https://doi.org/10.65025/ICAIC25235b

Segmentation of Pavement Cracks in North Cyprus

Albert David Bangura
abangura@baucyprus.edu.tr

Tolga Yirtici
Cyprus International University, Cyprus.

Abstract
This thesis explores the use of deep learning, specifically Convolutional Neural Networks (CNNs), for automated pavement crack segmentation in North Cyprus, addressing the need for efficient road maintenance. The study emphasizes the limitations of manual inspection and introduces a CNN-based U-Net architecture, developed to automate feature extraction, and enable more accurate and efficient crack segmentation. Implementing a U- Net model with custom layers to learn features without transfer learning forms the core of this methodology, utilizing ReLU and sigmoid activation functions, binary cross-entropy as the loss function, and the Adam optimizer. The model is evaluated using metrics including accuracy, precision, recall, F1 score, and IoU, achieving over 98% accuracy and demonstrating optimal performance with larger image sizes (256x256). These results highlight the potential of U-Net based crack segmentation systems to significantly improve road maintenance and safety.

Keywords
Convolution Neural Networks (CNNs), Crack Segmentation, Deep learning, U-Net Architecture, Road maintenance.

1. Introduction

This research aims to use artificial intelligence to improve road maintenance in Northern Cyprus. It underscores the poor quality of regional roads, which are riddled with cracks, and the manual detection methods that are simply inadequate. Instead, it proposes the use of Convolutional Neural Networks (CNNs) to detect and segment those cracks in an automated manner—an approach many researchers have argued is much more efficient and accurate than visual inspections. The road maintenance "problem" (the amount and the quality of surveillance and action necessary to keep a road drivable) is a common one, but this research covers a region with a specific set of geological and climatic conditions that are unusual in comparison to most of the other regions in the world where road maintenance is studied.

The road network in Northern Cyprus, especially in major urban areas like Nicosia, Kyrenia, and Famagusta, has a real tough time due to the huge number of vehicles and constrained infrastructure. Mainly built as a two-lane road system, the path network is just not able to handle the kinds of traffic volumes that exists today. This is compounded by the fact that the network has a much lower quality compared to international standards, which means it does not really have any of the features that make a road a safe and reliable means of transportation. Among the most notable "missing features" of the road network is the absence of anything that resembles a proper drainage system. This leads to quite a few water accumulation problems that can lead to cracks. Another major issue is the lack of safety pathways for pedestrians and people riding bikes.

A significant issue affecting the road network in Northern Cyprus is the prevalence of extensive cracks, which vary in size and visibility. While some are easily noticeable, others are more subtle and more difficult to detect, posing a hidden risks to road users particularly for international students and tourists who are unfamiliar with the region. Without timely intervention, these cracks worsen over time, leading to severe damage such as potholes, which degrade the overall quality of the road. This rapid deterioration emphasizes the urgent need for effective road management strategies to enhance infrastructure and improve road safety.

Maintaining roads is essential for ensuring safety, economic performance and environmental sustainability. Well-maintained roads help not only prevent accidents but also support environmentally friendly transportations reducing vehicle operating costs and fuel consumption, factors essential for economic growth. Addressing cracks and resurfacing roads expands the lifespan of road infrastructure, saving on long-term repair costs while minimizing environmental impact and lowering vehicle emissions. However, road cracks present significant challenges. They compromise safety by disrupting traffic flow and potentially damaging vehicles. Cracks also allow water to seep into the road base, weakening the structural integrity. Neglecting small cracks can lead to larger, more expensive road repairs, ultimately increasing maintenance costs and posing greater risks.

The role of timely crack detection in prolonging road life and improving safety.

Timely crack detection is vital for prolonging the lifespan of roads and ensuring the safety of all road users. Roads are subjected to various environmental and load-bearing stresses, leading to crack formation caused by factors such as temperature fluctuations, water infiltration, and continuous traffic. If left unaddressed, small cracks can expand and deteriorate further over time, resulting in severe damage like potholes and patches. Early detection allows for preventive renovation measures, such as sealing or filling cracks, which can effectively halt further deterioration.

From a safety perspective, timely crack detection prevents hazardous conditions that could lead to accidents. Visible cracks, specifically those allowed to evolve into larger defects, pose significant risks to drivers, cyclists, and pedestrians. For vehicles, cracks can reduce tire grip, cause vibrations, and lead to loss of control. Regular monitoring and early intervention mitigate these dangers, ensuring roads remain smooth, well-maintained, and safe for all users.

1.1. Problem Statement

1.1.1. Current Challenges in Manual Road Crack Detection

Manual road crack detection faces significant challenges; traditional methods rely on visual inspections conducted by personnel, which are prone to delays and inconsistencies due to human error and varying levels of expertise (Yaun, Shi, & Li, 2024). Moreover, manual detection often places inspectors in dangerous situations, particularly when working near traffic. These limitations underscore the importance of adopting automated and reliable detection methods to ensure the timely and accurate identification of road cracks, ultimately enhancing road safety and maintenance efficiency (Yaun, Shi, & Li, 2024).

Emerging methods, such as those leveraging deep learning, have demonstrated potential for detecting cracks at both sub-pixel and pixel levels. Early studies, including those by Wang et al. and Yaun, Shi, and Li, have shown promise, but current solutions remain limited. Issues such as low contrast, insufficient datasets, and inaccurate localization reduce the effectiveness of these models (2024). Despite these challenges, integrating conventional techniques with AI-based methods presents a promising path forward to improve the efficiency and accuracy of crack detection.

1.1.2. Research Motivation and Objectives

The motivation of using deep-learning in road maintenance is lies in its potential to provide efficient, accurate, and lead to proactive solutions. By analyzing extensive road network data, deep learning can detect cracks and other issues with far greater precision and effectiveness than traditional human inspections. These technologies enable both real- time monitoring and predictive maintenance; allowing potential problems to be identified and addressed before they escalate into significant damage, such as potholes. This not only extends the lifespan of urban infrastructure but also enhances safety by equipping workers with critical insights to avoid hazardous situations. In essence, deep learning revolutionizes road maintenance management by optimizing the handling of critical infrastructure and conserving valuable resources. The objective of this thesis is to develop a CNN-based system capable of accurately segmenting pavement cracks.

1.2. Scope Of Study

1.2.1. Geographical Focus on North Cyprus

Northern Cyprus presents an ideal setting for studying road crack detection due to its unique geographic and climatic conditions. The region faces several environmental challenges, including significant temperature fluctuations, torrential rains, and coastal humidity, all contributing to road degradation. Additionally, the rugged terrain and mountain ranges, such as the Kyrenia Mountains, further complicate road maintenance efforts. Implementing automatic crack detection systems in the Northern Cyprus context offers the potential to detect cracks promptly and accurately, enhancing the safety and efficiency of the road network. Moreover, such preventative road maintenance strategies can significantly reduce repair costs and extend the lifespan of the infrastructure, addressing critical challenges faced by the region.

1.2.2. Use of CNN as the Primary Tool for Segmentation.

Convolutional Neural Networks (CNN) have emerged as a central algorithm for road crack detection, gaining significant attention due to their remarkable accruable accuracy and effectiveness in image recognition (Elghaish, et al., 2021). Studies have shown that CNN models like U-Net can detect cracks with higher accuracy under diverse lighting and weather conditions. These models are capable of processing large volumes of data in short time, making them highly efficient. By automating the detection process, CNNs eliminate the need for manual inspections, which are inherently subjective and labor intensive (Benedetto et al., 2023). This automation not only enhances crack detection accuracy but also significantly improves the efficiency of road maintenance (Hacefendiolu & Başaa, 2022).

1.3. Significance of the Study

1.3.1. The Potential to Enhance Road Safety and Reduce Maintenance Costs.

Artificial Intelligence driven road management offers the potential of significantly enhance road safety while reducing costs through the application of deep learning and computer vision technologies. These advancements help prevent road defects and eliminate the need for time constraining and error-prone manual inspections. Automated detection enables more accurate and reliable maintenance, ensuring roads remain in better condition. Costs savings are maximized by extending the lifespan of road infrastructure and reducing the frequency and scale of repairs. Therefore, AI driven road maintenance not only delivers improved road quality but also optimizes resource allocation, resulting in substantial economic benefits.

1.3.2. Contribution to Research in AI-based Road Maintenance Systems.

AI research in road maintenance has made significant progress, greatly improving the efficiency and quality of semantic crack segmentation in road images. Deep learning techniques can now detect and classify patchy roads and potholes in real-time, contributing to the extended lifespan of road infrastructure. Incorporating AI into road maintenance also promotes sustainability by enabling proactive and efficient management of resources. This study does not only advance road maintenance practices in North Cyprus but also enriches existing literature by performing pavement crack segmentation on a dataset specifically collected from roads in the Northern Cyprus region.

Literature Review

Road crack segmentation with deep learning is a very hot field and there is a lot of literature on this subject. The section compared the work of some of these researchers to identify the main problems and strengths of each. Road infrastructure can no longer be underrated as an asset for growth and social mobility. Pavement crack is unsafe, and late maintenance can be expensive. Pavement crack detection used to be an observation task done manually with the naked eye, but the downside is that it's slow, subjective and error- prone (Lau, et al., 2020).

