Leveraging Perspective Transformation for Enhanced Pothole Detection in Autonomous Vehicles

by

Abdalmalek Aburaddaha

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Computer Engineering)
in the University of Michigan-Dearborn
2024

Master's Thesis Committee:

Associate Professor Samir Rawashdeh, Chair
Associate Professor Abdallah Chehade
Associate Professor Paul Watta

Abdalmalek Aburaddaha
abdmalek@umich.edu
ORCID iD: 0009-0004-8323-7501

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to several individuals whose support and contributions were invaluable throughout the process of completing this thesis.

First and foremost, I am deeply indebted to my supervisor, Professor Samir Rawashdeh, for his unwavering guidance, insightful feedback, and continuous encouragement. His profound knowledge, dedication, and mentorship have been instrumental in shaping my research and pushing me to achieve my full potential.

I extend my heartfelt appreciation to Dr. Zaid Alshair for his invaluable assistance and support throughout this project. His expertise and willingness to offer advice have been truly remarkable, and I am grateful for his contributions.

I would also like to acknowledge the unconditional love and support of my family, especially my parents and my beloved wife, Haneen. Their unwavering belief in me, patience, and sacrifices have been a constant source of motivation, enabling me to persevere through the challenges encountered during this journey.

Finally, I am thankful to all those who have directly or indirectly contributed to the completion of this thesis, whether through insightful discussions, technical assistance, or moral support.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ACRONYMS

ADAS Advanced Driver-Assistance Systems

AP Average Precision

AR Average Recall

AV Autonomous Vehicles

CNNs Convolutional Neural Networks

D.I. Destination Image

FR-CNN Faster Regional Convolutional Neural Networks

GANs Generative Adversarial Networks

GPUs Graphics Processing Units

HE Homography Estimation

IoU Intersection over Union

LiDAR Light Detection and Ranging

mAP mean Average Precision

MDOT Michigan Department of Transportation

MRTH Ministry of Road Transport and Highways

NHTTSA National Highway Traffic Safety Administration

NMS Non-Maximum Suppression

RANSAC Random Sample Consensus

RADAR Radio Detection and Ranging

ResNet Residual Network

RGB Red, Green, and Blue

RoI Range of Interest

RT Response Time

SGD Stochastic Gradient Descent

S.I. Source Image

SoTA State-of-The-Art

SSD Single Shot Detection

SV Stereo Vision

TP True Positives

FP False Positives

FN False Negatives

RPN Region Proposal Network

YOLOv5 You Only Look Once version 5

ABSTRACT

Poor road conditions, often resulting from inadequate maintenance or adverse weather, are a significant factor in road accidents. Additionally, human reaction time poses limitations in responding to unexpected hazards like potholes, which pose a considerable safety risk. Early detection of distant potholes is crucial to allow drivers, both human and autonomous, to react appropriately by taking avoiding actions or reducing speed, thereby minimizing vehicle damage and potential accidents. This thesis proposes a unique approach for improved pothole detection, particularly for distant ones, by leveraging the well-established You Only Look Once version 5 (YOLOv5) object detection model in conjunction with perspective transformation. This technique effectively enhances the visual prominence of distant potholes by virtually bringing them closer, facilitating their detection, and improving feature extraction, even under varying illumination conditions. Our approach achieves significant improvements in several metrics, exceeding 5% in terms of mean Average Precision (mAP) at various Intersection over Union (IoU) thresholds (0.5, 0.75, and 0.5–0.95) and recall. To the best of our knowledge, this is the first work to specifically address the challenge of distant pothole detection and utilize perspective transformation as a targeted approach to address this issue.

CHAPTER 1

Introduction

1.1 Pothole Detection Importance in Autonomous Vehicles

Potholes present a significant hazard to both vehicles and their occupants, potentially causing damage and posing safety risks [20, 17]. This underscores the importance of their detection, a task that is crucial for the operation of Autonomous Vehicles (AV). As AV are poised to transform the transportation landscape, their safety, and functioning hinges on their ability to effectively identify hazards as early as possible and have enough time for the vehicle or driver to respond. From the viewpoint of an AV, potholes are often treated as static objects, and they pose a threat to road users, especially at high velocities. Therefore, the implementation of an automated pothole detection method can generate a vast amount of data that can be utilized as input for highway maintenance packages [18, 4]. These packages can aid in the optimization of maintenance planning.

1.2 Road Accidents Due to Poor Road Conditions

As per a 2011 report on road accidents in India by the Ministry of Road Transport and Highways (MRTH), a total of 142,485 individuals lost their lives due to road accidents. Out of these, nearly 1.5% or approximately 2200 fatalities, were attributed to poor road conditions [4].

In 2020, the U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTTSA) recorded a sobering statistic: 38,824 lives were lost in traffic accidents nationwide.

This figure represents the highest number of fatalities since 2007. The fatality rate per 100 million vehicle miles travelled also surged to 1.34, marking a significant 21% increase from the previous year and reaching its highest level since 2007. Shockingly, about one-third of the roughly 33,000 annual traffic fatalities are attributed to poor road conditions [1]. Furthermore, 27% of major urban roads, including interstates, freeways, and arterial routes, are classified as substandard, providing drivers with unpleasantly bumpy rides. Only a mere 31% of roads are considered to be in good condition. The financial toll of potholes is also staggering, with drivers bearing a hefty $26.5 billion in vehicle damage costs in 2021 alone.

In 2021, Michigan experienced 1,131 traffic crash fatalities, representing a 4% increase from the 1,083 fatalities recorded in 2020 [14]. This marked the highest number of traffic fatalities in the state since 2005, when 1,129 fatalities were reported. Additionally, the number of crashes in Michigan rose from 245,432 in 2020 to 282,640 in 2021, indicating a significant 15% increase [14]. A study revealed that 40% of Michigan's roads are in poor or mediocre condition, with one in ten bridges classified as structurally deficient. This poor road infrastructure costs the average driver approximately $646 per year in vehicle repairs [8]. Michigan's pothole problem is particularly severe, with the state ranking third in the nation for the worst potholes, based on Google search statistics [5, 28]. To address this issue, the Michigan Department of Transportation (MDOT) allocates $6 million to $7 million annually to fill around 400,000 potholes. These statistics underscore the urgent need for effective pothole detection and response systems in AV to improve road safety.

1.3 Human Response to Potholes

When a driver encounters a pothole, the human Response Time (RT) to apply the brakes can vary. For simple tasks, the average human RT is often quoted as 0.2 seconds [39, 2]. However, for more complex tasks such as emergency braking when a pothole is detected, the RT is longer. It's important to note that these times can be greatly affected by the driver's alertness and expectation of the need to brake [30]. In contrast to human drivers, AV equipped with pothole detection systems

can identify and respond to potholes in real-time. These systems can modulate the vehicle's speed and positioning upon detecting a potential pothole, ensuring the vehicle stays within its lane. This rapid RT can significantly enhance road safety by enabling the vehicle to take immediate action to either avoid the pothole or minimize the impact.

1.4 Pothole Detection Methods in Autonomous Vehicles

The implementation of pothole detection systems in AV offers several advantages. Not only can it improve road safety by reducing the risk of accidents caused by potholes, but it can also contribute to more efficient road maintenance by providing accurate and timely data on pothole locations and severity. Moreover, detecting and avoiding potholes can reduce the fuel consumption, wear and tear, and maintenance cost of a vehicle [3]. In addition, it can indirectly decrease the total travel time in some cases [3].

A pothole is a common road surface issue, typically found in asphalt pavements, caused by a combination of factors including traffic wear and the effects of water. The process begins with water weakening the soil beneath the road, followed by traffic repeatedly passing over the weakened area, causing the pavement to break and dislodge. This action results in the formation of a depression or hole in the road surface as shown in figure 1.1, [16], which can pose hazards to vehicles and pedestrians alike. Potholes can exhibit a wide range of characteristics, with varying degrees of severity and potential hazards to vehicles and road users. As depicted in the figure 1.1, some potholes may possess substantial depth, creating a significant risk of damage or accidents for any vehicles traversing them. These deep cavities in the road surface can potentially cause loss of control, suspension damage, or even complete immobilization of vehicles, posing a severe threat to the safety of occupants and other road users alike.

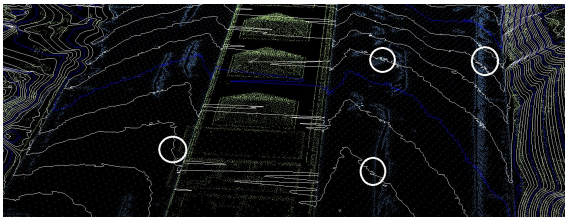
Moreover, the hazardous nature of potholes is further compounded when they become obscured by the presence of water. In such cases, the potholes are effectively rendered invisible to the human eye, significantly hindering their detection and avoidance. This phenomenon introduces an

Figure 1.1: Potholes Formation.[16]

additional layer of danger, as unsuspecting road users may inadvertently encounter these concealed obstacles, potentially leading to serious consequences.

Potholes are detected and observed in different ways, counting Human detection, Sensor based detection, Vibration based detection, and Vision-based detection which this paper is focusing on. The traditional method of pothole detection involves human inspection, figure 1.2b [38].

This method is labor-intensive, time-consuming, and can be dangerous for the inspectors. It also lacks consistency due to human error and is not efficient for large-scale detection [18]. Whereas Sensor-based detection methods, such as Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR), use electromagnetic waves to detect potholes. LiDAR uses light waves showing in figure 1.2a, [13], providing high-resolution data and precision [32]. However, it can be expensive, especially for small-scale applications [21]. RADAR, on the other hand, uses radio waves and is superior in terms of cost and ability to monitor large areas [32]. However, it has



(a) Pothole Detection by Lidar [13]

(b) Potholes Inspection Manually [38]

Figure 1.2: Potholes Detection Methods

a lower resolution compared to LiDAR, making it challenging to track and distinguish objects in crowded environments [25].

Vibration-based detection methods use accelerometers to detect potholes based on the vibration information of the acceleration sensors [18]. This method is cost-effective and suitable for real-time processing. However, it has limitations in providing the exact shape of potholes and could provide incorrect results, as road joints and hinges can be misidentified as potholes [12]. Moreover, this method is not suitable for detecting potholes in order to avoid or act towards reducing their effect on the vehicle. On the other hand, Computer vision techniques for pothole detection have gained popularity due to the accessibility and feasibility of cameras. These techniques use images or videos as input data and apply image-processing and deep learning techniques to detect potholes. While each method has its strengths and weaknesses, computer vision techniques offer significant advantages in terms of cost-effectiveness, precision, and the ability to integrate with other data sources for pothole detection.

1.5 General Perspective on Computer Vision for Robotics and Autonomous Vehicles

In robotics research, the goal of giving machines the ability to see their environment like humans do has long been pursued. Rule-based methods were the mainstay of early computer vision systems for robotics and AV. These techniques made use of manually designed features and algorithms created especially for regulated settings. Moreover, simple sensors like infrared or ultrasonic range finders were used to tackle tasks like obstacle avoidance and line following. Although these early approaches proved effective in controlled environments, they were not as flexible and adaptive as what is needed in real-life situations, which are by their very nature complicated and variable. Robotics computer vision has entered a new age with the introduction of powerful computing resources and advances in machine learning methods. A step toward more reliable object recognition was the introduction of methods like template matching, which compared pre-defined object templates with image data. These techniques, however, continued to have trouble with changes in perspective, lighting, and object appearance.

Convolutional Neural Networks (CNNs), in particular, and deep learning brought about the real revolution in computer vision. The architecture and operation of the biological visual cortex served as an inspiration for the development of CNNs in the 1980s. But early CNNs were constrained by the processing power available at the time and computationally costly. Furthermore, a lot of labelled data—which was frequently hard to come by and expensive to produce—was needed for them to be trained. However, early in the 2010s, a number of events came together to cause a turning point for CNNs. First off, the emergence of powerful GPUs (Graphics Processing Units) and other hardware breakthroughs greatly sped up the CNN training process. Second, the massively labelled datasets that were readily available—images with millions of labels, like ImageNet—provided the fuel needed to train these algorithms that were insatiably thirsty for data. Ultimately, the 2012 development of more complex CNN architectures, such as AlexNet, demonstrated the superior image recognition capability of deep learning. This breakthrough led to a

rapid advancement in CNNs. Researchers developed deeper and more complex architectures like VGG16 and Residual Network (ResNet), further pushing the boundaries of accuracy. Today, such pre-trained models, are a cornerstone of computer vision for robotics and AV. These models are trained on massive general image datasets and then fine-tuned on task-specific datasets related to robotics or AV. This approach allows them to leverage the vast knowledge learned from generic image data and specialize in the specific objects and scenarios relevant to the robotic or AV application. Hence, the benefits of pre-trained models are substantial, such as:

- **Faster Training**: Fine-tuning a pre-trained model on a smaller, task-specific dataset is significantly faster and requires less data compared to training a CNN from scratch. This is crucial for robotics and AV where acquiring large amounts of labelled data can be time-consuming and expensive.

- **Superior Performance**: Pre-trained models learn low-level features like edges and textures from the general image dataset, which serve as a strong foundation for learning task-specific features on the smaller, specialized dataset. This often leads to superior performance compared to training a CNN from scratch on the limited task-specific data alone.

- **Improved Generalizability**: By leveraging pre-trained knowledge, the model can better adapt to variations in lighting, perspective, and object appearance within the specific domain, leading to more robust and generalizable performance in real-world scenarios.

The evolution of CNNs, from their early limitations to the powerful pre-trained models of today, has been a game-changer for computer vision in robotics and AV. As deep learning research continues to push the boundaries, we can expect even more sophisticated and robust vision systems that will empower robots and AV to navigate the world with ever-greater autonomy and safety.

