeling using LabelImg	20
Figure 9. An example of YOLO annotation format.....	20
Figure 10. An example of Pascal VOC annotation format	21
Figure 11. The total number of labels for each category (Passos et al., 2020).....	22
Figure 12. The distribution of the labels obtained from two groups of datasets	23
Figure 13. Example of horizontal flip for data augmentation.....	24
Figure 14. An example of the changing the saturation	24
Figure 15. Examples of images containing trees and tree shadows for training data.....	25
Figure 16. Label distribution after data augmentation.....	26
Figure 17. Sample images taken with the smartphone camera in the preliminary study	26
Figure 18. Drone Flight Route (Google Maps).....	27
Figure 19. Drone used to collect images (DJI Mavic 2 Pro)	28
Figure 20. Example of road image collected using UAV	30
Figure 21. Example of Hacettepe University Beytepe Campus image using UAV	30
Figure 22. ArcGIS map of images taken from drone	31
Figure 23. A simple illustration of ANN network structure (drawn by author)	33
Figure 25. An illustration of the Convolution Neural Networks (CNN) architecture (drawn by author).....	36
Figure 26. The workflow of object detection (drawn by author).....	37
Figure 27. An illustration of object detection (Bochkovskiy vd., 2020)	38
Figure 28. Faster R-CNN structure (Ren et al., 2017).....	39

Figure 29. YOLO object detection system (Redmon et al., 2016).....	40
Figure 30. Comparison of YOLOv7 on the COCO dataset (Wang et al., 2022)	41
Figure 31. The architecture of U-Net (Ronneberger et al., 2015).....	42
Figure 32. IOU formulization.....	49
Figure 33. Confusion Matrix	50
Figure 34. Distribution of test data according to the number of labels	53
Figure 35. a) Classification loss b) Localization loss c) Regularization loss d) Total loss of the Faster R-CNN model with ResNet101.....	55
Figure 36. Pre-trained networks for YOLOv5	56
Figure 37. The visualization of mAP, precision and recall of the YOLOv5m6 model with the number of epochs during training	57
Figure 38. Loss values of the YOLOv5m6 model in terms of the number of epochs during training and validation.....	58
Figure 39. Confusion Matrix as a result of the YOLOv5m6 model for training	59
Figure 40. The visualization of mAP, precision and recall of the YOLOv7 model with the number of epochs during training.	60
Figure 41. Loss values of the YOLOv7 model with the number of epochs during training and validation	61
Figure 42. Confusion Matrix as a result of the YOLOv7 model for training	62
Figure 43. Comparison of true labels and predicted labels by YOLOv7 model.....	63
Figure 44. Loss values of the U-Net with ResNet34 crack segmentation model for training and validation	65
Figure 45. Accuracy of the U-Net with ResNet34 crack segmentation model for training and validation	65
Figure 46. Labels for drone test image.....	67
Figure 47. Prediction of drone test image with YOLOv7	68
Figure 48. Confusion Matrix as a result of the YOLOv7 model for testing	69
Figure 49. Examples of high degree a) pothole b) reflective crack c) alligator crack d) longitudinal crack.....	71
Figure 50. Distribution of the severity of defects according to class.....	72
Figure 51. Example of segmentation of high-level a) Alligator crack b) Longitudinal Crack	73
Figure 52. Road condition map of Beytepe Campus of Hacettepe University	75

LIST OF TABLES

Table 1. Existing datasets with crack image.....	16
Table 2. The images and their masks in the crack datasets	17
Table 3. Drone flight coordinates	28
Table 4. Drone Flight Records.....	29
Table 5. Parameters for the calculation of the GSD	32
Table 6. Pavement defects inspection criteria used in the state of California	44
Table 7. Pothole inspection criteria used in the Canadian province of British Columbia	45
Table 8. Pavement defects inspection limits used in this study	46
Table 9. Hyperparameters for each model.....	49
Table 10. The performances of the Faster R-CNN models	54
Table 11. The performances of the YOLOv5 models	56
Table 12. The loss values of YOLOv5m6 model	58
Table 13. The performances of YOLOv7 models	60
Table 14. The loss values of YOLOv7 model	61
Table 15. The performances of YOLOv7 models	64
Table 16. Comparison of U-Net Segmentation models.....	66
Table 17. YOLOv7 test results for each class	69
Table 18. The number of defects according to class and severity	71

SYMBOLS AND ABBREVIATIONS

Abbreviations

DIP	Digital Image Processing
AI	Artificial Intelligence
FHT	Fast Haar Transform
FFT	Fast Fourier Transform
MM	Mathematical Morphology
WT	Wavelet Transformation
MPS	Minimum Path Selection
MFCD	Multi-Scale Fusion Crack Detection
SVM	Support Vector Machine
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
NN	Neural Network
RF	Random Forest
MDA	Multiscale Dilated Attention
FFU	Feature Fusion Up Sampling
YOLO	You Only Look Once
SSD	Single Shot Detector
PID	Pavement Image Dataset
UAV	Unmanned Aerial Vehicle
DGCA	Directorate General of Civil Aviation
OSM	Open Street Map
GSD	Ground Sampling Distance

RPN	Region Proposal Network
ROI	Region of Interest
AP	Average Precision
CFD	Crack Forest Dataset
GPU	Graphics Processing Unit
API	Application Program Interface
IOU	Intersection Over Union
AUC	Area Under Curve
mAP	Mean Average Precision
YOLO	You Only Look Once
ResNet	Residual Networks

1. INTRODUCTION

Roads, which are one of the most critical structures all over the world, have a vital role in the development of the country's economy and human society. Therefore, countries value the construction of infrastructures and do not avoid providing the necessary budget for them. With the development of road construction extensively, the importance of road maintenance and rehabilitation has increased. The quality and sustainability of a road pavement directly determine the lifetime of that road. However, due to aging, environmental factors and road surface deterioration, different types of distress occur on the pavement surface. The most typical sorts of pavement distress that we commonly see on the road are cracks and potholes. They have a significant impact on the durability of the paved roads. These defects not only affect the quality of service and visual appearance of the structures but also give rise to steel corrosion by reaching the reinforcement, which shortens the lifetime of roads. At the same time, these defects can pose a potential threat to safe driving and may endanger human life. Therefore, it is a vital task to inspect these surface damages before performing maintenance and repair. It is crucial to identify the defects on time and accurately. Quick identification prevents possible dangers and is necessary to reach more accurate results. A manual visual inspection is the initial technique for identifying and categorizing pavement defects. Traditional visual inspection-based damage detection is expensive, inefficient, time consuming, labor intensive, and requires logistic planning. Nowadays, with the rapid expansion and increase of road networks, it is challenging to perform large-scale inspections. However, with the advent of unmanned aerial vehicles (UAVs), data collection has become easy. UAVs are used in many research areas because their ability to collect data quickly and without a lot of labor requirements. In addition to their efficiency and cost-effectiveness, UAVs can also perform large-scale inspections.

In response to this problem, many automated defect detection methods have been developed over the years to efficiently inspect defects on paved roads. This is an ongoing research topic and challenge for pavement condition researchers and computer vision community.

The difference of this thesis from the studies in the literature is combining the object detection and segmentation tasks.

This paper is organized as follows. The relevant studies based on image processing, machine learning, and deep learning are included in the literature review section. Chapter 3 explains available datasets for training and testing which includes images gathered by UAV at Hacettepe University Beytepe Campus. Chapter 4 introduces the methodology and applications' detailed explanation. Chapter 5 describes the results of analysis as well as comparison of different models. Finally, Chapter 6 contains a brief summary of the results of this thesis.

1.1. Objectives

Regular assessment of road damage is necessary to understand the performance of the roads and to carry out the necessary maintenance and repair work in a timely manner. Roads should be maintained regularly because they are constantly exposed to traffic loads and changes in environmental conditions. However, a lot of manpower is required to perform this assessment continuously. Therefore, the assessment of roads by drones can be done with less manpower.

This study's main aim is to automate the pavement condition assessment process by using drones. Therefore, to predict pavement distresses rapidly and accurately, a novel pavement assessment method based on deep learning frameworks is proposed. The goal of an automatic detection system is to perform continuous and cost-effective pavement evaluation.

Individual objectives can be listed as:

- Detection of damage with the object detection algorithm with the highest performance
- Determination the ground sampling distance for each image and creation of a new pavement assessment guide to quantify of the defects
- Determination of the total damaged area with the segmentation model with the highest performance
- Categorization of the severity levels of the defects
- Presentation of ArcGIS-based interactive road state map

2. LITERATURE REVIEW

Pavement surface condition assessment is the key factor in the maintenance and repair of roads. It is required to conduct a reliable and high-quality assessment of distresses in the pavement such cracks, potholes, and patches, and consequently, there are many studies in the literature that focus on pavement defect detection using state-of-the-art technologic developments over the years. With the technological developments, focal point of the methods used by researchers have changed. Many methods have been developed to automate the pavement damage detection, from image processing to machine learning methods, including deep learning methods which has been extensively used not only on pavement research but also in many other study fields from both the academy and the industry nowadays. At an early stage, automatic damage detection using digital image processing (DIP) have gained popularity with the advent of more powerful cameras that obtain higher-quality images, and numerous approaches based on image processing have been proposed by researchers. Later, machine learning methods drew rapid attention of many researchers who intended to automate the process. While image processing techniques can only focus on pixels of an image and understand unsubtle features, machine learning algorithms have the capability of learning deep features. With the growing trend in artificial intelligence (AI), deep-learning techniques became prominent, and they achieved tremendous success in various tasks (2017-2022).

2.1. Image Processing Techniques

Images taken from the roads may have not have the desired quality due to noise, blur, and uneven illumination. In early studies, various image processing techniques have been used to develop methods for performing high-quality pavement surface defect inspections. Several methods have been proposed to reduce the impact of noise on defect detection. The commonly utilized DIP techniques, such as threshold segmentation, edge detection, morphology, and wavelet transform can be used to separate the defects from the background and create a binary image classification.

Image segmentation is a procedure that focus on detecting boundaries such as edges, lines or, curves in the processed images. Threshold segmentation (Koch and Brilakis, 2011; Zhu et al., 2007) is a classical method of image segmentation. In threshold segmentation,

an image is divided into object and background by putting a pixel threshold. The method proposed by Koch and Brilakis, (2011) uses histogram-based thresholding to distinguish between portions of pothole images with defects and those without defects. They extracted a possible pothole form based on its geometric properties using morphological thinning and elliptic regression. Since the backdrop is lighter than the crack pixels, both dynamic and local thresholding methods have yielded in successful results . Oliveira and Correia, (2009) used the dynamic threshold to identify dark pixels in images. Tsai et al., (2010) used six methods for segmentation (Canny edge detection, a multiscale wavelet method, an iterative clipping method, statistical thresholding, dynamic optimization, and a crack seed identification method) to detect the structure of the crack that arise on the concrete pavement, and the effectiveness of each technique is evaluated in relation to each other. Consequently, the dynamic optimization-based method outperformed the other five methods.

Threshold segmentation can only result in good outcome if there are visible difference between pixels of the object and background. However, if the image contains complicated information, it may become difficult to perform segmentation using a threshold value on the pixel intensity of the image (Zhu et al., 2007). It is also a difficult task to determine the appropriate threshold value.

The edge detection methods were used in wide areas by many scholars. Commonly used edge detection filters can be listed as Canny, Sobel, Roberts, and Prewitt. In the field of automatic pavement damage detection, edge detection techniques were used specially to segment crack features. For instance, Zhao et al., (2010) applied Canny algorithm for pavement edge detection. Another implementation that uses edge detection algorithms is presented in Abdel-Qader et al., (2003), as they analyzed the cracks occurred on a bridge. They also compared the crack detection results for Canny, Sobel, fast Haar transform (FHT) and fast Fourier transform (FFT). However, the performed research showed that the edge detection methods may not give sufficient results in low-contrast images.

Mathematical morphology (MM) is also another productive image processing technique used for the detection of defects. Fundamental morphological operations can be listed as dilation, erosion, opening, and closing; however, it can only be used for binary operations. While the closing operator, which is the most popular in mathematical morphology, can be utilized to fill small gaps between crack pixels, the opening operation can be employed

o remove the undesired noises in the image (Fan et al., 2020). Tanaka, N. and Uematsu, (1998) proposed a method for detecting pavement cracks using these morphological approaches on the gray-scale images of the road surface. In the recent studies, for example Cubero-Fernandez et al., (2017), morphological operations were used during image preprocessing to obtain main features of the cracks , before the classification is performed. The wavelet transformation (WT) evolved into a powerful signal processing tool during the last two decades. Wavelets have been frequently employed to reduce the noises in images since the images basically consist of two-dimensional signals. 2D continuous wavelet transform, which is one of the image processing methods, was applied to detect crack and non-crack regions by Subirats et al. (2006). If there is a background with a strong texture, this proposed method could only be used for noise removal since these approaches are performed on individual pixels.

Analysis of collected raw video clips for pavement distress detection has been also studied. Huidrom et al., (2013) developed a robust method that uses a fast video segmentation algorithm called DFS and CDDMC to automatically detect and quantify potholes, cracks, and patches from video clips using various image processing techniques.

Another challenge to overcome in image processing is shadows that composed of complex noises and backgrounds. For that reason, Zou et al., (2012) proposed a method named CrackTree that uses image processing for crack detection. After removing the shadows while preserving crack pixels, they used a minimum spanning tree algorithm for crack identification.

Although all of the discussed image processing algorithms can detect cracks in images, they are not as effective as desired. In addition, not all image processing techniques can be adapted to various pavement images with different conditions. Image processing methods have been restricted to identifying distresses, but they have not yielded in efficient results for localizing and categorizing the detected distresses. The flowchart of image processing-based model for crack detection is shown in Figure 1.



Figure 1. The flowchart of image processing techniques for crack detection (Munawar et al., 2021)

2.2. Machine Learning Techniques

With the development of computer vision and artificial intelligence, automatic detection of cracks through a variety of machine learning methods has become possible for both pavements and other structures. The three main types of machine learning algorithms are Supervised learning (Liu, 2011), Unsupervised learning (Barlow, 1989) and Reinforcement learning (Szepesvári, 2010). During the last two decades, automatic pavement defect identification and analysis have been done using both supervised and unsupervised learning approaches. The main difference between unsupervised and supervised learning is whether the utilized data have a label or not. Supervised learning uses the labeled dataset, and then training occurs to classify data and predict outcomes by considering the previous experience. However, there is no labeled data in unsupervised learning, this technique finds hidden structures in unlabeled data by deciphering the similarity between data by itself.

Unsupervised algorithms can be effective when the existence and the location of the defect is unknown within the data. These algorithms are frequently employed to obtain useful information about data features and perform clustering. The main types of unsupervised learning algorithms include K-means (Yang et al., 2017), K-medoids (Park and Jun, 2009), and Neural Networks (Abiodun et al., 2018).

Amhaz et al. (2016) introduced an unsupervised learning algorithm for crack segmentation based on minimum path selection (MPS) by calculating the crack width using two-dimensional pavement image. However, the developed technique is incapable of dealing with a crack that has a complicated regional anatomy.

Akagic et al. (2018) suggested a new unsupervised learning method for detecting cracks in 2D images. After dividing the input images, the Otsu threshold and the maximum

histogram value were used on every sub-image. This method has performed well in low signal-to-ratio circumstances thanks to unsupervised learning.

Unsupervised learning-based MFCD, or the multi-scale fusion was proposed by Li et al. (2019). By calculating the highest average crack score across all scales, they attempted to increase the effectiveness of crack detection.

Oliveira and Correia, (2013) designed a system based on both image processing and machine learning. The system called CrackIT used threshold-based segmentation in the pre-processing step to distinguish cracked pixels from the background, and then used clustering techniques (unsupervised learning) to group patches of images that contain cracks.

In the case of supervised learning, the most common types of problems are the regression and classification. If there is data that needs to be categorized, classification algorithms can deal with the problem. The most popular classification algorithms are: Random Forest (Biau and Scornet, 2016), Nearest Neighbors (Indyk and Motwani, 1998), Support Vector Machines (SVM) (Hearst et al., 1998), Decision Trees (Kotsiantis, 2013), Logistic and Linear regression (Wright, 1995) and Neural Networks (Abiodun et al., 2018).

One of the most prevalent supervised learning techniques with binary classification is the Support Vector Machine (SVM) (Christopher J.C. Burges, 1998). Gavilán et al., (2011) presents a road distress detection model that uses a linear SVM-based classifier for different pavement types in Spain. In (Marques, 2012), three different pre-processing arrangements are applied to increase the smoothness of the image texture. A crack classification using SVM was then used to distinguish cracks and non-cracks.

Another supervised machine learning technique that is popular in classification and regression is Random Forest. Shi et al. (2016) introduced CrackForest, which is a road crack detection framework that was developed by using random structured forests, to deal with complex topological structures. They compared the proposed approach with other methods and concluded that it was effective in discriminating cracks from noises.

Cubero-Fernandez et al., (2017) applied decision tree heuristic algorithm for classification task once several image processing techniques are utilized to enhance pavement images.

2.3. Deep Learning Techniques

One of the subsets of machine learning is Deep Learning, which can simulate the computational capabilities of the human brain and develop decision-making patterns. Deep learning algorithms are based on neural networks, which is a set of algorithms that works similar to human brain. In recent years, upon further development, deep learning algorithms have been extensively used for research in various areas as they do not require much human interference.

The basic goals of the field of computer vision are classification, localization, detection, and instance segmentation. The difference between these tasks is illustrated in Figure 2 based on the output image. Road defect detection and classification has been the subject of numerous experiments with Deep Learning, especially based on Deep Convolutional Neural Networks.

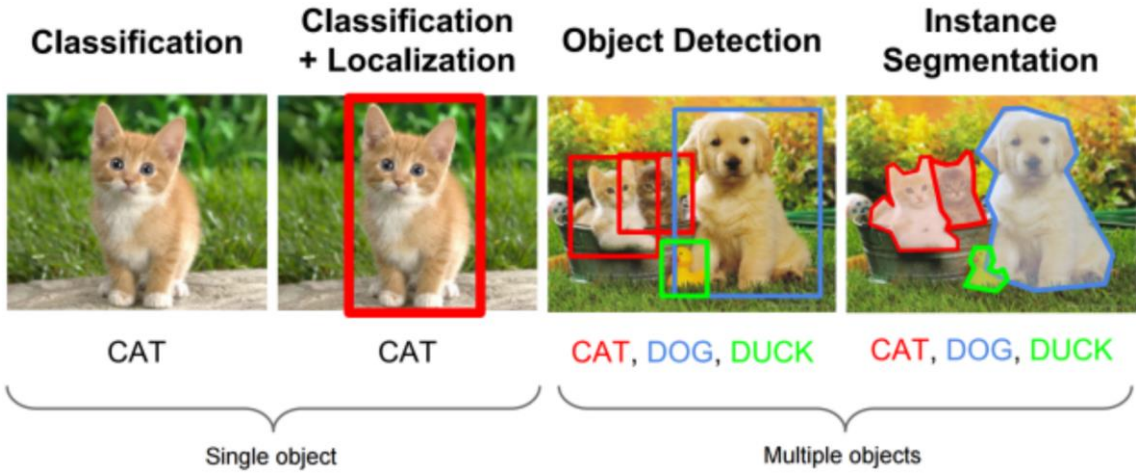


Figure 2. Image classification and localization for single object, object detection and segmentation for multiple objects (Jaiswal et al., 2020)

Convolutional Neural Networks, a method of deep learning (CNN), have great advanced especially in performing multi-class classification compared to the traditional methods. CNN algorithms are generally preferred over regular Artificial Neural Networks (ANN) since they require less computation. LeCun et al., (1998) designed the first neural convolutional neural network called LeNet for handwritten digit recognition (Figure 3).

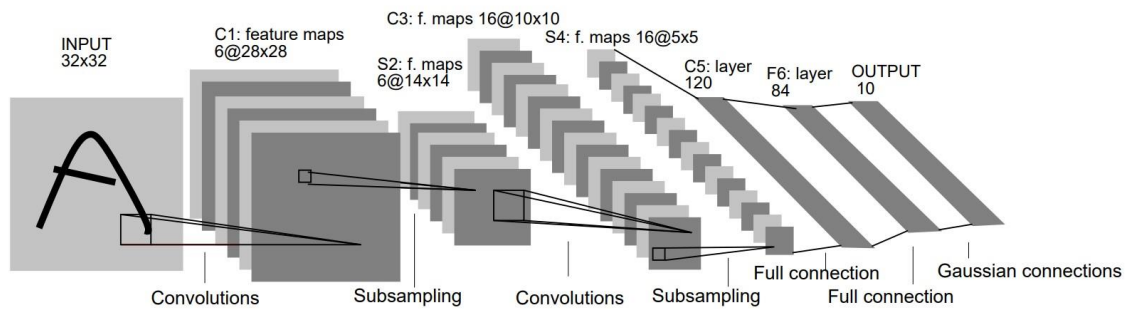


Figure 3. Architecture of LeNet-5, the first famous CNN architecture. (LeCun et al., 1998)

Binary Classification

Binary classification can be performed with deep learning. It is the simplest method to categorize the input into two classes. When a model performs binary classification, it examines the input data and determines which of two possible classifications to assign it.

As an example of binary classification, the road images acquired by smartphone were first divided into small patches. Later, the probability of these patches including cracks was calculated using deep CNN (Zhang et al., 2016). This study was pioneer in applying deep learning techniques to identify road cracks.

Gopalakrishnan et al. (2017) trained a deep learning model using an open-source database called ImageNet and transferred its learning ability to perform automatic pavement crack detection. The developed pavement crack detection model contains five convolutional blocks of the VGG16 architecture. Then, several machine learning classifiers such as Support Vector Machine (SVM), Random Forest (RF), and Neural Network (NN) were utilized. The performances of these classifiers are then compared. It is concluded that the single-layer neural network classifier based on pre-trained VGG-16 deep CNN model outperformed the other four classifiers.

Cha et al. (2018) improved a deep CNN to discriminate between crack pixels and non-crack pixels in concrete structures without utilizing image processing techniques for feature extraction. The results indicated that even if the images were taken under various conditions such as insufficient lighting or shadow, the performance of the method was more robust compared to the traditional edge detection techniques (i.e., Canny and Sobel).

CrackNet software based on deep learning was established by Zhang et al. (2017). The outputs of the proposed CNN, a pixel-level classification method, are class scores for all individual pixels. The difference between regular CNN and CrackNet is that CrackNet does not use any pooling layers. The architecture of the suggested CNN model, CrackNet, is illustrated in Figure 4. Later, to increase the computation speed of the model, the CrackNet software is improved into CrackNet II software (Zhang et al., 2018). CrackNet II uses a deeper architecture compared to CrackNet. CrackNet II not only provides faster detection, but it is also effective for detecting of small cracks.

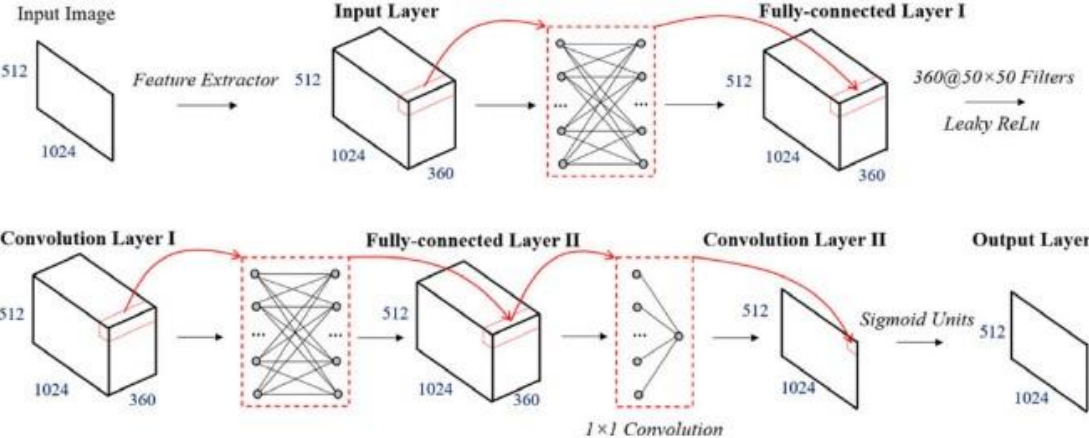


Figure 4. CrackNet model architecture (Zhang et al., 2017)

Yang et al., (2020) presented a novel network utilizes feature pyramid and hierarchical boosting to detecting cracks automatically. This novel approach provides an integration between context information and low-level features. CRACK500 dataset which contain five hundred images from the campus of Temple University was also published by the same group of researchers.

Multi-Classification

Several earlier studies focus only on the existence of defects. If it is desired to separate the cracks into groups or to differentiate various damage types, binary classification is not sufficient. For this reason, several researchers have focused on defect categorization

and defect severity level determination for years. As a result, many techniques based on CNN for multi-classification task have been presented.