Faced with these constraints, engineers have turned to computation to automate segmentation and detection of pavement cracks. DL algorithms – Convolutional Neural Networks (CNNs) in particular – are now the best prospects for crack segmentation– better, faster and more reliable (Lau, et al., 2020).

In response to these limitations, scientists have used computation for automated pavement crack segmentation. Deep learning (DL) algorithms – Convolutional Neural Networks (CNNs) in particular – have become the perfect candidates for this purpose, with great promise for making pavement crack detection more accurate, efficient and objective (Lau, et al., 2020).

2.1. Key Deep Learning Architectures

The most popular deep learning architectures explored for crack segmentation in roads are:

2.1.1. Fully Convolutional Networks

FCNs are another most used type for semantic segmentation, and they dispense fully connected layers with convolutional layers so that you can provide input images of unlimited size. We have tested the crack detection for FCNs of all backbones, VGG16, VGG19, ResNet50.

2.1.2. SegNet

SegNet is also a 100% convolutional network just like U-Net but with different decoder architecture. It has been used for crack segmentation in pavements such as concrete, asphalt, and bridge decks.

2.1.3. Other Architectures:

Other architectures have also been investigated like ResNet, DenseNet, PSPNet, DeepLabv3+ and GCN, all having their own advantages and disadvantages of accuracy, performance and complexity.

2.2. Datasets And Evaluation Metrics

A very important aspect of deep learning research is good datasets for training and testing. Some open-source crack detection datasets were created, each with its own features and drawbacks. Some commonly used datasets include:

2.2.1. CrackForest

Concrete crack data set with hand labeled ground truth.

2.2.2. AigleRN

A map of road crack images taken with street cameras.

2.2.3. Crack500

A map of road crack pictures from mobile phones.

2.2.4. TRIMMED

Collection of grayscale road crack images recorded with high-resolution line-scan camera.

2.2.5. CFTD

Collection of RGB road cracks image captured with consumer grade cameras.

To test the effectiveness of crack segmentation algorithms, scientists often measure it with a mix of different metrics such as precision, recall, F1-Score, Intersection over Union (IoU).

2.3. Challenges And Contributions from Previous Studies

Lau et al. (2020) introduced a U-Net CNN with a ResNet-34 encoder through transfer learning to overcome the inefficiencies of traditional crack detection methods like thresholding, morphology, and edge detection. Their model incorporated SCSE modules, progressive image resizing, and varying learning rates to optimize training. Tested on CFD and Crack500 datasets, the system achieved F1 scores of 96% and 73%, outperforming larger models such as Split-Attention Network.

Zhang et al. (2019) designed a context-aware semantic segmentation network that fused predictions from overlapping image patches using cross-state and cross-space constraints. This approach allowed pixel-wise crack segmentation without retraining for different image sizes. Applied to CFD, TRIMMD, and CFTD datasets, it achieved state-of-the-art Boundary F1 scores while processing images in 0.7 seconds each, demonstrating efficiency, accuracy, and scalability for infrastructure crack assessment across diverse conditions.

Gao et al. (2019) proposed a generative adversarial network (GAN)-based approach for crack segmentation, introducing U-GAN, CU-GAN, and FU-GAN, all using modified U-Nets as generators. The models were trained to distinguish real from fake crack images at the pixel level. On AigleRN, CFD, and HTR datasets, they outperformed existing methods in precision, though recall rates declined on noisy or blurry images. The study suggested advancing multi-scale convolution, feature fusion, and attention mechanisms.

Pereira et al. (2019) addressed the inefficiency and subjectivity of manual crack inspections by developing a U-Net-based deep learning system for pavement and pothole segmentation. Data were collected via smartphones, and experiments showed the model achieved 97% accuracy with mean mIoU of 0.85. Despite dataset limitations, U-Net performed exceptionally well outside its original medical imaging domain, with future improvements anticipated through larger, more varied datasets and comparisons with newer architectures.

Wang et al. (2021) compared eleven CNN-based semantic segmentation models, including FCN, PSPNet, UPerNet, and DeepLabv3+, using ResNet, VGG, and DenseNet backbones. Models were evaluated with IoU, accuracy, precision, recall, and F1 score. DeepLabv3+ with ResNet101 backbone performed best, particularly with spatial pyramid pooling. Results confirmed CNNs' superiority over heuristic methods, though noise and dataset variations remained challenges. The authors recommended DeepLabv3+ and GCN as robust solutions for practical crack detection.

Li et al. (2022) highlighted deep learning advances, showing CNNs like U-Net, HED, and SRN significantly outperform image processing for crack segmentation. They suggested integrating semantic segmentation with edge detection to overcome each method's weaknesses. This combined framework improves accuracy in identifying fine crack properties such as width and length. Such detailed measurement offers practical benefits for pavement maintenance and management, demonstrating deep learning's impact on safety and cost efficiency in infrastructure monitoring.

Zhang et al. (2023) emphasized the shift from handcrafted, image-processing methods toward CNN-based automated crack detection. Models such as U-Net, UperNet, ResUNet, and Pointrend produced strong results even under noisy and poorly lit conditions, proving more effective than earlier techniques. However, segmentation of small cracks and distinction from pavement features like stains or manhole covers remained difficult. Their study reinforced CNNs' strengths while acknowledging persistent challenges in fine-scale crack identification and differentiation.

Lee et al. (2019) proposed the Semantic Crack Segmentation Network (CSN) to address the limitations of edge-detection filters and early CNNs, which were restricted to patch-based methods and suffered from noise. CSN improved accuracy by processing entire images and avoiding sliding windows. To combat data scarcity, they generated synthetic crack images using Gaussian kernels and Brownian motion. Combining real and synthetic data enhanced training, leading to stronger segmentation performance in cluttered and complex scenes.

Jia (2023) criticized the slowness and inaccuracy of manual inspections and image processing for pavement crack identification. They proposed an enhanced U-Net with an Efficient Channel Attention (ECA) module in the encoder and FCNHead decoder. This configuration strengthened crack feature extraction and model generalization. Their method proved particularly effective in detecting small cracks under challenging pavement conditions, offering a practical solution for improving segmentation performance and reliability in infrastructure maintenance and monitoring tasks.

Benedetto et al. (2023) tackled crack segmentation challenges caused by noisy, obstructed, and overexposed images. They proposed a modified U-Net incorporating a ResNet50 encoder pre-trained on ImageNet, combined with residual structures to improve learning. Their model excelled in handling environmental obstructions and achieved more precise crack detection, especially for measuring crack width. Since crack width is essential for severity determination, the proposed system significantly enhanced the accuracy and usefulness of automated crack assessment tools.

Panella et al. (2022) reviewed deep learning models, focusing on CNNs like FCNs and U-Net for crack segmentation. They analyzed the trade-off between model complexity and loss of fine details due to pooling layers. U-Net's skip connections allow it to preserve spatial information across layers, enhancing accuracy and efficiency. The review underscored U-Net's biologically inspired design and superior performance in separating "what" and "where" information compared with simpler CNNs, making it a reliable segmentation architecture.

Al-Huda et al. (2023) introduced KTCAM-Net, a hybrid deep learning model combining classification and segmentation networks with Class Activation Maps (CAM). KTCAM-Net employed hybrid loss functions, crack boundary filtering, and overlapping fusion algorithms, producing fine crack localization even under noise and imbalance. Benchmark datasets confirmed its superior performance, particularly for detecting thin cracks. The model's flexibility extended beyond cracks, showing potential for broader surface defect detection in inspection tasks, highlighting its innovative contribution to segmentation methods.

Nguyen et al. (2021) proposed a two-stage CNN to address noisy, low-resolution crack images. The first stage used a five-layer CNN to localize cracks and remove artifacts, while the second encoder-decoder network segmented pixels in the localized regions. Tested on the 2StagesCrack dataset, the method outperformed single-stage models by handling low-quality images effectively. The authors concluded that this two-stage system was both accurate and computationally efficient for real-world pavement crack segmentation.

Chen et al. (2020) presented Progressive Contextual Segmentation Network (PCSN), built on SegNet with a VGG16 encoder pre-trained on ImageNet. They applied augmentation methods like flipping, rotation, and contrast scaling to improve dataset diversity. Compared to Mask R-CNN and FCN-8s, PCSN achieved superior crack detection under difficult conditions, balancing inference speed and segmentation quality. The study showed how input image size impacted performance, with larger inputs providing richer crack details at the cost of slower processing.

Fang et al. (2020) categorized crack detection into image processing, classical machine learning, and deep learning. They noted that while thresholding and edge detection are noise-sensitive, machine learning methods like SVMs lacked robustness. CNNs achieved major advances, though challenges included high computational costs, noisy faint cracks, and difficulty maintaining both global and local context. Their review underscored the advantages of deep learning but highlighted the ongoing need for efficiency improvements and better contextual data integration.