1.6 Challenges and Proposed Method

1.6.1 Challenges

This study delves into a novel approach for enhancing distant pothole detection using computer vision techniques. The primary goal is to provide ample reaction time for drivers, Advanced Driver-Assistance Systems (ADAS) equipped vehicles, and AV by enabling the early identification of these road hazards.

Detecting potholes from a distance using cameras presents a significant challenge for several reasons, impacting the effectiveness of computer vision algorithms. The following is a detailed breakdown of some of these challenges:

Reduced Pixel Size: As the distance between the vehicle and the pothole increases, the pothole occupies a smaller area within the captured image. This translates to fewer pixels representing the pothole, hindering the ability of computer vision algorithms to extract crucial features like depth and shape.

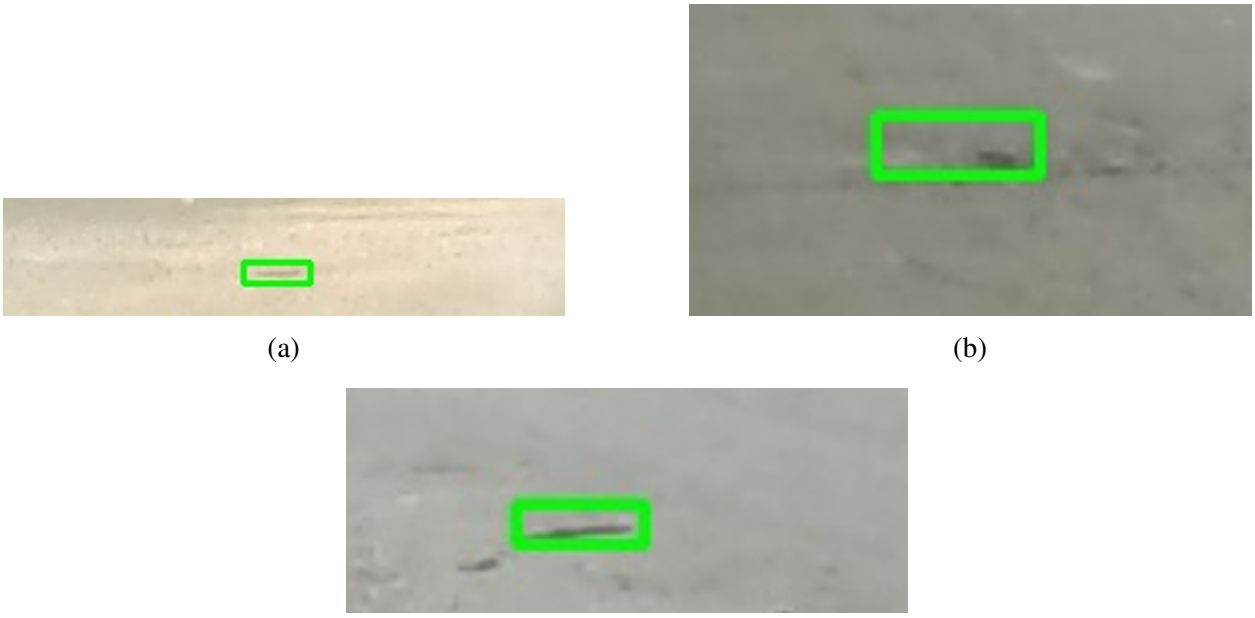
Considering the following equation for image resolution:

$$\text{Pixel size (mm)} = \frac{\text{Sensor size (mm)}}{\text{Image resolution (pixels)}} \tag{1.1}$$

where:

- **Pixel size (mm)**: The physical size of each pixel in millimeters.
- **Sensor size (mm)**: The physical size of the sensor in millimeters.
- **Image resolution (pixels)**: The number of pixels in the image, representing its width or height.

In this scenario, the sensor size remains constant, but as the distance between the camera and the pothole increases, the pothole occupies a smaller portion of the image frame, effectively reducing its image resolution. This reduction in resolution translates to fewer pixels representing the pothole,



(a) (b)

(c)

Figure 1.3: Data Limitations Examples

leading to a loss of detail. Consequently, algorithms struggle to differentiate between potholes and other road imperfections like cracks, shadows, or uneven pavement textures. This limitation can significantly impact the accuracy of pothole detection.

Limited Detail: Distant potholes appear smaller and less defined in the image. Examples are shown in 1.3. This lack of detail makes it challenging to identify crucial features essential for accurate detection. Important details such as:

- **Cracks**: Cracks surrounding or emanating from the pothole are vital indicators of its severity and potential for growth. However, at a distance, these cracks become blurred or invisible, making it difficult for algorithms to distinguish them from background noise.
- **Edges**: The edges of a pothole define its shape and size. However, distant potholes often appear with blurry or indistinct edges, hindering the ability of algorithms to accurately localize the pothole and determine its dimensions.
- **Depth**: Depth information is crucial for assessing the severity of a pothole and the potential

risk it poses. However, at a distance, monocular cameras (single camera systems) struggle to capture depth information effectively.

The limitations in capturing these details can lead to missed detections of actual potholes or false positives where other road imperfections are misidentified as potholes. Both scenarios can compromise safety on the road.

1.6.2 Proposed Method: Distant Pothole Detection with Vision and Perspective Transformation

This study tackles the challenges of distant pothole detection by introducing a novel approach that leverages a technique called perspective transformation. Perspective transformation is a powerful mathematical tool in computer vision,[40], which alters the viewing angle of potholes from the perspective of the vehicle or driver to a bird's-eye view. This alteration is based on the bounding box values and pixel locations of the potholes. By bringing distant potholes closer and magnifying the targeted objects, this technique improves feature extraction. By applying perspective transformation, we essentially create a magnified and standardized view of the pothole region within the image. This enhanced view empowers the computer vision algorithm to extract crucial features like cracks, edges, and depth variations more effectively, leading to a significant improvement in the accuracy of pothole detection, especially for those located at a distance. To the best of our knowledge, no previous study has exploited perspective transformation in this manner for such an application.

The remainder of this work is structured as follows

- **Chapter 2**: Offers a review of the background and pertinent literature.
- **Chapter 3**: Delves into the data utilized, and implementation.
- **Chapter 4**: Presents the findings of these experiments along with the evaluation metrics used.

- **Chapter 5**: The final chapter encapsulates the discussion, conclusion and outlines potential avenues for future research.

CHAPTER 2

Related Work

Many studies have explored various computer vision and image processing techniques for automated pothole detection from visual road imagery. The following sections cover some approaches many papers implemented to tackle the pothole detection problem.

2.1 Classical and Basic Vision Approaches

Nienaber et al. [24] propose an algorithmic approach by combining road colour modelling and Canny edge filtering to identify pothole contours, reporting 81.8% precision and 74.4% recall on test datasets captured via a Go-Pro camera. Their method leverages the intensity contrast and distinct edges surrounding potholes to delineate them accurately. This approach provides a lightweight pothole detection solution without needing expensive sensors or deep learning architectures. However, the reliance purely on edge and colour attributes may limit generalizability across diverse road types and lighting. Dynamic entities on roads can also impede performance and the solution was not validated for real-time usage. Nonetheless, it serves as useful preliminary evidence for the potential of basic computer vision techniques. We build on this study by examining more robust learning-based

Chen et al. [7] attempted a deep learning approach using location-aware convolutional neural networks (CNNs) for classifying potholed road regions from imagery. Their framework first selects pothole candidate areas using a Region Proposal Network (RPN), then classifies crops from those regions with a binary classifier – achieving a classification accuracy of 95.2%. Compared

to classical computer vision methods, this demonstrates the power of deep CNN models for learning robust pothole attributes. However, the two-stage pipeline can add computational expenses. Their approach also does not precisely localize potholes with bounding boxes, limiting real-world usability without further post-processing. Moreover, the method was also less effective on low-resolution images. We aim to overcome these limitations using more advanced one-shot detection models.

Pereira et al. [26] proposed a basic 4-layer CNN for pothole detection, claiming accuracy and precision of 99.8% and 100%, respectively. However, key limitations exist, including: the use of a basic CNN approach, which is probably slower than using a transfer learning model like You Only Look Once (YOLO) and might not be able to learn many features of complex objects like potholes. The lack of visual information to describe the model's performance and evaluation, and the absence of a separate test set also limit the study's comprehensiveness.

2.2 Two Stages Object Detection and Stereo Vision Approaches

In the study conducted by Abhishek Kumar [19], the Faster Regional Convolutional Neural Networks (Faster R-CNN) model was used for the detection of potholes, claiming the method can detect potholes at 100 meters. However, no details or proof were provided to substantiate this claim. Faster R-CNN, a popular model for object detection tasks. However, the paper did not provide any details about the training configuration or hyper-parameters used, which are crucial for understanding and replicating the study. Additionally, no quantitative results, such as accuracy or precision, are provided to demonstrate the performance of the approach, making it difficult to evaluate the effectiveness of the proposed method. A more rigorous evaluation of diverse test footage, with both quantitative and qualitative results, would be needed to fully assess the capabilities of this pothole detection technique. Nonetheless, the use of deep learning and models like Faster R-CNN shows promise for automated analysis of road conditions from video data. Further work building on these concepts and addressing the limitations above could advance the State-of-The-Art (SoTA)

in pothole detection.

Dhiman et al. [11] presented multiple approaches for pothole detection: Stereo Vision techniques SV1 and SV2, analysing road elevation and depth variation to identify defects; and deep learning methods LM1 and LM2 are based on Mask R-CNN and YOLOv2. SV1 and SV2 achieved 92.5% and 93.7% accuracy, respectively, on a 1,000 stereo image dataset. LM1 obtained 0.6 Average Precision (AP) at 0.5 IoU after 400 epochs. This AP is slightly higher than that achieved by other methods with fewer epochs. Moreover, it is still doubtful that the use of R-CNN along with other methods is suitable for real-time applications, on the contrary of using YOLOv5, as will be introduced in our study. On top of that, the accuracy of the SV techniques depended on the quality of the stereo images and the calibration of the cameras. Noise, distortion, or misalignment in the images could affect depth estimation and pothole detection.

Amita Dhiman's study [10] introduced a method for detecting significant road surface damage using SV. This method involved modelling the road plane directly in the image-disparity space, without the need to project a disparity image into 3D space. The study utilized the Random Sample Consensus (RANSAC) process to identify patterns in the data and fit a plane to these points, selecting the plane with the most in-layers as the best model. The distance from each pixel to this plane was then computed. However, the study did not address the hardware requirements necessary for implementing this model in a real-world scenario, and it was noted that this method might be slow and more computationally intensive that one step object detection

2.3 One Stage Object Detection and Generative Adversarial Networks (GANs) Approaches

In another study, Al-Shaghouri [33] explored real-time pothole detection using YOLOv3 and YOLOv4 models trained on custom datasets. The approaches were evaluated on 5000 images from internet sources and local Lebanon footage. YOLOv4 achieved the best performance with 85% precision, and 85.39% mAP at an IoU threshold of 0.25. However, the low IoU value in-

dicates potential limitations in detecting exact pothole regions. Additionally, model performance across varying distances is unknown. The training was computationally expensive, for 5000 iterations on a GPU. While showing promising accuracy, real-time feasibility and performance across diverse settings requires further investigation.

Maeda et al. [23] proposed using GANs to synthesize additional pothole training data paired with a Single Shot Detection (SSD) model for detection. Experiments showed a 2-5% F-score improvement when synthetic data was below 50% of the real dataset size. However, performance decreased with higher proportions of synthetic images. Despite these promising results, the study had several limitations. The use of GANs introduced computational overhead and increased training time. Moreover, training GANs effectively requires a delicate balance between the generator and discriminator, which can be challenging to achieve and may result in mode collapse or training instability. In addition to that, the reliance on synthetic data generation may introduce uncertainties regarding the model's generalization capabilities to real-world scenarios. Overall, strategic use of synthetic data generation may benefit pothole detection, but many open challenges remain.

An investigation of pothole detection using YOLOv3 and Sparse R-CNN under various challenging conditions was introduced by Bučko et al. [6]. It appears that the same dataset was used for both training and testing, limiting generalizability. Performance was significantly higher in daylight (mAP 0.74-0.79) compared to nighttime (mAP 0.31) and rain (mAP 0.07) at a resolution of 1080x1080. While showing promise for pothole detection in ideal conditions, severe degradation in difficult scenarios exposes limitations. Using identical train and test sets fails to indicate real-world viability. Additional bench-marking on distinct datasets and against current methods would better demonstrate capabilities.

Salaudeen et al. [31] proposed using an image enhancement GAN (ESRGAN) to improve pothole detection. Enhanced images were input to YOLOv5 and EfficientDet for localization. While super-resolution (SR) images increased mean average precision (mAP 50–95) for YOLOv5L (32%) and EfficientDet (26%) compared to low-resolution, which achieved up to 10.6% average precision (AP) with EfficientDet and 12% with YOLOv5, GANs add computational overhead and

risk over-fitting and are highly dependent on the quality of the training data. Furthermore, the use of EfficientDet with YOLOv5 reduced the speed of object detection.

Rastogi et al. [29] Introduced a modified YOLOv2 model for pothole detection to mitigate vanishing gradients and unlearn insignificant features learned during training. Training used smartphone images taken very close to potholes. While achieving 87% precision and 89% recall, reliance on non-representative data is a major limitation. Images captured at close range from above fail to indicate the model's applicability for real-world performance, such as AV.

CHAPTER 3

Methodology and Dataset Preparation

3.1 Methodology

This section talks about the methods and data used in this study. Two methods were used together. The first one is YOLOv5 [37], which is a well-known detection algorithm for real time applications, and the second method is the automated perspective transformation which improves the detection precision and accuracy of the detection model used in this study.

3.1.1 Detection Algorithm-YOLOv5

In the realm of computer vision, object detection has been revolutionized by the advent of deep learning. Among the various models that have emerged, the YOLO (You Only Look Once) series has gained significant attention due to its balance of speed and accuracy. The fifth version, YOLOv5, is a robust deep learning model built on PyTorch. This model is used in this study due to its efficiency, speed, and low computational needs.