Fan et al. (2018) offered a structured prediction method using Deep Learning CNN to automate the multi-label classification problem in crack detection. They have not used any preprocessing techniques but instead used raw images. To deal with the problem of unstable data, the developed model was tested on two different data sets. It was found that the proposed approach has strong generalizability even for images with particularly complex backgrounds and crack patterns.

Song et al., (2019) proposed deep multi-scale convolutional features to detect high-level cracks by using the multiscale dilated attention (MDA). Afterwards, the crack resolutions were recovered using the Feature Fusion Up Sampling (FFU) module. Cracks were separated into groups for block, alligator, transverse, and longitudinal categories during the second phase. In addition, the severity of them was evaluated as a function of branch spacing and crack width.

Object Detection

In recent years, object detection applications have also been used for the road damage detection since they can perform both classification and localization tasks. The most commonly used CNN-based algorithms for object detection are Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2017), Single Shot MultiBox Detector (SSD) (Liu et al., 2016), and You Only Look Once (YOLO) (Redmon et al., 2016). All of these techniques aim to define a bounding box around the target object.

The main advantage of YOLO is that it can locate and classify objects with a single CNN. Hereby, it works faster than other listed algorithms.

Majidifard et al. (2020) proposed a comprehensive new dataset named pavement image dataset (PID). Images were extracted as wide-view and top-view from Google API Street view, and they were manually labeled into nine categories. YOLO and Faster R-CNN were utilized to automatically identify and categorize road damages. The results were then compared with each other.

Another object detection-based multi-classification method that uses CNN architecture was developed by Maeda et al. (2018). They prepared an extensive dataset that includes 9053 road images with eight damage types captured by a smartphone. Then, SSD architecture was used to perform automatic damage detection and classification tasks. The proposed SSD framework employed Inception V2 and MobileNet as the backbone feature extractor modules.

In (Du et al., 2021), a large-scale dataset collected by a high-resolution camera installed on a vehicle was prepared. The dataset includes 45,788 distress images with various illumination for seven damage classes. Subsequently, a technique for identifying and categorizing roadway deterioration based on the Yolov3 network was proposed to automate feature extraction and increase detection speed. In addition, this study compared the Yolo Network, Faster R-CNN, and SSD algorithms regarding speed and precision. The results showed that Yolov3 is relatively faster than other algorithms.

The applications of computer vision-based techniques have not only been used for pavement monitoring but also been applied for structural condition assessment and inspection (i.e., concrete cracks, concrete spalling, fatigue cracks in the steel, steel corrosion) (Spencer et al., 2019). For example, Cha et al., (2018) proposed a Faster R-CNN-based structural defect detection model to categorize various damages such as corrosion and delamination of steel in bridges.

Image Segmentation

Image segmentation is another popular technique used for differentiating objects in an image. For image segmentation, each pixel is labeled with a corresponding category. In recent years, this method has been used in pavement damage inspection automation.

Li et al., (2021) presented a semantic segmentation model, that uses advanced pixel-level recognition, to automate pavement distress classification. They created an extensive pavement distress image dataset containing 10,097 images and labeled these images under six categories including potholes, patches, block cracks, longitudinal cracks, transverse cracks, and alligator cracks. The U-Net and Resnet architecture was then trained to detect the areas of damage in the images. When the binary classification was

considered, the results obtained by this method showed good pixel accuracy. However, this method is not efficient in predicting the shapes and locations of the six distress.

A technique for segmenting and detecting pavement cracks was suggested by (Liu et al., 2020), which utilizes a combination of the modified YOLOv3 and U-Net. First, an object detection algorithm was used to classify four types of distresses, and the crack regions detected in the first step were then used as an input for the segmentation step. The goal of these steps is to increase the accuracy of the model. After comparing the proposed two-step method with other detection and segmentation techniques, it was found to be more successful.

3. DATASETS

Pavement surfaces deteriorate over the years due to many reasons such as heavy traffic, natural disasters, construction faults and environmental conditions. The common types of the distresses can be divided into three main groups: cracking, disintegration, and surface treatment distresses. These types of damage differ depending on the materials of the pavement. Figure 5 shows the types of pavement distress for both flexible and rigid surfaces as a summary.

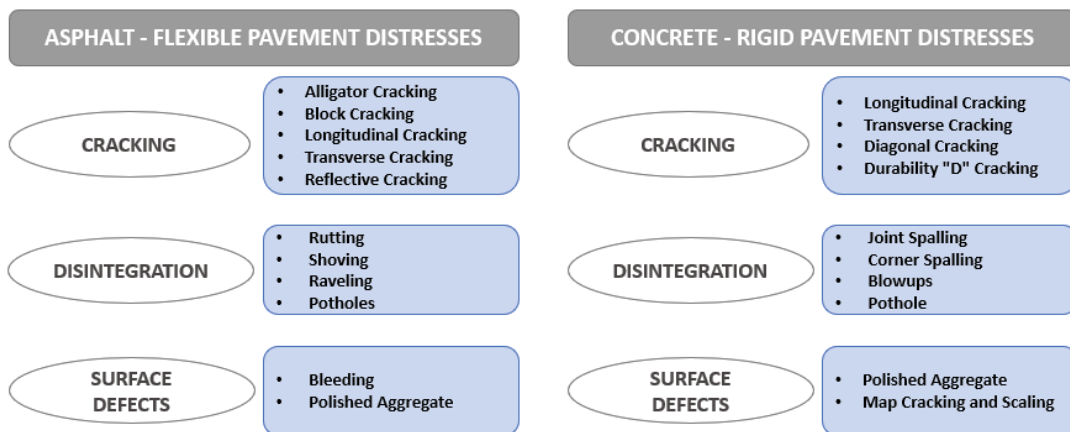


Figure 5. The common distress types for flexible and rigid pavements

Since these deteriorations occur very slow at the beginning, they only require regular maintenance to maintain the level of service of the road. If not maintained and improved in a timely manner, deterioration will accumulate and lead to very expensive maintenance and improvement work. Therefore, pavement maintenance is important to extend the useful life of a road. Although there are a variety of defects in different pavements, cracks are the most common damage that occurs on both types of roads. Cracking of the road surface allows moisture to penetrate the subgrade, which can lead to premature deterioration of the road structure. To maximize the life expectancy of a road, it is desirable to minimize the damaging effects of cracking. This can be achieved by early detection and repair of cracks before extensive deterioration occurs. If cracks are not repaired in a timely manner, maintenance costs can increase as the crack progresses.

Pothole is another common type of damages that occurs on the road surface. Potholes are also caused by the progression of cracks that are not maintained in a timely manner. Potholes deface the road structure and shorten the life of the road. Failure to maintain a crack in a timely manner will result in more wear and potholes. Potholes are also important for safe driving. They lead to impaired driving comfort and accidents, especially since visibility is very poor in the dark. They pose a greater risk, especially to vehicles such as bicycles and motorcycles, and can even cause fatal accidents. Therefore, proper and timely road maintenance has become an important service that extends the life of the road and also reduces risks.

There are numerous open-access and private datasets in the literature that contain these types of damage. Especially with the widespread use of digital image processing methods, studies using these datasets have increased. Since it is easy to distinguish the crack structure from any background, effective studies have been conducted to detect cracks on asphalt and concrete surfaces. As a result of these studies, open access datasets are presented. Since the advent of machine learning and deep learning methods, the need for more images has arisen. To meet this need, large-scale datasets of road defect images from different angles and multiple damage classes have been developed. To choose the best dataset for this research, a detailed literature review was conducted. Many open-access datasets related to pavement conditions have been found and prepared for efficient use in this study.

3.1 Available Datasets

Pavement distress data are traditionally collected by walking on the road and evaluating the required criteria. However, this procedure takes a long time, and the outcomes are inaccurate. Automated data collection systems are evolving as technology develops. A truck or robot with cameras are among the most commonly used methods for data collection in pavement inspection. Since the installation of these trucks is very costly, they are not preferred by researchers. Nowadays, drones are increasingly used for pavement data collection (Zhu et al., 2022).

Some datasets that are collected manually have been found in the literature. However, most of them have only 2 types (with crack and without crack) of labels. These are some examples of them; CrackForest Dataset (Shi et al., 2016), Crack500 (Yang et al., 2020),

CrackTree (Zou et al., 2012), GAPs (Eisenbach et al., 2017) and DeepCrack (Liu et al., 2019) . The images in these datasets were taken with smartphone cameras and contain close-up photos of the crack structure. The number of images and pixel sizes in these datasets are listed in Table 1. Most of them contain images with a size of 448x448, while Crak500 and GAPs contain crack images with a high pixel size of 3264x2448 and 1920x1080, respectively.

Table 1. Existing datasets with crack image


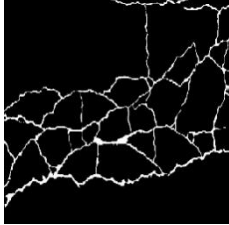



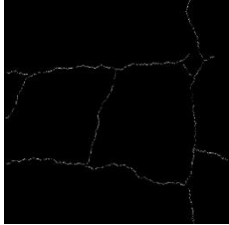
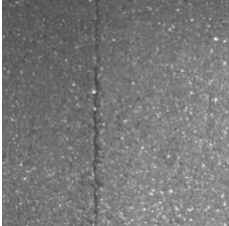
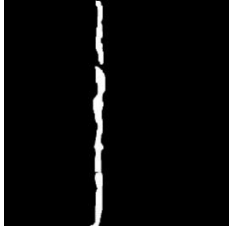

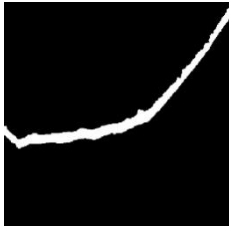
Dataset	Number of Images	Pixel Size
Crack Forest Dataset (CFD)	118	448x448
Crack500	500	3264x2448
CrackTree	206	448x448
GAPs v1	1969	1920x1080
DeepCrack	537	448x448

Datasets containing only the crack structure are generally used in the literature for image processing methods. Image processing methods such as thresholding, edge detection, and segmentation have been applied to the images in these datasets. Researchers have generally utilized these datasets to extract the crack region based on the assumption that the crack pixel is darker than its background. Recently, with the increasing interest in deep learning, such datasets have been used with deep learning algorithms. In particular, segmentation based on deep learning networks have been applied to crack images. Since the crack structure has specific pixels, studies were conducted on different surfaces. These studies were not limited to asphalt and concrete, but were also applied to marble (Vrochidou et al., 2022), rock (Chen et al., 2020), and plastic surface (Kien et al., 2019).

The datasets listed in Table 1 contain not only the raw images but also their masks. Each crack image has a binary image as a mask, labeled at the pixel level. Some of the raw images and masks of the datasets examined in this study are shown in Table 2.

In this study, segmentation was performed using deep learning methods to obtain a pixel output of road damage. Images and their masks from the five datasets listed in Table 2 were used to train the segmentation model.

Table 2. The images and their masks in the crack datasets

Dataset	Raw Image	Mask
Crack Forest Dataset (CFD)		
Crack500		
CrackTree		
GAP v1		
DeepCrack		

The first goal of this work is to detect different types of pavement damage. Therefore, datasets containing multiple types of pavement distresses were searched and two datasets were found. One of them is created by Majidifard et al., (2020). There are 7237 images of pavement damage in this dataset, including nine different types of damage. The images in this dataset were taken from two different viewpoints. Sample images from the dataset are displayed for each category in Figure 6.

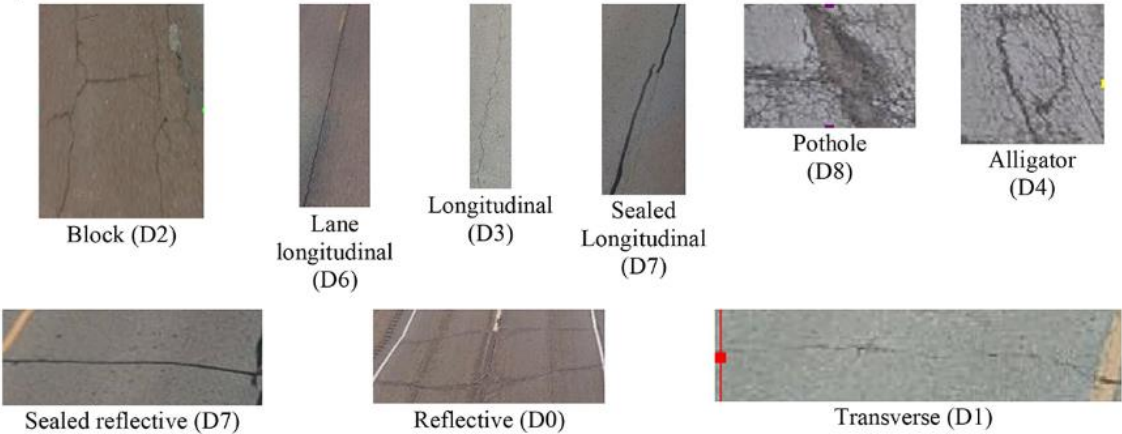


Figure 6. Samples of the images in the dataset presented by Majidifard et al., (2020)

Another dataset was created by Passos et al., (2020) with support from the Brazilian National Department of Transportation Infrastructure (NDTI). These images were extracted from videos recorded by a Highway Diagnostic Vehicle (HDV) with a high-resolution camera. The total of 2235 images extracted from the video in this dataset have a resolution of 1280x729. Although generally consists of pothole images, it also contains some types of cracks such as alligator and longitudinal cracks, as shown in Figure 7.



Figure 7. Samples of the images in the dataset introduced by Passos et al., (2020)

Flexible pavements such as asphalt are common on existing highways in Turkey. The most common pavement damages in Turkey are potholes, longitudinal cracking, alligator cracking, and reflective cracking. Therefore, this study aims to investigate a multi-classification task focusing on these four types of defects in object detection. Thus, the aforementioned datasets were prepared for the object detection model.

3.2. Dataset Preparation

The size of the dataset is the key factor for the effective execution of a deep learning model. Since the above two datasets generally contain the desired defect types, the use of these datasets would be more appropriate. The images containing these crack types were extracted and classified into four categories. In order to train the object detection model, the collected images should be used as input to the damage detection models. This study employed a number of deep learning-based object detection algorithms. Therefore, depending on the algorithm used for different formats, the images are labeled according to their classes. This labeling process is explained in detail.

No labeling process is required to build the segmentation model. This is because the publicly available crack datasets also contain their images and masks.

3.2.1. Labeling

Object detection task based on deep learning algorithms requires that the object contained in the image can be assigned to its class. The image may contain multiple objects belonging to different types, so each object must be labeled separately. To train the object detection model, bounding boxes should be created around the location of each object. The labeling process can be done manually for each image using various freely available software tools. In this study, a software called LabelImg was utilized for labeling. This graphical tool allows labeling images and to prepare them for deep learning models. Thanks to LabelImg, the coordinates of the object locations are recorded in different formats for each algorithm. An illustration of the LabelImg software's labeling procedure can be seen in Figure 8.



Figure 8. An example of a labeling using LabelImg

YOLO Format

In order to perform object detection with the YOLO architecture, the images must conform to the YOLO format. In this format, the upper right corner of the images is considered a (0,0) point, and the lower left corner is considered a (1,1) point, so labeling is based on pixel coordinates between 0 and 1. Using the LabelImg software, the information about the object is stored in a file TXT for YOLO format in the order shown below.

[class] [x-coordinate of the center of the object] [y-coordinate of the center of the object] [width of the object] [height of the object]

```
3 0.36279296875 0.5171875 0.0595703125 0.065625
2 0.439453125 0.5421875 0.083984375 0.14375000000000002
2 0.43603515625 0.3140625 0.0947265625 0.25625000000000003
2 0.77294921875 0.74765625 0.1142578125 0.1859375
```

Figure 9. An example of YOLO annotation format

In YOLO format, the damage class is symbolized by a number, as shown in Figure 9. The damage classes were used in the labeling in this study as shown below:

0	—————>	Longitudinal Crack
1	—————>	Reflective Crack
2	—————>	Alligator Crack
3	—————>	Pothole

Pascal VOC Format

To use the Faster R-CNN architecture for training the object detection model, the input data must be in Pascal VOC format. In this format, the annotations are stored in an XML file. This XML file contains directory information for each image, the coordinates of the corresponding bounding-box, and the type of defect, as shown in Figure 10.

```
<annotation>
  <filename>potholedataset_(196).jpg</filename>
  <folder>C:/Tensorflow3/workspace/training_demo
    (196).xml</folder>
  - <size>
    <width>1024</width>
    <height>630</height>
    <depth>3</depth>
  </size>
  - <object>
    <name>Alligator</name>
    - <bndbox>
      <xmin>778</xmin>
      <ymin>424</ymin>
      <xmax>1022</xmax>
      <ymax>629</ymax>
    </bndbox>
  </object>
</annotation>
```

Figure 10. An example of Pascal VOC annotation format

Asphalt pavements are widely used on existing highways in Turkey. Deformations such as cracks and potholes usually occur in these flexible pavements. Therefore, it was considered appropriate to use the dataset prepared by Passos et al., (2020) in this study. The labeling process was carried out for four classes: Pothole, longitudinal crack, alligator crack, and reflective crack. First, images from the dataset presented by Passos et al., (2020) were manually labeled, resulting in 1524 images collected for this project. The total number of labels created for each class in the dataset is shown in Figure 11.

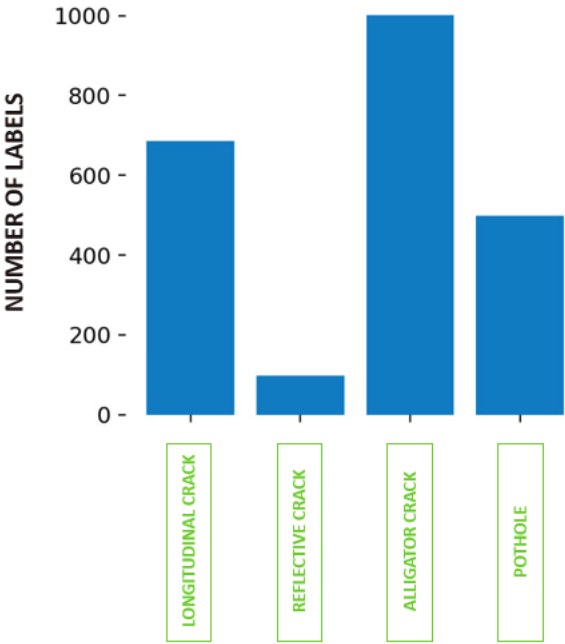


Figure 11. The total number of labels for each category (Passos et al., 2020)

The balance of tags belonging to each class is one of the most crucial elements influencing an object detection model's accuracy. However, after the labeling process for this dataset, it was found that the numbers of tags were not evenly distributed. In particular, it was found that the number of reflective cracks was very low. This is because the dataset mostly consists of pothole, alligator cracking, and longitudinal crack types. To address this issue, another dataset (Majidifard et al., 2020) was also labeled for four categories and added to the training data. In this way, a total of more than 200 labels for reflective cracks were added. The final version of the class distribution is shown in Figure 12.

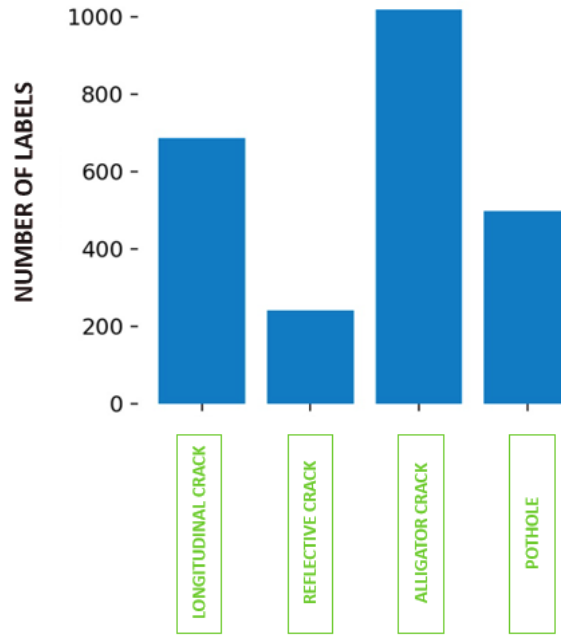


Figure 12. The distribution of the labels obtained from two groups of datasets

The collected images are used as input to the damage detection models for training the models. In this study, it was recommended to use two deep learning-based object detection models.: YOLOv5 and Faster R-CNN. Since each model has a different annotation format, the images are labeled according to a compatible format.

3.2.2. Data Augmentation

Prior to building deep learning-based object detection models, several image augmentation techniques were applied to the images to be included in the training. The performance of the model is adversely affected by an irregular distribution of label numbers. It was estimated that this situation may decrease the overall accuracy of the model. Therefore, to balance the distribution, some image augmentations techniques were applied to the images, especially to reflective cracks.

In order to create a variety of images in the model, both geometric changes and differences in pixels can be applied to the images. In this study, the geometry of the image is important to determine the crack type. This is because the direction in which the crack extends determines its class. Therefore, only horizontal flip could be applied as a geometric modification. An example of this application can be found in Figure 12.



Figure 13. Example of horizontal flip for data augmentation

Changing the brightness of the image is another technique of data augmentation. This technique allows the model to recognize the image under different illuminations. In this study, the brightness was changed at a rate of 30%. This created new images that were 30% brighter and darker.

Another method is to change the saturation level. Saturation is the depth or intensity of color in an image. In this work, the saturation was changed by 50%. A sample image of this change is shown in Figure 14.



Figure 14. An example of the changing the saturation

In the last step of data augmentation, some images were converted to black and white format and added to the dataset. In this way, more data was obtained.

During the dataset preparation phase, some preliminary studies were conducted to build models for automatic road damage detection. In these studies, it was found that the model made false detections in images with trees or tree shadows. It was found that it perceived areas with tree branches and shadows as cracks. For this reason, some small images with objects such as trees and tree shadows were added to the dataset without labels. The training data was enlarged in this way. This allows the model to produce more accurate results. Examples of these images can be seen in Figure 15.



Figure 15. Examples of images containing trees and tree shadows for training data

After the data augmentation for training the model, 342 more images were added to the dataset. Thus, the uneven distribution of labels was eliminated. The dataset then contained a total of 1930 images. The final distribution of the dataset according to the numbers of labels for each category is shown in Figure 16.

As a result of these applications, the total number of images to be used for building an object detection model was increased to 1930 images. Then, the dataset was split into a training and a validation dataset in a ratio of 80/20. The validation data is used to evaluate the performance of the model after training, while the training data is used as input to the model. Therefore, the number of training and validation images was calculated to be 1544 and 386, respectively.

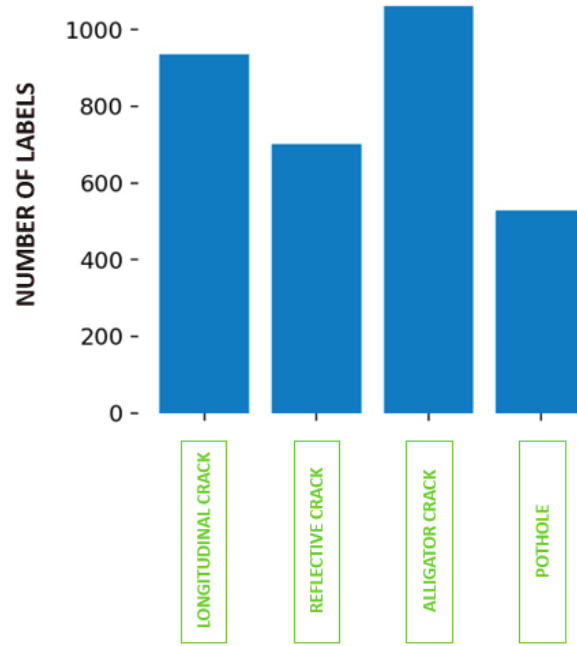


Figure 16. Label distribution after data augmentation

3.3. Data Collection

The Beytepe Campus of Hacettepe University was chosen as the place to work on this project. The campus was built on a 5,877,628 m² plot of land along the 20th kilometer of the Ankara-Eskişehir highway. Before conducting a drone flight, the roads on the Beytepe Campus were examined for feasibility. It was found that there are many damages that need to be maintained and repaired. During the preliminary surveys, pictures of the damaged areas were taken with a smartphone camera. Figure 17 shows examples of these images.



Figure 17. Sample images taken with the smartphone camera in the preliminary study

After this preliminary study, it was decided on which routes the flight with the drone would be carried out. The route on which the study was to be conducted was selected as the main street of the dormitories on the Beytepe Campus. The estimated flight distance was planned about 1400 m. The map of the flight route was taken from Google Maps and is illustrated in Figure 18.