Yu et al. (2021) compared one-stage and two-stage deep learning detection models for crack detection. Two-stage models like Faster R-CNN achieved high accuracy by using region proposal networks, while one-stage models such as YOLO and SSD traded some accuracy for speed, making them more suitable for industry. Both approaches struggled when applied to UAV-acquired images, which are much larger than standard datasets like ImageNet or VOC, revealing limitations in scalability for field conditions.

Ali et al. (2021) examined CNNs for automated, real-time crack detection in buildings, highlighting their advantages over costly and dangerous manual inspections. CNNs could classify images, localize cracks with bounding boxes, and perform pixel-level segmentation. However, segmentation accuracy was hampered by class imbalance, inconsistent lighting, and obstacles in images, which biased networks against crack pixels. Their study emphasized these practical challenges while confirming CNNs' effectiveness for crack detection tasks in diverse real-world applications.

Yan et al. (2022) addressed the difficulty of detecting cracks in low-light conditions by proposing CycleADC-Net. The system first employed CycleGAN to translate dim images into brighter ones without altering crack structure, then applied a dual-channel encoder-decoder with attention to merge global and local signals. This approach improved detection accuracy in poorly lit scenarios, enabling robust crack segmentation where conventional deep learning models typically fail due to lighting limitations.

Xu and Liu (2022) solved the challenge of limited training data by employing DCGAN to generate synthetic pavement crack images. Expanding the dataset from 1,608 to 6,000 images improved CNN classification accuracy from 80.75% to 91.61% while reducing class imbalance. Their findings demonstrated the value of generative models for creating realistic training data, enhancing both model performance and dataset diversity in pavement crack detection applications, particularly when real-world samples are scarce.

Chen et al. created a dataset of 10,000 manually labeled pavement and bridge crack images to address the lack of public datasets. Using SegNet with a VGG16 encoder and data augmentation, they developed PCSN, which achieved mean average precision of 83%, outperforming Mask R-CNN (42%) and FCN-8s (67%). Larger input sizes improved segmentation performance, though smaller inputs offered faster inference. Their work demonstrated the importance of dataset quality and architecture design for reliable crack detection.

2.4. Addressing Data Limitations

2.4.1. Data Augmentation

Most research works use data augmentation to scale the size and number of training datasets, decreasing overfitting, and achieving model generalization. Certain common augmentation tricks include rotation, flip, scaling, and cropping images.

2.4.2. Cross-Dataset Testing

To test models for their adaptation to different imaging scenarios and crack types, some researchers have run cross-dataset testing — training models on one dataset and running them against another.

2.5. Handling Complex Noise and Background

2.5.1. Modules for Attention

Attention modules such as Convolutional Block Attention Module (CBAM) or Efficient Channel Attention module (ECA)) are added to CNN architectures for feature extraction and attention to specific image areas. These modules are used to help the model deconstruct cracks from noise and other non-relevant features.

2.5.2. Residual Structures

Residual structures like ResNet have been demonstrated to improve performance of deep learning models by allowing training of deeper networks. Adding skip connections that skip one or more layers is how residual structures solve the vanishing gradient problem and make the information flow throughout the network.

2.6. Computational Efficiency

2.6.1. Improving Network Design

Network engineers are constantly looking for new ways to tweak network design, so it runs faster and with accuracy. Such as searching for lightweight architectures, delimiting parameters, and learning from the effective training strategies.

Although pavement crack segmentation is making great progress, there are still obstacles, and more research needs to be done:

2.6.2. More Capable and Generalizable Models

Typical models can't handle the complexities of real-world scenarios like lighting conditions, shadow interference, water stains, crack types, etc. The research needs to be more robust and generalizable models that can manage these problems well.

2.6.3. Crack Width Estimation

Most research is centered around crack segmentation, but the crack width estimate is important for pavement cleaning and repair. Our next studies should try to build crack width estimation into the segmentation equations.

2.6.4. Instant Crack Detection

The real-time crack detection is required in applications like autonomous road monitoring and maintenance planning. The studies should keep developing computationally efficient models and algorithms that can perform in real time on resource limited machines.

Deep learning has taken the pavement crack segmentation process to a whole new level, with promise for automating pavement inspection and maintenance. Various CNN architectures and methods have been able to show promising results but there's more research that needs to be done on robustness, generalization, crack width estimation, and real-time performance. Researchers noted that in future research, deep learning for pavement crack segmentation will be ever more integral to road safety and resilience.

Three Background

3.1. Computer Vision

Computer vision is the area of artificial intelligence (AI) that lets computers and systems process visual data in the world like digital pictures and videos. Through ingesting and visualizing these data, computer vision systems can do the things humans do most of the time, such as object detection, image classification and segmentation tasks.

Computer vision is as old as it gets: research began in the 1960s, to help machines see even simple pictures. The digital image, computational capability and algorithmic development that has followed over the decades have led to the advancement of the industry. Notably, deep learning and neural networks have enabled significantly higher image recognition and detection.

3.1.1. Core Techniques and Technologies

Today's computer vision is dependent on some important technologies:

- Machine Learning & Deep Learning: These are methods that can learn from millions of images and increase patterns and predictions.
- Convolutional Neural Networks (CNNs): CNNs are neural networks trained specifically on pixels and so, are ideal for image classification.
- Computer vision has a huge number of applications in many different sectors:
- Healthcare: Performs medical imaging for diseases and abnormalities (X-ray, MRI to diagnose).
- Auto industry: Creating safe perception and control of the environment with object recognition and lane-recognition for autonomous cars.
- Manufacturing: Supporting automatic inspection and quality control, where product defects are found on production lines.
- Safety and Surveillance: Creating surveillance with face detection and activity detection to ensure security and safety.
- Current research in computer vision is about to make it more accurate, more efficient, and more applicable in various fields. Combining computer vision with other AI capabilities including natural language processing and robotics, there are more opportunities for

creative and usable solutions across many industries.

3.2. Machine Learning and Deep Learning

Machine learning (ML) is a subset of artificial intelligence (AI) that helps systems learn from data, find patterns, and make decisions without any human involvement. With algorithms and statistical representations, ML makes it possible for computers to perform tasks that cannot be programmed for each action in the program.

Deep learning on the other hand is a branch of machine learning (ML) in artificial intelligence (AI) based on algorithms modelled after the brain's structure and activity — artificial neural networks. These algorithms search for patterns and make choices based on layers of data from which more advanced features are extracted.

In terms of the theory, deep learning has been around since the 1940s, when neural network models were first built. Researchers such as Frank Rosenblatt first developed the perceptron, an early neural network architecture capable of rudimentary pattern recognition, in the 1950s and '60s. But computational restrictions and theoretical problems held up progress.

These regained momentum in the 1980s when the backpropagation algorithm was developed to train multi-layer neural networks better. But even with all these improvements, it wasn't until the 2000s, when the computing power became more powerful and the availability of more data, that deep learning really started making big gains.

3.3. Convolutional Neural Networks

CNNs are a type of deep learning model that works on data with a grid structure like images. They have been the key technology of computer vision and enables computers to perform tasks such as image and video recognition, object recognition and segmentation. In 2012, the field made its breakthrough when AlexNet by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton won the ImageNet Large Scale Visual Recognition Challenge. AlexNet's accomplishment demonstrated the power of CNN in tackling advanced image classification problems (Draelos, 2019).

3.3.1. Core Components and Architecture

An ordinary CNN architecture consists of the following key players:

- Convolutional Layers: These layers perform convolutional functions on the input, thus identifying spatial hierarchies and local trends in the data.
- Activation Functions: Non-linear functions such as ReLU (Rectified Linear Unit) are used to introduce non-linearity to the model and allow it to learn complex patterns.
- Pooling Layers: Pooling layers perform down-sampling to reduce the dimension of the data and make the representations more tractable and stable with slight translations.
- Fully Connected Layers: After a few convolutional/pooling layers, high level neural network reasoning occurs through fully connected layers that links each neuron of a layer to each neuron of a subsequent layer.
- CNNs are used in a variety of applications, some of them include:

    - Image & Video Recognition: Widely applied in face recognition software, object detection and video analysis.
    - Medical Imaging: CNNs are used to diagnose illnesses based on medical images (MRIs, CT scans, X-rays etc.).
    - Autonomous Vehicles: They make self-driving cars to see and interpret the world by analyzing images from cameras.
- Natural Language Processing: CNNs are used for text classification (sentiment detection, spam detection).

3.4. Semantic Segmentation of Pavement Cracks

This is the task of categorizing every pixel in an image to correctly determine and mark which regions of an image is a crack. This is necessary for the maintenance of infrastructure and transportation safety.

Pavement crack detection has always been done manually with some simple image processing like edge detection and thresholding. These methods were promising as early solutions but had low accuracy, particularly in multi-light, multi-noise environments. The advent of deep learning and the improvement of computation power has ushered a more efficient way of detecting cracks in images using more advanced techniques such as U- Net.

3.5. U-Net Architecture

U-Net first published in "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Ronneberger et al., (2015) is a special kind of convolutional neural network (CNN) to perform image segmentation efficiently. It has become an ideal architecture for image segmentation tasks, which are to find each pixel in a picture and classify it into a single category such as crack or no-crack. U-Net is a pipeline with two main sections: a decoder and an encoder.