YOLOv5 is a member of the YOLO family, and it includes different versions: small (s), medium (m), large(l), and extra-large(x). Each one of them provide better accuracy when the bigger one is used but requires more time to train.

There are several reasons on why we selected YOLOv5 over the other previous versions of YOLO. And here we are showing a better comparison between the YOLOv5 and its predecessors, followed by a table 3.1 that summarizes the comparison.

- **Speed and Accuracy First:**
  - **Lightweight Architecture:** Compared to YOLOv4, YOLOv5's simplified architecture reduces model complexity. This provides faster inference speeds result from this, which makes it appropriate for real-time applications on devices with limited resources. Although YOLOv4 provided some speed enhancements over YOLOv3, YOLOv5 places a higher priority on speed.
  - **Enhanced Training Efficiency:** To enhance training effectiveness, YOLOv5 makes use of strategies like focus modules and data augmentation. As a result, training times can be shortened by enabling quicker model convergence.
- **Flexibility and Usability:**
  - **PyTorch Framework:** The PyTorch framework, which is well-known for being user-friendly and easily customizable, is used by YOLOv5. Because of this, YOLOv5 is more developer friendly than YOLOv4, which is dependent on the Darknet, a less popular framework.
  - **Modular Design:** The design concept of YOLOv5 is modular. Pre-defined building blocks that can easily change out or adjusted to meet unique needs are used to construct the model. This makes it possible to modify the model with greater flexibility to accommodate various hardware platforms or tasks.
- **Additional Benefits:**
  - **Better Prediction of Anchor Boxes:** YOLOv5 uses Anchor Boxes for object detection, nevertheless it has made enhancements in the prediction of these boxes. Compared to earlier versions, this could result in bounding box localization that is more accurate.
  - **Emphasis on Community Development:** YOLOv5's developers place a high value on building a vibrant community around the model. This promotes teamwork, expedites the fixing of bugs, and stimulates the development of new features and expansions.

Table 3.1: Summary Comparison Between YOLOv5 and Its Predecessors, YOLOv4,v3

Feature	**YOLOv3**	**YOLOv4**	**YOLOv5**
Framework	Darknet	Darknet	PyTorch
Focus	Object Detection	Speed and Accuracy	Speed and Accuracy
Architecture	More Complex	Lighter than YOLOv3	Lightest
Training Speed	Slower	Faster than YOLOv3	Fastest
Inference Speed	Slower	Faster than YOLOv3	Fastest
Ease of Use	Less User-Friendly	Less User-Friendly	More User-Friendly
Community Focus	Lower	Lower	Higher

The architecture and stages of YOLOv5 follow a similar architecture to previous YOLO versions, consisting of a backbone network followed by detection heads. The backbone network is responsible for extracting features from the input image, while the detection heads are responsible for predicting bounding boxes and class probabilities. The backbone network in YOLOv5 is based on the CSPDarknet53 architecture, which is a variant of the Darknet neural network. CSPDarknet53 is known for its efficiency and effectiveness in feature extraction, making it well-suited for object detection tasks, and it includes 53 convolutional layers. YOLOv5 uses a modified version of the YOLO head, which consists of a series of convolutional layers followed by a final detection layer. Moreover, the exact number of convolutional layers can vary depending on the configuration of the model (e.g., YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x). The following section shows in more details how YOLOv5 works:

Modern object detection algorithms like YOLOv5 employ a distinct strategy from previous techniques. YOLOv5 completes object detection in a single forward pass through the neural network as opposed to passing through several steps of region proposal and classification. Because of this, it operates extremely quickly and effectively, which is essential for real-time applications like surveillance systems and autonomous driving. As shown in 3.1 The expected input image format is a batch of images as input. Each image is like a puzzle piece. The image shape is (m, 224, 224, 3):

- **m** represents the number of puzzle pieces (or batches).

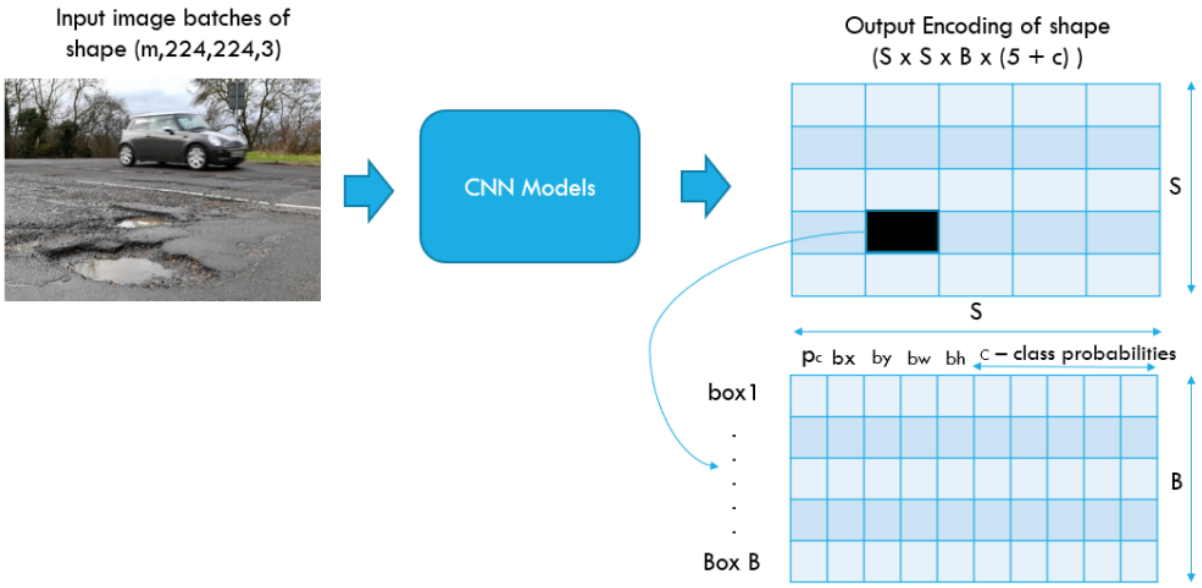


Figure 3.1: One Shot Detection Using YOLOv5 [27]

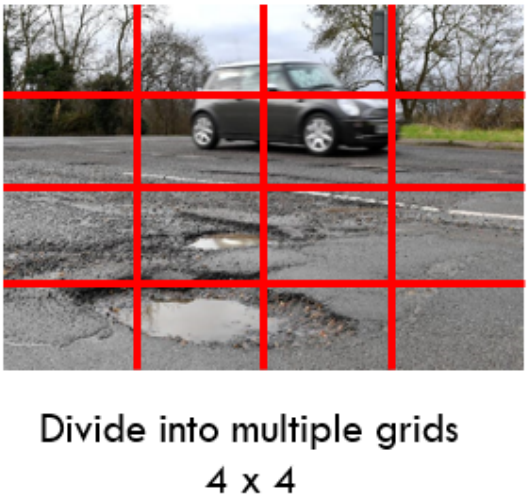
- **224** is the size of each piece (both width and height).
- **3** channels stand for Red, Green, and Blue (RGB) colours.

When you feed an image into the YOLOv5 model, the first thing the model does is divide the image into a grid, let's say a 4x4 grid for example as shown in 3.2a. Each cell in this grid is responsible for detecting objects whose center falls within that cell's boundaries. Instead of analyzing the image in stages, this grid structure enables the model to reason about the full image at once.

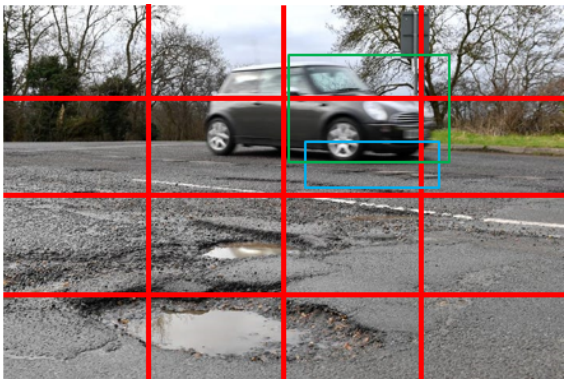
Every cell generates numerous bounding box predictions (assuming two in one cell for simplicity), a confidence score for each box, and a prediction on the presence of an object. Known as "anchor boxes," these bounding boxes are available in multiple aspect ratios to fit a range of sized and shaped items. Throughout the training process, the model is built to pick up on the proper anchor box sizes and shapes, which enables it to adjust to the many item kinds that it must detect.

Let's examine the predictions made by each cell. For each anchor box, the output of the cell is:

- **An object confidence score**: How confident the model is that an object exists within that



(a)



(b)

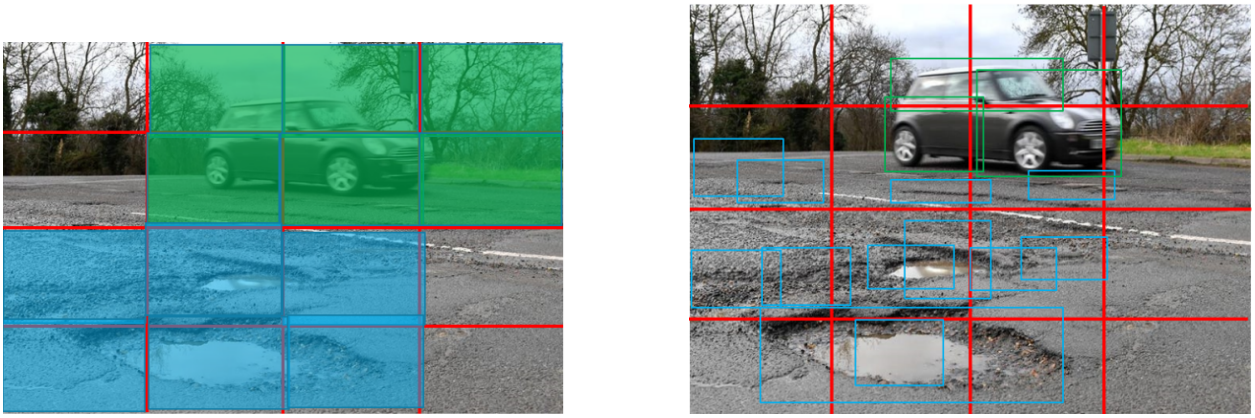
Figure 3.2: YOLOv5 Steps 2 (a) and 3 (b) [27]

box.

- **Bounding box coordinates**: Usually expressed as (x, y, width, height) values, these values indicate the expected box's position and size with respect to the cell.
- **Class probabilities**: A vector of probabilities that indicates the chance of each potential class—car, pedestrian, pothole, etc.—being present in that box.

Therefore, a cell will produce two distinct anchor boxes, each with its own confidence score, bounding box coordinates, and class probability vectors, if it detects both a car and a pothole inside its bounds, as shown in 3.2b.

In addition to predicting the coordinate of the bounding boxes and their confidence scores, a class probability coloured map is predicted as well, which shows the likelihood of the presence of a class in each cell, for example you can see in 3.3a that the vehicle class is in cell number 2, 3, 4 while the pothole class is in cell number 9, 10, 11, and so on. This map enables the network to assign a class map to each of the bounding boxes. Moreover, several overlapping bounding boxes are generated for each cell as shown in the 3.3b. "But what about all those overlapping boxes?" at



(a) (b)

Figure 3.3: YOLOv5 Steps 4 (a) and 5 (b) [27]

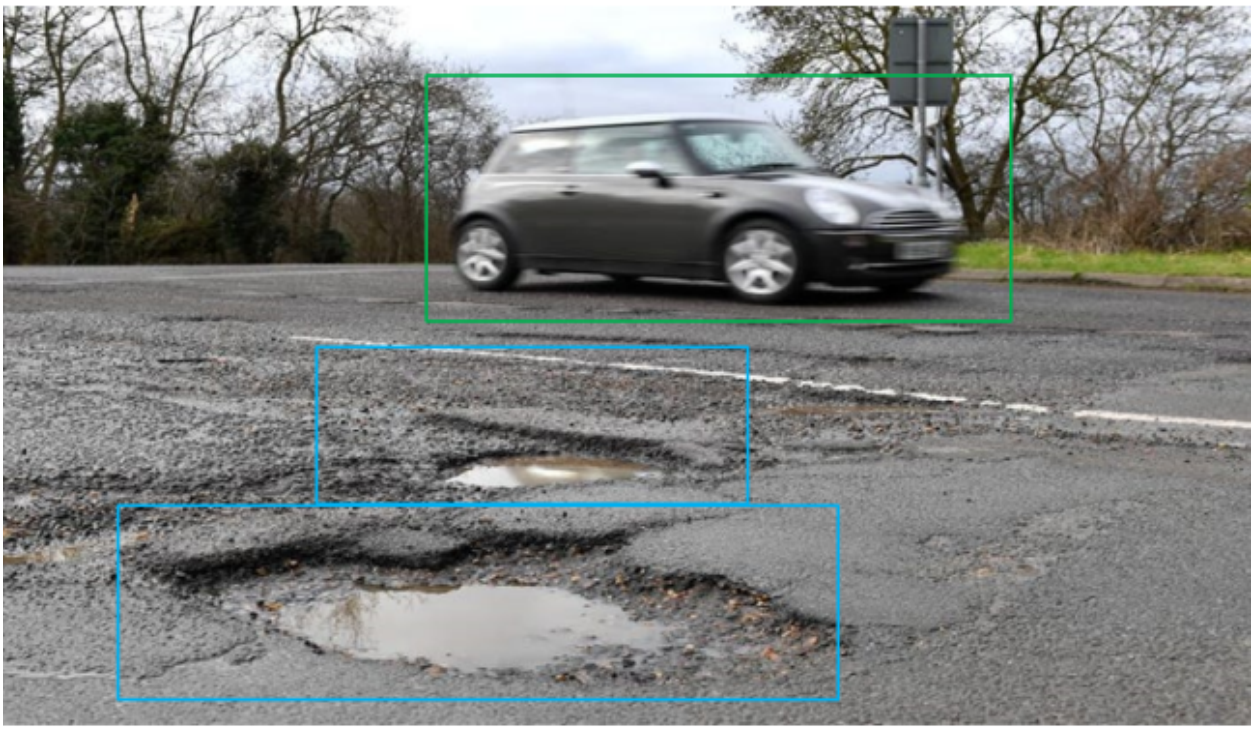


Figure 3.4: Final Detection After Applying NMS to Reduce the Number of Overlapped Bounding Boxes [27]

this point. The Non-Maximum Suppression (NMS) method is used in this situation. NMS is a post-processing stage that retains only the most reliable predictions, removing redundant, overlapping boxes.

