World Electric Vehicle Journal

Article

Performance Evaluation of You Only Look Once v4 in Road Anomaly Detection and Visual Simultaneous Localisation and Mapping for Autonomous Vehicles

Jibril Abdullahi Bala [1,2,*], **Steve Adetunji Adeshina** [3] and **Abiodun Musa Aibinu** [2,4]

[1] Department of Electrical and Electronics Engineering, Nile University of Nigeria, Abuja 900001, Nigeria
[2] Department of Mechatronics Engineering, Federal University of Technology, Minna 920211, Nigeria; abiodun.aibinu@futminna.edu.ng or maibinu@gmail.com
[3] Department of Computer Engineering, Nile University of Nigeria, Abuja 900001, Nigeria; steve.adeshina@nileuniversity.edu.ng
[4] Summit University, Offa 250101, Nigeria
[*] Correspondence: 201233001@nileuniversity.edu.ng or jibril.bala@futminna.edu.ng

Abstract: The proliferation of autonomous vehicles (AVs) emphasises the pressing need to navigate challenging road networks riddled with anomalies like unapproved speed bumps, potholes, and other hazardous conditions, particularly in low- and middle-income countries. These anomalies not only contribute to driving stress, vehicle damage, and financial implications for users but also elevate the risk of accidents. A significant hurdle for AV deployment is the vehicle's environmental awareness and the capacity to localise effectively without excessive dependence on pre-defined maps in dynamically evolving contexts. Addressing this overarching challenge, this paper introduces a specialised deep learning model, leveraging YOLO v4, which profiles road surfaces by pinpointing defects, demonstrating a mean average precision (mAP@0.5) of 95.34%. Concurrently, a comprehensive solution—RA-SLAM, which is an enhanced Visual Simultaneous Localisation and Mapping (V-SLAM) mechanism for road scene modeling, integrated with the YOLO v4 algorithm—was developed. This approach precisely detects road anomalies, further refining V-SLAM through a keypoint aggregation algorithm. Collectively, these advancements underscore the potential for a holistic integration into AV's intelligent navigation systems, ensuring safer and more efficient traversal across intricate road terrains.

Keywords: autonomous vehicles; deep learning; road anomaly detection; visual SLAM; YOLO v4

check for updates

Citation: Bala, J.A.; Adeshina, S.A.; Aibinu, A.M. Performance Evaluation of You Only Look Once v4 in Road Anomaly Detection and Visual Simultaneous Localisation and Mapping for Autonomous Vehicles. *World Electr. Veh. J.* **2023**, *14*, 265. https://doi.org/10.3390/wevj14090265

Academic Editor: Ghanim A. Putrus

Received: 9 August 2023
Revised: 25 August 2023
Accepted: 12 September 2023
Published: 18 September 2023

1. Introduction

Electric vehicles (EVs) and autonomous vehicles (AVs) have recently become a major direction for development due to climate change and the resulting emission restrictions in nations around the world [1–3]. These vehicles are expected to make up thirty-one percent of the global vehicle fleet by 2050 [4]. Research in intelligent road transportation systems mainly focuses on these vehicles due to their versatility and increased throughput. However, with the developments in the EV and AV industry, there have been rising concerns about driver safety by manufacturers, operators, and researchers [5,6].

Road accidents are accountable for about 1.2 million fatalities, annually, according to the World Health Organization (WHO) [7]. A major factor influencing the high rate of road crashes is poor road infrastructure [6,8]. In many low- and middle-income countries, road networks are frequently compromised by anomalies such as potholes, cracks, and unmarked speed bumps. While a considerable amount of resources are directed towards the upkeep of these roads, the manual monitoring techniques currently in place are both cost-intensive and fatigue-inducing. Furthermore, without the ability to accurately map and navigate these dynamic environments in real time, the potential benefits of automated systems remain untapped. This highlights a pressing need not only for advanced anomaly detection but also for efficient Visual Simultaneous Localisation and Mapping (V-SLAM) systems. Implementing V-SLAM would enable real-time, dynamic responses to road conditions, bridging the gap between current infrastructure challenges and the promise of autonomous solutions.

In the field of road surface characterisation and anomaly detection, there have been numerous attempts to develop effective techniques for identifying and locating anomalies in road surfaces. However, many of these approaches have limitations in terms of their applicability for practical implementation. In particular, a significant proportion of these methods treat the problem as a classification task, which while capable of indicating the presence of an anomaly, is unable to provide a precise location of where the anomaly has been detected [9–11]. Furthermore, while there are a few techniques that leverage object detection models, they tend to be limited in their ability to detect only a single type of road anomaly, which falls short of the requirements for a comprehensive road profiling system [12,13].