3.5.1. Encoder

The encoder analyses the image and extracts important features, creating a condensed summary of the visual data.

3.5.2. Decoder

The decoder takes the condensed information from the encoder and gradually reconstructs the image but with a focus on highlighting those specific features related to cracks.

3.5.3. Skip Connections

This is what makes the U-net very effective, it links the encoder and the decoder stages. These connections allow the network to retain both the fine grain details from the original image and the high-level understanding gained by the encoder. It is combining the big picture and those tiny details.

3.5.4. Convolutional Layers

Responsible for extracting features from the image like edges, textures and patterns

3.5.5. Activation function

Determines whether a neuron should fire or not. So, it is not just about detecting a feature but about deciding how important that feature in the context of crack detection

3.5.6. Pooling layers

Responsible for down sampling the image, making computations more manageable while retaining the most important information. It is like creating a more efficient summary of the image data without losing those crucial details.

3.5.7. Dropout

This helps prevent a common problem in machine learning called overfitting. Overfitting is like when a student memorizes the answers for a test but cannot apply that knowledge to new problems. So, dropout layers help the model generalize better to accurately identify cracks in any image. Even with the best architecture, a model needs guidance to learn effectively, that is where the loss function and optimizer comes into play.

3.5.8. Loss Function

Think of the loss function as coach that yells at a team when they mess a play. It measures how far off the model's predictions are from the actual cracks in the images, guiding the model to improve its performance.

3.5.9. Optimizer

Think of the optimizer as the coach constantly adjusting strategies and techniques to minimize those mistakes, in this case to minimize the loss.

3.5.10. Intersection Over Union (IoU)

The intersection over union is how we know the model actually works. It measures how well the models predicted crack overlap with the actual crack in the image. Perfect score of 1 means the model's prediction perfectly matches the real crack. The model developed in this thesis achieves some seriously impressive IoU scores demonstrating its accuracy in identifying and delineating those cracks.

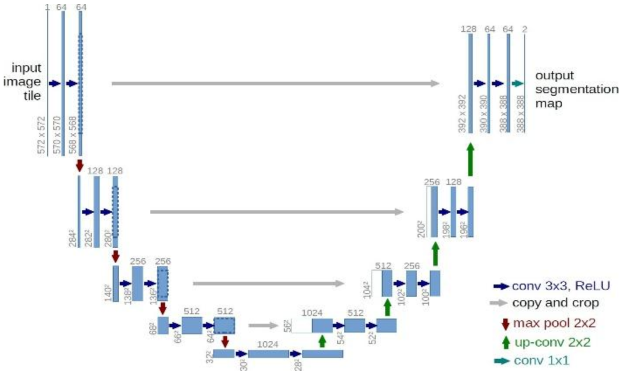


Figure 3.1: A standardized U-Net architecture, which lays a core premise to this thesis
Source: (https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/, 2015)

Methodology

Developing a deep learning-based solution for the automatic segmentation of cracks in pavement images serves as the central focus of this study. Framed as a binary image segmentation task, the process involves classifying each pixel in an input image as either part of a crack or a non-crack region.

This chapter outlines the methodology employed to achieve accurate segmentation of pavement cracks using deep learning. Key steps include data preprocessing, normalization, model training, validation, testing, and evaluation. The aim is to create and train a robust model that delivers precise and reliable results, paving the way for efficient and automated road maintenance solutions.

4.1. Dataset

4.1.1. Data Description

The dataset consists of annotated images designed for pavement crack detection and classification. It contains 893 images, each labeled with polygonal annotations for various types of pavement distresses. The annotations which were done by an expert include geometric coordinates for the regions of interest, enabling precise identification of defects. The dataset is organized and timestamped, indicating updates and task details, making it suitable for training and validating computer vision models in infrastructure inspection. The dataset contains the following classes: Fatigue cracking, Block cracking, Edge cracking, Wheel path longitudinal cracking, non-wheel path longitudinal cracking, Transverse cracking, Patch, Potholes, Manholes, Bumper removal, Bleeding, Raveling. However, since this this study performs binary segmentation, therefore all the twelve classes were regarded as cracks and the background as none crack, resulting into two classes.

4.1.2. Data Preprocessing

Pavement crack images and their corresponding ground truth were generated. The dataset is divided into training, validation, and test subsets for model training and evaluation. The training is conducted for various sets of parameters with 90% of the images for training and validation (10% of the training dataset used for validation), and 10% used for testing.

Images and masks were resized to a fixed resolution to standardize the input for the U-net architecture. For grayscale images, a channel dimension was added to ensure compatibility with the network architecture. Masks are converted to binary (0 or 1) and stored as single-channel images where 0 indicates none crack and 1 crack regions.

4.2. Model Design

A U-Net architecture was chosen for its effectiveness in semantic segmentation tasks, particularly for infrastructure images where fine-grained details are essential. Custom layers were defined to allow the model to learn the features in the dataset from the ground up without any transfer learning. This resulted to a more simplified and efficient model. Below is a description of the layers in the developed model. The U-Net model consists of the following detailed layers and components:

4.2.1. Input and Scaling

The input layer accepts images with dimensions (IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS). A Lambda layer normalizes pixel values by dividing every pixel value by 255.0 to scale them between 0 and 1.

4.2.2. Contraction Path

This path progressively down samples the input while extracting features using convolutional and pooling operations:

- Layer 1: Two convolutional layers with 64 filters of size 3x3, followed by a dropout rate of 0.1. Padding is set to same, maintaining the spatial dimensions.
- MaxPooling: A pooling operation with a 2x2 window reduces the spatial dimensions by half.
- Layer 2: Two convolutional layers with 128 filters of size 3x3, followed by dropout rate of 0.2.
- MaxPooling: A 2x2 pooling layer reduces dimensions further.
- Layer 3: Two convolutional layers with 256 filters of size 3x3, followed by dropout of rate 0.3.
- MaxPooling: A 2x2 pooling layer reduces dimensions again.

4.2.3. Bottleneck

This section represents the deepest part of the network, capturing high-level abstract features: Two convolutional layers with 512 filters of size 3x3, followed by a dropout layer rate 0.4.

4.2.4. Expansive Path

This path upsamples the feature maps and combines them with corresponding feature maps from the contraction path via skip connections to recover spatial information

- Layer 5: A transposed convolution layer upsamples the feature maps, halving the number of filters to 256. The output is concatenated with features from Layer 3. Two convolutional layers with 256 filters 3x3 refine the combined features, with dropout of rate 0.3.
- Layer 6: Another transposed convolution upsamples to 128 filters. The output is concatenated with Layer 2. Two convolutional layers with 128 filters 3x3 follow, with dropout rate 0.2.
- Layer 7: A final transposed convolution upsamples to 64 filters. The output is concatenated with Layer 1. Two convolutional layers with 64 filters 3x3 refine the features, with dropout of rate 0.1.

4.2.5. Output Layer

A final convolutional layer with 1 filter of size 1x1 and a sigmoid activation function produces a single-channel output, representing the pixel-wise probability map for segmentation. Overall, the network maintains critical spatial information by concatenating skip connections from the contraction path to the expansive path, effectively leveraging both low-level details and high-level features. The number of filters doubles in the down sampling path to enhance feature representation and halves symmetrically in the up sampling path to match the input dimensions.

4.2.6. Activation Functions

In an artificial neural neuron, an activation function decides whether to fire a neuron or not. In other words, is my output equal to 1 or 0. One if it is activated and zero if it is not activated in the case of a binary problem.

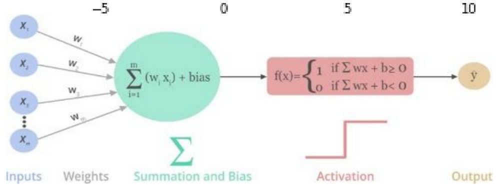

x_1
x_2
x_3
x_n
w_1
w_2
w_3
w_n
$\sum_{i=1}^{m}(w_i x_i) + \text{bias}$
$f(x)=\begin{cases}1 & \text{if } \Sigma wx+b \geq 0\\ 0 & \text{if } \Sigma wx+b < 0\end{cases}$
$\hat{y}$
Σ
Inputs
Weights
Summation and Bias
Activation
Output

Figure 4.1: Representation of a simple neural network with one node
Source : (https://www.mdpi.com/2071-1050/16/23/10756, 2024)

4.2.7. Sigmoid Activation Function

The output layer uses a single convolution operation with a sigmoid activation function to generate pixel-wise probabilities for the presence of cracks. The sigmoid function is a mathematical function commonly used in machine learning and neural networks. It maps any input value to a value between 0 and 1, making it ideal for binary classification problems or tasks where probabilities are required.