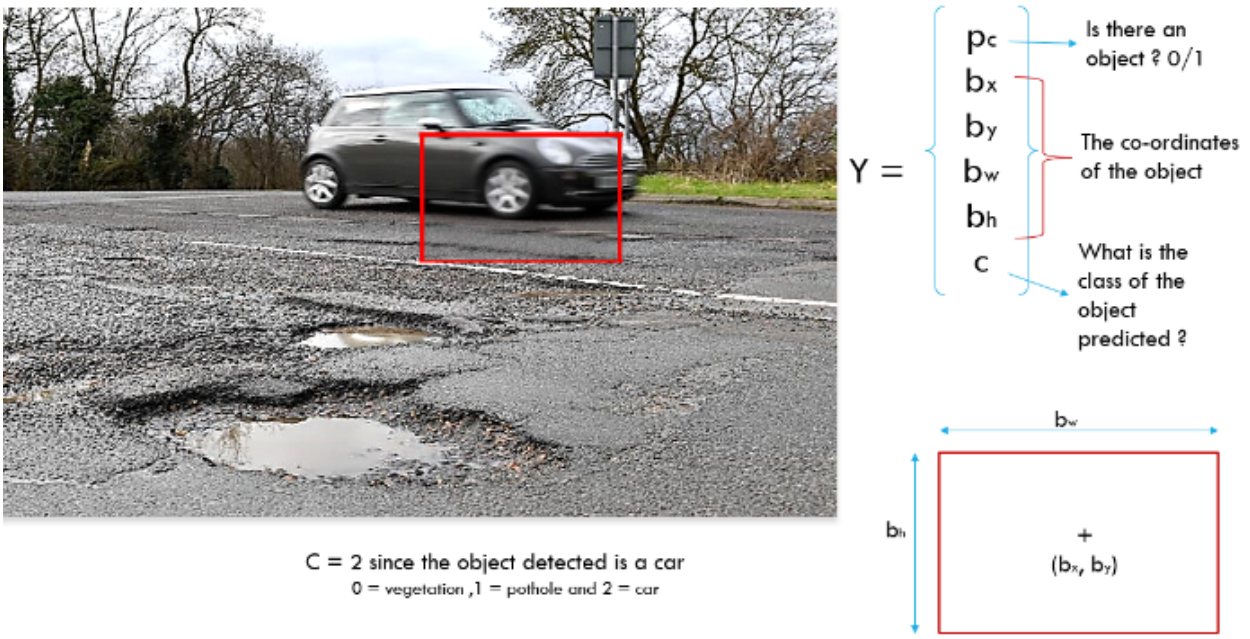


Figure 3.5: Final Output Showing a Detected Vehicle Bounding Box With Its Confidence Level Along With The Bounding Box Coordinates, Dimensions, and Class [27]

A clean set of detections will be created when NMS effectively removes the boxes beneath the most confident ones as shown in 3.4. In order to accomplish this, it computes an intersection-over-union (IoU) score for each pair of boxes. If the IoU is greater than a predetermined threshold, the box with the lower confidence score is suppressed.

YOLOv5 produces a set of bounding boxes as its end result, each having an image's precise coordinates, a confidence score, and a class label, as shown in 3.5. Then, this output can be applied to a number of other tasks, including surveillance, pothole identification on roadways, and autonomous driving.

3.1.2 Perspective Transformation

Perspective transformation is a key technique in the fields of computer vision and image processing, essential for modifying images to represent different viewing angles. This process adjusts the image's perspective to simulate how it would appear if the viewer's position were to shift. [36] Perspective transformation alters an image's viewpoint without retaining angles, lengths, or the

(a) (b)

Figure 3.6: Examples of Perspective transformation. (a) Original image, (b) Transformed image.

parallel nature of lines. However, it upholds the principles of col-linearity and the intersection of lines, ensuring that lines stay straight despite the transformation. The essence of this process lies in a transformation matrix [34]. This matrix recalculates the coordinates of the image's original points, producing a new coordinate set that depicts the same locations within the context of the altered image perspective as shown in figure 3.6.

This technique finds widespread applications, such as rectifying images distorted by wide-angle lenses or unconventional angles, integrating virtual elements into real scenes for augmented reality, enhancing object recognition and tracking accuracy by accommodating perspective changes, and generating 3D models from multiple 2D images captured at different viewpoints. Moreover, the benefits of using this technique include making the far objects closer in our case, and enhancing the presence of the features for every object as shown in figure 3.6. Understanding the process involves several stages. Firstly, feature detection and matching identify and correspond key points like corners and landmarks in both source and target images. Next, homography estimation utilizes these matched features to calculate a transformation matrix (homography) that maps points between the images. Finally, image warping applies the homography matrix to warp the source image pixels, generating the transformed image.[9, 15, 35].

The magic behind perspective transformation lies in a 3×3 matrix called the homography matrix (H). This matrix holds the transformation parameters that map points from the source

image (original perspective) to the target image (desired perspective).

The structure of the homography matrix (H) can be represented as follows:

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Each element (h_{ij}) within the matrix contributes to the transformation:

- h_{11}, h_{12}, h_{13}: Affect the x-coordinate of the transformed point.
- h_{21}, h_{22}, h_{23}: Affect the y-coordinate of the transformed point.
- h_{31}, h_{32}, h_{33}: Homogenization factor (usually set to 1).

The core mathematical relationship between a point in the source image (t) and its corresponding point in the transformed image (t') is described by the Homography Equation as follows:

$$t' = H \cdot t \tag{3.1}$$

Where:

- t': $(x', y', 1)$ represents the coordinates of the point in the transformed image.
- H: 3×3 homography matrix.
- t: $(x, y, 1)$ represents the coordinates of the corresponding point in the source image.

As shown in equation 3.1, It essentially signifies a linear transformation that maps points from one perspective to another based on the calculated homography matrix.

Finally, in a process called Homography Estimation (HE), calculating the homography matrix (H) requires corresponding points between the source and target images. These points are typically identified using feature detection algorithms that locate key-points like corners and edges. Once

enough corresponding points are identified, mathematical techniques estimate the homography matrix that best maps these points from the source to the target image.

In the context of pothole detection for AV, perspective transformation using homography can be particularly advantageous in several aspects as follow:

- **Enhanced Feature Extraction**: By virtually bringing distant potholes closer in the transformed image, homography can improve the extraction of pothole features, leading to more accurate detection, especially for potholes further down the road.
- **Bird's-Eye View Creation**: Homography can be used to warp the road image into a bird's-eye view, potentially simplifying pothole detection algorithms by providing a more standardized view of the road surface.

3.2 Dataset

The dataset provides a realistic representation of South African road conditions from a driver's vantage point [24]. Some examples of the training dataset images from the positive and negative classes are shown in Figures 3.7 and 3.8, illustrating the appearance of pothole damages and intact pavement surfaces captured in the dataset. The positive pothole class comprises 1,119 JPEG images extracted from video footage captured with a GoPro camera mounted inside a vehicle's windshield. The negative non-pothole class is larger 2,658 images which makes the train dataset totalling 3777, and there is 628 images for the test-set as shown in figure 3.9. All images share dimensions of 3680×2760 pixels. As seen in these sample images, factors like slightly different lighting conditions, and road type mimic what automated pothole detection systems integrated into vehicles would encounter in practice.

Figure 3.7 presents three representative image samples from the dataset, illustrating the diverse lighting conditions and pothole characteristics encountered in real-world scenarios. These examples highlight the challenges posed by varying illumination levels and pothole appearances, which must be effectively addressed to achieve robust and accurate pothole detection.

(a) (b) (c)

Figure 3.7: Examples of different images under different lighting conditions. (a) Image with sun rays towards the camera and low light covers the potholes, and (b) Clear light condition, and (c) Shadow covers potholes.

(a) (b)

Figure 3.8: Examples of different negative images. (a), (b) two images with no potholes included.

Firstly, sample (a) depicts a scenario with intense lighting, potentially caused by direct sunlight or strong artificial illumination. In such conditions, the high contrast and strong shadows can introduce complexities in identifying and distinguishing potholes from other road surface features. Effective pothole detection algorithms must be capable of adapting to these challenging lighting scenarios, ensuring reliable performance even in the presence of harsh illumination.

In contrast, sample (b) showcases normal lighting conditions, where the potholes exhibit a different appearance characterized by a sandy composition. This sample underscores the diverse nature of potholes, which can manifest with varying textures, colors, and material compositions.

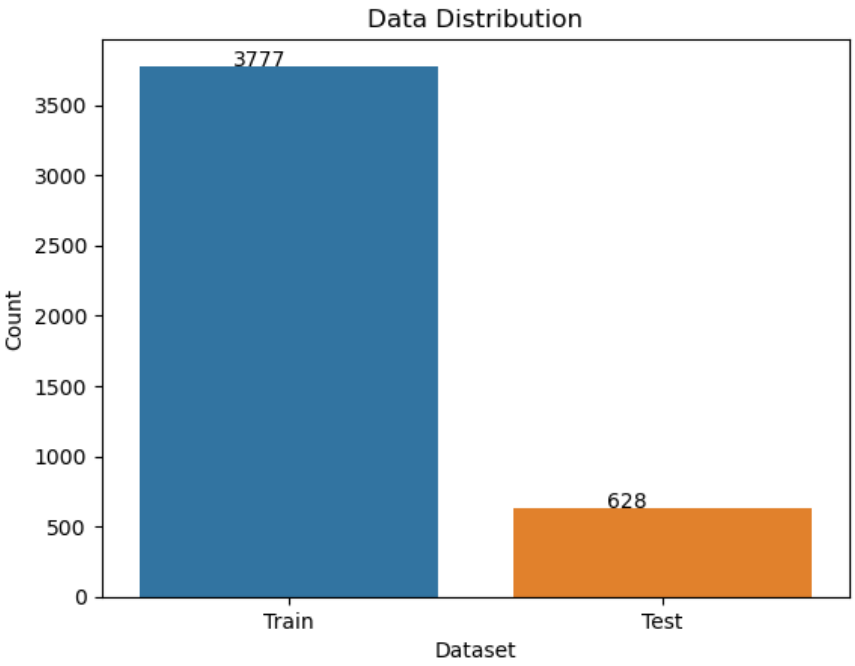


Figure 3.9: Data distribution, that shows the training and testing dataset size.

Successful pothole detection systems must be designed to recognize and accurately classify these diverse pothole types, accounting for the unique visual features and characteristics associated with each variation.

Lastly, sample (c) presents a low-lighting condition, where the presence of shadows negatively impacts the visibility and amplification of pothole features. In such scenarios, the reduced contrast and diminished illumination can obscure crucial details, hindering the accurate identification and localization of potholes. Effective algorithms must be capable of compensating for these different lighting conditions, employing effective techniques to ensure reliable pothole detection even in sub-optimal lighting environments.

The diverse range of lighting conditions and pothole appearances represented in these image samples highlights the complexity of the pothole detection task and the need for robust and adaptable algorithms. By incorporating advanced computer vision and machine learning techniques, it becomes possible to develop intelligent systems capable of addressing these challenges, enabling accurate and reliable pothole detection across a wide range of real-world scenarios.

The dataset we are using in this project suffers in some cases from poor and unreliable labeling as shown in figure 1.3. So, improvement of training data is one of the key factors affecting the efficiency of deep learning models, which includes YOLO, when using it in pothole identification, as the case in our study. When labels are not well-made or inconsistent, they can make the model's capacity to learn and generalize much harder. It should be noted that the image consists of numerous locations for potholes, and YOLO uses labeled bounding boxes to localize and measure these potholes and learn from them. If we have mislabeled or imprecise bounding boxes, then the model will likely learn wrong features. This can result in:

- **Reduced Accuracy**:The presence of unreliable labels, can distract a model's attention from essential characteristics of potholes and focus on irrelevant details. As a consequence, the overall detection accuracy could be less than perfect, as the system might skip real potholes or miss-classify some other road defects as such. In addition, the use of incorrect labels during training can cause the model to overemphasize pothole characteristics, making it difficult to detect a pothole among many others. This is one way through which detection accuracy will generally decrease because the actual potholes will be detected as other imperfections. Moreover, in real-world situations, YOLO models have difficulty in generalizing from poorly labeled data.

- **Limited Generalizability**: YOLO models based on mislabeled training data do not easily learn to cope with new variations they will face in actual scenarios. In these situations, the model might succeed on a specific test set of training data that has errors associated with it but then fail with unseen variation, potholes, or different environment settings. The narrow scope of generalizability makes the model unfit for real-world applications.

- **Degraded Performance on Specific Pothole Types**: When labels for potholes are inadequate and fail to capture the subtle differences between various types (such as small cracks versus large, deep potholes), it can significantly impact the model's ability to accurately detect specific categories. The model may become biased towards the types of potholes that

were prevalent in the training data with more accurate labels, while neglecting other important variations.

- **Ensuring Robust Pothole Detection**: To address these issues and ensure robust pothole detection, it is crucial to use high-quality, well-annotated datasets for training YOLO models. This process may involve manual data cleaning efforts to correct existing label errors or exploring techniques like active learning to focus labeling efforts on the most informative data points.