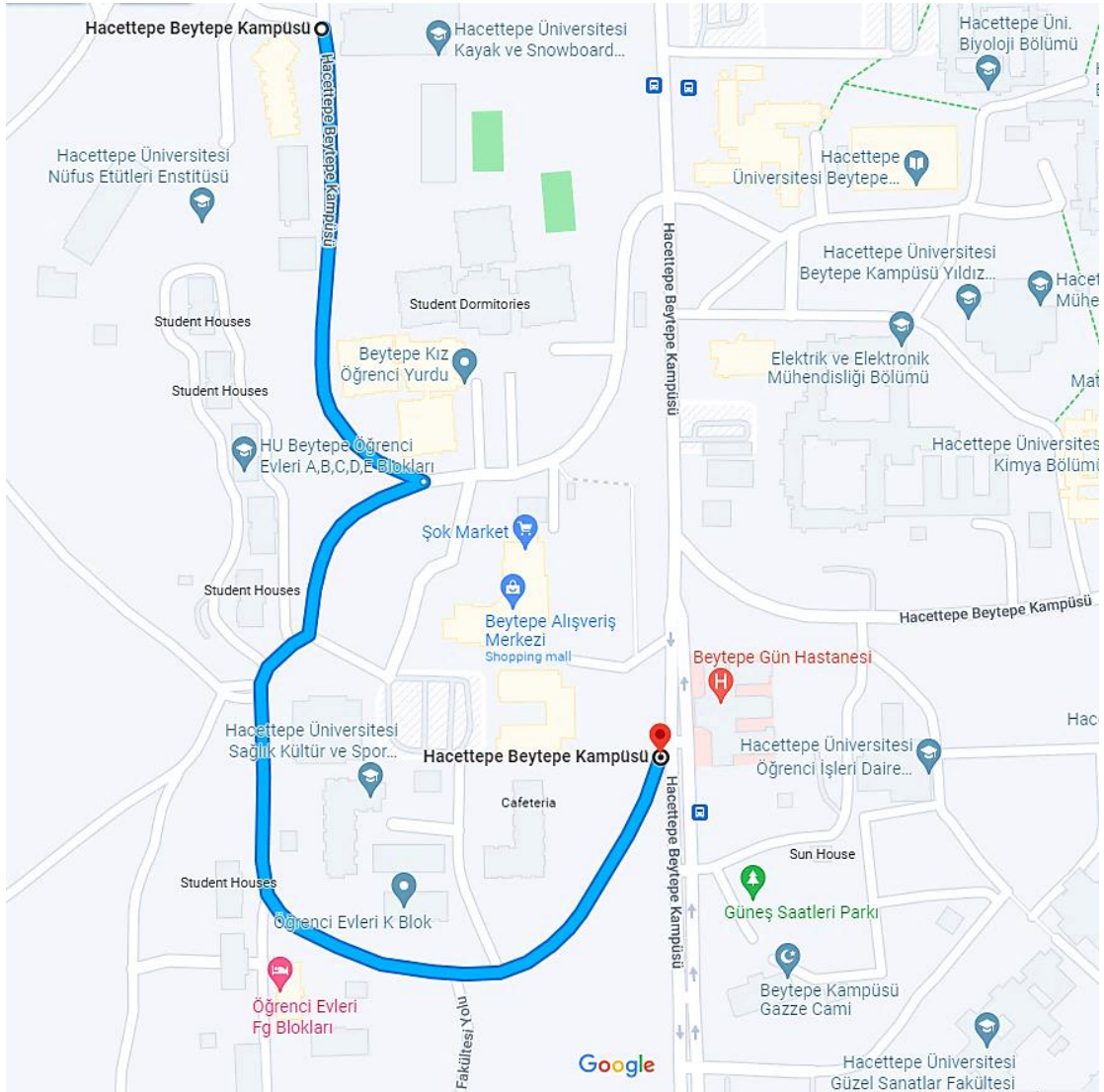


Figure 18. Drone Flight Route (Google Maps)

Today, several research are using unmanned aerial vehicles (UAVs) because these tools are fast and inexpensive. In addition, UAVs have recently been supported by the development of artificial intelligence technologies and have taken a larger place in human life.

Data acquisition was performed using a drone to capture high-resolution images of road deformations quickly and safely. This project used the DJI Mavic 2 Pro unmanned aerial vehicle, one of DJI's new drone models. It features a high-resolution camera. The drone weighs about 907 grams and has a length, width, and height of 322×242×84 mm when the wings are spread out (Figure 16). The drone's integrated camera has a 1-inch CMOS sensor and a resolution of 20 megapixels.



Figure 19. Drone used to collect images (DJI Mavic 2 Pro)

The necessary permissions were obtained from the authorized bodies of Directorate General of Civil Aviation (DGCA) and Hacettepe University for the UAV flight. For safety reasons, the flight was preferably conducted when the university is outside of academic hours and vehicle traffic is low. The drone was flown on the main road in the region of the dormitories of the Beytepe Campus of Hacettepe University. Continuous images and video clips were collected from this region. The start and end coordinates of the flight are listed in Table 3. The flight took place over a total distance of 1361 meters.

Table 3. Drone flight coordinates

	Flight Start Coordinate	Flight End Coordinate
Latitude	39; 52; 1.64430000001459575	39; 52; 12.6842999999935557
Longitude	32; 43; 58.3614999999989000	32; 43; 50.653900000000046258

Although the flight was initially launched autonomously, there were interruptions in the GPS signal because the flight area was in military airspace. For this reason, the UAV flight was conducted manually and parallel to the road to ensure safety. The flight altitude of the UAV could not be determined due to the manual flight and the lack of LIDAR integration, but it was flown at altitudes between 8 and 20 meters above the ground surface and these altitudes were recorded. Due to the low battery and loss of the GPS signal, four separate flights were conducted. The total flight time was 31 minutes. The total distance flown, total time, and number of images recorded for these flights are shown in Table 4.

Table 4. Drone Flight Records

Flight	Distance (meter)	Duration (minutes)	Road Photo Number
First Flight	616	12	30
Second Flight	205	6	21
Third Flight	299	6	23
Fourth Flight	241	7	28

During these flights, a total of 102 images of the road images and the campus images were collected from the flight area with. These images are 5472×3648 pixels in size. The pavement image taken in this flight is shown in Figure 20, and a photo of the Hacettepe University campus is shown in Figure 21 as an example.



Figure 20. Example of road image collected using UAV



Figure 21. Example of Hacettepe University Beytepe Campus image using UAV

3.3.1. Processing the Collected Images

The images captured by the drone were transferred to the map using ArcGIS Pro, a geographic information system-based software (Figure 22). This transfer process was based on the coordinate information of the images. The capture points of the 102 photos collected during the four flights mentioned in Table 4 are visualized on this map. OpenStreetMap (OSM) was used as the map base.



Figure 22. ArcGIS map of images taken from drone

Ground sampling distance (GSD) is a fundamental concept for any drone flight for research and inspection purposes. It forms the basis for many important flight planning decisions. This value can be used to determine from what altitude to fly to achieve the required resolution and quality of the desired data. This value is also important for the flight safety of the camera.

Ground sampling distance represents the size of one pixel on the ground. Ground sampling distance is usually specified in cm/pixel. This helps to find the amount of ground or surface area covered by a drone image. Two major variables are used to calculate the GSD: the flight altitude and the camera parameters. The camera sensor width, focal length, and image resolution are the camera parameters used to calculate the GSD. The formula for calculating the GSD is as follows.

$$\text{Ground Sampling Distance (GSD)} = \frac{\text{Sensor Width} * \text{Height (Altitude)}}{\text{Focal Length} * \text{Image Width}}$$

In order to calculate the ground sampling distance of each image, the distance of the camera to the ground at the time of capture must be known. However, this value is not the same for each image due to manual flight. For this reason, the altitude of each image was determined using the Google Earth application since the coordinates were known. In addition, the altitude of the point where the image was taken, that is, the altitude at which the drone is flying at that moment, is also included in the information of the photo. Thus, the distance to the ground was calculated by calculating the difference between the two altitudes. Using this data, the ground sampling distance was calculated for each image and used to determine the severity of the road damage.

The parameters of the camera connected to the drone used in this study and the information about the images obtained after the flight are summarized in Table 5.

Table 5. Parameters for the calculation of the GSD

Image Width (pixels)	5472
Image Height (pixels)	3648
Focal Length (mm)	28
Sensor Width (mm)	13.3
Sensor Height (mm)	8.8
Height (m)	Variable

4. METHODOLOGY

4.1. Deep Learning

Deep learning is a method that is now widely used in many areas of artificial intelligence. Deep learning, one of the subsets of machine learning algorithms, uses multilayer artificial neural networks (ANN). The learning process of this system is based on the previously given data, learns on this basis, and provides results. Compared to other types of machine learning, this method can learn information directly from data such as photos, videos, audio, or text, and predict outcomes. Its applications include classification, object detection, and natural language processing.

The structure of ANN is inspired by the way synapses work in the human brain. To process information, it uses a similar structure of nodes and connections. These networks perform operations on input data (x_0) received from another neuron and transmit the received output to another neuron or a layer. Artificial neural networks in the deep learning architecture operate according to this logic. Figure 23 shows a simple illustration of this structure.

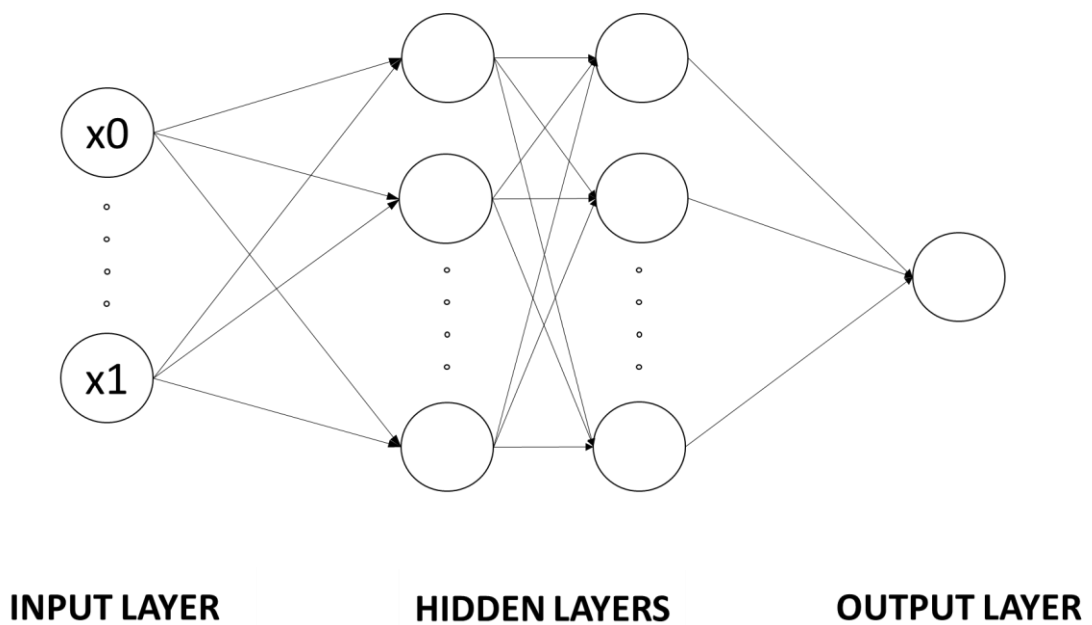


Figure 23. A simple illustration of ANN network structure (drawn by author)

Other techniques based on machine learning and image processing require upfront preprocessing. Deep learning techniques on the other hand, can work directly with raw data. They can process raw data without requiring preprocessing, because they transfer the relevant features of the desired object to be found within the algorithm. In order to achieve accurate results in deep learning, a large amount of data is required. While image classification algorithms can only categorize the data, object detection algorithms can both classify and locate the object.

In recent years, image classification and object detection in computer vision applications have evolved with the development of technologies. Great progress has been made thanks to access to large data sets and faster training on GPUs. In addition, libraries offered by large technology companies (Google, Facebook, Microsoft) have facilitated further studies in this area.

4.2. Convolutional Neural Networks

With the growth of data and the desire to extract more meaningful information from the data, optimization is required to extract attributes. Deciphering the connections between neurons and layers in this big data with a classical artificial neural network model and using the learned parameters presents a tremendous computational challenge. For that reason, CNN are introduced by LeCun and Bengio, (1995) to the field of computer vision. Convolutional neural networks work as a mathematical operation by using one or more layers. Convolution can effectively minimize the training complexity of the network model as well as the weights associated with the network connections and parameters, making it easier to train.

While the input data is an image, the properties of the neural network become pixel values. So, the model reads in the pixel values of an image, performs feature extraction and classification. However, the features of an image are not only the pixel values, but also the relationships between pixels, or in other words, image features are edges, lines, corners, or basically the shapes and patterns of an image. In order for classification to be performed with an artificial neural network, the artificial neural network must be given the features (feature extraction) that represent the relevant data. Selecting the right features is quite difficult, especially when the input data is an image. However, in

convolutional neural networks, the convolutional layer performs the feature selection automatically.

The convolutional neural network consists of two main parts: The first is a convolutional and pooling mechanism that divides the image into features and analyzes them, and the second is a fully connected layer that uses the output and predicts the class to identify the image.

The convolutional layer is added to the neural network and the network is now called a convolutional neural network. The purpose of the convolutional layer is to extract features from an image using a mathematical operation called convolution. The dot product between the kernel and the subarray of an input image of the same size as a kernel is calculated by the convolutional layer. The single pixel value of the output image is then determined by adding all the values obtained from the dot product. This procedure is continued for each kernel until the entire input image is covered. An illustration of this operation can be seen in Figure 24.

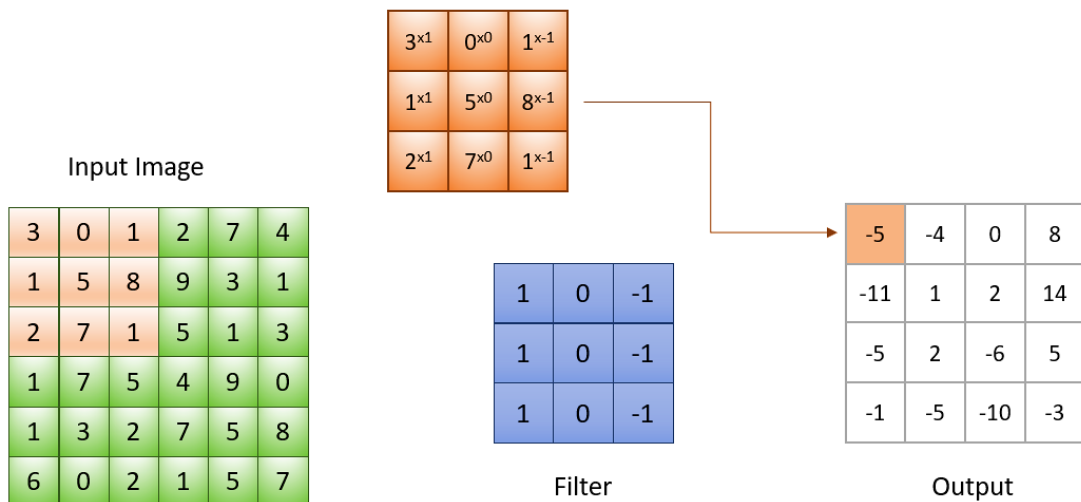


Figure 24. An illustration of a 3x3 filter with convolution operation

The pooling layer, the second layer of the architecture, aims to decrease the size of the matrix. This helps compress the amount of data and parameters to avoid overfitting. In the last layer, the image is converted to a single-column vector using a fully connected

layer. This generated flattened output is taken as input and the working logic of the classical artificial neural network is applied. Then, a fully connected structure is created by connecting each input to a neuron. After the model identify dominant and specific low-level features in images throughout a range of epochs, it can categorize the image and predict the class. The architecture of convolutional neural networks is shown in Figure 25.

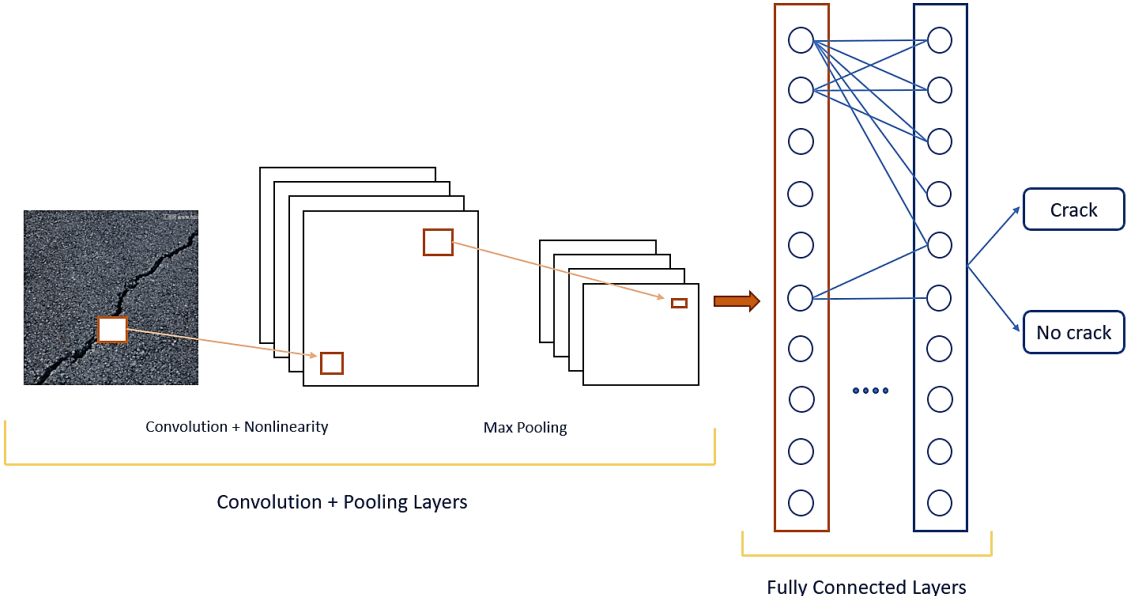


Figure 25. An illustration of the Convolution Neural Networks (CNN) architecture (drawn by author)

4.3. Object Detection

The image classification task is determining the class of a particular object. In the Object localization task, the position the objects are predicted, and bounding boxes are drawn around their extend. Object detection consists of these two tasks, categorizing the class of each object and locating one or more things in an image.

This study is about the classification and localization the pavement defects using images captured by a drone. For this reason, the images in the previously mentioned datasets were used as input data to train the object detection model. Object detection algorithms based on deep learning can learn the characteristics of all the pixels in the marked region and identify the class of this region thanks to the convolutional layers in the neural

network architecture. Thanks to these networks, the algorithm creates a map about the relationship between the data and stores it in memory. Thus, when a new image is presented to the trained model, the algorithm can predict the position and the class of the object to be found. The working principle of this algorithm is illustrated in Figure 26.

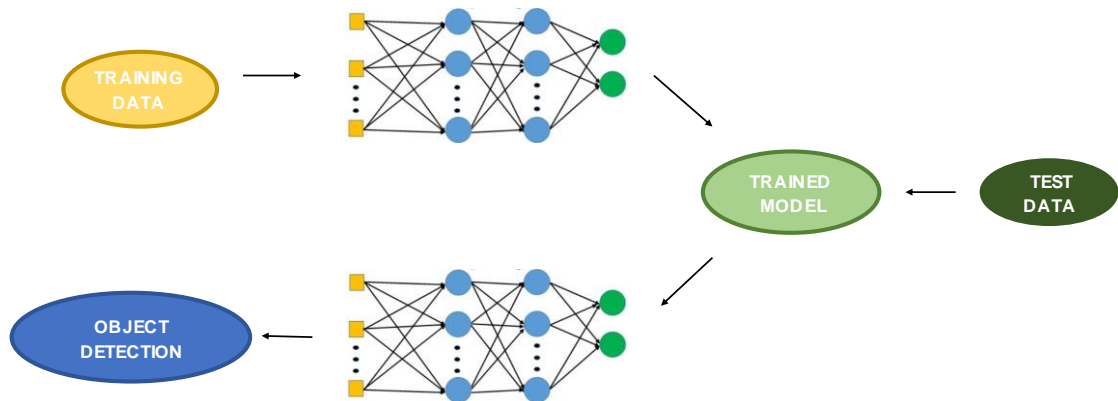


Figure 26. The workflow of object detection (drawn by author)

Deep learning-based object detection models are often split into two types: One-Stage Object Detection and Two-Stage Object Detection. Two-Stage object detector models have another part that finds possible regions, which distinguishes them from One-Stage object detector models. In the second part, the model works on the possible regions to classify them. However, One-Stage Object Detection models does all parts in one stage.

The most popular architectures for two-stage object recognition are R-CNN(Ibragimov et al., 2020), Fast R-CNN, and Faster R-CNN. As suggested by its name, the Faster R-CNN, which is the fastest model among them, is widely used today due to its high model performance. One-Stage Object algorithms include Single Shot Detector (SSD), RetinaNet, and YOLO versions.

The object detection models consist of two sections: a backbone and a head. While the backbone is responsible for learning features, the head creates bounding boxes around the desired objects and finds the class based on the learned features. ResNet, VGG and Mobile Net are the best-known examples of backbones. They can be trained on ImageNet and these pre-trained weights can be used in object detection algorithms. Some recent algorithms also include the neck part, which is the layer between the backbone and the

head. This part extracts more information from the different layers of the backbone and passes it to the head. In this study, two heads were preferred: YOLO and Faster R-CNN. A general flowchart of object detection models is illustrated in Figure 27.

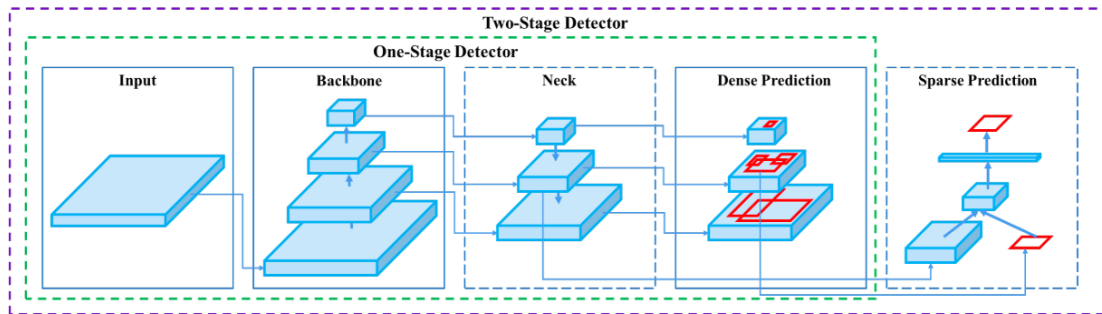


Figure 27. An illustration of object detection (Bochkovski vd., 2020)

Since the purpose of this study is to compare the one-stage and two-stage object detection models and to determine the model with the best performance, Faster R-CNN and YOLO versions were selected for this study. These object detection architectures examined in this study are listed below.

4.3.1. Faster R-CNN

The Faster R-CNN algorithm is an example of a two-stage object detection model that is the evolution of Fast R-CNN (Girshick, 2015). In the Faster R-CNN network, an additional neural network layer was added, which solved some problems of Fast R-CNN. The new neural network layer is called the Region Proposal Network (RPN).

In this algorithm, convolutional layers first create features from the input image. Then, the candidate regions where the object can be found are estimated by RPN, and the feature map is created using fully convolutional neural networks. The output of RPN is used as input to the Region of Interest (ROI) pooling layer. Then, object detection is performed for each predicted region. The object detection phase is performed by R-CNN. These phases are shown in Figure 28.

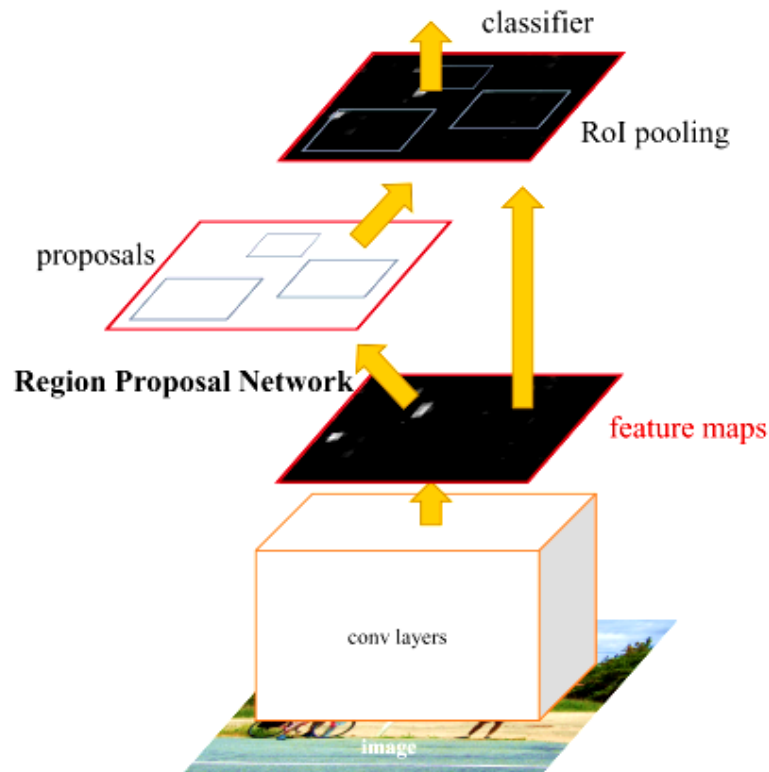


Figure 28. Faster R-CNN structure (Ren et al., 2017)

In order to train a model with Faster R-CNN, ResNet50 and ResNet 101 which are pre-trained networks were utilized as a backbone. More than a million photos from the ImageNet database were used to train these convolutional neural networks.