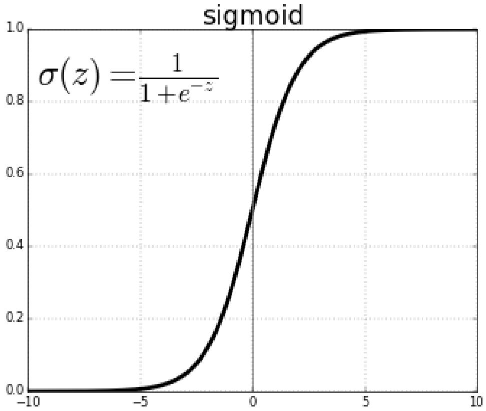


Figure 4.2: Sigmoid function
Source: (https://www.quora.com/What-is-the-ReLU-layer-in-CNN, 2020)

4.2.8. ReLU Activation Function

ReLU is the most used activation function in the convolutional layer of a convolutional neural network especially for image classification, objection and segmentation. The ReLU activation function is expressed mathematically below. If z is greater than 0, the output is z else if z is less than or equal to 0, the output is 0.

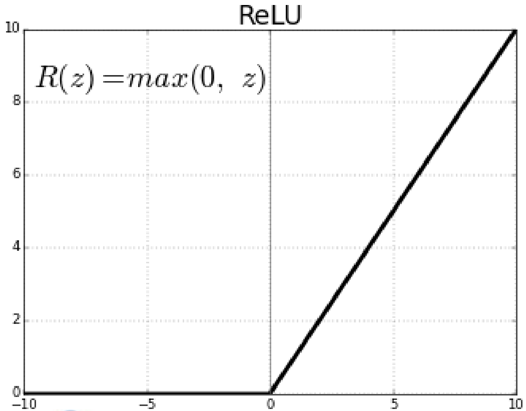


Figure 2.3: ReLU Activation Function
Source: (https://www.quora.com/What-is-the-ReLU-layer-in-CNN, 2020)

4.2.9. Loss Function

Also known as the cost function or error function quantifies the error between output of the algorithm and the given target value. Binary cross-entropy is used as the primary loss function for segmentation tasks.

4.2.10. Optimizer

An optimizer updates the model in response to the output of the loss function. Optimizers assist in minimizing the loss function. The Adam optimizer was used due to its effectiveness, and it is popularly used in the convolutional neural networks. Adam stands for Adaptive moment estimation. According to Kingma and Ba (2015) Adam is computationally efficient with little memory requirements and typically require little to no tuning.

4.3. Experimental Setup

The study involved training, validating, testing and evaluating the U-net model using a diverse dataset of pavement images containing various crack types and severities. The dataset was divided into three subsets: training, validation, and testing. The training subset was used to optimize the model's parameters, while the validation subset was employed to monitor the training process and prevent overfitting. The testing subset, consisting of unseen images, was used to test the model's performance on new data. Various hyperparameters were investigated to optimize the U-net model's performance, including:

4.3.1. Image Size

Experiments were conducted with different image dimensions (width and height), including 128 and 256 pixels.

4.3.2. Batch Size

The number of images processed in each training iteration was varied, exploring batch sizes of 8, 16, 32, and 64.

4.3.3. Number of Filters

The number of convolutional filters in the U-net architecture was adjusted to explore the impact on performance.

4.3.4. Bottleneck Dropout

An increase in dropout of 0.5 was introduced at the bottleneck of the U-net architecture to explore its impact on mitigate overfitting. Overfitting randomly drop out a proportion of neurons during training.

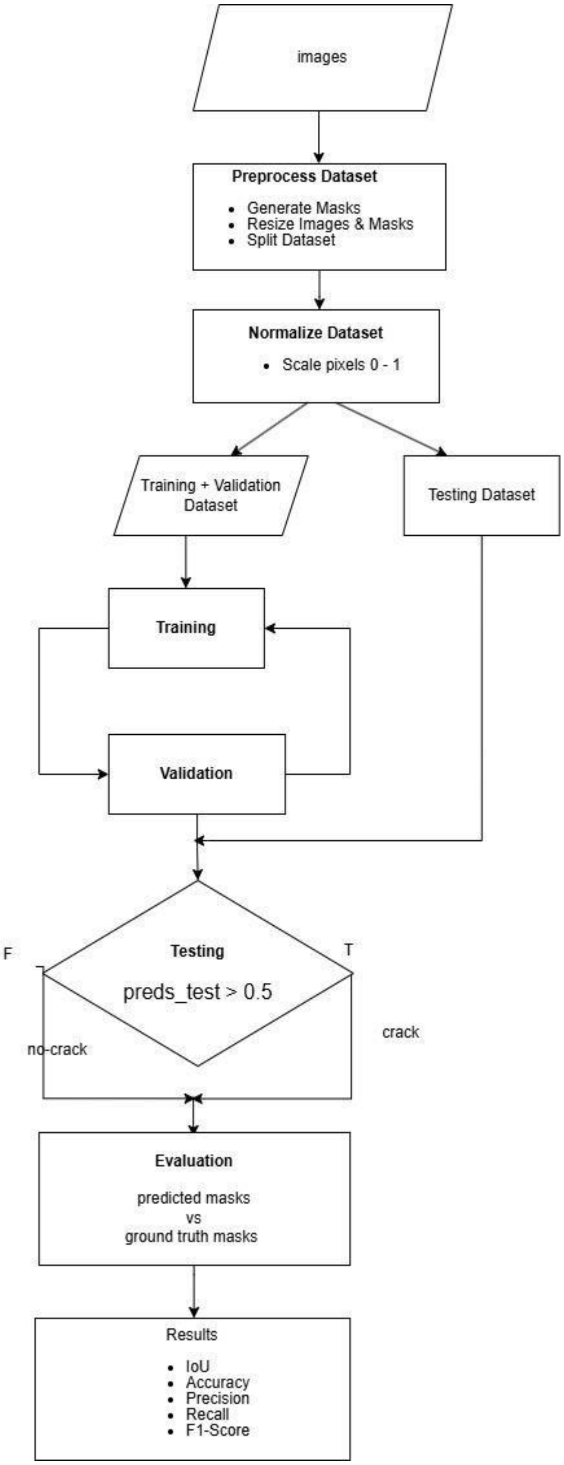


Figure 4.4: Flowchart of Methodology

Below is the pseudocode for the flowchart of the methodology shown above START

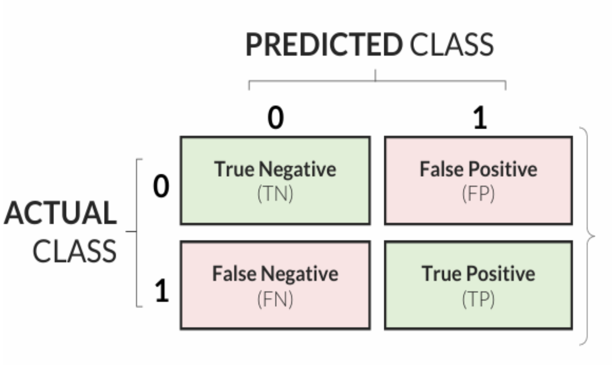
```
#Preprocess images
    ✓ Generate masks
    ✓ Resize images and masks
    ✓ Split dataset # Normalize dataset
    ✓ Scale pixels values between 0 and 1 #Train and
      validate the Model
For I = 1 : EPOCHS
For j = 1 : BATCHES
    ✓ Train the model
    ✓ alidate the model End
End
# Test the model
For i = 1 : PREDICTIONS
If preds > 0.5
Crack

End
Else
No-crack
# Evaluate the model
COMPARE Ground Truth Masks vs. Predicted Masks
DISPLAY RESULT
END
```

4.4. Evaluation Metrics

4.4.1. Confusion Matrix

A Confusion Matrix is a table summarizing the frequency of predicted vs. actual pixels for a given set of data. This is the most common and concise way to evaluate performance and compare models against one another. A confusion matrix can be used to derive several types of model performance metrics, including accuracy, precision and recall.



ACTUAL CLASS \ PREDICTED CLASS	0	1
0	True Negative (TN)	False Positive (FP)
1	False Negative (FN)	True Positive (TP)

Figure 4.5: Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad - - - - \qquad 4.1$$

Accuracy helps answer the question, of all predictions, what percentage were correct? However, accuracy alone is not enough to properly evaluate a classification model when one class is rare, a model may have high accuracy but a useless model. Precision and recall add context and may be superior to accuracy in some cases. This is true for many segmentation and classification problems, where the focus is on predicting one class correctly than the other. For example, in pavement crack segmentation, the goal is to segment cracks. A very small percentage of regions on an image is actually going to be cracks. So, the modelling effort is going to be on segmenting cracks, and not on regions with no cracks. Precision and recall are metrics that focus on how well the model predicts the positive class, that is cracks in this case.

4.4.3. Precision

Precision answers the question, of all predicted positives (crack pixels), what percentage were correct?

$$Precision \; = \frac{TP}{TP + FP} \qquad --------- \qquad 4.2$$

4.4.4. Recall

Recall answers the question, of all actual positives (crack pixels), what percentage were predicted correctly. Recall is an important metric when we are more concern with False Negatives which is the case in the context of pavement crack segmentation.

$$Precision \; = \frac{TP}{TP + FN} \qquad --------- \qquad 4.3$$

4.4.5. F1 Score

F1 score measures the harmonic mean of precision and recall. In mathematics the harmonic mean, which is favored for means of ratios as it measures an equal weight on both ratios. It will thus be lower than the standard arithmetic mean.

$$Precision \; = 2\frac{Precision * recall}{Precision + recall} \qquad ---- \qquad 4.4$$

4.4.6. Intersection over Union (IoU)

This metric compares the area of overlap between the predicted masks and the actual or ground truth masks to the total area of the two masks combined. The IoU ranges from 0 (no overlap) to 1 (perfect overlap).