3.2.1 Dataset Preparation

As previously noted, the training dataset comprises two classes: positive, consisting of 1119 images containing potholes, and negative, consisting of 2658 images without potholes. The dataset includes images with extraneous elements, such as the dashboard, which may be incorrectly classified and detected as potholes. Additionally, a significant portion of the images is occupied by the sky, as depicted in Figures 3.7 and 3.8, resulting in increased computational requirements and training time for the model. To mitigate these issues and optimize computational efficiency, we aimed to remove unnecessary elements that could impair the model's performance. To achieve this, we proposed cropping the images from both the top and bottom.

To determine the optimal cropping locations and minimize the loss of bounding boxes, we analyzed the distribution and percentage of bounding boxes across the dataset. Our analysis revealed that 99% of the bounding boxes fell within the 1200–1800-pixel range. Values above this range typically correspond to the dashboard area, which could be erroneously identified as potholes, while values below this range include sky regions, further increasing computational overhead.

Figure 3.10 provides a visual comparison between the original image and two different cropping strategies, highlighting the impact of image composition on computational efficiency and model performance in the context of pothole detection. This illustration underscores the importance of preprocessing techniques and their role in optimizing the overall detection pipeline.

In the original image (a), it is evident that both the vehicle's dashboard and the sky occupy

Figure 3.10: An image example, before and after cropping. (a) Original image, (b) This represents the image after cropping the dashboard shows how the sky still occupying a big area of the image, (c) The resulted cropped image from both sides.

a significant portion of the frame, potentially introducing irrelevant information and increasing the computational load during training and inference phases. Utilizing such uncropped images for pothole detection tasks would result in unnecessary computational overhead and prolonged training times, ultimately affecting the overall performance and efficiency of the model. To address this issue, a cropping strategy was employed, as depicted in (b), where the image was cropped from the bottom to eliminate the dashboard region. This approach mitigates the risk of falsely detecting dashboard elements as potholes, a common source of erroneous predictions. However, while this cropping strategy successfully removes the dashboard, it still leaves a substantial portion of the image dedicated to the sky, which contributes minimal information relevant to pothole detection and continues to impose an unnecessary computational burden on the model.

Consequently, a more focused cropping strategy was adopted, as shown in (c), where the im-

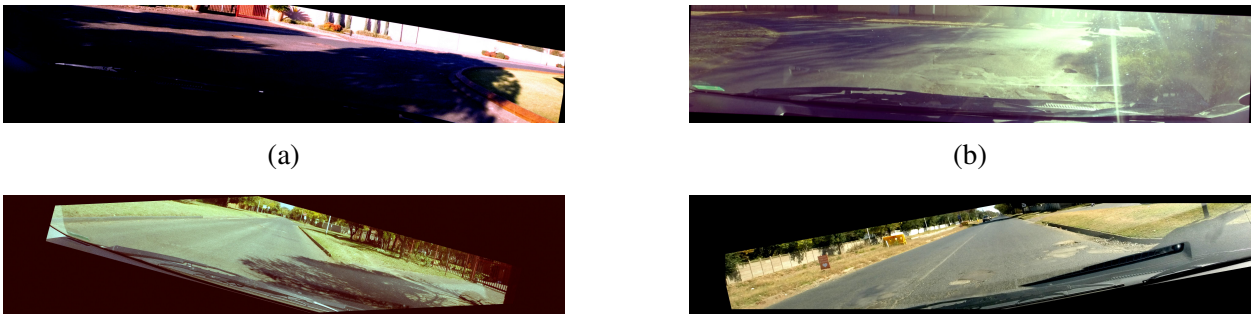
age is cropped to concentrate exclusively on the road surface. This strategic cropping approach ensures that the model's attention is directed solely towards the region of interest, where potholes are most likely to be present. By eliminating irrelevant regions such as the sky and dashboard, the computational complexity is significantly reduced, enabling faster inference times and more efficient utilization of computational resources. The cropping strategy illustrated in (c) not only improves computational efficiency but also enhances the model's ability to accurately detect and localize potholes. By focusing exclusively on the road surface, the model can better leverage its feature extraction and pattern recognition capabilities, identifying subtle cues and distinguishing characteristics that are critical for reliable pothole detection.

This comparative analysis highlights the importance of preprocessing techniques and their impact on model performance and computational efficiency. By carefully selecting and applying appropriate cropping strategies, researchers and practitioners can optimize their pothole detection pipelines, achieving faster inference times, reduced computational complexity, and improved overall accuracy, ultimately contributing to the development of more robust and efficient road maintenance solutions.

To ensure the robustness and generalization capabilities of the proposed pothole detection model, the dataset was strategically partitioned into distinct subsets for training, validation, and testing purposes. Specifically, 20% of the data was allocated for the testing subset, while the remaining 80% was divided between the validation and training sets, adhering to best practices in machine learning model development. Moreover, recognizing the importance of data diversity and variability in training effective deep learning models, a comprehensive set of augmentation techniques was employed to expand the training dataset and introduce a wide range of conditions and variations as shown in 3.11 and 3.12. This approach aimed to enhance the model's resilience to noise, mitigate the risk of over-fitting, and improve its ability to generalize to real-world scenarios.

The augmentation techniques employed encompassed a diverse array of transformations, including:

1. Random adjustments to image brightness and contrast levels, simulating varying lighting



(a) (b)

(c) (d)

Figure 3.11: Augmented Cropped Examples

(a) (b) (c)

Figure 3.12: Augmented Transformed Examples

conditions encountered in practical applications.

2. Simulation of motion blur, mimicking the effects of camera or object movement during image capture.

3. Application of median and average blurriness filters, accounting for potential blur introduced by environmental factors or camera limitations.

4. Random flipping and cropping operations, introducing variations in orientation and composition to improve invariance.

These augmentation techniques were applied systematically to the training dataset, effectively increasing the number of samples and introducing valuable diversity in terms of image characteris-

tics, lighting conditions, and potential noise factors. Figures 3.11 and 3.12 provide visual examples of the augmentation techniques applied to both the cropped images and transformed images, respectively, illustrating the breadth and effectiveness of the employed approach. Furthermore, to ensure balanced representation and mitigate class imbalance, the number of images in the positive class (containing potholes) was increased through augmentation to match the number of images in the negative class (without potholes), resulting in a total of 2658 images for training and evaluation purposes.

By rigorously partitioning the dataset, applying comprehensive augmentation techniques, and ensuring class balance, the proposed methodology adheres to best practices in machine learning model development, laying a solid foundation for robust and accurate pothole detection performance. This systematic approach not only enhances the model's ability to generalize to diverse real-world scenarios but also mitigates potential biases and over-fitting issues, ultimately contributing to the development of a reliable and effective pothole detection solution.

CHAPTER 4

Experimental Design

4.1 Implementation

The effective implementation of the perspective transformation technique hinges on the ability to produce transformed images that are sufficiently clear and devoid of deformations that could potentially impede the feature extraction process performed by the model. Consequently, the selection of the appropriate transformation angle and dimensions is of substantial importance, as it directly impacts the quality and usability of the transformed images for the subsequent training and inference phases. Failure to optimize these parameters could result in the unintended loss of critical objects within the image, adversely affecting the model's ability to accurately detect and localize potholes.

To address these challenges and ensure the generation of high-quality transformed images, a systematic approach was adopted. This section demonstrates the steps taken to achieve optimal transformations, reduce object loss, and adequately cover the required distance to ensure comprehensive coverage of all potholes present in the original images. Furthermore, it provides insights into the training process and the evaluation methodologies employed to assess the model's performance. The careful selection of transformation parameters involved a meticulous exploration of various angles and dimensions, aiming to strike a balance between preserving image clarity, minimizing distortions, and maximizing the visible area of the road surface. Extensive empirical evaluations were conducted to identify the optimal transformation configurations that minimized

the loss of critical objects, such as potholes, while simultaneously ensuring that the transformed images retained sufficient visual fidelity for effective feature extraction and model training. Moreover, to ensure comprehensive coverage and detection of potholes across varying distances, the transformation process was designed to adequately capture and represent the entire road surface within the field of view. This approach aimed to mitigate the risk of overlooking potholes located at different distances from the camera, thereby enhancing the model's ability to detect and localize these hazards accurately and consistently.

Following the generation of transformed images adhering to the optimized parameters, a rigorous training process was undertaken. This involved the careful selection of appropriate training hyper-parameters, such as learning rates, batch sizes, and regularization techniques, to facilitate effective model convergence and generalization. Additionally, robust evaluation methodologies were employed to assess the model's performance on both the transformed and original image datasets, enabling a comprehensive analysis of its pothole detection capabilities under various conditions.

4.1.1 Perspective Transformation Approaches

Achieving a bird's-eye view transformation of images involves selecting the appropriate 4 source points to create the Region of Interest (RoI) in the source image (S.I.) and determining the rectangular dimensions in the destination image (D.I.) as shown in the figure 4.1. It is crucial to include potholes that are located at a considerable distance in each image and to obtain a clear image with minimal deformation. Initial attempts to select the right RoI resulted in 90-degree bird's-eye view images that only covered a short distance in front of the vehicle, exhibiting significant deformation sometimes, and omitting the potholes that were farther away, as illustrated in the figure 4.2 Where this example highlights the consequences of sub-optimal perspective transformations and underscore the importance of carefully selecting the appropriate transformation parameters. These visual representations illustrate the detrimental effects of improper transformations on the quality of the transformed images and the subsequent impact on the model's ability to accurately detect and localize potholes.

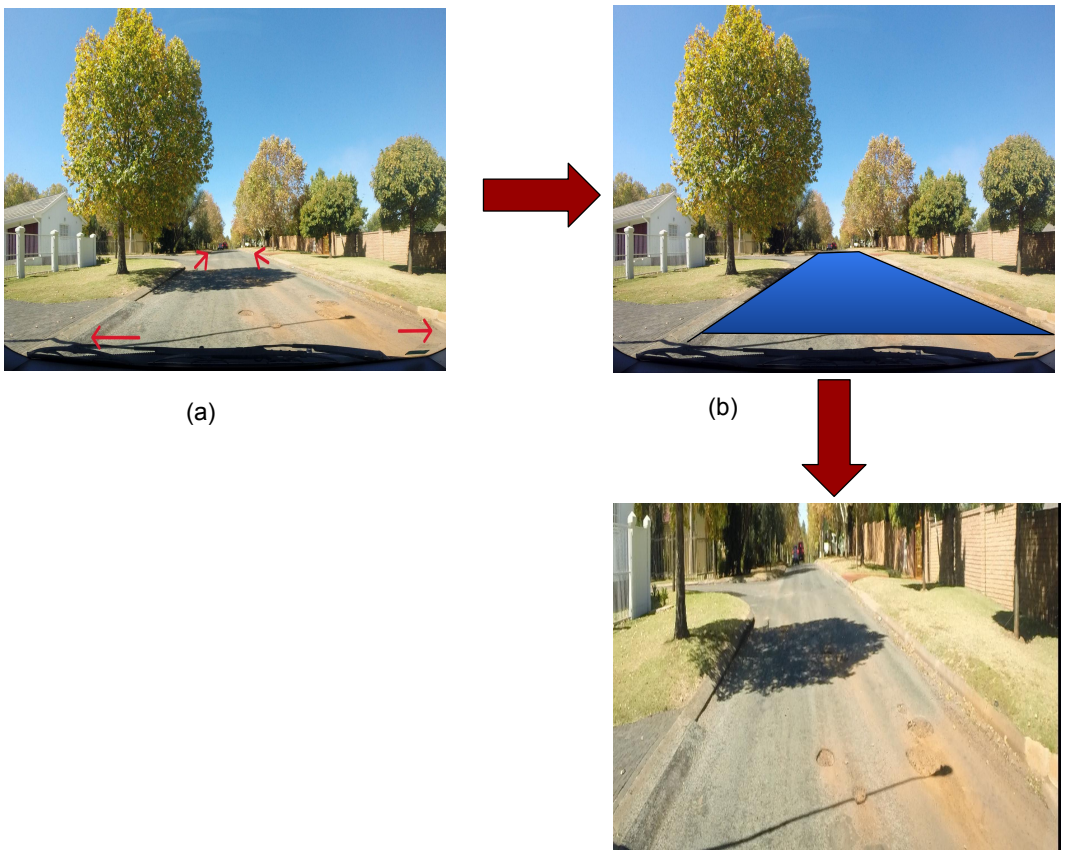


Figure 4.1: Steps for image perspective transformation. (a) Selecting 4 source points (b) Creating ROI to be transformed, (c) Result of transformation.