4.3.2. YOLOv5

The You Only Look Once (YOLO) algorithm was initially introduced by Redmon et al., (2016) for fast object detection in real time. YOLO and its versions are algorithms that perform one-stage object detection. They can directly recognize the desired object using convolutional networks. The most important feature that distinguishes the YOLO algorithm from the other algorithms is its ability to perform object detection in real-time. It is also superior to its alternatives in terms of speed, as the algorithm predicts the bounding boxes of an image at once. How this algorithm detects the objects is explained in Figure 29.

YOLO uses the Darknet framework, a structure trained on ImageNet, with 53 network layers as the backbone for feature extraction. In the YOLOv5 architecture, the algorithm is constructed by using PyTorch as the coding base. This is much more user-friendly and easier. For this reason, YOLOv5 became the most widely used version of YOLO.

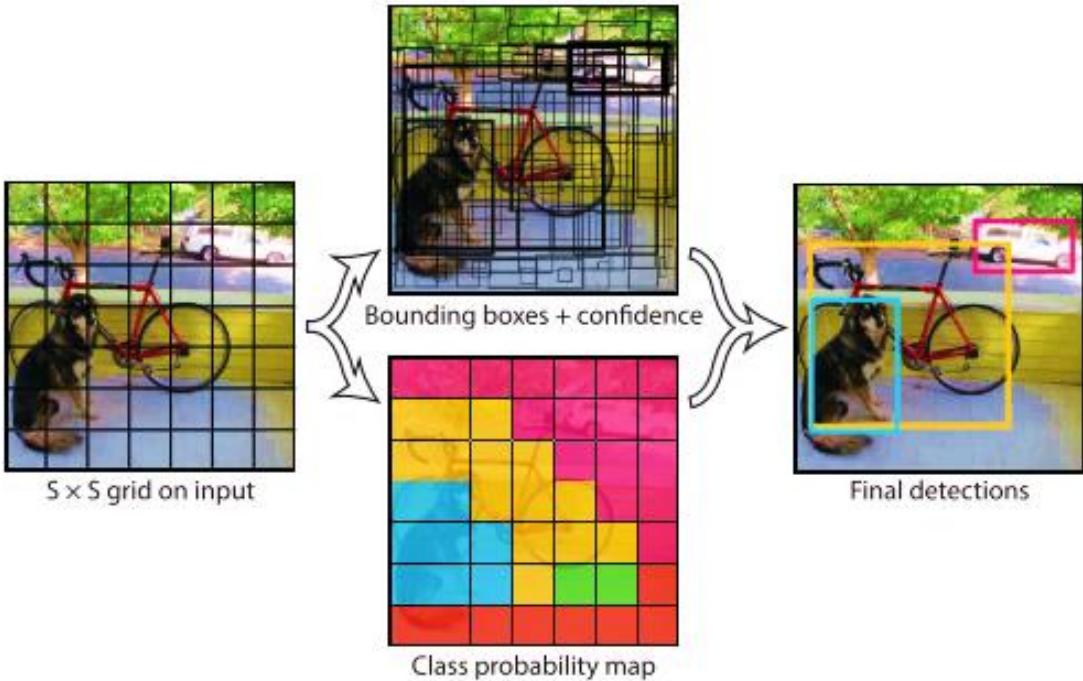


Figure 29. YOLO object detection system (Redmon et al., 2016)

Four different pre-trained models of the YOLO algorithm were used in this study. The YOLOv5s, YOLOv5s6, YOLOv5m, and YOLOv5m6 models were selected based on their performance on COCO data.

4.3.3. YOLOv7

YOLOv7 is the latest version of the YOLO series by Wang et al., (2022). With this release, the field of object detection makes enormous progress. YOLOv7 also focuses on optimizing the training process by using trainable Bag of Freebies. In this way, the real-time object recognition parameters can be reduced by about 40% and the computer usage can be reduced by 50%, resulting in faster inference speed and higher recognition accuracy. While training costs decrease, model accuracy and speed of inference increase.

Like YOLOv5, YOLOv7 uses PyTorch as the coding base. In the MS COCO Object Detection task, YOLOv7 outperforms other convolutional networks and earlier versions of its series in terms of average precision (AP) and inference time (Figure 30). There are also some pre-trained weights available for YOLOv7 object detection. The most suitable weights were employed for automatic defect detection: YOLOv7, YOLOv7-tiny and YOLOv7-X.

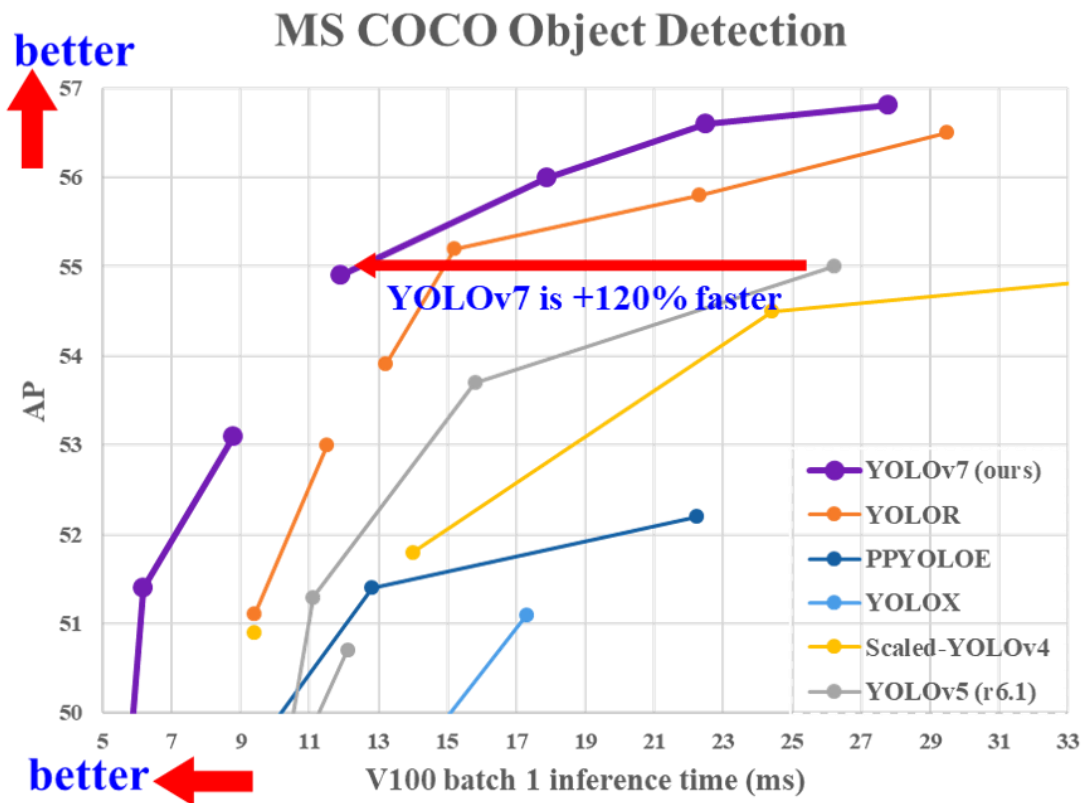


Figure 30. Comparison of YOLOv7 on the COCO dataset (Wang et al., 2022)

4.4. Object Segmentation

The goal of image segmentation is splitting images into several smaller parts by labeling each pixel of an image. These parts, or multiple segments, help the image segmentation model learn the task. The basic requirement for image segmentation is the use of masks in the training phase. These masks can be a binary image consisting of zero or non-zero values of pixels. The output has the same size as the input image, in which each pixel is labeled. Therefore, it can be called a pixel-level classification.

In this study, CrackForest Dataset (CFD) (Shi et al., 2016), Crack500 (Yang et al., 2020), Cracktree (Zou et al., 2012), GAPs (Eisenbach et al., 2017) and DeepCrack (Liu et al., 2019) were used to train the segmentation model. 1500 randomly selected images from these datasets were used. Before the training step, these images were resized to 448x448.

4.4.1. U-Net

U-Net, a convolutional network-based network structure, was introduced by Ronneberger et al., (2015) and used for biomedical image segmentation. The success of U-Net in biomedical image segmentation has led to the use of this model in many segmentation studies. In this study, the U-Net architecture was preferred for crack segmentation.

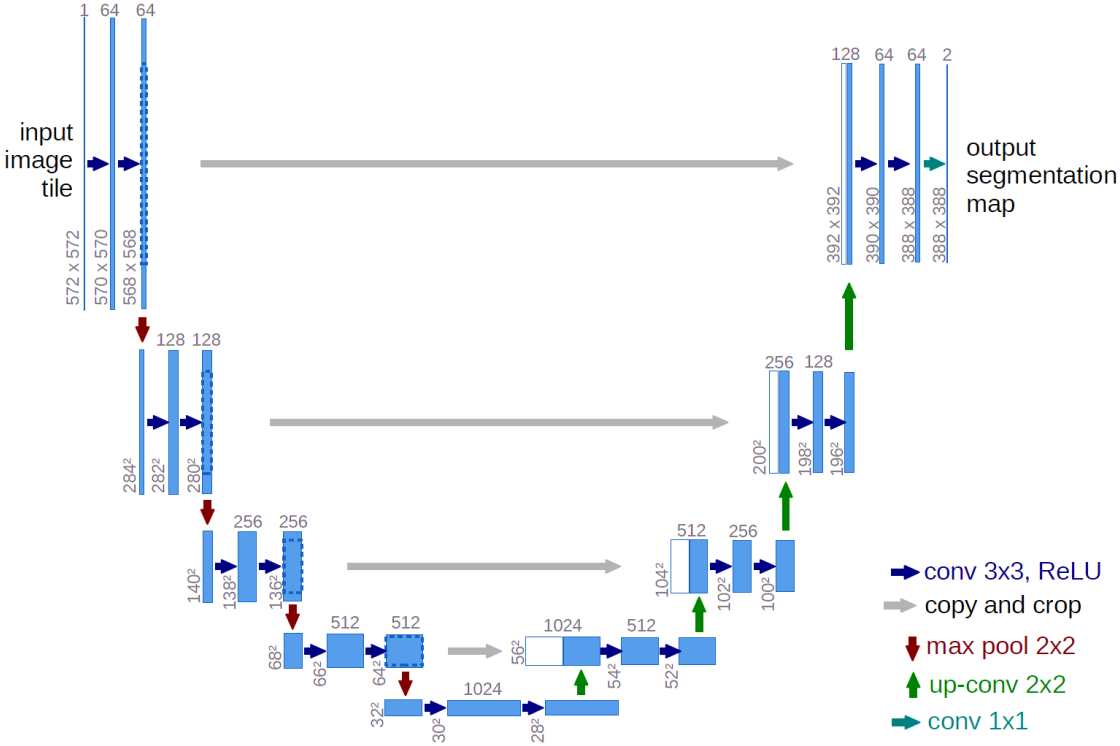


Figure 31. The architecture of U-Net (Ronneberger et al., 2015)

The U-Net architecture includes two paths: contracting and expansive, as shown in Figure 31. The first part consists of classical convolutional neural networks (conv) and max-pooling layers, and the network is constrained by applying 3x3 convolutions in each part. This part is called the encoder and is used to understand the model image. This part can

also be referred to as the feature extractor. It generates the desired mask by understanding the input data with filters. The second part is the decoder part, where the object requested by the model is learned. In this part, a 2x2 upward convolution is performed and the number of features is halved. This layer is mainly used to check the localization. In the last layer, a 1x1 convolution is performed and the output equals the quantity of label classes. The U-net owes its name to the U-shaped image that is created in each stage of this algorithm.

The feature extractor networks in the first part can be replaced by pre-trained networks. In this study, the encoder part was replaced by two different pre-trained convolutional neural networks: VGG-16 (Simonyan and Zisserman, 2014) and ResNet50 (He et al., 2015). These networks were trained with the ImageNet dataset so that they are able to extract the required features. The VGG-16 network was designed to diminish the amount of parameters and speed up the training time in convolutional layers. In ResNet50 (Residual Networks) network, there are some additional layers to figure out complex problems. ResNet50 is often preferred for transfer learning. These deep learning networks were trained to develop a model for crack segmentation, and their results are presented in the results section.

4.5. Quantification of Pavement Defects

There are certain criteria for determining the maintenance and repair of road damage. The criteria for each class of damage generally relate to the geometric structure of the damage and the area it occupies. Damage is classified according to the severity of the damage based on these criteria.

Each country or region has different guidelines for repairing road damage. To find guidelines, an extensive search of the literature was conducted, and many guidelines were found around the world. First, standards for road damage repair were found primarily for the U.S. states: Oregon, Pennsylvania, Florida, Washington, California, Indiana, and North Carolina. Additional guidelines were also found for the Canadian provinces of British Columbia and Ontario. The guidelines for each state are listed in the Appendix-1 for alligator cracking, longitudinal cracking, reflective cracking, and potholes, respectively.

It was investigated whether there is a specification for road damage inspection in Turkey. Engineers and experts from Ankara Metropolitan Municipality, Ankara Keçiören Municipality, Antalya Metropolitan Municipality, and the General Directorate of Highways were consulted on this issue. However, investigation revealed that there was no guide currently in use. Therefore, for this study, a new guide compatible with the metric system was prepared for four classes of damage, using the guidelines in America and Canada as a reference.

Table 6. Pavement defects inspection criteria used in the state of California

Distress Type	Severity	Limits for California
Longitudinal Crack	Low	Average crack width <0.25 in
	Medium	0.25 in ≤ Average crack width ≤0.5 in
	High	Average crack width > 0.5 in
Reflective Crack	Low	Average crack width <0.25 in
	Medium	0.25 in ≤ Average crack width ≤0.5 in
	High	Average crack width > 0.5 in
Alligator Crack	Low	Average crack width <0.25 in
	Medium	0.25 in ≤ Average crack width ≤0.5 in
	High	Average crack width > 0.5 in
Pothole	Low	0 <number<2
	Medium	2 ≤ number ≤ 5
	High	number> 5

The criteria used in the State of California for longitudinal cracks, transverse cracks, alligator cracks, and potholes are shown in Table 6. These limits were converted to the metric system for use in this study. However, the limits for potholes in this guide were not considered adequate. For this reason, the pothole limits in the guide used in the Canadian province of British Columbia, listed in Table 7, were used as an additional limit. Since the images are two-dimensional, the depth of the cracks and pits could not be measured. Therefore, the diameter and number of potholes are used to determine the damage limit for the pothole. The reference limits used in this study are given in Table 8 in the metric system.

Table 7. Pothole inspection criteria used in the Canadian province of British Columbia

Distress Type	Severity	Limits for British Columbia
Pothole	Low	Pothole < 175 cm ² in area (~15cm ø) and less than 25mm deep
	Medium	Pothole > 175 cm ² in area (~15cm ø) and 25mm to 50mm deep
	High	Pothole > 175 cm ² in area (~15cm ø) and greater than 50mm deep

The limits in Table 8. allow the usability of the road to be determined in the region where the study was obtained. Based on these criteria, the degree of damage is determined, and then maintenance and repair are performed according to the degree. For example, if the degree of damage is considered high, the maintenance of damage must be carried out urgently.

Table 8. Pavement defects inspection limits used in this study

Distress Type	Severity	Limits for British Columbia
Longitudinal Crack	Low	Average crack width <0.635 cm
	Medium	$0.635 \text{ cm} \leq \text{Average crack width} \leq 1.27 \text{ cm}$
	High	Average crack width > 1.27 cm
Reflective Crack	Low	Average crack width <0.635 cm
	Medium	$0.635 \text{ cm} \leq \text{Average crack width} \leq 1.27 \text{ cm}$
	High	Average crack width > 1.27 cm
Alligator Crack	Low	Average crack width <0.635 cm
	Medium	$0.635 \text{ cm} \leq \text{Average crack width} \leq 1.27 \text{ cm}$
	High	Average crack width > 1.27 cm
Pothole	Low	0 < number of potholes < 2, Diameter < 15 cm
	Medium	$2 \leq \text{number of potholes} \leq 5,$ Diameter > 15 cm
	High	Number of potholes > 5, Diameter > 15 cm

4.6. Environmental Setup

Setting up the required environment is the first step before starting to train the object detection model. Python is the most common programming language used when creating deep learning models. Python must be utilized on the computer before to develop an

object detection model. In this study, Anaconda, a free and open-source software developed to use Python language, was used as the development environment.

A powerful graphics processing unit (GPU) is also important when working with larger neural networks and a large amount of data. They allow processing in small steps in parallel and are therefore faster than CPUs. All training experiments are performed using GPUs on a laptop with 16 Gb of RAM and a graphics card called NVIDIA GeForce GTX 1660Ti and Windows operating system. To use GPUs, the required CUDA and cuDNN were installed. While CUDA 10.2 was used for training the YOLO models, CUDA 11.2 was used for training the Faster R-CNN models.

Two frameworks were used to create an environment for YOLO versions and Faster R-CNN: Tensorflow and PyTorch. TensorFlow's object detection application program interface (API) provides an environment for building models for deep learning-based object detection. A Faster R-CNN architecture is also available in that framework. The YOLO is an object detection algorithm that uses PyTorch as its coding foundation. Thus, the installation of the PyTorch and the Tensorflow was completed on the Anaconda software. In addition, PyTorch was also used to create a crack segmentation model with U-Net.

4.7. Metrics

Before training the model, the hyperparameters should first be determined. These are the variables that shape the structure of the network. There is no fixed value for them, they vary depending on the task and the dataset. The experiment should continue with trying the different hyperparameters to find the best ones. The hyperparameters used in this study are listed in below.

After training, the performance of each model should be evaluated according to certain metrics. Based on these metrics, conclusions should be drawn to compare the models in terms of their efficiency. Generally, the accuracy value is considered as the accuracy, but the accuracy is not sufficient when the data sets are unstable and small. Therefore, the evaluation metrics of COCO (Lin et al., 2014) were used. There includes a brief explanation of evaluation criteria such the confusion matrix, recall, precision, and F1 score.

4.7.1. Hyperparameters

When the deep learning network is trained, certain parts of the data are included in the training. Once the first group is trained, the performance of the model is tested. Then backpropagation is performed according to the result, and the results are updated. The model is trained again with new data and the results are updated again. This process is continued in this way trying to get the best result. Each of these steps is called an epoch. The number of epochs varies contingent on the number of parameters of the algorithm and its speed.

How much data is entered is one of the most important parameters. Although the accuracy of the model increases with increasing image size, using a large image size leads to a decrease in the training speed of the model. Therefore, 640x640 images were preferred for all trainings in this study and the size of the images was changed to 640x640.

Another parameter is the batch size. The batch size allows the model to train with small pieces. Splitting the dataset into small parts during the training phase is called a batch. This can speed up the training because less memory is required. Although a large batch size increases the accuracy of the model, a lot of memory is required for this process. Therefore, considering the capacity of the computer, the optimal batch size is set to 4.

Learning rate and momentum coefficient are the other hyperparameters. The learning rate is a parameter that should be used during the training of the model. It ensures to determine the weighting result for each class during the training phase. This value can be set as a fixed value or as a value that increases or decreases with time. In this study, the learning coefficient was initially set at 0.01, but as a variable that decreases with time. The momentum coefficient was set at 0.9.

Another important parameter is the Intersection Over Union (IOU), which indicates the accuracy value of the created boxes. The exact bounding box of the damage is compared to the bounding box obtained after testing the model. The proportion of the overlapping area to the overall area is equal to the IoU value. This value must be determined before training the model. The model needs this value when it validates itself during training. While boxes below this value are deleted, boxes above the value are kept. The symbolized formula for calculating the IOU value is illustrated in Figure 32.

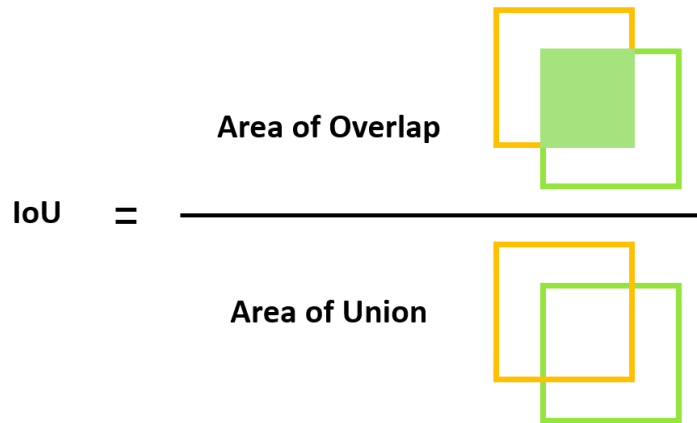


Figure 32. IOU formulization

The parameters mentioned above are the same for both algorithms, but the number of epochs differs. The hyperparameters used for each model are summarized in Table 9.

Table 9. Hyperparameters for each model

Model	Image Size	Batch size	IoU	Learning Rate	Momentum Coefficient	Epochs
YOLOv5	640x640	4	0.4	0.01	0.9	30
YOLOv7	640x640	4	0.4	0.01	0.9	100
Faster R-CNN	640x640	4	0.4	0.01	0.9	25000

4.7.2. Evaluation Metrics

Confusion Matrix

The predicted and actual values can be compared using at the confusion matrix. There are four different outputs, which are explained below in relation to the subject under study.

- True Positive:* Predicting damage to the image that is actually damaged
- True Negative:* Predicting that there is no damage in the image that is not actually damaged
- False Positive:* Claiming that there is damage to the image that is not actually damaged
- False Negative:* Claiming that there is no damage to the image that is actually damaged

The Confusion Matrix can be defined as the output of these values in a general frame. The general framework of this matrix is shown in Figure 33.

PREDICTED	Negative	False Negative	True Negative
	Positive	True Positive	False Positive
		Positive	Negative
		ACTUAL	

Figure 33. Confusion Matrix

Precision

The precision value is the proportion of those correctly predicted positively to the objects predicted to be positive. A high precision value is an important criterion that shows that the performance of the model is good. The formula of this metric is given below.

$$\textit{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

Recall

The recall value is the ratio of correctly predicted positive objects to whole objects belonging to same class.

$$\textit{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}}$$

There is a curve that illustrates the relationship between precision and recall. The effectiveness of the binary classification model may also be assessed using it. Both high precision and high recall are desirable properties for the algorithm. Therefore, the Area Under Curve (AUC) can be used as an alternative metric in some cases.

F1-Score

The harmonic mean between the Precision and Sensitivity is represented by the F1 score. The F1 value is a value between 0 and 1. If this value is close to 1, it means that the accuracy of the model is high. The formalization of this value is explained below.

$$\textit{F1 Score} = 2 * \frac{\textit{precision} * \textit{recall}}{\textit{precision} + \textit{recall}}$$

mAP

Mean Average Precision (mAP) is generally exploited metric for evaluating the model, especially in the object detection algorithms such as Faster R-CNN, YOLO, Mask R-CNN. This value is the average of the precision value calculated for each class. It takes a value between 0 and 1. This metric is calculated with the following general formula.

$$mAP = \frac{1}{N} \sum AP_i$$

The mAP values are considered to find the percentage of correct predictions in the model. In this study, these values were observed to compare three different object detection models in terms of their performance.

Loss

Basically, the loss function is the measure of the relationship between the dataset and the algorithm. The loss function should be chosen before training begins. This metric is required for the model to efficiently determine the model error. The loss value represents a probability value between 0 and 1. It is calculated using a logarithmic function based on how much the predicted value deviates from the true value. Thus, if the model does not predict well, the loss value will be close to 1 due to the large difference between the actual value and the predicted value. The loss value of a good model should be close to 0. So, it is always tried to minimize the loss by changing the parameters.

The formula of the most common lost function The Mean Squared Error is described below. While Y_i represents the actual value, Y' represents the model prediction.

$$MSE = \frac{1}{N} \sum (Y_i - Y')^2$$

5. RESULTS

5.1. Training Results

Object detection models were trained using the previously prepared road damage detection dataset. To evaluate and compare the training results of the models, the previously mentioned evaluation metrics are calculated for each model. The training performances of the Faster R-CNN, YOLOv5, and YOLOv7 algorithms are elaborated in this chapter using Precision, Recall, mAP, and Training Time.

The training results of crack segmentation models with U-Net were also be mentioned in this part.

The trained models that performed the best in automatic road damage detection were examined using the test images. A total of 281 images were used for the test of defect detection. The distribution of this test data according to the number of labels is shown in Figure 34.