Furthermore, existing techniques in Visual-SLAM have salient limitations in their implementation. First, is the issue of reliability of these methods in outdoor environments. Techniques such as radar- and laser-based systems provide no semantic information about the environment [14–17]. Additionally, even though significant research has been carried out in the area of object detection [18] and semantic segmentation [19], V-SLAM implementations in highly dynamic sceneries such as road networks and highways have not been exhaustively explored.

In this research, we present an integrated model tailored for autonomous vehicles (AVs) that encompasses both road surface characterisation and Visual Simultaneous Localisation and Mapping (V-SLAM). The cornerstone of this paper lies in its deep learning methodology, adept at identifying and pinpointing infrastructural anomalies within roads. Conceptualised under the framework of object detection, this model possesses the capacity to recognise up to three distinct road anomalies while simultaneously offering visual localisation and mapping capabilities. The rest of this paper is organised according to the following sections: Section 2 contains a review of recent related work, whereas the study methodology is presented in Section 3. Section 4 presents the results, analysis, and discussion, while Section 5 gives the conclusion and future study directions.

2. Literature Review

2.1. Road Anomaly Detection

2.1.1. Traditional-Based Methods

Road surface characterisation involves identifying unique features of the road infrastructure. This process also involves identifying instances that deviate from the standard road setting. In real-world applications, these road anomalies include potholes, cracks, swellings, stripping, and unmarked speed bumps [9]. The detection and avoidance of these anomalies is crucial since late detection can lead to vehicle damage or road accidents. Therefore, it is pertinent that an intelligent transportation scheme is able to identify not only the presence of anomalies on the road but also their location on the road [20]. These actions will aid transport authorities to obtain real-time information about the road infrastructure, enable a vehicle to manoeuvre these anomalies via a suitable control or navigation method, and facilitate the incorporation of the model into a relevant visual odometry technique.

In the literature, several methods have been implemented in the detection of road anomalies. One of these techniques is the use of sensor data for anomaly identification. In this case, the vehicle utilises sensors such as accelerometers, gyroscopes, lidar, ultrasonic, and radar sensors to perceive their environments and to detect road anomalies. In some cases, these sensors are combined or fused to utilise different sensor readings for more accurate results.

In [21], an application based on crowd sensing was designed for detecting the condition of roads. The technology estimates the position of potholes and speed bumps using acceleration data from road users' cell phones. The program, dubbed CRATER, has a 95% and 90% success rate of respectively detecting speed bumps and potholes. However, the program had a 5% false detection rate for speed bumps and 10% for potholes.

For the detection of speed bumps and the lowering of vehicle speed, an intelligent system based on smartphone technology was devised in [22]. The device detects speed bumps with a gravity sensor, and the third equation of motion was used for speed reduction. Data were collected using crowdsourcing and a variety of vehicles. Although the system produced good results, constraints connected with this study include the lack of consideration of the width and depth of potholes and speed bumps.

In [23], using vehicle driving noise, a non-compression auto-encoder was used for identifying road surface anomalies. The authors suggested the non-compression auto-encoder (NCAE) deep learning-based anomaly detection platform, which was cost-effective and operated in real time. Through convolutional operations, the developed platform can predict backward and forward time-series causality data. Furthermore, the architecture outperforms the compared anomaly detection methods in the aspect of the Area Under Receiver Operating Characteristic Curve (AUROC). When compared to vision-based approaches, the high cost and complexity of gathering sensor data is difficult. This strategy also makes determining the type of anomaly challenging.

Additionally, [24] devised a system for detecting speed bumps using accelerometric characteristics and a Genetic Algorithm (GA). The authors created a unique approach for detecting road irregularities (i.e., speed bumps). A GPS sensor, an accelerometer, and a gyroscope sensor put in an automobile were used in this approach. Data are obtained from the sensors after the car has driven through numerous streets. GA is then utilised to create a logistic model that successfully identifies road anomalies using a cross-validation technique. In a blind evaluation, the suggested model had a 0.9714 accuracy, a less than 0.018 false positive rate, and an AUROC of 0.9784. However, the aforementioned limitations of sensor-based methods apply here as well.

The authors in [25] created an innovative technique for identifying road bumps using an accelerometer-based Android program. This program analyses accelerometric sensor data obtained from several roadways to assess the correctness of the recommended approach. The study implemented a noise threshold to differentiate between phone shaking and accelerometric data, which is not an efficient process.

Furthermore, [26] focuses on the development of a new algorithm for detecting and characterising potholes and bumps from signals acquired using an accelerometer. The proposed algorithm utilises a wavelet-transformation-based filter to decompose the signals into multiple scales and then applies a spatial filter to the coefficients to detect road anomalies. The characterisation of these anomalies is achieved using unique features extracted from the filtered wavelet coefficients. The results of the analyses show the effectiveness of the proposed algorithm in accurately detecting and characterising potholes and bumps.