In the top right corner, the transformed image exhibits a high degree of deformation, which severely distorts the visual features and characteristics of the potholes present in the scene. Such distortions can potentially hinder the model's ability to effectively extract and comprehend the relevant features, ultimately reducing its pothole detection accuracy and overall performance. Excessive deformation can obscure crucial details and introduce artifacts that may be misinterpreted by the model, leading to erroneous predictions and false positives. On the other hand, the transformed image in the bottom right corner demonstrates a different issue. While the level of deformation appears to be less severe compared to the previous example, the transformation fails to capture a sufficiently long distance within the image frame. This limitation results in the exclusion of potholes located further away from the camera, leading to incomplete detection and potential

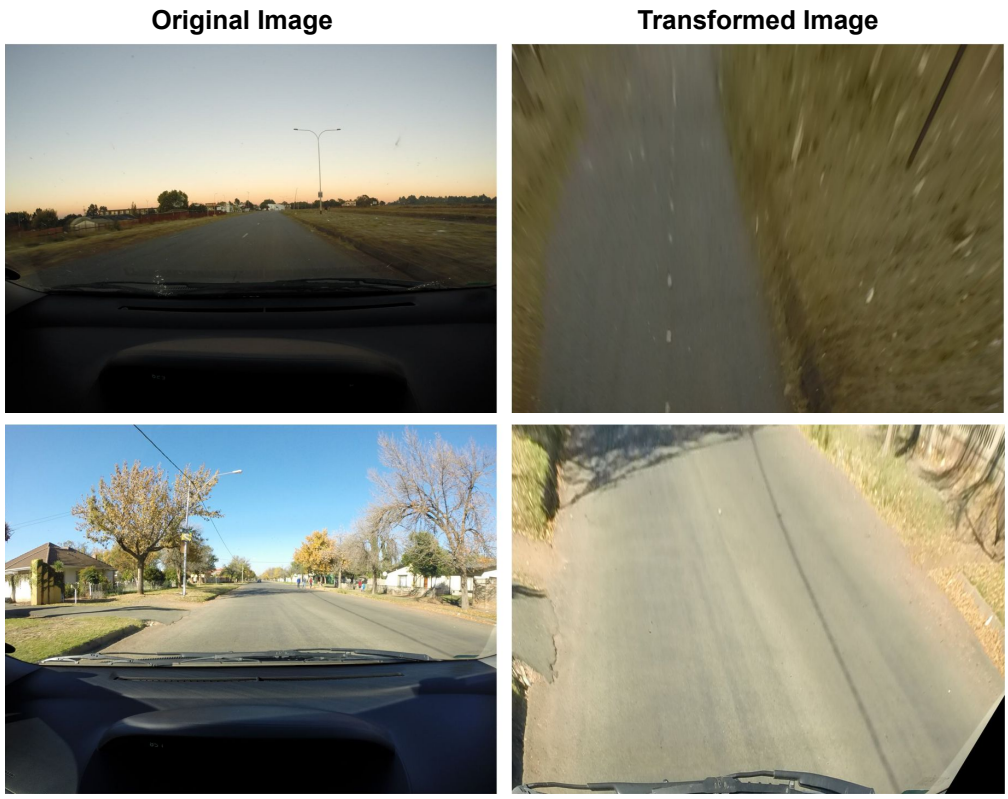


Figure 4.2: Failed transformation attempt. (a) Original image, (b) 90-degrees transformed image with low resolution and covering a short distance.

oversights in various scenarios and image samples. The inability to effectively detect and localize distant potholes can have significant consequences in real-world applications, as it may lead to a failure to identify potential hazards and subsequently delay necessary maintenance and repair efforts. Furthermore, this limitation could compromise the model's overall robustness and generalization capabilities, as it may struggle to adapt to scenarios where potholes are present at varying distances from the camera viewpoint.

These examples highlight the importance of striking a delicate balance when applying perspective transformations. Excessive deformation can distort visual features and hinder feature extraction, while insufficient coverage can result in the omission of critical objects, such as distant potholes. Consequently, a meticulous approach to selecting the appropriate transformation parameters is essential to ensure the generation of high-quality transformed images that preserve

the integrity of the visual features and provide comprehensive coverage of the scene. Hence, we iteratively refined the RoI manually to identify optimal source points applicable to most images for the best transformation outcomes. Ultimately, we achieved satisfactory results with a transformation angle of less than 90 degrees, ensuring that distant potholes were consistently included in all images. Nevertheless, this approach occasionally resulted in missing some closer potholes, losing almost 8% of near potholes, which we accepted since that our focus is on the far ones. This dataset formed the foundation for training and evaluating the model's performance before moving towards automating the transformation process, a direction we intend to explore in future research. Additionally, with the labelled dataset, we developed a function that can transform any image in the dataset without omitting any potholes, thereby ensuring high-quality transformation outputs compared to the regular transformation method, as depicted in Figure 4.3 using ground truth bounding boxes. Where the automated process maintained all potholes with better image quality, avoiding the loss of any, while the regular transformation occasionally missed one or more potholes in some cases.

Talking about how this automatic perspective transformation works briefly, where we are explaining that in more details in the future work section: We created a function that facilitates the automated perspective transformation of images and their corresponding bounding boxes. Initially, it sets up output directories for the transformed images and bounding boxes. Subsequently, all bounding box coordinates are extracted from text files. Source points are then computed based on the minimum and maximum x and y values of all bounding boxes, while destination points are predefined for the new transformation. The function sequentially processes each image, converting it to RGB format, computing the perspective transformation matrix (M), and applying the transformation using cv2.warpPerspective based on this equation 3.1. Concurrently, the bounding box coordinates are transformed using the same matrix to align them with the transformed image. Finally, the transformed images and bounding boxes are saved to the specified output directories. This automated approach ensures that the transformed images maintain accurate spatial relationships with the original images, thereby improving the efficiency of image processing tasks.

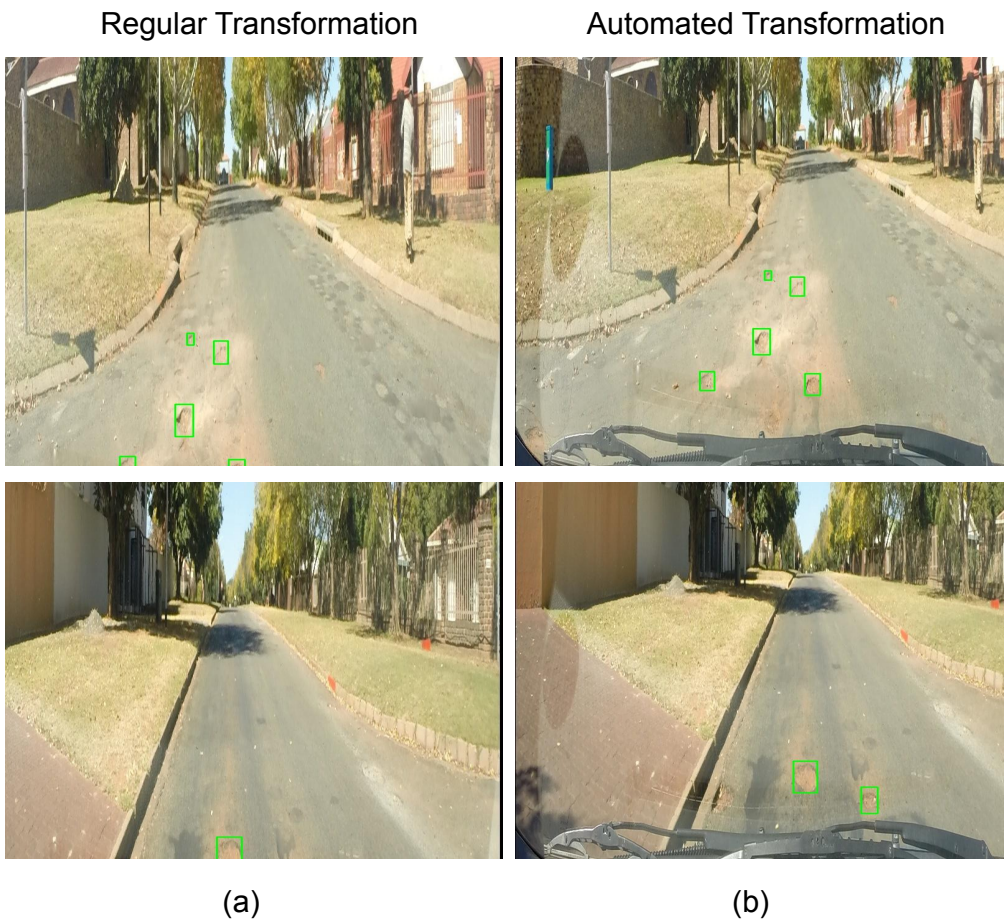


Figure 4.3: Regular Vs Automated Transformation. (a) Images transformed manually losing some potholes ,(b) Images transformed Automatically while maintaining all potholes.

As illustrated in Figure 4.3, a comparative analysis between the regular transformation approach (a) and the proposed automated transformation technique (b) reveals the superiority of the latter in pothole detection and image quality preservation. The visual examples clearly demonstrate that the regular transformation method fails to accurately capture and identify all potholes present in the original ground truth images, particularly those in close proximity to the camera viewpoint. This limitation is evident in both image samples presented in section (a), where several near-field potholes are notably absent from the transformed representations. Furthermore, a closer inspection of the regularly transformed images reveals a higher degree of distortion and deformation when compared to their automated counterparts. The presence of noise and artifacts is more pronounced in the images resulting from the regular transformation approach, potentially hindering

subsequent processing and analysis steps. Conversely, the automated transformation technique exhibits a greater ability to faithfully reproduce and detect all potholes present in the ground truth images, including those located in the near-field region. The transformed images display a higher level of stability and consistency, with minimal distortion or deformation observed. Notably, the shapes and sizes of objects within the auto-transformed images closely resemble their real-world counterparts, ensuring accurate representation and facilitating downstream tasks such as object detection and segmentation. Moreover, the automated transformation approach demonstrates a more judicious allocation of image real estate, dedicating a larger portion of the frame to the road surface where potholes are likely to occur. In contrast, the regular transformation method tends to include excessive sky regions, which may be deemed superfluous for pothole detection applications.

However, it is important to acknowledge a potential drawback of the automated transformation technique. The inclusion of vehicle components, such as windshield wipers, in the transformed images may introduce additional noise and increase the risk of false positive detections. Nonetheless, this issue can be readily addressed through simple post-processing steps, such as cropping the image from the bottom to exclude the problematic regions while preserving the relevant pothole bounding boxes. Overall, the proposed automated transformation approach exhibits superior performance in terms of pothole detection accuracy, image quality preservation, and efficient utilization of the available frame space. These advantages, combined with the ability to mitigate potential drawbacks through straightforward post-processing steps, make the automated transformation technique a compelling choice for pothole detection and road surface analysis applications.

4.1.2 Model Training

The training process for the pothole detection model involved a comprehensive evaluation of three variants of the YOLOv5 architecture, namely YOLOv5-small, YOLOv5-medium, and YOLOv5-large. These models were trained on two distinct datasets: the original cropped images and the perspective-transformed images, each accompanied by their respective bounding box annotations. To ensure thorough training and convergence, each model underwent a rigorous training of 100

epochs. The optimization process was driven by the Stochastic Gradient Descent (SGD) algorithm, which demonstrated superior performance compared to the widely used Adam optimizer. The learning rate was set to 0.01, a value empirically determined to facilitate effective model convergence and generalization. Notably, the training process revealed a significant performance advantage when utilizing the perspective-transformed images as opposed to the base cropped images as shown in the table 4.1 below. For instance, when training the YOLOv5-large model on the regular cropped images, a precision value of 0.957 was achieved. However, when the training was performed on the perspective-transformed images, the precision metric improved substantially, reaching 0.986. Similarly, the recall metric also exhibited a notable improvement, increasing from 0.89 to 0.91. In addition to the standard single-class training approach, an alternative strategy was explored by training the YOLOv5-small model on a modified dataset, where the single pothole class was subdivided into three distinct classes: Near, Medium, and Far. This subdivision was based on the top-left y-coordinate values of the bounding boxes, effectively categorizing potholes according to their perceived distance from the camera viewpoint.

The class subdivision process resulted in a balanced distribution of instances across the three classes, as illustrated in Figure 4.4. Specifically, the y-coordinate ranges corresponding to each class were defined as follows:

- For cropped images:
  - **Far** class: if $4 \leq y \leq 235$, resulting in 1100 far bounding boxes
  - **Medium** class: if $236 \leq y \leq 370$, resulting in 937 medium bounding boxes
  - **Near** class: if $371 \leq y \leq 771$, resulting in 1060 near bounding boxes
- For the transformed images:
  - **Far** class: if $0 \leq y \leq 350$, resulting in 1168 far potholes
  - **Medium** class: if $351 \leq y \leq 550$, resulting in 993 medium bounding boxes
  - **Near** class: if $551 \leq y \leq 800$, resulting in 936 near bounding boxes

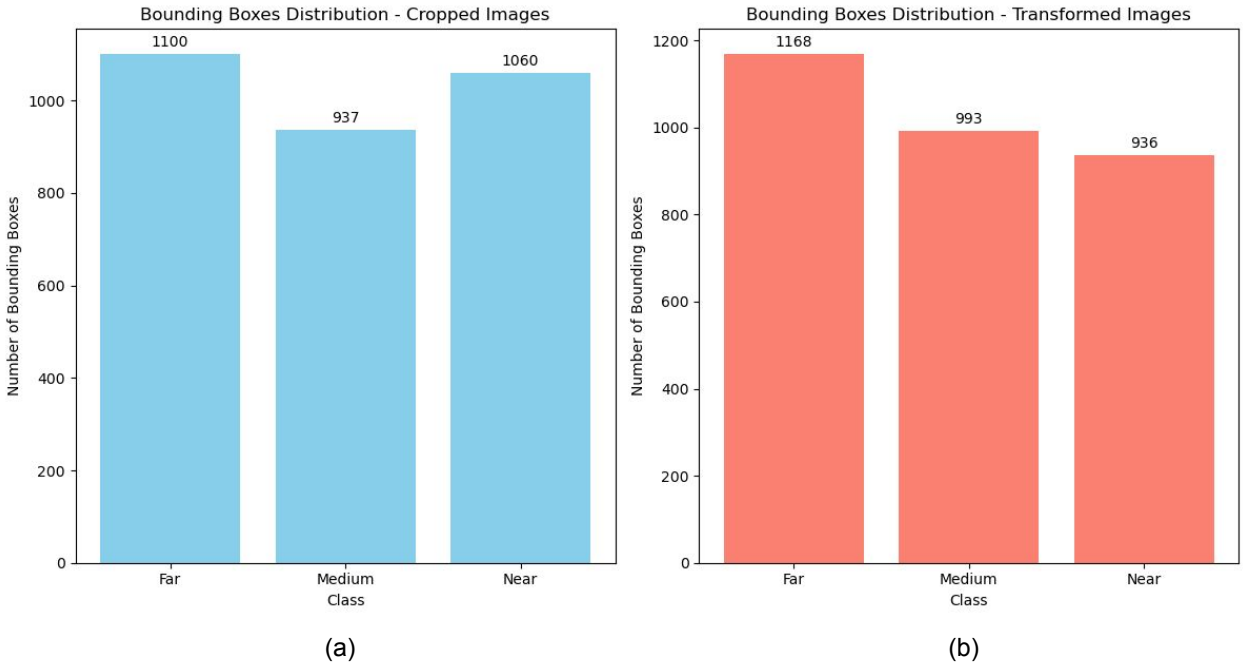


Figure 4.4: (a) Class count in the cropped images dataset, (b) Class count in the transformed images dataset.