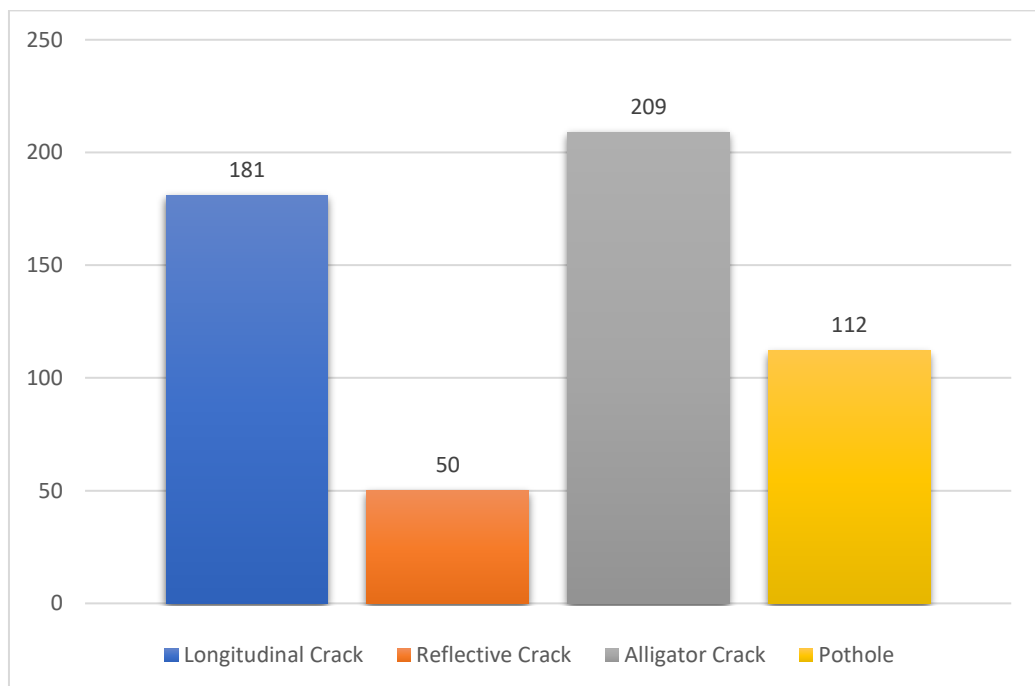


Figure 34. Distribution of test data according to the number of labels

5.1.1. Faster R-CNN Defect Detection

In this study, the Faster R-CNN architecture was trained with the prepared dataset. Two popular pre-trained networks ResNet50 and ResNet101 were employed for automatic damage detection. The performances of these Faster R-CNN models are compared in Table 10 with respect to the evaluation metrics.

Table 10. The performances of the Faster R-CNN models

Model	Recall	Loss	mAP	Training time (h)
ResNet 50	0.52	0.035 (training) 0.29 (validation)	0.44	6.42
ResNet 101	0.38	0.007 (training) 0.147 (validation)	0.59	16.21

Since the size of the ResNet50 network structure is smaller, it is compatible if the dataset contains an object that can be easily distinguished, or whose features can be easily classified. However, determining the pavement defect class and extracting its features requires a more detailed study. Since the number of pixels of these objects is small and the number of variables in the images is high, it is difficult to detect them with a small network. Therefore, an average precision value (mAP) of 0.44 was obtained in the study using the ResNet50 network.

The training result of the ResNet101 model was much better, as can be seen in Table 10. This is because the ResNet101 network has quite a large architecture compared to the ResNet50 network. Therefore, it can learn the smallest details of objects. The mean precision value (mAP) in this training was found to be 0.59. Although the average precision value is high, it is quite slow in terms of training time.

Loss values for Faster R-CNN models are considered in three different ways: Classification loss, Localization loss, and Regulation loss. To evaluate all these losses together, it would be more precise to examine the total loss value. The loss values found as a result of training the Faster R-CNN model with ResNet101 are shown in Figure 35

by number of epochs. These graphs were created using Tensorboard, a tool offered by Google that can be used to visualize the network structure and many parameters. These graphs belong to the loss values formed in the validation phase. After completing the training at the end of 25,000 cycles, it was observed that the loss values approached 0.

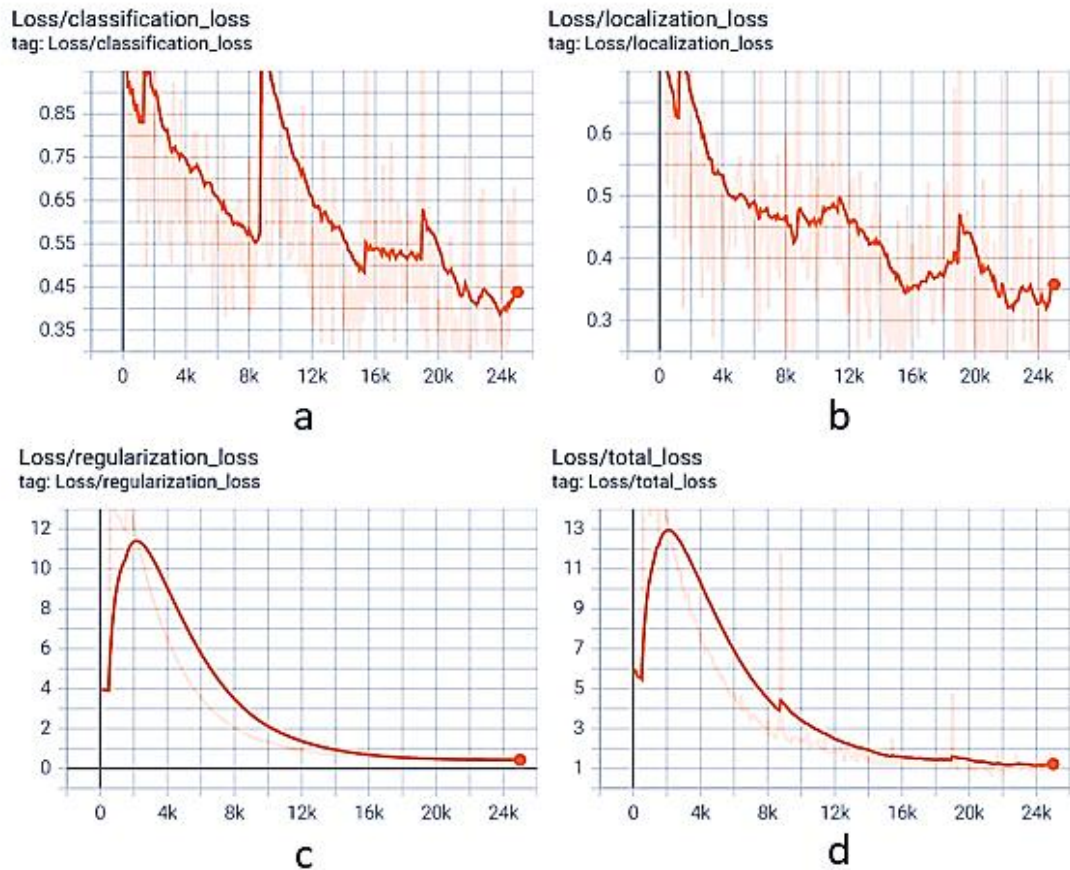


Figure 35. a) Classification loss b) Localization loss c) Regularization loss d) Total loss of the Faster R-CNN model with ResNet101

5.1.2. Yolov5 Defect Detection

Several pre-trained layers can be utilized for the YOLOv5 object detection model. These networks are ordered from the smallest to the largest in terms of size as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Figure 36 provides an overview of the performance of these pre-trained weights on the dataset COCO. Larger models such as Yolov5x and Yolov5x6 generally perform better in all situations. However, they require more CUDA memory to train and they run very slowly, since they have more parameters.

In the literature, YOLOv5s/m is generally preferred for mobile classifications. Considering the capacity of the computer on which the study was performed, the use of YOLOv5s, YOLOv5m and their versions (YOLOv5s6, YOLOv5m6) was chosen.

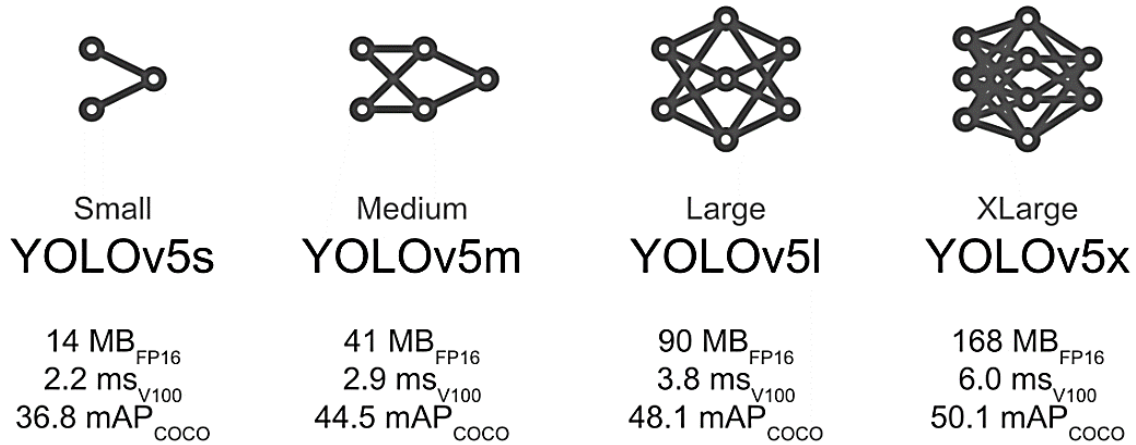


Figure 36. Pre-trained networks for YOLOv5

Training results for the YOLOv5s, YOLOv5m, YOLOv5s6, and YOLOv5m6 models are compared in Table 11 in terms of precision, sensitivity, F1 score, mean precision, and training time. Test results for the YOLOv5 models were visualized using Wandb (Weights and Biases) (Biewald, 2020).

Table 11. The performances of the YOLOv5 models

Model	Precision	Recall	F1-Score	mAP	Training Time (h)
YOLOv5s	0.62	0.60	0.61	0.60	3.72
YOLOv5s6	0.58	0.61	0.59	0.61	3.78
YOLOv5m	0.58	0.62	0.60	0.61	6.33
YOLOv5m6	0.68	0.62	0.61	0.65	6.74

It was observed that training time increased with increasing model size and complexity. Since the YOLOv5s weighting is smaller than the YOLOv5m weighting, the training speed is almost half. In addition, the average precision value (mAP) in each model was over 60%. Considering the ratios of precision, recall and F1 value, the performance of the models is close to each other. However, although the training time was slow, the model with the highest performance in terms of all metrics was the YOLOv5m6.

Figure 37 visualize the progression of mAP, precision and recall of the YOLOv5m6 model with the validation data during training. While these initially increased, they remained constant thereafter.

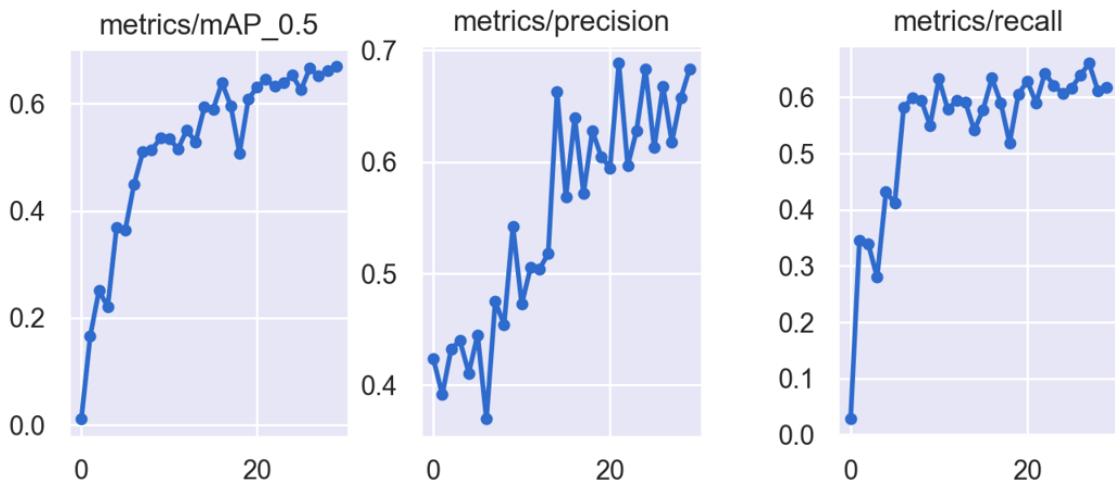


Figure 37. The visualization of mAP, precision and recall of the YOLOv5m6 model with the number of epochs during training

The loss values obtained after testing the YOLOv5m6 model with validation data during training are summarized in Table 12 for training and validation. The changing of these values over 30 epochs is shown graphically in Figure 38.

The bounding box losses, object losses, and classification losses for validation and training of the YOLOv5 defect detection model were determined using the experiment tracking tool Wandb. After completing the training at the end of 100 epochs, it was observed that the loss values approached 0.

Table 12. The loss values of YOLOv5m6 model

	Bounding Box Loss	Object Loss	Classification Loss
Training	0.0301	0.0201	0.0181
Validation	0.0167	0.0175	0.0012

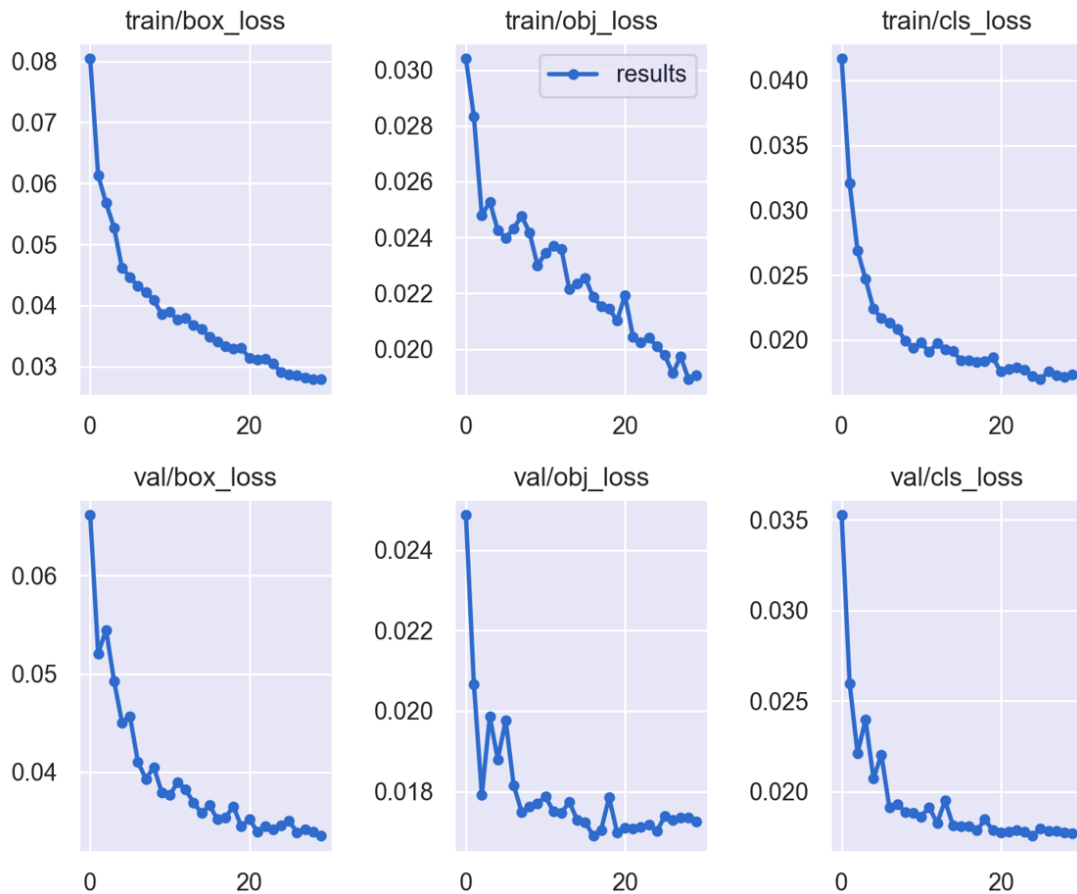


Figure 38. Loss values of the YOLOv5m6 model in terms of the number of epochs during training and validation

In this model training, the average precision for all classes in the validation data was 0.65. During the training of the model, an evaluation study was performed with validation images and the average precision values for each class were determined separately.

To understand the relationship between the classes, confusion matrix is determined. The correct prediction rates of each class in this model are given in the confusion matrix in Figure 39. When looking at the confusion matrix, it was found that the class with the

highest rate was the pothole with 0.77. Since the images of potholes contain clearer information about the pixels and the area occupied in the image is larger, they were easily identified using the model. The damage class with the lowest proportion was the longitudinal crack with a value of 0.45.

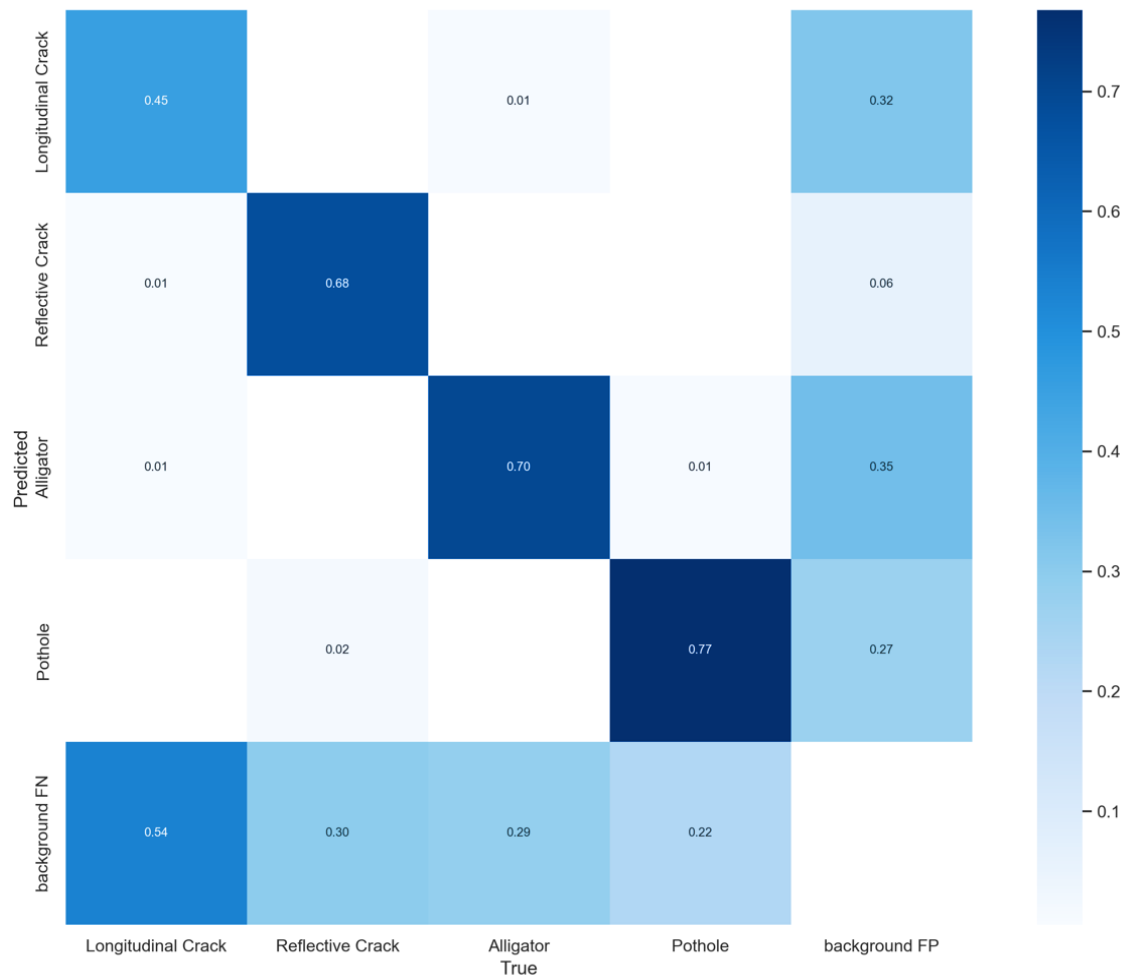


Figure 39. Confusion Matrix as a result of the YOLOv5m6 model for training

5.1.3. Yolov7 Defect Detection

Multiple pre-trained layers can be used when training the YOLOv7 object detection model. There are many models in the YOLOv7 family. However, considering the capacity of the computer and its memory, three models that have the smallest parameters were selected. In this study, the normal model of YOLOv7, YOLOv7-tiny and YOLOv7-X weights and their pre-trained weights were utilized. While YOLOv7-tiny is the smallest

network among the YOLOv7 family, YOLOv7-X consists of larger networks. In the Table 13, the suggested YOLOv7 architectures are compared using the evaluation metrics.

Table 13. The performances of YOLOv7 models

Model	Precision	Recall	mAP	Training Time (h)
YOLOv7	0.64	0.62	0.67	6.23
YOLOv7_training	0.67	0.57	0.66	6.04
YOLOv7-tiny	0.57	0.55	0.58	6.30
YOLOv7x	0.61	0.54	0.59	6.22
YOLOv7x_training	0.62	0.52	0.56	6.27

As for the mAP values, the most successful model, YOLOv7, has a mean average precision of 0.67. The training graphs of this model are shown in Figure 40. The progression of mAP, precision and recall of the YOLOv7 model is illustrated in Figure 40. After showing an increase up to a certain point, they remained stable.

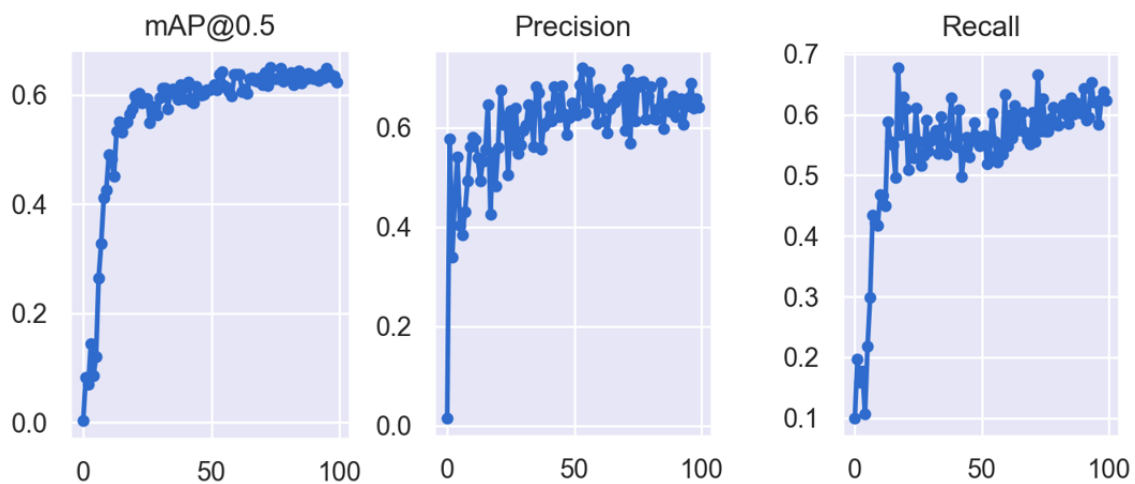


Figure 40. The visualization of mAP, precision and recall of the YOLOv7 model with the number of epochs during training.

The final loss values obtained for the training and validation of the YOLOv7 model are shown in Table 14. The variation of the loss values during 100 cycles is shown graphically in Figure 41.