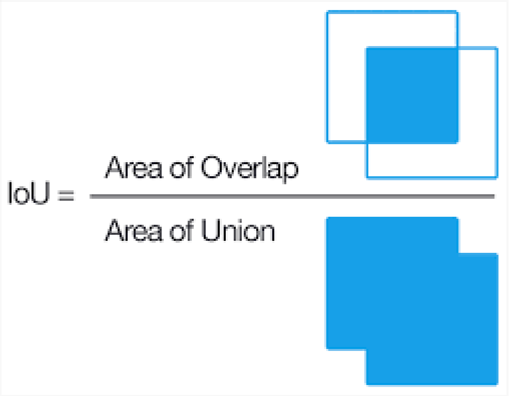


Figure 4.6: Intersection over Union
Source:(https://wiki.cloudfactory.com/docs/mp-wiki/metrics/iou-intersection-over- union, 2024)

4.5. Implementation Details

Tools, frameworks, and hardware/software specifications.

The following tools and frameworks were used to develop and train the model:

- TensorFlow and keras
- Visual studio code using the Jupiter notebook extension to create notebooks
- Python and other python libraries

The model was trained on a PC with an Nvidia GPU with the following hardware specifications:

- 16 GB RAM
- 16 GB NVIDIA GeForce RTX 4060 GPU
- 512 GB SS

Results and discussion

5.1. Results

The performance of the U-Net model was assessed using various metrics, including accuracy, precision, recall, F-score, and Intersection over Union (IoU) score. The U-net model demonstrated promising results in segmenting pavement cracks. Table 1 present the detailed results for the testing phase. This table provide a comprehensive overview of the model's performance across different experiments, allowing for a comparative analysis of the impact of various hyperparameter configurations

Table 5.1: Testing Results

Exp	**Accuracy**	**Precision**	**Recall**	**F1Score**	**Epoch**	**Batch size**	**IoU score**
1	98.68	82.97	76.88	79.81	56	8	66.40
2	98.63	82.55	75.67	78.96	52	32	65.24
3	98.55	85.42	68.94	76.30	62	64	61.68
4	98.63	82.53	75.74	78.99	34	8	65.27
5	98.61	84.88	71.93	77.87	56	32	63.77
6(bn do = 0.5)	98.65	82.29	76.80	79.45	58	16	64.42
7(W&H = 256)	98.66	85.57	72.37	78.42	25	8	64.50
8	98.67	81.47	78.34	79.87	48	16	66.49
9 (32 filters layer)	98.54	80.52	75.23	77.79	61	32	63.65
10 (16 filters layer)	98.67	83.33	75.87	79.43	36	8	65.88

Table 5.2: Confusion Matrix

Experiment 1	
393528	2191
3209	10672
Experiment 2	
393499	2220
3377	10504
Experiment 3	
394086	1633
4312	9569
Experiment 4	
393493	2226
3368	10313
Experiment 5	
393517	2202
3799	10082
Experiment 6	
393425	2294
3220	10661
Experiment 7	
393493	2226
3368	10313
Experiment 8	
1573225	9859
11982	43334
Experiment 9	
393193	2526
3438	10443
Experiment 10	
1574690	8394
13346	41970

5.1.1. Results From Literature

Below is a list of papers taken from previous studies. The results of the evaluation metrics from these studies have been shown in table

1. Generative Adversarial Networks for Road Crack Image Segmentation
2. Deep Learning-Based Semantic Segmentation Methods for Pavement Cracks
3. U-Net-Based CNN Architecture for Road Crack Segmentation
4. Semi-supervised semantic segmentation network for surface crack detection
5. Hybrid deep learning pavement cracks semantic segmentation
6. Two-stage convolutional neural network for road crack detection and segmentation

Table 5.3: Results from previous studies

Title	Accuracy	Precision	Recall	F1-score
1	N/A	75.06	55.19	60.48
2	76.24	52.73	75.95	80.89
3	N/A	85.34	68.13	75.77
4	N/A	82.39	56.88	67.30
5	N/A	64.10	64.10	62.40
6	N/A	77.00	75.00	72.00

5.1.2. Visualization of Segmentation Results

Predictions on random samples from the training, validation, and test datasets were visualized. This included:

- Input image: This is an unseen original image from the test dataset
- Ground truth mask: This is the mask generated for the original image.
- Predicted mask: This is the image with only the crack regions painted by the U-Net model.

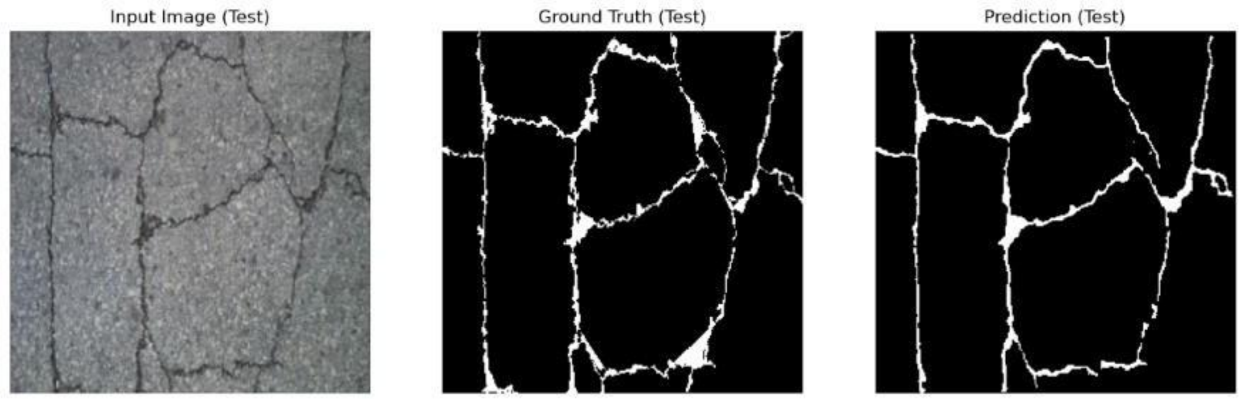
The visualization corroborated the results from the evaluation metrics which indicated high precision and recall, affirming the model's capability to segment pavement cracks accurately. By employing robust preprocessing, custom metrics, and extensive visualization, the workflow ensures reliable segmentation results suitable for practical deployment.

5.1.3. Thresholding

The sigmoid output of the model was thresholded at 0.5 to convert probabilities into binary predictions.

5.1.4. Inference Results

The trained model was tested on a separate test set. A random test image is displayed with its corresponding ground truth and predicted masks to verify the correctness of the predictions.



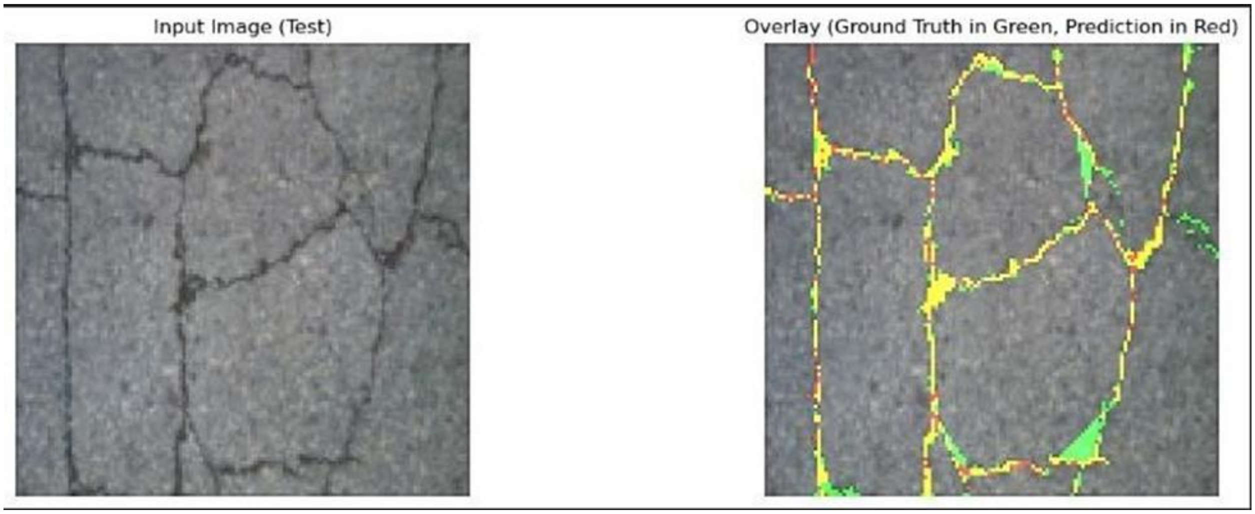
Input Image (Test)

Ground Truth (Test)

Prediction (Test)

Figure 5.1: Screenshot of inference result

To clearly visualize which regions of the image the model has been able to clearly segment as crack and those that the model missed, the ground truth mask coloured in green and predicted mask coloured in red has been overlayed over each other. When red and green are combined, the resulting color is yellow. Therefore, the regions coloured yellow indicates the pixels that the model correctly predicted as cracks. Secondly, the regions coloured red indicate pixels that the model wrongly predicted as cracks and finally the regions coloured green are those cack pixels could not pick up.