In [27], the detection and identification of road anomalies and obstacles in the road infrastructure using data collected from an Inertial Measurement Unit (IMU) installed in a vehicle is presented. The authors evaluate the use of Convolutional Neural Network (CNN) for this task, as well as the use of time-frequency representation (spectrogram) as input to the CNN instead of the original time domain data. The proposed approach was tested on an experimental dataset collected from 12 vehicles driving over 40 km of road and showed improved results compared to previous shallow machine learning algorithms and the use of CNN on time domain data. The authors report an identification accuracy of 97.2% after extensive optimisation of the CNN algorithm and the spectrogram implementation.

2.1.2. Deep Learning-Based Methods

Several research studies implemented road anomaly detection using machine learning (ML) methods. These ML techniques were implemented both in sensor-based methods and visual methods. An ML method for the determination of road surface anomalies using some sensors of a smartphone was presented in [12]. Using gyroscope, GPS, and accelerometer data obtained from cellphones, the author investigated different supervised ML approaches for efficiently classifying the conditions of road surfaces. The work concentrated on the classification of three major labels, which are smooth roads, potholes, and deep transverse cracks. The study also found that using characteristics from all three dimensions of the sensors produced more accurate findings than utilising only one axis. Furthermore, the model performance was assessed with respect to deep neural networks. The DT and SVM methods had smaller classification times than the neural-network-based methods. Loss of accuracy and precision was observed, resulting from the small dataset and disproportional distribution of class instances. In addition, the use of three types of sensors increases the complexity of the system.

In [13], using multispectral images from unmanned aerial vehicles, an asphalt pavement pothole and fracture detection system was created. The approach displayed the spectral and spatial characteristics of road abnormalities, and ML techniques such as SVM, ANN, and random forest were utilised to differentiate between undamaged and damaged pavements. The classification accuracy was 98.3 percent; however, the system was unable to identify cracks of less than 13.54 mm width due to the limits of spatial resolution of the UAV pavement photos.

ML techniques have shown impressive results in vehicle perception tasks. However, in ML methods based on classification such as Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), the effectiveness of the algorithm relies on the input data representation and the feature extraction method implemented [28]. To tackle the limitations of traditional ML algorithms, deep learning (DL) has been widely implemented in classification and object detection tasks, especially in the area of AV perception.

The authors in [29] proposed to develop a dual-stage YOLO v2-based road marker detector capable of operating in real time and possessing lightweight spatial transformation-invariant categorisation. The authors presented a two-staged technique to handle distorted road marker recognition and balance performance metrics such as recall and precision. The developed spatial transformation layer was able to tolerate road markings in the second stage which were distorted, resulting in enhanced accuracy. The built network was able to run at a speed of 58 FPS on a single GTX 1070 under varied scenarios. The two-stage model had an 86.5 percent mean average precision, whereas the RM-Net model had a 97.5 percent accuracy. The metrics were shown to be superior to standard classification and detection approaches. Lanes and road boundaries were not taken into account in this study, leaving potential for further research in that area.

In addition, [10] using street-level photos and geographical information, developed a road environment categorisation model. Based on a deep convolutional neural network (CNN), the research presents a novel framework for autonomous systems capable of the identification of street-level photos in a road scene. The model was pre-trained on the ImageNet dataset before it was trained on the KITTI dataset using transfer learning. The model classified urban, rural, and highway street photos with an accuracy of 86 percent. The approach assessed the various types of roadways. However, the road conditions were not investigated, leaving a void for future study directions.

A technique for road crack identification using multi-scale Retinex, which was combined with wavelet transform, was developed in [30]. To eliminate the halo formed by the Retinex technique and to reduce picture distortion, the wavelet transform was incorporated into the standard multi-scale Retinex method. The system had a recognition accuracy of 95.8 percent, which was higher than the traditional algorithm's accuracy of 75.1 percent. The approach solely targets cracks, not other road oddities like potholes or speed bumps.

Furthermore, a technique for road anomaly and drivable area detection using a dynamic fusion module (DFM) was developed in [11]. The work created a road anomaly and drivable area detection standard for mobile robots by comparing existing contemporary single modal and semantic segmentation CNNs based on data-fusion and utilising six visual feature modalities. Furthermore, a novel module known as the DFM that can be simply implemented in existing data-fusion networks to successfully and efficiently fuse diverse types of visual characteristics was developed. The approach was capable of distinguishing between drivable areas and those with road abnormalities. When tested against other published methodologies on the KITTI dataset, the model had an average accuracy of 94.05 percent. Vehicles and pedestrians were the anomalies studied in this study; road problems were not examined.