Table 14. The loss values of YOLOv7 model

	Bounding Box Lost	Object Loss	Classification Loss
Training	0.0249	0.0055	0.0008
Verification	0.0447	0.0236	0.0065

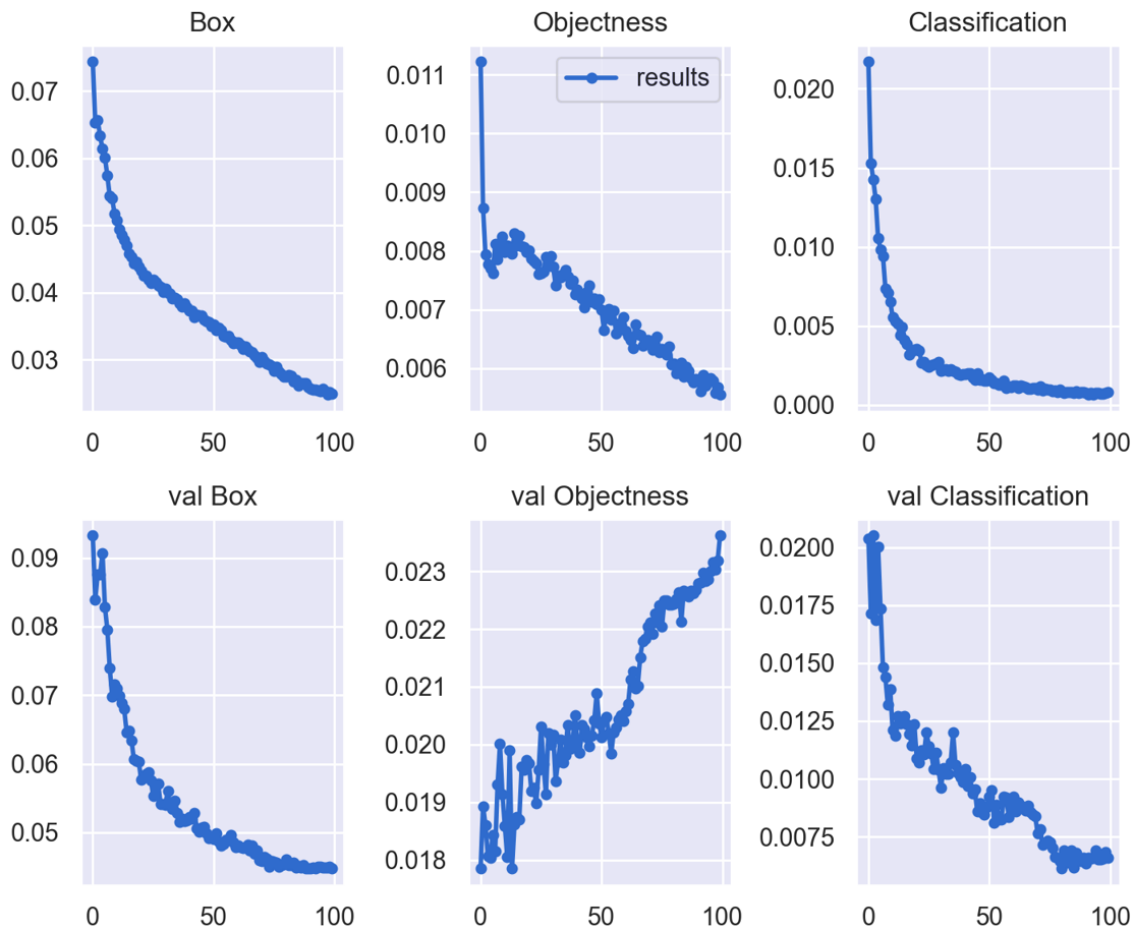


Figure 41. Loss values of the YOLOv7 model with the number of epochs during training and validation

When training the YOLOv7 model, mean average precision value of 0.67 was obtained for the validation data for all classes. The correct prediction rate obtained for each class is shown in Figure 42 in the confusion matrix. It was found that the class with the highest rate was alligator cracks with 0.72. This value is followed by reflective cracks with 0.71. While the potholes have a rate of 0.67, the longitudinal cracks have the lowest value of 0.60.

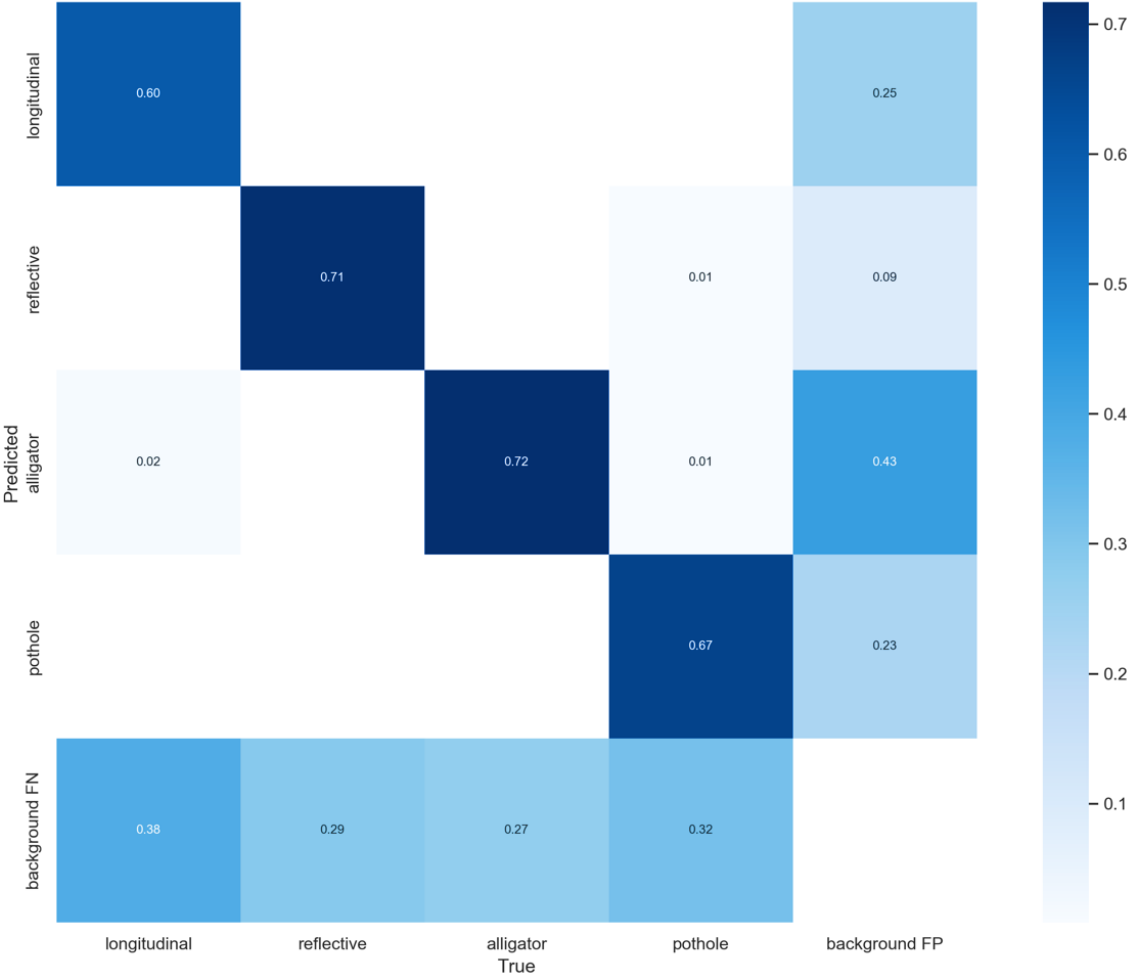
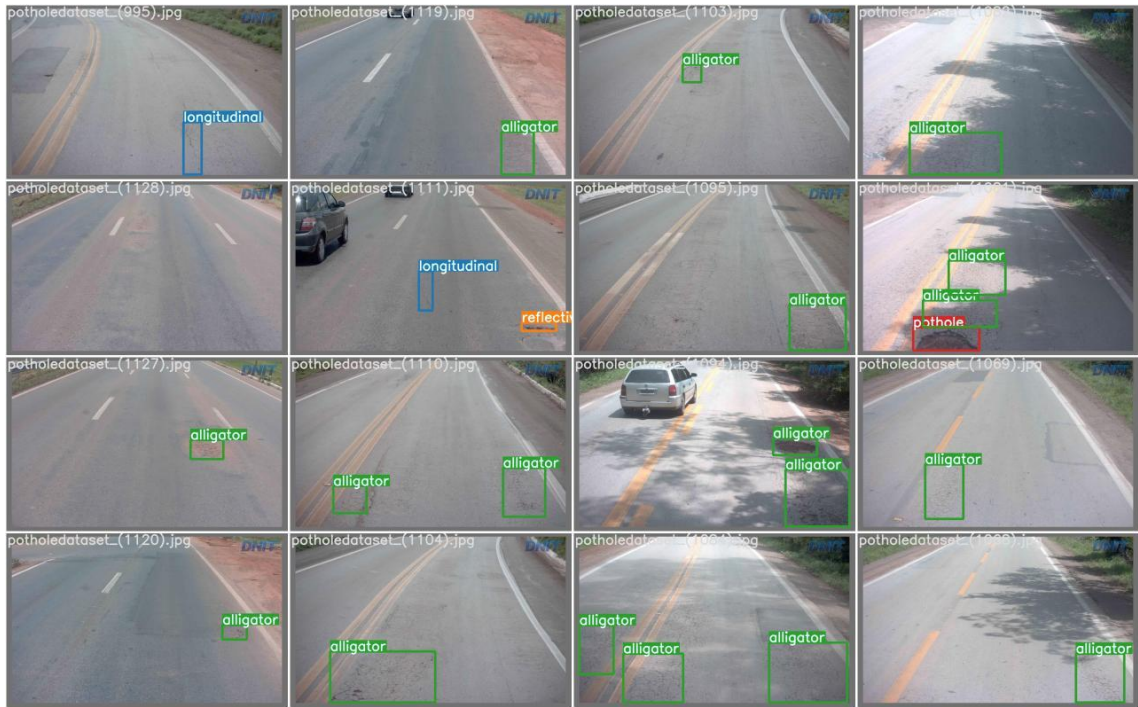
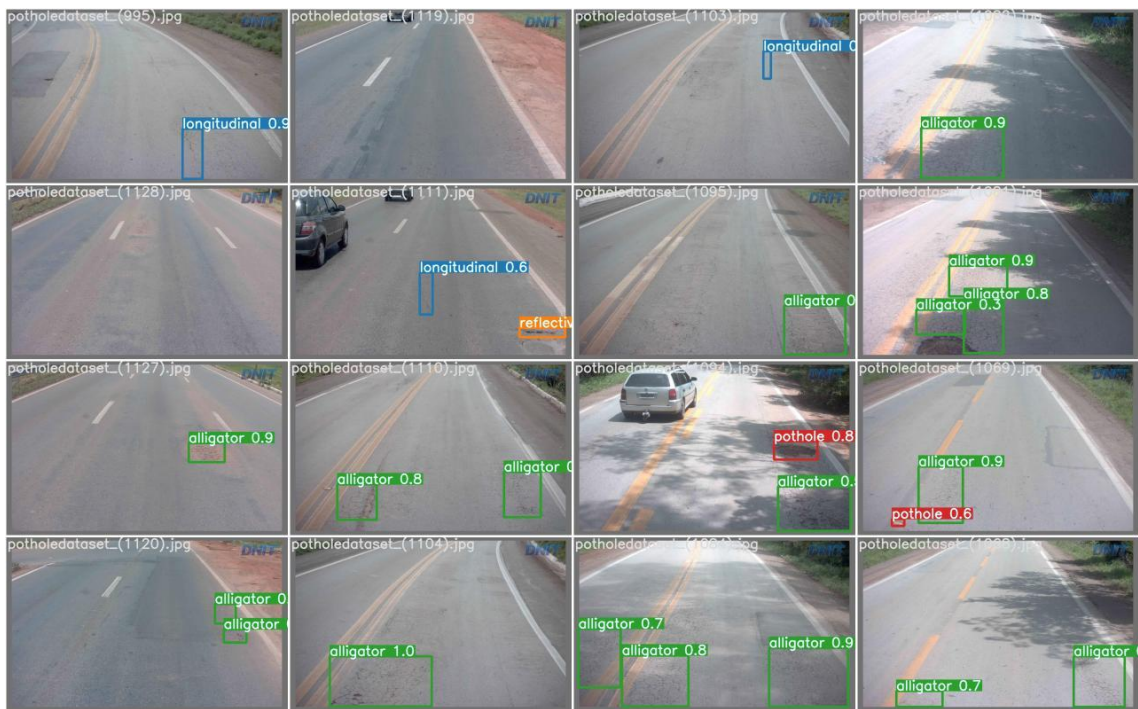


Figure 42. Confusion Matrix as a result of the YOLOv7 model for training

The actual labels and the predicted labels of a validation image are compared in Figure 43. Although the model successfully detected most defects, some images containing small size damages failed to produce accurate results.



(a) Ground truth of the validation image



(b) Predicted labels of the validation image

Figure 43. Comparison of true labels and predicted labels by YOLOv7 model

5.1.4. U-Net Crack Segmentation

The crack segmentation model is trained with the U-Net architecture to extract the area of the defects from the image. Two different pre-trained networks are used for this architecture: VGG16 and ResNet34. The performances obtained by testing the trained models with validation images are compared in Table 15.

Table 15. The performances of YOLOv7 models

Model	Number of Epochs	Loss Value	Accuracy	Training Time (h)
ResNet34	100	0.028 (training)	0.988 (training)	6.15
		0.122 (validation)	0.954 (validation)	
VGG16	100	0.034 (training)	0.986 (training)	7.31
		0.259 (validation)	0.937 (validation)	

There is not much difference between the training time of the two segmentation models. However, when training with the ResNet34 network, a model with 95.4% accuracy was obtained, which is higher than the VGG16 model. The loss values for training and validation of the crack segmentation model with ResNet34 are shown in Figure 44 with 100 epochs during training. The training accuracy and the validation accuracy of the U-Net segmentation model with ResNet34 are illustrated in Figure 45.

Both training and validation loss decreased up to 20 epochs. Thereafter, the training loss continues to decrease while the validation loss slightly increases. This is due to the overfitting of the model. It means that the model memorizes the training data and therefore cannot predict the test images well. However, a good result was obtained in the validation data with an accuracy of 0.95.

Test results of this model on untrained images from Crack500 and CFD datasets are compared in Table 16. As expected from the results, the model trained with ResNet34 is more sensitive to noise. So, it gave more a better result than VGG16.

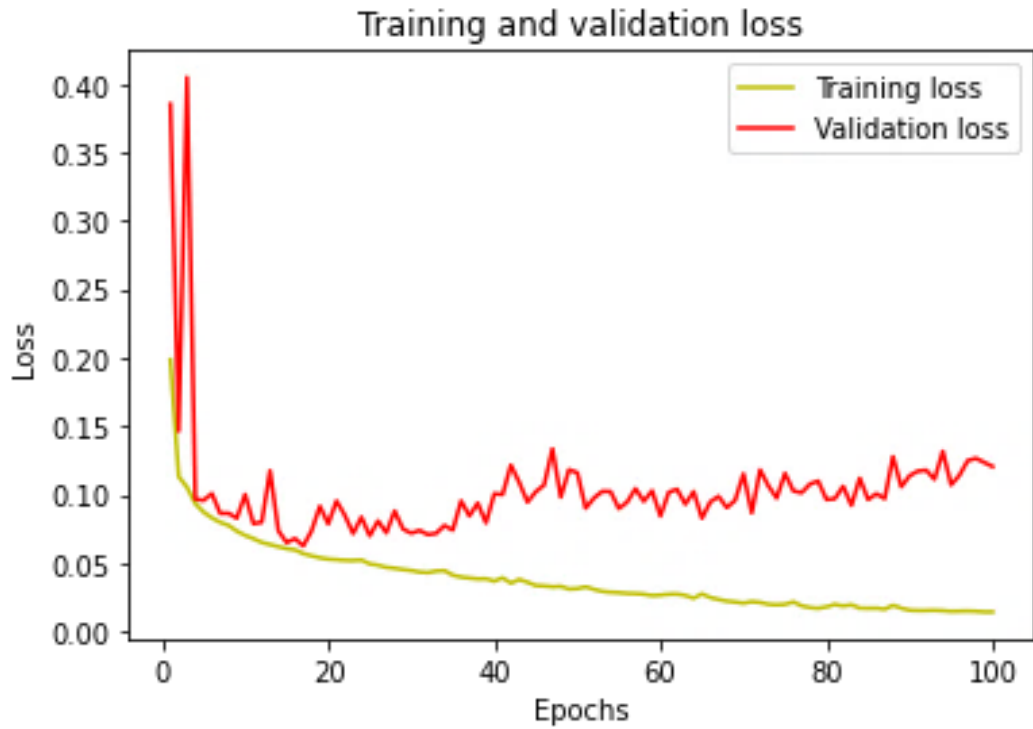


Figure 44. Loss values of the U-Net with ResNet34 crack segmentation model for training and validation

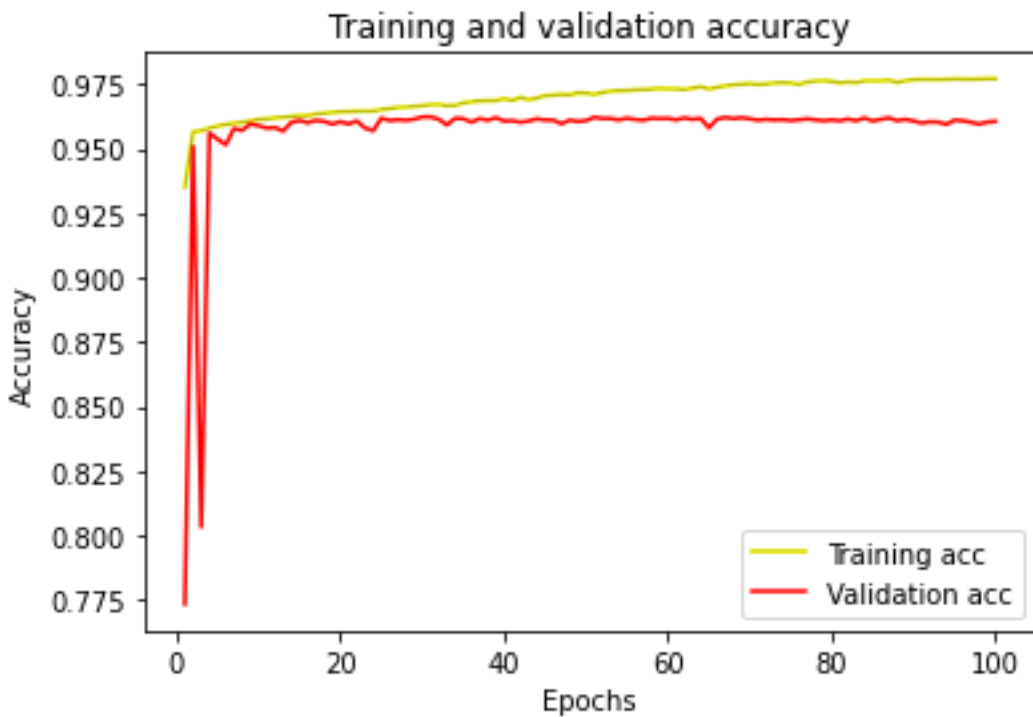



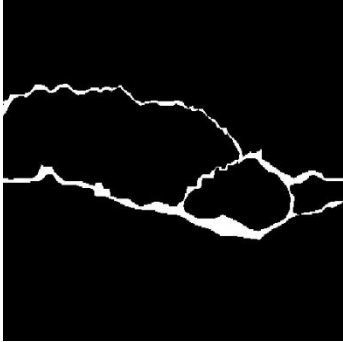
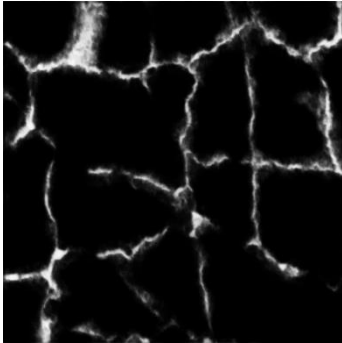
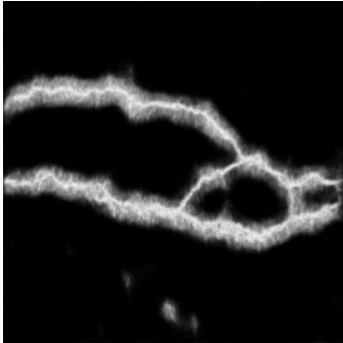
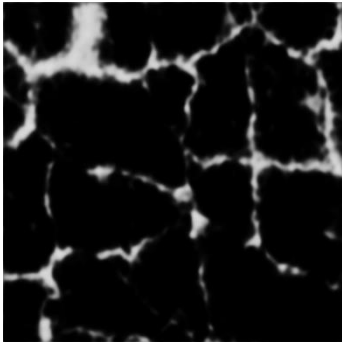
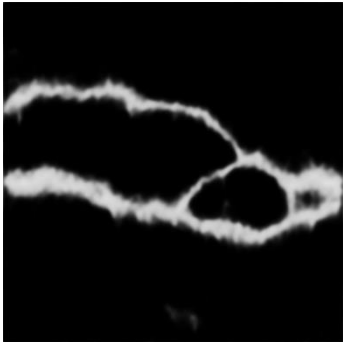


Figure 45. Accuracy of the U-Net with ResNet34 crack segmentation model for training and validation

Table 16. Comparison of U-Net Segmentation models

	CRACK500	CFD
Image		
Mask		
VGG16 Prediction		
ResNet34 Prediction		

5.2. Test Results

5.2.1. Defect Detection

The object detection model trained with the deep learning method was used to detect damage on the images taken by drone from the Beytepe Campus of Hacettepe University. The YOLOv7 model with the highest performance was selected for road damage assessment. First, the images from the region were manually labeled for four classes. Examples of labelled images can be found in Figure 46. The labels predicted by YOLOv7 model for this image are shown in Figure 47.

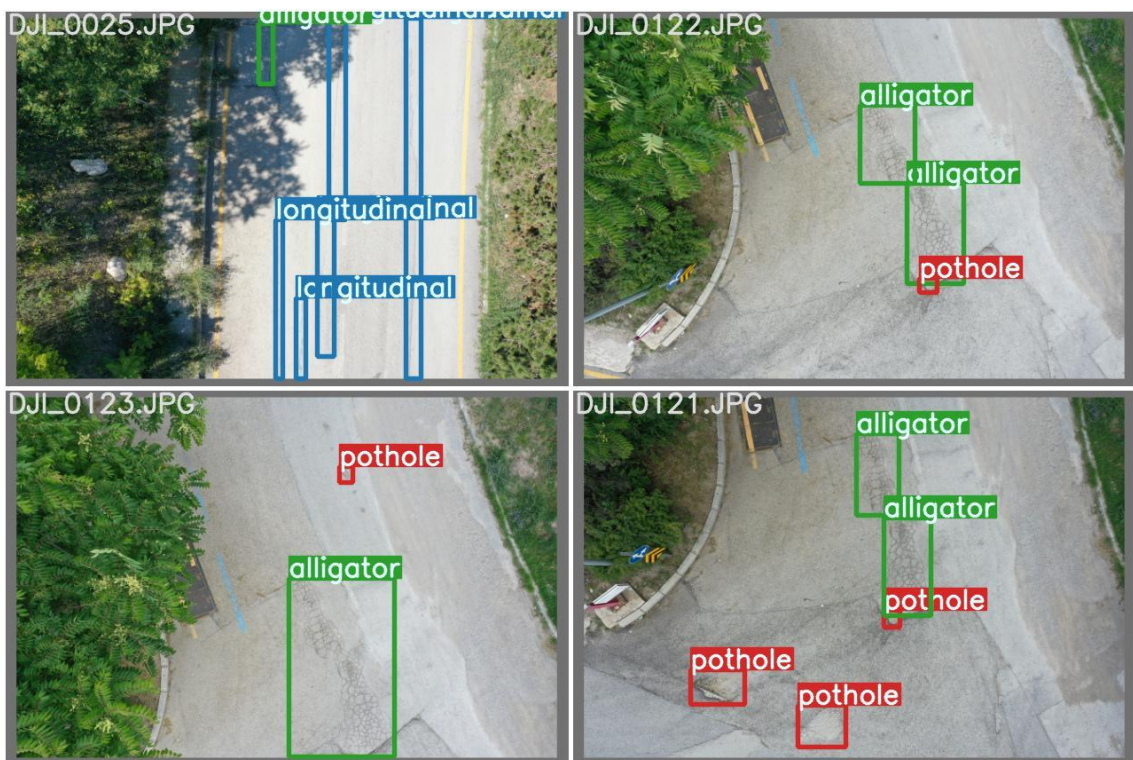


Figure 46. Labels for drone test image

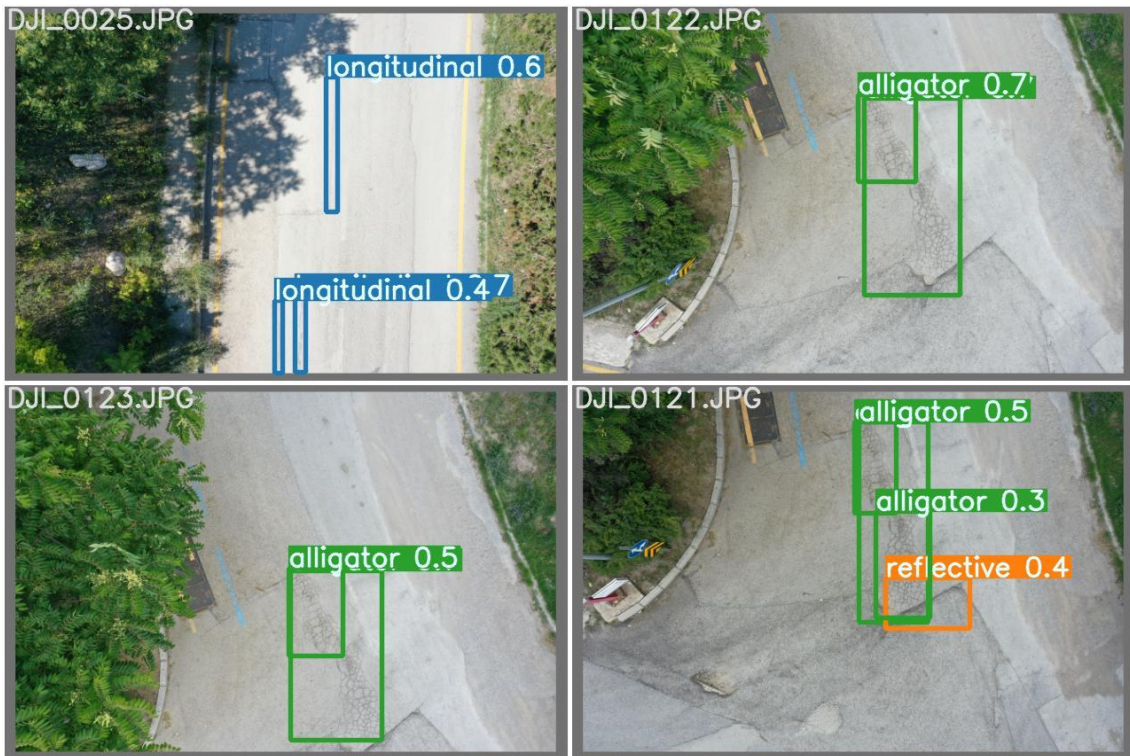


Figure 47. Prediction of drone test image with YOLOv7

In these predictions, it was found that the model could not recognize the potholes. This is because the model was trained with close-up images. Therefore, the pixels in the images taken by the drone have low resolution. So, the model scarcely distinguishes these pixels when making predictions. Another drawback of the model is that it could not detect damage in shadow areas. This is because the model was trained with less training data, with shadows and cracks in the road.