Input Image (Test)

Overlay (Ground Truth in Green, Prediction in Red)

Figure 5.2: Overlayed Screenshot of Inference Result 1

The results inferred that the U-Net was able to learn the features and as a result could segment cracks in any image that contains crack regions. Below are additional inference screenshots.

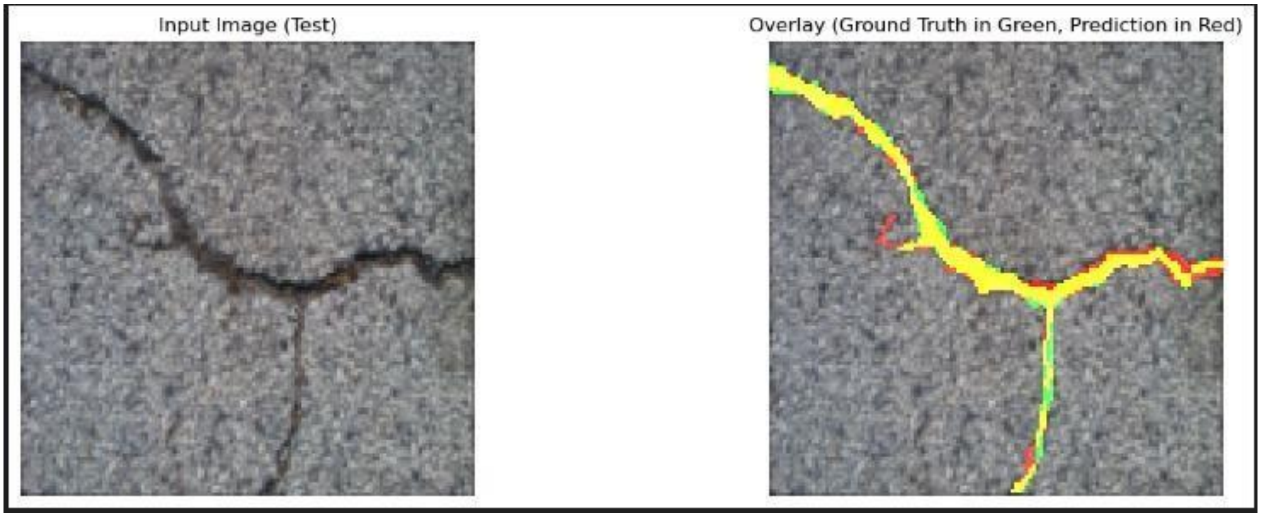


Figure 5.3: Overlayed Screenshot of Inference Result 2

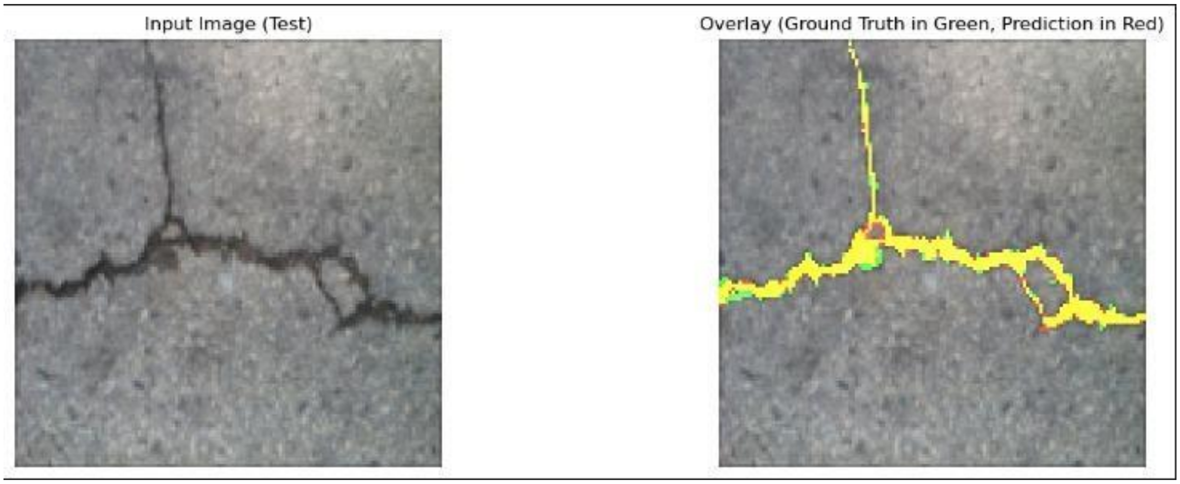


Figure 5.4: Overlayed Screenshot of Inference Result 3

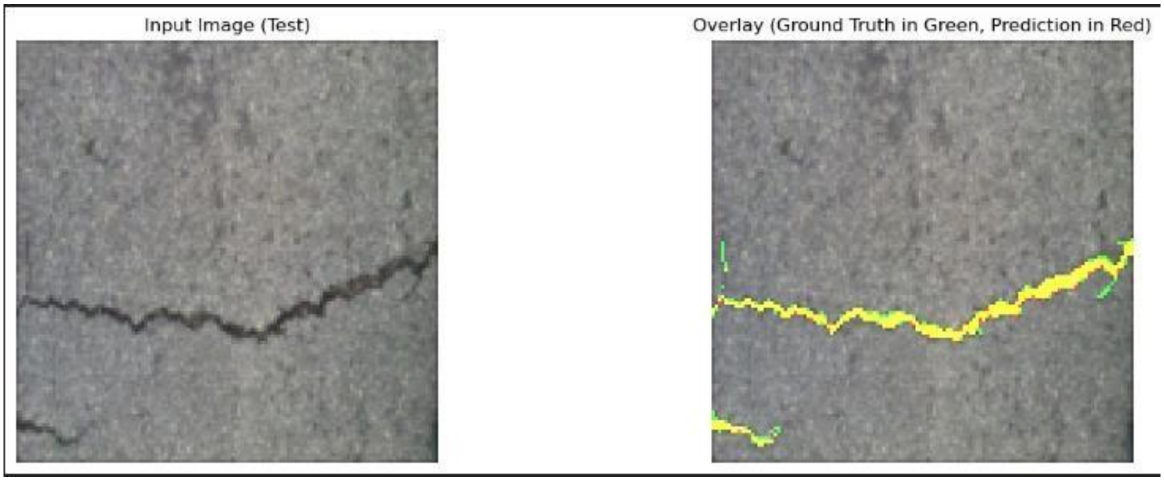


Figure 5.5: Overlayed Screenshot of Inference Result 4

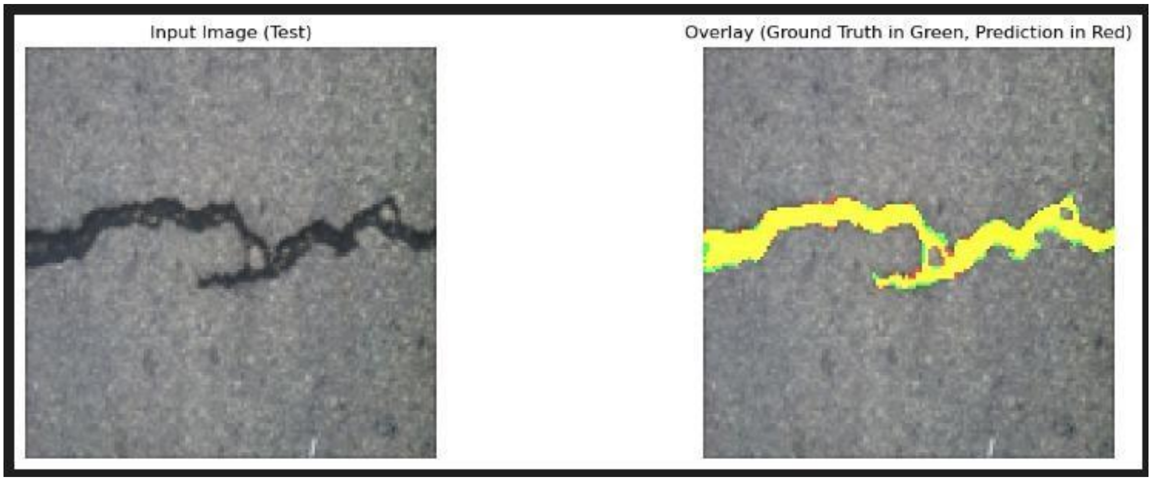


Figure 5.6: Overlayed Screenshot of Inference Result 5

5.2. Discussion

5.2.1. Analysis of Testing Results

Table 3 presents the results on the testing set, providing the most crucial evaluation of the model's performance on unseen data. Similar to the validation results, the testing accuracy remained consistently high, ranging from 98.51 to 98.67. The precision, recall, F-score, and IoU scores demonstrated variations based on the hyperparameters. The model with a batch size of 16 and image dimensions of 256 x 256 achieved the highest IoU score of 66.49, signifying its effectiveness in accurately segmenting cracks on new data.