This class subdivision approach not only introduced a level of granularity in the detection process but also aimed to improve the model's ability to accurately localize and categorize potholes based on their perceived distance from the camera. By explicitly training the model to recognize and differentiate between near, medium, and far potholes, the potential for enhanced detection accuracy and distance estimation was explored.

It is noteworthy that the perspective transformation process, while beneficial for enhancing the model's performance, resulted in the unintended loss of approximately 8% of the bounding box annotations in the transformed dataset. This challenge, however, was subsequently addressed through the development of an automatic transformation method, which aimed to mitigate the loss of bounding box information during the transformation process.

The training of the YOLOv5-small model on the datasets with subdivided classes (Near, Medium, and Far) yielded significant improvements in the model's ability to detect and distinguish between potholes at varying distances. Particularly noteworthy was the enhancement in detecting the "Far Potholes" class, where the mean Average Precision (mAP) at an Intersection over Union

Table 4.1: Comparison between the baseline and the perspective transformed models.

Model	Class	Bounding Boxes	Precision	Recall	$\text{mAP}_{50\%}$	$\text{mAP}_{50:95\%}$
Baseline	All	629	0.683	0.617	0.642	0.292
	Near	181	0.614	0.754	0.701	0.338
	Med	205	0.673	0.631	0.652	0.310
	Far	243	0.764	0.466	0.571	0.227
Perspective Transform	All	533	0.638	0.626	0.661	0.344
	Near	151	0.650	0.603	0.662	0.368
	Med	175	0.592	0.656	0.656	0.369
	Far	207	0.672	0.618	0.667	0.295

(IoU) threshold of 50% (mAP50%) exhibited a substantial improvement, increasing from 0.571 to 0.667 when training on the perspective-transformed dataset. Moreover, the training process revealed considerable improvements in the mAP50-95% metric across all classes when utilizing the perspective-transformed dataset. For instance, the mAP50-95% values for the small, medium, and large classes increased from 0.338, 0.31, and 0.227 to 0.368, 0.369, and 0.295, respectively. Additionally, the overall mAP50-95% across all classes showed a notable improvement, rising from 0.292 to 0.344, as presented in Table 4.1.

These quantitative results underscore the effectiveness of the perspective transformation approach in enhancing the model's ability to accurately detect and localize potholes, particularly those located at varying distances from the camera viewpoint. By introducing a additional level of complexity and diversity through the transformed images, the model was better equipped to learn and generalize to real-world scenarios, where potholes may appear at different distances and angles. Furthermore, the development of the automatic transformation method addressed the issue of bounding box loss encountered during the initial transformation process. By mitigating this challenge, the automatic transformation approach ensured the preservation of valuable annotation data, thereby enhancing the training process and potentially improving the model's overall performance. Additionally, the systematic evaluation of different training strategies, combined with the incorporation of perspective transformations and class subdivisions, not only yielded quantitative

improvements in pothole detection accuracy but also laid the foundation for a more robust and comprehensive pothole detection system. These advancements contribute to the development of efficient and effective road objects detection, ultimately enhancing road safety and minimizing the potential risks associated with unaddressed potholes.

CHAPTER 5

Evaluation Metrics and Test Results

5.1 Metrics

To evaluate the performance of the trained models, a comprehensive testing phase was conducted on a dataset comprising 628 images. Given that the YOLO object detection algorithm outputs its predictions in the format of (center coordinates, width, height), while the ground truth bounding boxes were defined as (top-left coordinates, width, height), a conversion step was necessary to enable a direct comparison between the predicted and ground truth bounding boxes. This conversion facilitated the plotting of both predictions and ground truth boxes on the same scale, as illustrated in Figure 5.1, and enabled the computation of essential metrics for model evaluation based on their intersection over union (IoU), mean Average Precision (mAP), and Average Recall (AR). In the field of machine learning, particularly in object detection tasks, mean Average Precision (mAP) and Average Recall (AR) are critical metrics for assessing the performance of object detection algorithms. The mAP metric, widely adopted in computer vision applications, is calculated by averaging precision values across various recall levels. Precision, in this context, indicates the proportion of true positive detections among all positive detections, while recall measures the proportion of true positive detections among all ground truth positives. A high mAP score signifies accurate object detection performance across different recall levels, a crucial requirement for applications demanding high precision and recall, such as object detection tasks. On the other hand, Average Recall provides an aggregate measure of recall across all classes in a multi-class classi-

Figure 5.1: Plot of area overlap between the ground truth bounding boxes in green, and the predicted bounding boxes in red.

fication or object detection task. It offers a holistic view of a model's performance across various categories, reflecting its ability to identify relevant instances among all actual positives. Precision and recall are fundamental metrics in object detection tasks, offering deeper insights into model performance beyond mere accuracy measures.

Precision is calculated as the ratio of True Positives (TP) to the sum of TP and False Positives (FP), as illustrated by Equation 5.1. Conversely, recall is computed as the ratio of TP to the sum of TP and False Negatives (FN), as shown in equation 5.2.

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

The Intersection over Union (IoU) metric plays a pivotal role in the evaluation of object detection algorithms, as it quantifies the overlap between the predicted bounding boxes and the ground truth bounding boxes. The IoU is calculated as the ratio of the area of overlap between the two

bounding boxes to the area of their union, as expressed by Equation 5.3.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{5.3}$$

In object detection tasks, the IoU metric is commonly employed as a criterion for determining whether a predicted bounding box should be considered a true positive detection. A predetermined threshold, typically set at 0.5 or higher, is used to classify a prediction as a true positive based on its overlap with the ground truth bounding box. In the context of this study, an IoU threshold of 0.5 was adopted, ensuring that only predictions with a sufficient degree of overlap with the ground truth were considered for evaluation. Moreover, a high IoU value indicates a strong correspondence between the predicted and ground truth bounding boxes, signifying accurate object localization and bounding box estimation.

It is important to note that the choice of the IoU threshold can significantly impact the evaluation results and the trade-off between precision and recall. A lower IoU threshold may result in a higher number of true positive detections but potentially include more false positives, leading to a higher recall but lower precision. Conversely, a higher IoU threshold can increase precision by reducing the number of false positives but may also lead to a lower recall due to the stricter overlap requirement. Additonally, In the context of pothole detection, the IoU metric provides valuable insights into the accuracy and localization capabilities of the trained models. By analyzing the IoU distributions and setting an appropriate threshold, we can strike a balance between precision and recall, tailoring the evaluation criteria to meet the specific requirements of the application scenario.

The incorporation of the IoU metric, in conjunction with other performance measures such as mean Average Precision (mAP) and Average Recall (AR), contributes to a comprehensive evaluation framework for object detection algorithms. This framework enables a thorough assessment of the models' strengths and limitations, facilitating informed decision-making and guiding future improvements in the pothole detection pipeline.

5.2 Test Results

In this section, we present a comprehensive analysis of the evaluation results obtained by comparing the performance of three variants of the YOLOv5 architecture: YOLOv5-small, YOLOv5-medium, and YOLOv5-large. The evaluation process was conducted in two phases: initially, we compared the models' performance using a single class (pothole) between the base cropped image dataset and the transformed image dataset. Subsequently, we extended our analysis to incorporate three distinct classes (near, medium, and far) while evaluating the YOLOv5-small model using the same datasets. The results presented in tables 5.1 and 5.2 demonstrate a significant enhancement in various performance metrics, including mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds (0.5–0.9, 0.5, and 0.75), as well as Average Recall (AR) at the same thresholds, when utilizing the transformed image dataset compared to the base cropped image dataset for the single-class model. Moreover, the results illustrate comparable performance improvements when utilizing the three-class dataset, particularly for the medium and far classes, with certain metrics exhibiting enhancements exceeding 10%. However, it is noteworthy that no significant improvement was observed for the near class, which can be attributed to the majority of bounding boxes lost during the transformation process belonging to this class. This issue is effectively addressed when employing the automatic transformation approach, which is discussed in subsequent sections.

The accompanying figures, 5.2 and 5.3, provide visual examples that highlight the model's improved ability to detect distant (far) potholes when trained on the transformed dataset, resulting in fewer missed detections compared to the base model. This improvement is particularly evident in the prediction results for the three-class model. On top of that, the quantitative results presented in Tables 5.1 and 5.2 underscore the efficacy of the perspective transformation approach in enhancing the model's overall performance. By leveraging transformed images that simulate a bird's-eye view of the road surface, the model's capacity to accurately detect and localize potholes at varying distances is significantly improved. This enhancement is particularly noteworthy for the detection of distant potholes, which can be challenging due to the reduced visual cues and potential occlu-

Table 5.1: Single-Class Configuration Results, Using Three Yolov5's Versions

Model Configuration		**Metric**					
		$AP_{50:95\%}$	$AP_{50\%}$	$AP_{75\%}$	$AR_{50:95\%}$	$AR_{50\%}$	$AR_{75\%}$
Baseline	Small	**0.167**	**0.437**	**0.090**	**0.150**	**0.220**	**0.220**
	Medium	**0.176**	**0.429**	**0.101**	**0.154**	**0.231**	**0.230**
	Large	**0.177**	**0.448**	**0.100**	**0.149**	**0.228**	**0.228**
Transformed	Small	**0.232**	**0.531**	**0.158**	**0.195**	**0.294**	**0.294**
	Medium	**0.238**	**0.533**	**0.170**	**0.195**	**0.303**	**0.303**
	Large	**0.238**	**0.539**	**0.165**	**0.198**	**0.301**	**0.301**

Table 5.2: Three-Class Configuration Results with Small Yolov5

Model	**Pothole Proximity**	**Metric**					
		$AP_{50:95\%}$	$AP_{50\%}$	$AP_{75\%}$	$AR_{50:95\%}$	$AR_{50\%}$	$AR_{75\%}$
Baseline	Near	**0.196**	**0.504**	**0.110**	**0.191**	**0.274**	**0.274**
	Medium	**0.155**	**0.409**	**0.062**	**0.223**	**0.259**	**0.259**
	Far	**0.108**	**0.315**	**0.055**	**0.147**	**0.184**	**0.184**
Transformed	Near	**0.124**	**0.273**	**0.113**	**0.134**	**0.155**	**0.155**
	Medium	**0.236**	**0.481**	**0.195**	**0.298**	**0.364**	**0.364**
	Far	**0.200**	**0.489**	**0.141**	**0.208**	**0.286**	**0.286**

sions. The improved performance achieved by utilizing the transformed dataset can be attributed to several factors.

First, the transformed images provide a more comprehensive and uniform representation of the road surface, mitigating the effects of perspective distortion present in the original cropped images. This transformation effectively normalizes the appearance of potholes across varying distances, enabling the model to learn and generalize more effectively. Second, the transformed images offer a wider field of view, allowing the model to capture a larger portion of the road surface within a single image. This expanded view facilitates the detection of potholes at greater distances, which may be partially or completely occluded in the original cropped images due to the limited viewpoint. Furthermore, the incorporation of the three-class subdivision (near, medium, and far) introduces

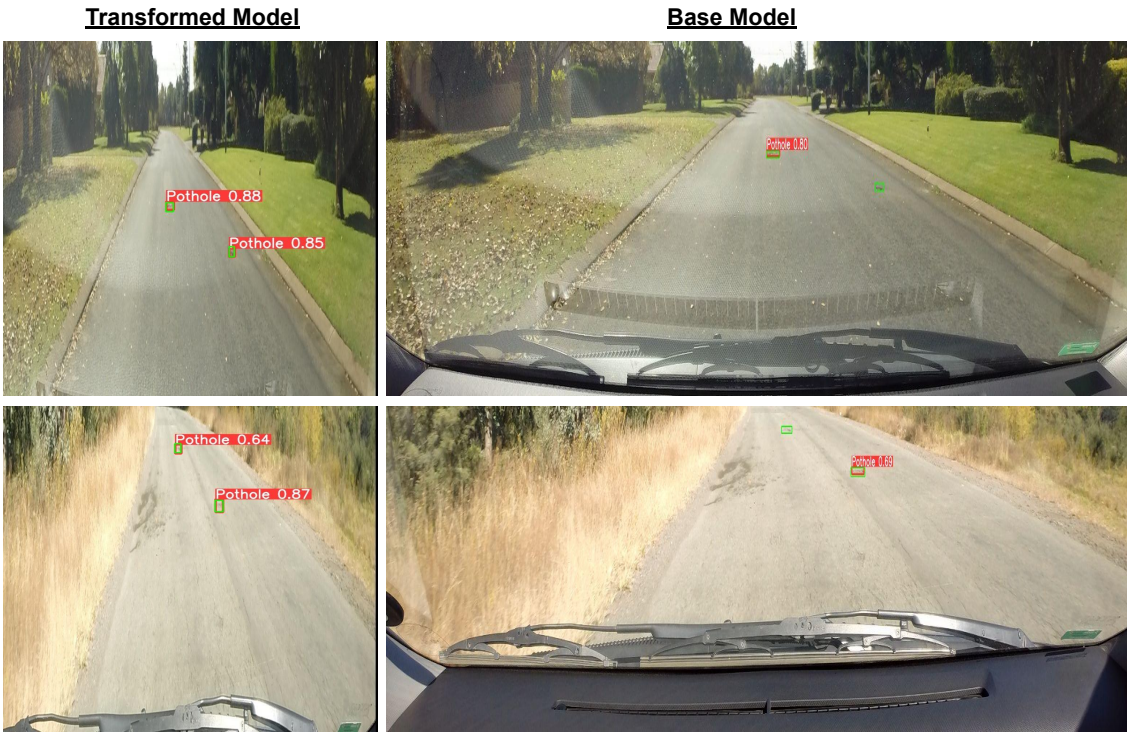


Figure 5.2: Testing results showing how the perspective transformation model surpasses the baseline model performance when testes on one class. Detected labels (Red), ground truth labels (Green). (a), Transformation model detects all potholes in the images, especially the far potholes. (b) Baseline model misses some potholes, especially the far ones.

an additional level of granularity in the detection process, encouraging the model to learn and differentiate between potholes based on their perceived distance from the camera viewpoint. This approach not only enhances the overall detection accuracy but also provides valuable insights into the distance estimation capabilities of the model.