The results of each class for precision, recall, and mAP obtained by testing the YOLOv7 model with drone imagery are shown in Table 17. Although the mAP value of the model during training was 0.67, the mAP value during testing for the drone images was 0.42. This is due to the different angle of the images between the training data and the test data. The training images were taken from a vehicle, so it contains close-up images. However, since the images taken by the drone were taken from a greater distance, these images have the damage with a lower pixel resolution. For this reason, the trained model performed poorly in the test images.

Table 17. YOLOv7 test results for each class

Class	Precision	Recall	mAP
All Classes	0.51	0.45	0.42
Longitudinal Crack	0.60	0.43	0.44
Reflective Crack	0.37	0.43	0.34
Alligator Crack	0.55	0.47	0.48
Pothole	0.53	0.45	0.43

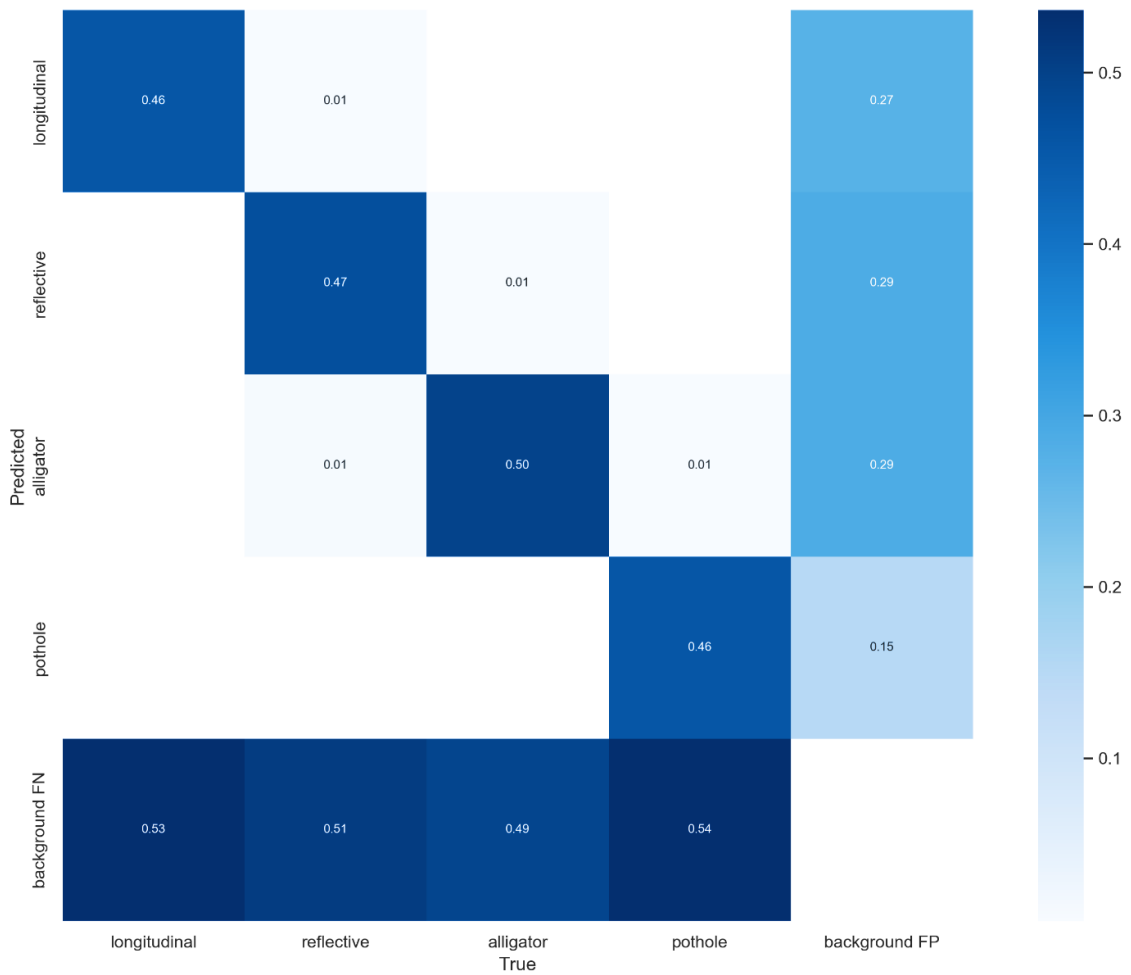


Figure 48. Confusion Matrix as a result of the YOLOv7 model for testing

The test performance of the YOLOv7 model can be examined for each class using the confusion matrix shown in Figure 48. The confusion matrix of the test images was created with a confidence score of 0.15 and an IoU threshold of 0.45. The model has the highest value of 0.5 for the class of alligator cracks as a result of the test. While the reflective cracks class has a success rate of 0.47, the potholes and longitudinal cracks classes have the lowest success rate of 0.46.

5.2.2. Segmentation of Defects

The damaged regions that are predicted by the model, i.e., bounding boxes drawn by the model, were recorded separately. To determine the severity of damages, damage areas were examined by segmentation model and the damage level for each image was determined.

The severity of damages was calculated using the reference limits described in Table 8. In this study, three classes were used to characterize the damage status of roads: Low, Medium, and High. Low damage is defined as areas that do not require maintenance and repair, while medium damage is defined as areas that require maintenance and repair. Defect classified as high describes areas where maintenance and repair urgently needed.

After determining the damaged areas with the YOLOv7 model, the bounding boxes drawn by the model around the damaged area were taken as separate images. To find the severity level of the damage, each damage image that was found by the YOLOv7 model was measured. MATLAB image processing tools were used to measure the width of the cracks and potholes in these images. The ground sampling distance for each image was calculated using an Excel spreadsheet created with the coordinates of the images. The crack widths calculated in pixels were converted to centimeters using this value. Then, the damage was classified into three classes based on reference limits: low, medium, and high.

These procedures were applied to the images of the Beytepe dataset, and a total of 528 damages were detected and labeled according to the severity of the damage. Table 18 shows the total number of detected defects according to their class and severity. Longitudinal cracks were mostly encountered in the region. A total of 271 longitudinal cracks were found, of which only 2 were classified as high. Similarly, the majority of reflective cracks were classified as low and only 3 were classified as high. The damage

class, whose maintenance and repair are urgent, is crocodile cracks with 20 damage. The least common damages class was potholes. A total of 15 potholes were found, of which only 1 had a high grade. Examples of damage with a high degree of damage are shown in Figure 46 for each class.

Table 18. The number of defects according to class and severity

Damage Class/Severity	Low	Middle	High	Total
Longitudinal Crack	249	20	2	271
Transverse Crack	97	14	3	114
Fatigue Cracks	68	40	20	128
Hole	14	0	1	15
Total	428	74	26	528

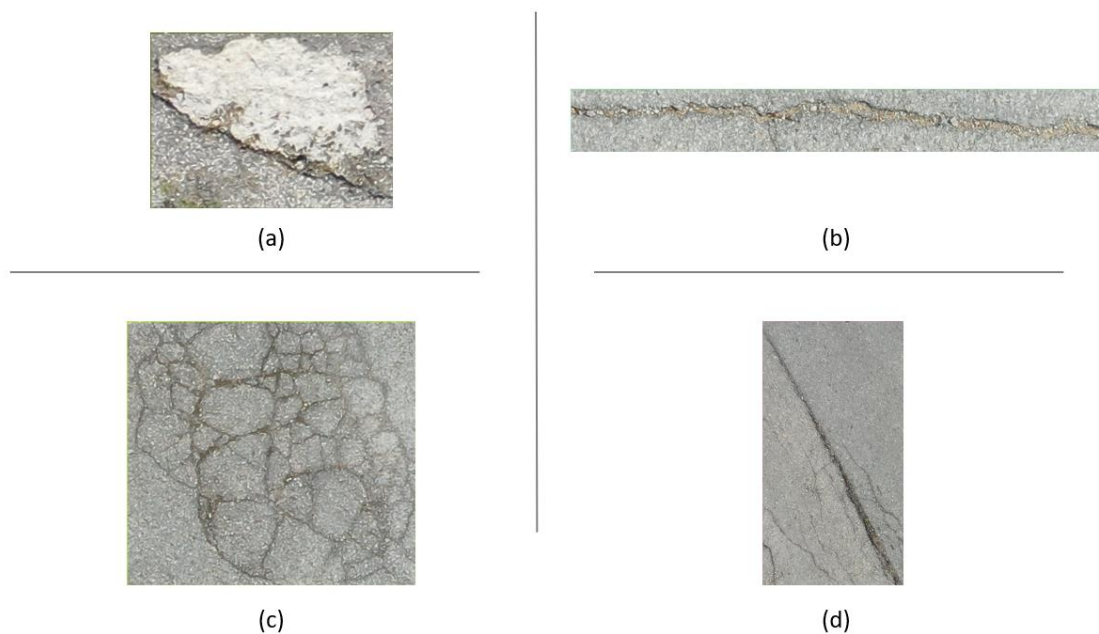


Figure 49. Examples of high degree a) pothole b) reflective crack c) alligator crack d) longitudinal crack

The distribution of these numbers by class and severity is illustrated in the diagram in Figure 50. As can be seen from the diagram, there are mainly longitudinal cracks in the region, which account for more than half of the total damage.

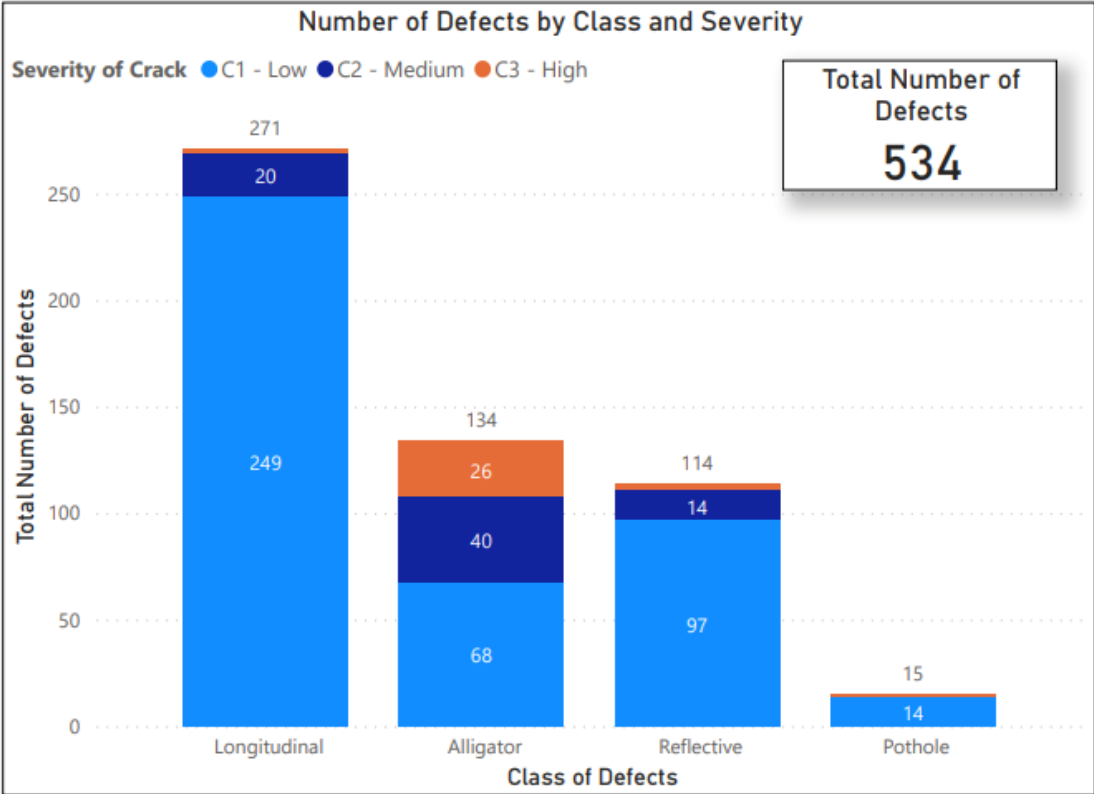


Figure 50. Distribution of the severity of defects according to class

This study attempts to quantify the area of damage regions using U-Net ResNet34 that is the highest performance crack segmentation model. This is significant to determine in advance the amount of material needed to repair road damage and to speed up this process.

The U-Net ResNet34 crack segmentation model was trained on close-up crack images and achieved the best results. However, since the test images were taken with a drone from a very high altitude, the damaged areas have low pixels. So, the crack segmentation model could not achieve success on these images. For that reason, the segmentation model was applied only to high-severity defects and the total area of this defects was subtracted.

Binary masks of the high-grade defects were obtained using the U-Net ResNet34 crack segmentation model. An example of this process can be found in Figure 51.

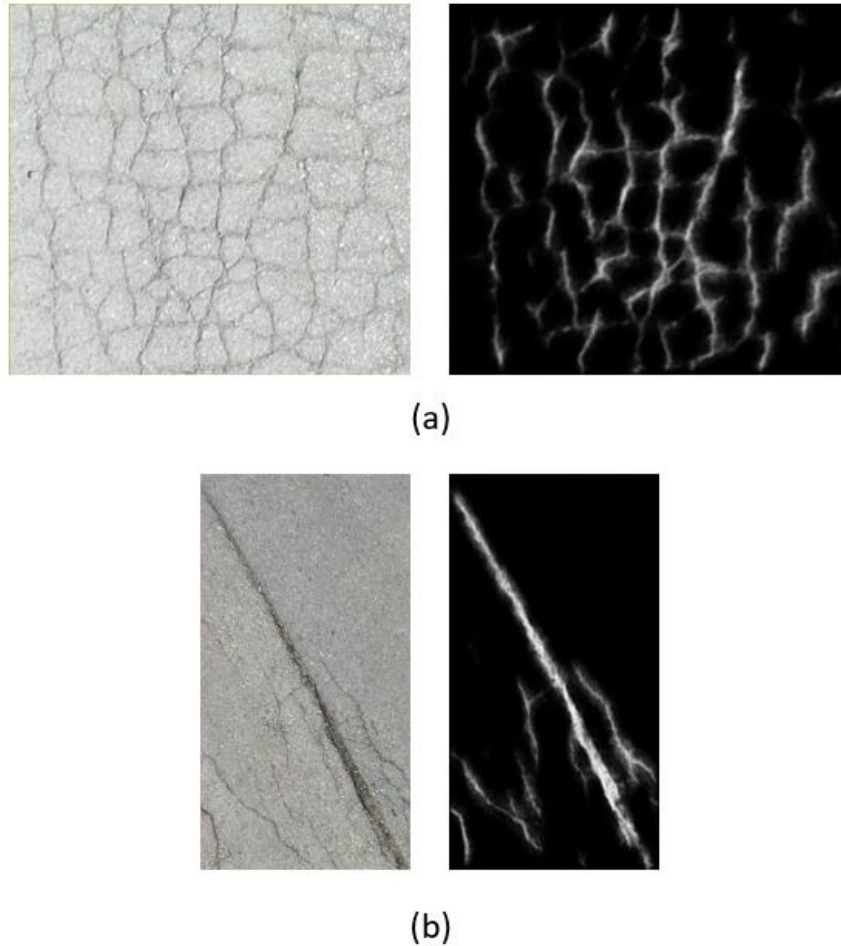


Figure 51. Example of segmentation of high-level a) Alligator crack b) Longitudinal Crack

To calculate the area of cracked regions, the white areas in the binary mask images were extracted as pixels. The white areas found in pixels were converted to square centimeters by multiplying the ground sampling distance of each image. After this analysis, the total area to be repaired was calculated as 3.87 square meters.

Although the total area of damaged areas could not be extracted by this method, a preliminary study was desirable by subtracting the area of high-grade damage. However, this model can be further improved by training the crack segmentation model with different and more data sets. Thus, the applicability with small pixels will increase.

5.3. Performance Evaluation

Following this study, an interactive map was created to show the severity of road damage and regional road conditions for visual interpretation. The map created with ArcGIS is explained in detail in this section.

In the dataset collected by the drone, each image also contains the coordinate information of the location where it was taken. Therefore, the coordinates of the images were inserted as points on the map using ArcGIS Pro software.

Damage was assessed individually considering the reference limits and then the severity was decided for each image. The coordinate points were coded in three colors according to the degree of damage of the images: green, yellow and red. The green points represent only the low damage region, the yellow points represent the low and medium damage region, and the red points represent the region with at least one high damage level.

A total of 66 images of roads on the Beytepe campus of Hacettepe College were used to assess the damage and were labeled according to the severity. As shown in Figure 52, these images were mapped to the point where the coordinate information is located. It is expected that this map, which contains information about road conditions, will be of great help to road maintenance and repair organizations. This map has been released through the link below.

<https://www.arcgis.com/home/item.html?id=a4901233b6e04fbabbb4b1ccd03c7886>

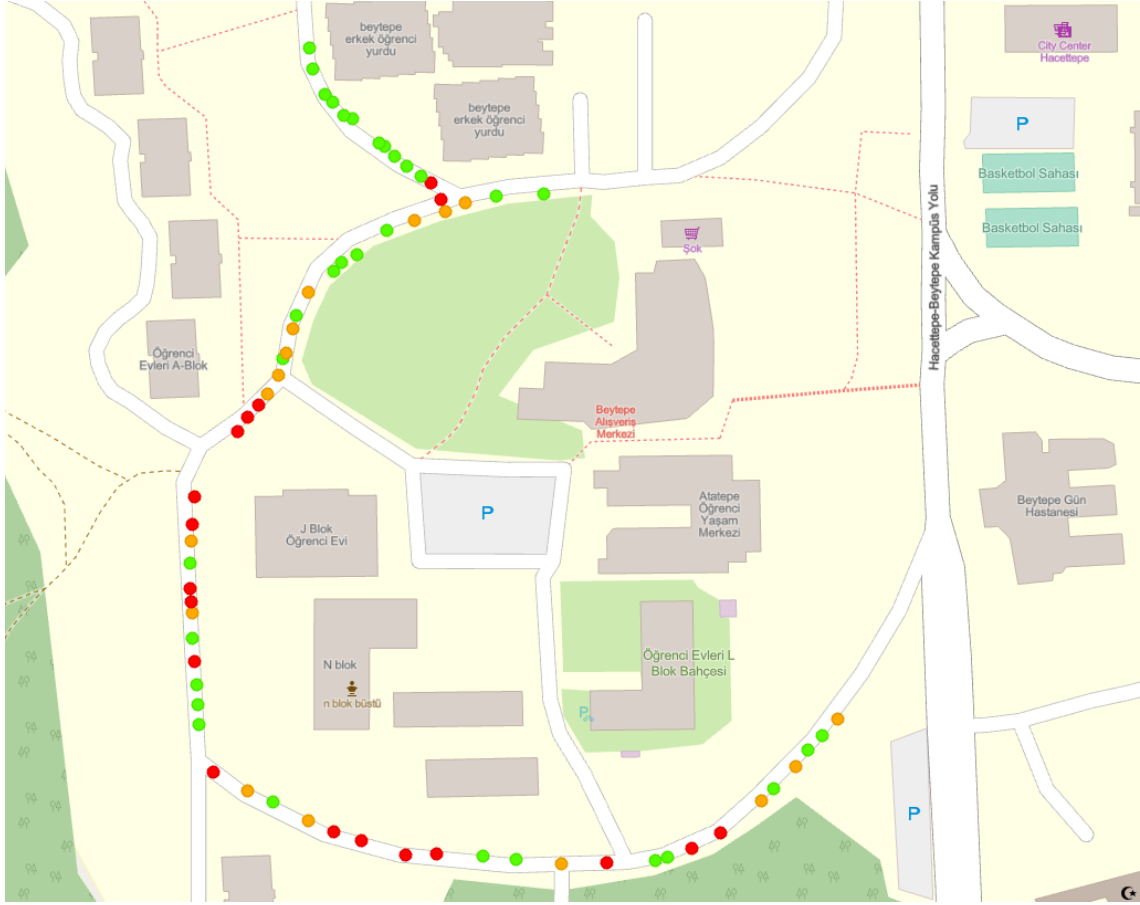


Figure 52. Road condition map of Beytepe Campus of Hacettepe University

5.4. Discussion/Comparison

To select the best model for defect detection, the evaluation metrics were monitored. In terms of mAP, the Faster R-CNN models gave significantly worse results than the YOLO models. Moreover, despite the small size of the model, the training time is longer compared to the YOLO model. Since the training time of the Faster R-CNN model is longer and the mean precision value (mAP) is lower than the YOLO models, it was decided that the use of this model is not efficient.

It was found that the YOLOv5 models were superior to the Faster R-CNN model in terms of model efficiency, prediction success, and code readability. In both training time and testing time, the YOLOv5 models achieved great success. The YOLOv5 models were found to be faster than the Faster R-CNN model, especially in terms of training time.

From the comparison of the YOLO architectures, it can be concluded that the results of the YOLOv7 models outweigh those of the YOLOv5 models. The YOLOv7 architectures

have the same speed as the YOLOv5 model in terms of training time. However, the average precision value was more successful than them. In addition, the mAP values are much higher than the Faster R-CNN and YOLOv5 models. For these reasons, YOLOv7 was selected for the application of the test images.

From the confusion matrices it can be deduced that the main problem of the proposed models seems to be that many objects are not recognized, i.e., they are considered as background.

Regarding the segmentation models, the training times of the two segmentation models did not differ significantly. In contrast, after training with the ResNet34 network, a model with higher accuracy was produced that outperformed the VGG16 model. For this reason, the U-Net with the ResNet34 crack segmentation model was selected for use with drone imagery.

6. CONCLUSION

In this study, the damage types were classified considering the most common deformations on highways in Turkey and the labelling was performed in different formats to be used in multiple algorithms. Moreover, image augmentation was applied to the dataset to increase the accuracy of the model.

Following this, different deep learning models were trained with tagged images and the performance of these models was compared using numerous evaluation metrics. The weights of the YOLOv7 model with the highest performance were recorded and tested at the Beytepe campus of Hacettepe University, which was selected as the pilot region for damage assessment.

A preliminary study was conducted to determine the condition of the roads on the Beytepe campus of Hacettepe University, which is the pilot area. Thanks to the information obtained through the preliminary study, an effective drone flight was conducted. Owing to the large area of Hacettepe University's Beytepe campus, the desired quantity and quality of images could be achieved by flying over the main roads.

It was found that damage assessment can be easily performed using photos taken by high-resolution cameras using UAVs. In the future, it is planned to conduct this study in a larger area. As a result of this study, it was found that the detection of damage using drone images is possible in larger areas.

Subsequently, the predicted road damage was used to determine the usability of the road. Therefore, a new damage assessment guide was created that was compatible with the model used. Thus, the current condition of the roads on the Beytepe Campus of Hacettepe University could be determined. Defect was detected using deep learning models, and according to this determination, the severity of damage was calculated manually on all images. Unfortunately, a one-to-one spatial inference of all detected damages could not be made. However, spatial inferences could be made for images with a high degree of damage.

Finally, an ArcGIS-based map was created to show the condition of the studied roads as a visual interpretation. This map can be used as a reference for organizations responsible for assessing road conditions and can be used prior to road maintenance and repair. It is

expected that the results of this study will lead to innovations and technology adaptations in the field of road damage maintenance.