5.2.2. Impact of Hyperparameters

The experimental results highlight the influence of hyperparameters on the U-net model's performance. The choice of image size, batch size, number of filters, and bottleneck dropout all contributed to variations in the evaluation metrics.

Key observations regarding the impact of hyperparameters include:

- Image size: Increasing the image size from 128 x 128 to 256 x 256 generally resulted in improved performance, particularly in terms of IoU score. Larger images provide more contextual information, allowing the model to better capture crack features and boundaries.
- Batch size: The optimal batch size varied depending on other hyperparameters. In some cases, smaller batch sizes led to better results, potentially due to improved generalization. However, in other cases, larger batch sizes were beneficial.
- Number of filters: Adding more filters to the U-net architecture generally improved performance, particularly the IoU score. Increasing the number of filters enhances the model's capacity to learn complex features, leading to more accurate segmentation.
- Bottleneck dropout: Introducing a dropout layer at the bottleneck of the U-net architecture improved performance in some experiments, suggesting its effectiveness in mitigating overfitting and enhancing generalization.

The findings of this study demonstrate the effectiveness of the U-net architecture for automated pavement crack segmentation. The model achieved high accuracy and promising IoU scores on both the validation and testing sets, indicating its ability to generalize to unseen data. The choice of hyperparameters played a significant role in optimizing performance, emphasizing the importance of careful hyperparameter tuning. The results suggest that U-net based crack segmentation systems have the potential to significantly improve road maintenance and safety inspections by enabling automated, efficient, and accurate crack detection.

Conclusion

In conclusion, this thesis highlights the transformative potential of deep learning, particularly through the U- Net architecture, in addressing the pressing challenges of road maintenance. The model developed showed impressive evaluation and inference results in segmenting pavement cracks. The study validates the effectiveness of U-Net in learning complex patterns and accurately distinguishing between crack and non- crack pixels. This capability underscores the model's utility in tackling real-world infrastructure issues. The research further emphasizes the critical role of hyperparameter tuning in optimizing model performance. Key factors such as image size, batch size, the number of filters, and dropout layers significantly influence the accuracy and efficiency of crack segmentation, with larger image sizes (256x256) yielding notably better results. This insight provides valuable guidance for future advancements in similar applications of deep learning.

Beyond its technical contributions, the study demonstrates the immense potential of automated crack detection systems to revolutionize road maintenance practices. By reducing reliance on time-consuming, error-prone manual inspections, these systems enhance efficiency, improve safety, and minimize risks to inspection personnel. This aligns with broader goals of modernizing infrastructure management and ensuring sustainable road safety solutions. This work also makes a meaningful contribution to the growing body of research on AI-driven infrastructure maintenance. It showcases the application of deep learning techniques to a critical problem—segmentation of pavement cracks—and lays a strong foundation for future studies. The findings are particularly relevant for regions like Northern Cyprus, where high traffic volumes and limited infrastructure exacerbate road maintenance challenges.

Ultimately, this thesis not only confirms the feasibility and efficiency of U-Net-based crack segmentation but also opens pathways for further exploration. Future research could focus on extending the model's capabilities to estimate crack dimensions, enable real-time detection, and enhance robustness under diverse environmental conditions. These advancements hold the promise of making automated road maintenance systems even more practical, scalable, and impactful in addressing global infrastructure needs.

References

1. Elghaish, F., Talebi, S., Abdellatef, E., Matarneh, S. T., & others. (2021). Developing a new deep learning CNN model to detect and classify highway cracks. Journal of Engineering Design and Technology. https://doi.org/10.1108/JEDT-04-2021-0192
2. Gao, Z., Peng, B., Li, T., & Gou, C. (2019). Generative Adversarial Networks for Road Crack. International Joint Conference on Neural Networks., 8.
3. Jia, S. (2023). Semantic segmentation of pavement cracks based on an improved U-Net. Journal of Computing and Electronic Information Management, 10(3), 5.
4. Lau, S. L., Chong, E. K., Yang, X., & Wang, X. (2020, June 19). Automated Pavement Crack Segmentation Using U-Unet-Based Convolutional Neural Network. IEEE, 88, 8.
5. Lee, D., Kim, J., & Lee, D. (2019). Robust Concrete Crack Detection Using DeepLearning- Based. International Journal of Aeronautical and Space Sciences (2019) 20:287–299, 287–299. doi:https://doi.org/10.1007/s42405-018-0120-5
6. Pereira, V., Tamura, S., Hayamizu, S., & Fukai, H. (2019). Semantic Segmentation of Paved Road and Pothole Image Using U-Net Architecture. IEEE, 4.
7. Wang, J.-J., Liu, Y.-F., Nie, X., & Mo, Y. (2021). Deep convolutional neural networks for semantic segmentation of cracks. Willet, 18. doi:https://doi.org/10.1002/stc.2850
8. Zhang, X., Rajan, D., & Story, B. (2019). In this paper, we consider context-dependent deep convolutional semantic segmentation network for cracking in infrastructure. Old crack-detection methods based on vision are not precise and cannot generalize well to various situations. These limitations a. Willey, 21. doi:10.1111/mice.12477
9. Li, P., Xia, H., Zhou, B., Yan, F., & Guo, R. (2022). A method to improve the accuracy of pavement crack identification by combining a semantic segmentation and edge detection model. Applied Sciences, 12(9), 4714. https://doi.org/10.3390/app12094714
10. Zhang, Y., Gao, X., & Zhang, H. (2023). Deep learning-based semantic segmentation methods for pavement cracks. Information, 14(3), 182. https://doi.org/10.3390/info14030182
11. Di Benedetto, A., Fiani, M., & Gujski, L. M. (2023). U-Net-based CNN architecture for road crack segmentation. Infrastructures, 8(5), 90. https://doi.org/10.3390/infrastructures8050090
12. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 234-241). Springer, Cham.
13. Panella, F., Lipani, A., & Boehm, J. (2022). Semantic segmentation of cracks: Data challenges and architecture. Automation in Construction, 135, 104110. https://doi.org/10.1016/j.autcon.2021.104110
14. Al-Huda, Z., Peng, B., Algburi, R. N. A., Al-antari, M. A., AL-Jarazi, R., & Zhai, D. (2023). A hybrid deep learning pavement crack semantic segmentation. Engineering Applications of Artificial Intelligence, 122, 106142.https://doi.org/10.1016/j.engappai.2023.106142
15. Nguyen, N. H. T., Perry, S., Bone, D., Le, H. T., & Nguyen, T. T. (2021). Two-stage convolutional neural network for road crack detection and segmentation. Expert Systems with Applications, 186, 115718. https://doi.org/10.1016/j.eswa.2021.115718
16. Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T., & Wang, P. (2020). Pavement crack detection and recognition using the architecture of SegNet. Journal of Industrial Information Integration, 18, 100144. https://doi.org/10.1016/j.jii.2020.100144
17. Draelos, R. (2019, April 13). The history of convolutional neural networks. Glass Box. https://glassboxmedicine.com/2019/04/13/a-short-history-of-convolutional-neural- networks/

18. Fang, F., Li, L., Gu, Y., Zhu, H., & Lim, J.-H. (2020). A novel hybrid approach for crack detection. Pattern Recognition, 107, 107474. https://doi.org/10.1016/j.patcog.2020.107474
19. Ali, R., Chuah, J. H., Talip, M. S. A., Mokhtar, N., & Shoaib, M. A. (2021). Automatic pixel-level crack segmentation in images using fully convolutional neural network based on residual blocks and pixel local weights. Engineering Applications of Artificial Intelligence, 104, 104391. https://doi.org/10.1016/j.engappai.2021.104391
20. Ali, R., Chuah, J. H., Talip, M. S. A., Mokhtar, N., & Shoaib, M. A. (2021). Automatic pixel-level crack segmentation in images using fully convolutional neural network based on residual blocks and pixel local weights. Engineering Applications of Artificial Intelligence, 104, 104391. https://doi.org/10.1016/j.engappai.2021.104391
21. Yan, Y., Zhu, S., Ma, S., Guo, Y., & Yu, Z. (2022). CycleADC-Net: A crack segmentation method based on multi-scale feature fusion. Measurement, 204, 112107. https://doi.org/10.1016/j.measurement.2022.112107
22. Xu, B., & Liu, C. (2022). Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples. Measurement, 196, 111219. https://doi.org/10.1016/j.measurement.2022.111219
23. Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T., & Wang, P. (2020). Pavement crack detection and recognition using the architecture of SegNet. Journal of Industrial Information Integration, 18, 100144. https://doi.org/10.1016/j.jii.2020.100144