While the results demonstrate the advantages of the perspective transformation approach, it is important to acknowledge the potential limitations and areas for further improvement. For instance, the loss of bounding boxes during the transformation process, particularly for the near class, highlights the need for more robust and adaptive transformation techniques to minimize data loss and ensure comprehensive coverage of all potholes, regardless of their distance from the camera which is clearly addressed and solved when using the automated perspective transformation that we are going to present its results in our future work.

Figure 5.3: Testing results showing the perspective transformation model surpasses the baseline model performance when tested on 3 classes (Near, Medium, and Far). Detected labels (Red 'far', Orange 'Medium'). (a), Transformation model detects all potholes in the images, with a focus on the far potholes. (b) Base model misses far and medium potholes in many cases.

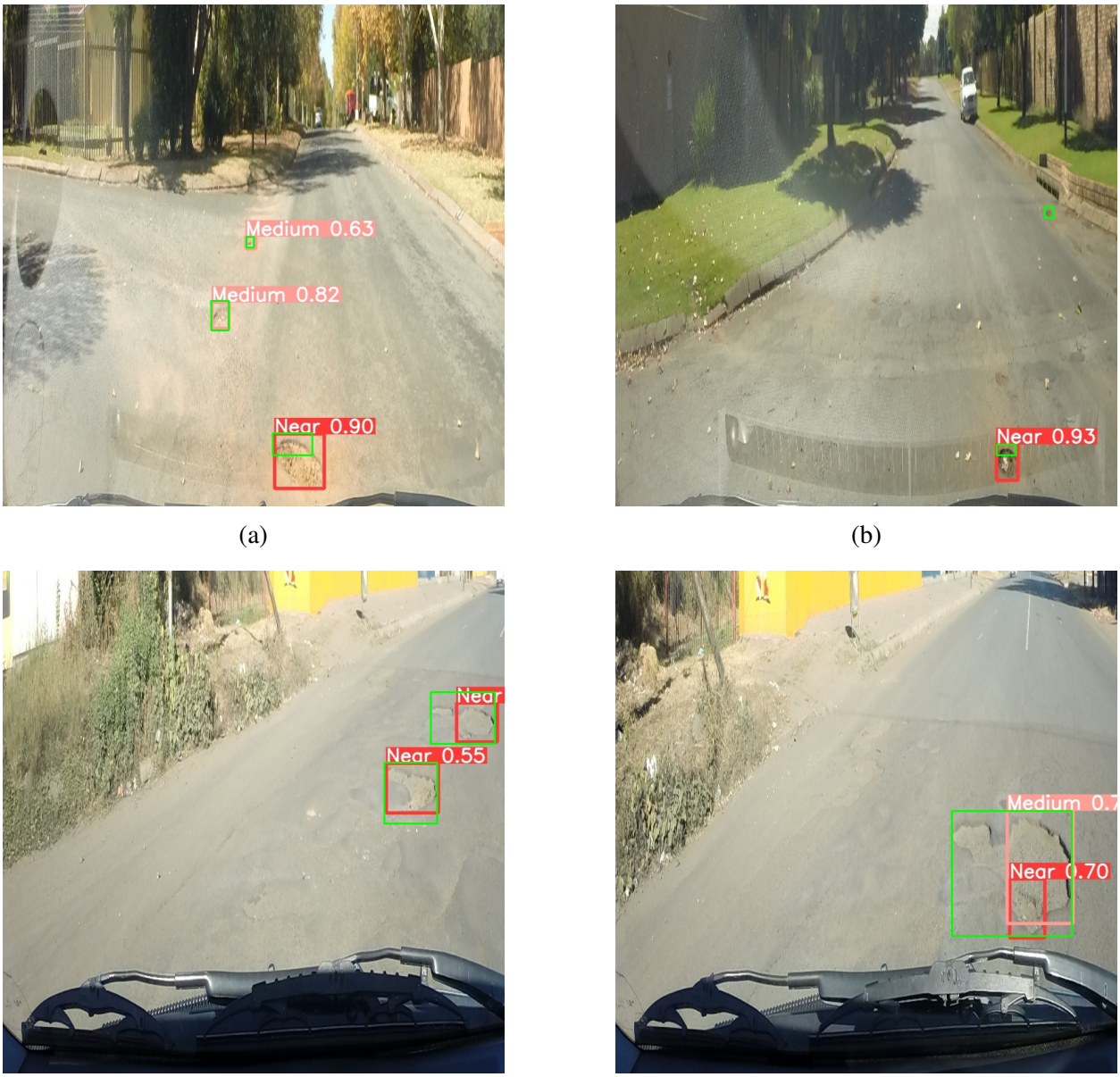
Overall, the evaluation results presented in this section provide compelling evidence for the effectiveness of the proposed methodology, leveraging perspective transformations and class subdivisions to enhance the accuracy and robustness of pothole detection systems.

CHAPTER 6

Discussion, and Conclusion

6.1 Discussion and Future Work

Our results demonstrate the effectiveness of regular perspective transformation in improving detection performance, both in single-class and multi-class scenarios. However, we anticipate that automated perspective transformation could offer even greater benefits across a wider range of datasets. By reducing deformation and minimizing object loss, this approach could significantly enhance the detection of near potholes, a class where our current method falls short. Furthermore, the models were trained on a relatively small training dataset, primarily due to a substantial portion being allocated for validation purposes. Adjusting the data distribution to allocate more data for training would likely improve the model's performance. Finally, it is crucial to acknowledge the impact of inconsistencies in the data labeling process, which had a detrimental effect on the model's performance. Specifically, several instances were identified where the provided labels only covered the edges or corners of potholes, failing to encompass the entirety of their features. This partial labeling approach resulted in a significant loss of valuable information, potentially confusing the model during the training process and hindering its ability to comprehend and learn the true features necessary for accurate pothole detection. Moreover, another labeling issue encountered was the presence of multiple potholes within a single bounding box annotation. This approach not only obscured the individual characteristics of each pothole but also compromised the integrity of edge features, which are crucial for the model to effectively identify and localize these road haz-



(a) (b)

(c) (d)

Figure 6.1: Poor Labels Examples in Green Compared to Detections in Red

ards. Accurate edge detection and delineation are essential for precise bounding box estimation and reliable pothole identification, making the preservation of these features paramount.

These labeling inconsistencies and inaccuracies introduced noise into the training data, potentially leading to sub-optimal model performance and reduced generalization capabilities. Figure 6.1 provides visual examples of these labeling issues, illustrating cases where labels cover only

partial pothole features or encompass multiple potholes within a single bounding box.

While our study focused on a specific dataset and detection algorithm (YOLOv5), further investigation involving different algorithms such as SSD [22] and datasets with varying lighting and weather conditions is warranted. This broader exploration would provide more comprehensive evidence of the benefits of perspective transformation in object detection tasks. Looking ahead, we are planning to expand our research to include the integration of image captioning with pothole detection. This integration could revolutionize driver assistance systems by providing drivers with clear and actionable information about road hazards. For example, the system could alert drivers with messages like ”pothole detected ahead, please slow down” or ”pothole on the right side near the sign, proceed with caution.” This integration of detection and captioning offers several advantages. It provides drivers with a more detailed and understandable warning, allowing them to respond appropriately based on the severity and location of the hazard. Overall, this research opens exciting possibilities for improving driver safety and convenience through advanced image analysis techniques.

6.2 Conclusion

This study fills a critical gap in existing research by focusing on the detection of far potholes, a key element in enhancing vehicle safety and minimizing damage caused by these road hazards. Our findings underscore the importance of prioritizing the detection of distant potholes to improve overall driving safety. By employing a perspective transformation technique in conjunction with the YOLOv5 detection algorithm, we achieved significant improvements in pothole detection precision, particularly for medium and far potholes, with some metrics showing improvements exceeding 10%. Our results highlight the effectiveness of using transformed images over regular ones, particularly in accurately detecting distant potholes. These findings lay the groundwork for further advancements in pothole detection technology and its application in systems such as path planning and navigation. Future research will focus on addressing the remaining challenges and

implementing these advancements in real-world driving scenarios.

BIBLIOGRAPHY

[1] AAA Newsroom. Aaa: Potholes pack a punch as drivers pay $26.5 billion in related vehicle repairs, Accessed 2022.

[2] ARCCA. Human reaction time in emergency situations, 2022.

[3] Shreyas Balakuntala and Sandeep Venkatesh. An intelligent system to detect, avoid and maintain potholes: A graph theoretic approach. *arXiv preprint arXiv:1305.5522*, 2013.

[4] Sachin Bharadwaj Sundra Murthy and Golla Varaprasad. Detection of potholes in autonomous vehicle. *IET Intelligent Transport Systems*, 8(6):543–549, 2014.

[5] Bridge Michigan. It's pothole season, and michigan's are among nation's worst, study finds, 2022.

[6] Boris Bučko, Eva Lieskovská, Katarína Zábovská, and Michal Zábovskỳ. Computer vision based pothole detection under challenging conditions. *Sensors*, 22(22):8878, 2022.

[7] Hanshen Chen, Minghai Yao, and Qinlong Gu. Pothole detection using location-aware convolutional neural networks. *International Journal of Machine Learning and Cybernetics*, 11(4):899–911, 2020.

[8] ClickOnDetroit. Michigan roads: How bad is the pothole problem?, 2019.

[9] Lowe David. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[10] Amita Dhiman, Hsiang-Jen Chien, and Reinhard Klette. Road surface distress detection in disparity space. In *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2017.

[11] Amita Dhiman and Reinhard Klette. Pothole detection using computer vision and learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3536–3550, 2019.

[12] Eduzaurus. Pothole detection methods, 2023.

[13] Geoawesomeness. Application of mobile lidar on pothole detection, 2013.

[14] GetCircuit. Worst roads in america, Accessed 2022.

[15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[16] The Independent. Potholes could be fixed by roads that 'eat' repair materials. *The Independent*, 2019.

[17] Road Innovation. The impact of road potholes: Addressing the challenge for safer and smoother journeys, Accessed 2022.

[18] Young-Mok Kim, Young-Gil Kim, Seung-Yong Son, Soo-Yeon Lim, Bong-Yeol Choi, and Doo-Hyun Choi. Review of recent automated pothole-detection methods. *Applied Sciences*, 12(11):5320, 2022.

[19] Abhishek Kumar, Dhruba Jyoti Kalita, Vibhav Prakash Singh, et al. A modern pothole detection technique using deep learning. In *2nd International Conference on Data, Engineering and Applications (IDEA)*, pages 1–5. IEEE, 2020.

[20] Attorney Steve Lee. A pothole can damage your vehicle, your passengers, and you, Accessed 2022.

[21] LiDAR Radar. Advantages and disadvantages of lidar, 2023.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[23] Hiroya Maeda, Takehiro Kashiyama, Yoshihide Sekimoto, Toshikazu Seto, and Hiroshi Omata. Generative adversarial network for road damage detection. *Computer-Aided Civil and Infrastructure Engineering*, 36(1):47–60, 2021.

[24] S Nienaber, Marthinus J Booysen, and RS Kroon. Detecting potholes using simple image processing techniques and real-world footage. 2015.

[25] Outsight. How does lidar compare to cameras and radars?, 2023.

[26] Vosco Pereira, Satoshi Tamura, Satoru Hayamizu, and Hidekazu Fukai. A deep learning-based approach for road pothole detection in timor leste. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pages 279–284. IEEE, 2018.

[27] Bayesian Quest. Build your computer vision application - part v: Road pothole detector using yolov5. https://bayesianquest.com/2022/07/25/build-you-computer-vision-application-part-v-road-pothole-detector-using-yolo-v5/, July 2022.

[28] QuoteWizard. Pothole damage costs us drivers $3 billion dollars per year, 2023.

[29] Roopak Rastogi, Uttam Kumar, Archit Kashyap, Shubham Jindal, and Saurabh Pahwa. A comparative evaluation of the deep learning algorithms for pothole detection. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE, 2020.

[30] Remodel or Move. Should you hit the brakes when going over a pothole?, 2023.

[31] Habeeb Salaudeen and Erbuğ Çelebi. Pothole detection using image enhancement gan and object detection network. *Electronics*, 11(12):1882, 2022.

[32] Piotr Samczynski and Elisa Giusti. *Recent Advancements in Radar Imaging and Sensing Technology*. MDPI, 2021.

[33] Anas Al Shaghouri, Rami Alkhatib, and Samir Berjaoui. Real-time pothole detection using deep learning. *arXiv preprint arXiv:2107.06356*, 2021.

[34] Zaid El Shair and Samir Rawashdeh. High-temporal-resolution event-based vehicle detection and tracking. *Optical Engineering*, 62(3):031209, 2022.

[35] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[36] The AI Learner. Perspective transformation, 2020.

[37] Ultralytics. Yolov5 documentation, 2020.

[38] University of Minnesota Twin Cities. Talking potholes. University of Minnesota Twin Cities, 2023.

[39] Visual Expert. Reaction time, 2024.

[40] Junshu Zhang, Jindong Zhang, Botao Chen, Jian Gao, Shanwei Ji, Xiaolong Zhang, and Zhaodan Wang. A perspective transformation method based on computer vision. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 765–768, 2020.