The goal is to contribute to the literature by combining object detection and segmentation tasks in the field of road condition assessment.

In the future, this study can be applied to a wide range of fields. In addition, the drone videos could be analyzed for defect detection in real time using object detection models.

7. REFERENCES

- Abdel-Qader, L., Abudayyeh, O., Kelly, M.E., 2003. Analysis of edge-detection techniques for crack identification in bridges. *J. Comput. Civ. Eng.* 17, 255–263. [https://doi.org/10.1061/\(Asce\)0887-3801\(2003\)17:4\(255\)](https://doi.org/10.1061/(Asce)0887-3801(2003)17:4(255))
- Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A.E., Arshad, H., 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, e00938. <https://doi.org/10.1016/j.heliyon.2018.e00938>
- Akagic, A., Buza, E., Omanovic, S., Karabegovic, A., 2018. Pavement crack detection using Otsu thresholding for image segmentation. 2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc. 1092–1097. <https://doi.org/10.23919/MIPRO.2018.8400199>
- Amhaz, R., Chambon, S., Idier, J., Baltazart, V., 2016. Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection. *IEEE Trans. Intell. Transp. Syst.* 17, 2718–2729. <https://doi.org/10.1109/tits.2015.2477675>
- Barlow, H.B., 1989. Unsupervised Learning. *Neural Comput.* 1, 295–311. <https://doi.org/10.1162/neco.1989.1.3.295>
- Biau, G., Scornet, E., 2016. A random forest guided tour. *Test* 25, 197–227. <https://doi.org/10.1007/s11749-016-0481-7>
- Biewald, L., 2020. Experiment Tracking with Weights & Biases. Softw. available from wandb. com 1–5.
- Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection.
- Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O., 2018. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput. Civ. Infrastruct. Eng.* 33, 731–747. <https://doi.org/10.1111/mice.12334>
- Chen, J., Zhang, D., Huang, H., Shadabfar, M., Zhou, M., Yang, T., 2020. Image-based

segmentation and quantification of weak interlayers in rock tunnel face via deep learning. *Autom. Constr.* 120, 103371. <https://doi.org/10.1016/j.autcon.2020.103371>

Christopher J.C. Burges, 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* 2, 121–167.

Cubero-Fernandez, A., Rodriguez-Lozano, F.J., Villatoro, R., Olivares, J., Palomares, J.M., 2017. Efficient pavement crack detection and classification. *Eurasip J. Image Video Process.* 2017. <https://doi.org/ARTN 3910.1186/s13640-017-0187-0>

Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., Kang, H., 2021. Pavement distress detection and classification based on YOLO network. *Int. J. Pavement Eng.* 22, 1659–1672. <https://doi.org/10.1080/10298436.2020.1714047>

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., Gross, H., 2017. How to get pavement distress detection ready for deep learning? A systematic approach, in: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2039–2047. <https://doi.org/10.1109/IJCNN.2017.7966101>

Fan, Z., Li, C., Chen, Y., Di Mascio, P., Chen, X.P., Zhu, G.J., Loprencipe, G., 2020. Ensemble of Deep Convolutional Neural Networks for Automatic Pavement Crack Detection and Measurement. *Coatings* 10, 152. <https://doi.org/ARTN 15210.3390/coatings10020152>

Fan, Z., Wu, Y., Lu, J., Li, W., 2018. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network.

Gavilán, M., Balcones, D., Marcos, O., Llorca, D.F., Sotelo, M.A., Parra, I., Ocaña, M., Aliseda, P., Yarza, P., Amírola, A., 2011. Adaptive road crack detection system by pavement classification. *Sensors* 11, 9628–9657. <https://doi.org/10.3390/s111009628>

Girshick, R., 2015. Fast R-CNN. *Proc. IEEE Int. Conf. Comput. Vis. 2015 Inter*, 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>

Gopalakrishnan, K., Khaitan, S.K., Choudhary, A., Agrawal, A., 2017. Deep Convolutional Neural Networks with transfer learning for computer vision-based

- data-driven pavement distress detection. *Constr. Build. Mater.* 157, 322–330.
<https://doi.org/10.1016/j.conbuildmat.2017.09.110>
- He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition.
<https://doi.org/10.48550/ARXIV.1512.03385>
- Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B., 1998. Support vector machines. *IEEE Intell. Syst. their Appl.* 13, 18–28.
<https://doi.org/10.1109/5254.708428>
- Huidrom, L., Das, L.K., Sud, S.K., 2013. Method for Automated Assessment of Potholes, Cracks and Patches from Road Surface Video Clips. *Procedia - Soc. Behav. Sci.* 104, 312–321. <https://doi.org/10.1016/j.sbspro.2013.11.124>
- Ibragimov, E., Lee, H.J., Lee, J.J., Kim, N., 2020. Automated pavement distress detection using region based convolutional neural networks. *Int. J. Pavement Eng.* 0, 1–12.
<https://doi.org/10.1080/10298436.2020.1833204>
- Indyk, P., Motwani, R., 1998. Approximate nearest neighbors 604–613.
<https://doi.org/10.1145/276698.276876>
- Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F., 2020. A Survey on Contrastive Self-Supervised Learning. *Technologies* 9, 2.
<https://doi.org/10.3390/technologies9010002>
- Kien, B.H., Iba, D., Ishii, Y., Tsutsui, Y., Miura, N., Iizuka, T., Masuda, A., Sone, A., Moriwaki, I., 2019. Crack detection of plastic gears using a convolutional neural network pre-learned from images of meshing vibration data with transfer learning. *Forsch. im Ingenieurwesen/Engineering Res.* 83, 645–653.
<https://doi.org/10.1007/s10010-019-00354-5>
- Koch, C., Brilakis, I., 2011. Pothole detection in asphalt pavement images. *Adv. Eng. Informatics* 25, 507–515. <https://doi.org/10.1016/j.aei.2011.01.002>
- Kotsiantis, S.B., 2013. Decision trees: A recent overview. *Artif. Intell. Rev.* 39, 261–283.
<https://doi.org/10.1007/s10462-011-9272-4>
- LeCun, Y., Bengio, Y., 1995. Convolutional networks for images, speech, and time series. *Handb. brain theory neural networks* 3361, 255–258.

- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2323. <https://doi.org/10.1109/5.726791>
- Li, D., Duan, Z., Hu, X., Zhang, D., 2021. Pixel-Level Recognition of Pavement Distresses Based on U-Net. *Adv. Mater. Sci. Eng.* 2021. <https://doi.org/10.1155/2021/5586615>
- Li, H., Song, D., Liu, Y., Li, B., 2019. Automatic Pavement Crack Detection by Multi-Scale Image Fusion. *IEEE Trans. Intell. Transp. Syst.* 20, 2025–2036. <https://doi.org/10.1109/TITS.2018.2856928>
- Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- Liu, B., 2011. Web Data Mining. *Web Data Min.* 63–64. <https://doi.org/10.1007/978-3-642-19460-3>
- Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V.C.S., Ding, L., 2020. Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Comput. Civ. Infrastruct. Eng.* 35, 1291–1305. <https://doi.org/10.1111/mice.12622>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single shot multibox detector, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
- Liu, Y.H., Yao, J., Lu, X.H., Xie, R.P., Li, L., 2019. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing* 338, 139–153. <https://doi.org/10.1016/j.neucom.2019.01.036>
- Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H., 2018. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Comput. Civ. Infrastruct. Eng.* 33, 1127–1141. <https://doi.org/10.1111/mice.12387>
- Majidifard, H., Jin, P., Adu-Gyamfi, Y., Buttlar, W.G., 2020. Pavement Image Datasets:

- A New Benchmark Dataset to Classify and Densify Pavement Distresses. *Transp. Res. Rec.* 2674, 328–339. <https://doi.org/Artn036119812090728310.1177/0361198120907283>
- Marques, A.G.C.S., 2012. Automatic road pavement crack detection using SVM. Lisbon, Port. Diss. Master Sci. Degree Electr. Comput. Eng. Inst. Super. Técnico.
- Munawar, H.S., Hammad, A.W.A., Haddad, A., Soares, C.A.P., Waller, S.T., 2021. Image-based crack detection methods: A review. *Infrastructures* 6, 1–20. <https://doi.org/10.3390/infrastructures6080115>
- Oliveira, H., Correia, P.L., 2013. Automatic Road Crack Detection and Characterization. *IEEE Trans. Intell. Transp. Syst.* 14, 155–168. <https://doi.org/10.1109/Tits.2012.2208630>
- Oliveira, H., Correia, P.L., 2009. Automatic road crack segmentation using entropy and image dynamic thresholding. *Eur. Signal Process. Conf.* 622–626.
- Park, H.S., Jun, C.H., 2009. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* 36, 3336–3341. <https://doi.org/10.1016/j.eswa.2008.01.039>
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2016-Decem, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. <https://doi.org/10.48550/arxiv.1505.04597>
- Shi, Y., Cui, L.M., Qi, Z.Q., Meng, F., Chen, Z.S., 2016. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* 17, 3434–3445. <https://doi.org/10.1109/Tits.2016.2552248>
- Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints arXiv:1409.1556*.
- Song, W.D., Jia, G.H., Jia, D., Zhu, H., 2019. Automatic Pavement Crack Detection and

- Classification Using Multiscale Feature Attention Network. *IEEE Access* 7, 171001–171012. <https://doi.org/10.1109/Access.2019.2956191>
- Spencer, B.F., Hoskere, V., Narazaki, Y., 2019. Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. *Engineering* 5, 199–222. <https://doi.org/10.1016/j.eng.2018.11.030>
- Subirats, P., Dumoulin, J., Legeay, V., Barba, D., 2006. Automation of Pavement Surface Crack Detection using the Continuous Wavelet Transform. <https://doi.org/10.1109/ICIP.2006.313007>
- Szepesvári, C., 2010. Algorithms for reinforcement learning, *Synthesis Lectures on Artificial Intelligence and Machine Learning*. <https://doi.org/10.2200/S00268ED1V01Y201005AIM009>
- Tanaka, N. and Uematsu, K., 1998. A Crack Detection Method in Road Surface Images Using Morphology. *IAPR Work. Mach. Vis. Appl.* 154–157.
- Tsai, Y.C., Kaul, V., Mersereau, R.M., 2010. Critical Assessment of Pavement Distress Segmentation Methods. *J. Transp. Eng.* 136, 11–19. [https://doi.org/10.1061/\(Asce\)Te.1943-5436.0000051](https://doi.org/10.1061/(Asce)Te.1943-5436.0000051)
- Vrochidou, E., Sidiropoulos, G.K., Ouzounis, A.G., Lampoglou, A., Tsimperidis, I., Papakostas, G.A., Sarafis, I.T., Kalpakis, V., Stamkos, A., 2022. Towards Robotic Marble Resin Application: Crack Detection on Marble Using Deep Learning. *Electronics* 11, 3289. <https://doi.org/10.3390/electronics11203289>
- Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. <https://doi.org/10.48550/arXiv.2207.02696>
- Wright, R.E., 1995. Logistic regression., in: *Reading and Understanding Multivariate Statistics*. American Psychological Association, Washington, DC, US, pp. 217–244.
- Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M., 2017. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. *34th Int. Conf. Mach. Learn. ICML 2017* 8, 5888–5901.
- Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., Ling, H., 2020. Feature Pyramid and

- Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* 21, 1525–1535. <https://doi.org/10.1109/tits.2019.2910595>
- Zhang, A., Wang, K.C.P., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J.Q., Li, B., 2018. Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. *J. Comput. Civ. Eng.* 32, 1–14. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000775](https://doi.org/10.1061/(asce)cp.1943-5487.0000775)
- Zhang, A., Wang, K.C.P., Li, B.X., Yang, E.H., Dai, X.X., Peng, Y., Fei, Y., Liu, Y., Li, J.Q., Chen, C., 2017. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. *Comput. Civ. Infrastruct. Eng.* 32, 805–819. <https://doi.org/10.1111/mice.12297>
- Zhang, L., Yang, F., Zhang, Y., Zhu, Y., 2016. Road crack detection using deep convolutional neural network. <https://doi.org/10.1109/ICIP.2016.7533052>
- Zhao, H., Qin, G., Wang, X., 2010. Improvement of canny algorithm based on pavement edge detection. *Proc. - 2010 3rd Int. Congr. Image Signal Process. CISP 2010 2*, 964–967. <https://doi.org/10.1109/CISP.2010.5646923>
- Zhu, J., Zhong, J., Ma, T., Huang, X., Zhang, W., Zhou, Y., 2022. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* 133, 103991. <https://doi.org/10.1016/j.autcon.2021.103991>
- Zhu, S., Xia, X., Zhang, Q., Belloulata, K., 2007. An image segmentation algorithm in image processing based on threshold segmentation. *Proc. - Int. Conf. Signal Image Technol. Internet Based Syst. SITIS 2007* 673–678. <https://doi.org/10.1109/SITIS.2007.116>
- Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S., 2012. CrackTree: Automatic crack detection from pavement images. *Pattern Recognit. Lett.* 33, 227–238. <https://doi.org/10.1016/j.patrec.2011.11.004>

APPENDIX

Appendix 1 -Criteria for Road Damage Assessment

Distress Type	OREGON (2019)		PENNSYLVANIA (2019)		FLORIDA		WASHINGTON	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
ALLIGATOR CRACKING	Low	An area of cracks with no or only a few connecting cracks. Cracks are not spalled or sealed. No pumping is evident. Includes Wheel Path Longitudinal Cracks.	Low	Average Crack Width \leq hairline (Fatigue cracking consisting of only longitudinal cracks in the wheel path.)	Low	Fine, longitudinal hairline cracks running parallel to each other with no or only a few interconnecting cracks.	Low	Branched, longitudinal, discontinuous thin cracks are beginning to interconnect and form the typical alligator pattern with no spalling.
	Moderate	An area of interconnected cracks forming a complete pattern. Cracks may be slightly spalled or sealed. No pumping is evident.	Medium	Average Crack Width $>$ hairline and \leq 0.25 in (Fatigue cracking consisting of longitudinal and interconnecting cracks typically forming a diamond shaped, chicken wire or alligator's hide pattern.)	Medium	Further development of light alligator cracking into a pattern or network of cracks that may be slightly spalled. Well-defined pattern of interconnecting cracks, where all pieces are firmly held in place.	Medium	Cracking is completely interconnected and has fully developed an alligator pattern. Some spalling may appear at the edges of cracks. The cracks may be greater than 1/4-inch wide, but the pavement pieces are still in place.
	High	An area of moderately or severely spalled interconnected cracks forming a complete pattern. Pieces may move when subjected to traffic. Cracks may be sealed. Pumping may be evident.	High	Average Crack Width $>$ 0.25 in (Fatigue cracking consisting of longitudinal and interconnecting cracks typically forming a diamond shaped, chicken wire or alligator's hide pattern.)	High	Network or pattern cracking has developed so that pieces are well-defined and spalled at the edges; some of the pieces are loose and rock under traffic.	High	The pattern of cracking is well developed. Spalling is very apparent at the crack. Individual pieces may be loosened and may rock under traffic. Pieces may be missing. Pumping of fines up through the cracks may be evident.

Distress Type	CALIFORNIA		INDIANA		NORTH CAROLINA		BRITISH COLUMBIA (2016)		ONTARIO (2016)	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
ALLIGATOR CRACKING	Low	average crack width <1/4 inch or crack sealed with the sealant in good condition and the width cannot be determined	Low	Cracks not spalled or sealed, pumping not evident, cracks not interconnected.	Light	Longitudinal disconnected hairline cracks about 1/8 inch wide running parallel to each other; initially may be only a single crack in the wheel path or edge of pavement but could also look like an alligator pattern.	Low	N/A	Very slight	Multiple cracks begin to develop short interconnecting cracks and cause alligator pattern forming. May include depression up to 6 mm. Crack width is up to 3mm.
									Slight	Alligator pattern established with block corners fracturing. May include depression 7-12 mm. Crack width is between 3 and 12mm.
	Medium	1/4 inch ≤ average crack width ≤ 1/2 inch	Moderate	Cracks form interconnected area, slight spalling, cracks may be sealed, pumping not evident.	Moderate	Longitudinal cracks in wheel path(s) or edge of pavement forming an alligator pattern; cracks may be lightly spalled and are about 1/4 inch wide.	Moderate	Interconnected cracks forming a complete block pattern; slight spalling and no pumping	Moderate	Alligator pattern established with spalling of blocks. May include depression 13-19 mm. Crack width is between 13 and 19mm.
	High	average crack width >1/2 inch	High	Area of moderately or severely spalled interconnected cracks forming a pattern, pieces dislodged, cracks may be sealed, and pumping may be evident.	Severe	Cracking has progressed so that pieces appear loose with severely spalled edges; cracks are about 3/8 to 1/2 inch wide or greater; potholes may be present.	High	Interconnected cracks forming a complete block pattern, moderate to severe spalling, pieces may move, and pumping may exist.	Severe	Blocks begin to lift. Small potholes from missing blocks. May include depression 20-25 mm. Crack width is between 20 and 25mm.
									Very Severe	Polygon blocks lifting. Potholes from missing blocks. May include depression greater than 25 mm. Crack width is greater 25mm.

Distress Type	OREGON (2019)		PENNSYLVANIA (2019)		FLORIDA		WASHINGTON	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
LONGITUDINAL CRACKING	Low	Crack widths < 0.125", no spalling, and no measurable faulting; or well-sealed and with a width that cannot be determined	Low	Average crack width > hairline and ≤ 0.25 in wide	Low	General condition: 1/ non-Filled, mean < 1/4" 2/ Filled: any Width,	Low	The cracks have very little or no spalling along the edges and are less than 1/4-inch in width. If the cracks are sealed and the width of the crack prior to sealing is invisible, they should be classified as Low Severity.
	Moderate	Crack widths ≥ 0.125" and < 0.5"; or with spalling < 3"; or faulting up to 0.5".	Medium	Average crack width ≤ 0.25 in wide, spalling ≥ 2.0 in wide for ≤ 50% length	Medium	General condition: 1/ Any Width, 2/ If Filled crack, 3/ If Non-filled, mean >1/4" 4/ Light cracking appears near the crack or at the corner of intersecting cracks.	Medium	The cracks have little or no spalling, but they are greater than 1/4-inch in width. There may be a few randomly spaced low severity connecting cracks near the main crack or at the corners of intersecting cracks.
	High	Crack widths ≥ 0.5"; or with spalling ≥ 3"; or faulting ≥ 0.5"	High	Average crack width > 0.25 in wide, spalling ≥ 2.0 in wide for > 50% length	High	Can be any Width	High	Cracks are spalled and there may be several randomly spaced cracks near the main crack or at the corners of intersecting cracks. Pieces are visibly missing along the crack. At some point, this longitudinal cracking becomes alligator cracking.

Distress Type	CALIFORNIA		INDIANA		NORTH CAROLINA		BRITISH COLUMBIA (2016)		ONTARIO (2016)	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
LONGITUDINAL CRACKING	Low	average crack width <1/4 inch or crack sealed with the sealant in good condition and the width cannot be determined	Low	Unsealed crack with mean width ≤ 0.25 in. Sealed crack with sealant in good condition.	N/A	Low	Single cracks with no spalling; mean unsealed crack width < 5mm	Very slight	Single crack less than 3 mm.	
								Slight	Single crack from 3 mm to 12 mm.	
	Medium	1/4 inch ≤ average crack width ≤ 1/2 inch	Moderate	Crack with mean width > 0.25 in. and ≤ 0.75 in. Crack with mean width ≤ 0.75 in. and adjacent random cracking at low severity levels.		Medium	Single or multiple cracks; moderate spalling; mean unsealed crack width 5-20mm	Moderate	13 mm to 19 mm width for single cracks, or multiple cracks starting.	
									High	average crack width >1/2 inch
	Very Severe	Greater than 25 mm wide for single cracks, or multiple cracks with spalling developed. May begin to alligator.								

Distress Type	OREGON (2019)		PENNSYLVANIA (2019)		FLORIDA		WASHINGTON	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
TRANSVERSE CRACKING	Low	An unsealed crack with a mean width of $\leq 0.25''$; or a sealed crack with sealant material in good condition and the width cannot be determined.	Low	Average Crack Width $>$ hairline and ≤ 0.25 in	Low	General condition: 1/ non-Filled, mean $< 1/4''$ 2/ Filled: any Width,	Low	The cracks have very little or no spalling along the edges and are less than 1/4-inch in width. If the cracks are sealed and the width of the crack prior to sealing is invisible, they should be classified as Low Severity.
	Moderate	Any crack with a mean width $> 0.25''$ and $\leq 0.75''$; or any crack with a mean width $< 0.75''$ in and adjacent low severity random cracking.	Medium	Average Crack Width > 0.25 in and ≤ 0.5 in	Medium	General condition: 1/ Any Width, 2/ If Filled crack, 3/ If Non-filled, mean $> 1/4''$ 4/ Light cracking appears near the crack or at the corner of intersecting cracks.	Medium	The cracks have little or no spalling, but they are greater than 1/4-inch in width. There may be a few randomly spaced low severity connecting cracks near the main crack or at the corners of intersecting cracks.
	High	Any crack with a mean width $> 0.75''$; or any crack with a mean width $\leq 0.75''$ and adjacent moderate to high severity random cracking.	High	Average Crack Width > 0.5 in	High	Can be any Width	High	Cracks are spalled and there may be several randomly spaced cracks near the main crack or at the corners of intersecting cracks. Pieces are visibly missing along the crack.

Distress Type	CALIFORNIA		INDIANA		NORTH CAROLINA		BRITISH COLUMBIA (2016)		ONTARIO (2016)	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
TRANSVERSE CRACKING	Low	average crack width <1/4 inch	Low	Unsealed crack with mean width ≤ 0.25 in. Sealed crack with sealant in good condition.	Light	Cracks, usually only transverse, are less than 1/4 inch wide and are not spalled; block pattern may not be visible yet; transverse cracks usually 10 to 20 feet apart. Cracks have little or no spalling and joints are usually not bumped up.	Low	Single cracks with no spalling; mean unsealed crack width < 5mm	Very slight	Less than 3 mm single crack.
									Slight	Single crack 3 mm to 12 mm.
	Medium	1/4 inch ≤ average crack width ≤ 1/2 inch or crack sealed with the sealant in good condition and the width cannot be determined	Moderate	Crack with mean width > 0.25 in. and ≤ 0.75 in. Crack with mean width ≤ 0.75 in. and adjacent random cracking at low severity levels.	Moderate	Block pattern may be visible with blocks 10 square feet or greater present; cracks are 1/4 inch to 1/2 inch wide; cracks may or may not be spalled; transverse cracks usually 5 to 20 feet apart; joints may be bumped up 1/2 inch over concrete.	Medium	Single or multiple cracks; moderate spalling; mean unsealed crack width 5-20mm	Moderate	13 mm to 19 mm single crack, or multiple cracks even if crack opening is less than 13mm. Cracks starting to develop cupping or lipping, barely noticeable bump.
	High	average crack width >1/2 inch	High	Crack with mean width > 0.75 in. Crack with mean width ≤ 0.75 in., adjacent random cracking, and moderate or high severity levels.	Severe	Cracks may be severely spalled with smaller blocks 2 to 10 square feet present; cracks usually greater than 1/2 inch wide; transverse cracks may be 1 to 2 feet apart throughout portions of the surface; cracks may be bumped up more than 1/2 inch.	High	Single or multiple cracks; severe spalling; mean unsealed crack width >20mm, alligator	Severe	20 mm to 25 mm single crack, or multiple cracks even if crack opening is less than 20 mm but greater than 13 mm. Cracks have developed cupping or lipping with noticeable bump.
									Very Severe	Greater than 25 mm single crack, or multiple cracks even if crack opening is less than 25 mm but greater than 20 mm. Cracks have fully developed cupping or lipping, and spalling has occurred. Bump or thump.

Distress Type	OREGON (2019)		PENNSYLVANIA (2019)		FLORIDA		WASHINGTON	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
POTHoles	Low	Depth < 1" (Typically delamination of thin patch or seal coat creating a shallow pothole.)	N/A	N/A	N/A	N/A	N/A	N/A
	Moderate	1" ≤ Depth ≤ 2" (Remains within top lift of wearing course.)						
	High	Depth > 2" (Extends beyond top lift of wearing course.)						

Distress Type	CALIFORNIA		INDIANA		NORTH CAROLINA		BRITISH COLUMBIA (2016)		ONTARIO (2016)	
	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits	Severity	Limits
POTHoles	Low	0 < number < 2	Low	Less than 1.0 in. deep.	N/A	N/A	Low	Pothole < 175 cm ² in area (~15cm ø) and less than 25mm deep	N/A	N/A
	Medium	2 ≤ number ≤ 5	Moderate	1.0 to 2.0 in. deep.			Medium	Pothole > 175 cm ² in area (~15cm ø) and 25 to 50mm deep		
	High	number > 5	High	More than 2.0 in. deep.			High	Pothole > 175 cm ² in area (~15cm ø) and greater than 50mm deep		