The authors in [31] developed a CNN-based pothole detection system using thermal imaging. The study looked at the feasibility of using thermal imaging in pothole identification. A comparison of the self-built CNN algorithm and existing pre-trained models was also performed, with the results revealing that pictures were accurately recognised, with the highest accuracy value of 97.08 percent utilising one of the pre-trained CNN-based residual network models. The pothole identification problem was modeled as a classification challenge rather than an object detection operation in the study. As a result, the potholes could not be identified in the photograph.

Similarly, in [32], a CNN for detecting potholes in vital road infrastructure was demonstrated. The research suggests a unique use of CNN using accelerometer information for the identification of potholes. Data are captured using an iOS-based smartphone put on a car's dashboard and running a specialised app. The results reveal that the proposed CNN technique outperforms previous models in terms of computing complexity and accuracy in pothole identification. The model has a 98 percent accuracy. The pothole identification problem was modeled as a classification challenge rather than an object detection operation in the study. As a result, the potholes could not be identified in the photograph. Furthermore, obtaining sensor data is difficult due to its high cost and complexity when compared to vision-based solutions.

To address the challenges posed by anomalies in road surfaces, [33]proposed a deep learning approach that uses various models including convolutional neural networks, LSTM networks, and reservoir computing models to automatically identify different types of road surfaces and to distinguish potholes from other destabilisations caused by speed bumps or driver actions. The experiments conducted using real-world data showed promising results and a high level of accuracy in solving both problems.

Furthermore, [20] built a DL-based edge AI-based automatic identification and categorisation method for road irregularities in VANET. The authors introduced a new approach based on VANET and edge AI for the automated identification of irregularities in roads by AVs and communication of relevant information to oncoming vehicles. ResNet-18 and VGG-11 are used for the identification and classification of roads with anomalies and plain roads without abnormalities. The model exhibited accuracy, precision, and recall values of 99.92 percent, 99.85 percent, and 99.85 percent, respectively. The study modelled the pothole detection problem as a classification task and not an object detection operation. Thus, the potholes could not be localised in the image.

2.2. Visual-SLAM

V-SLAM is a technology in which an autonomous navigation system employs a vision sensor to build and update a map of an unfamiliar area while tracking its location and orientation inside that environment [17,34]. Camera data, as opposed to other sensor data such as lidar, may give rich and extensive information, which improves high-level operations [35]. In a world reference frame, the camera path is represented as a collection of relative positions. Landmarks, which are objects or keypoint elements in each frame, reflect the surroundings. In static situations, landmarks stay stationary, but in dynamic environments, landmarks change location [35]. In this research, two main challenges have been highlighted regarding V-SLAM's applicability [36]:

- **Reliability in Outdoor Environments**: V-SLAM's reliability, especially in outdoor settings, requires further enhancement. The limitations of lidar and radar sensors in extreme climatic conditions, combined with their exorbitant prices, render them unsuitable for exterior conditions [14–16]. Although laser scans offer high precision and strong resistance against interferences, they lack the capability to provide semantic details about their surroundings [17]. While V-SLAM methods aim to address these gaps, they remain vulnerable to environmental factors, such as varying light conditions [37].

- **Operability in Dynamic Scenes**: Conventional SLAM and V-SLAM methods typically operate under the presumption of a static environment. This assumption is often not accurate, leading to V-SLAM methods that are designed for static scenes to falter in dynamic settings [38]. Such dynamic scenes often feature moving elements that must be accounted for during localisation and mapping processes. For instance, ORB-SLAM cannot differentiate whether the feature points extracted belong to stationary or moving objects [17]. Even though extensive research has been directed towards object detection [18] and semantic segmentation [19], the exploration of V-SLAM in highly fluid environments like roads and highways remains insufficient. There remains a pertinent need for autonomous systems to gain a comprehensive understanding of dynamic scenarios and to interact appropriately with moving elements [19,39].

3. Research Methodology

3.1. Road Anomaly Detection

3.1.1. Model Overview

The road surface characterisation via detection of road anomalies is achieved with a deep learning-based technique. In the developed object detection framework, we commence with image acquisition, leveraging cameras to capture real-time visuals. These images undergo essential pre-processing, focusing on data augmentation, cleaning, and validation. Subsequently, salient features are extracted by the deep learning model. Utilising the trained model, the objects in the images are identified and classified. To refine these preliminary findings, post-processing techniques, like non-maximum suppression, are deployed. The interpreted data from detected objects then guide the vehicle's subsequent reactions, ensuring timely and appropriate responses. This intricate yet swift process underscores the system's efficacy in real-time scenarios, providing vehicles with a dynamic, responsive tool for seamless navigation and safety. The developed model identifies potholes, cracks, and unmarked speed bumps. The flow diagram of the system operation is presented in Figure 1, while the pipeline for the model is presented in Figure 2.

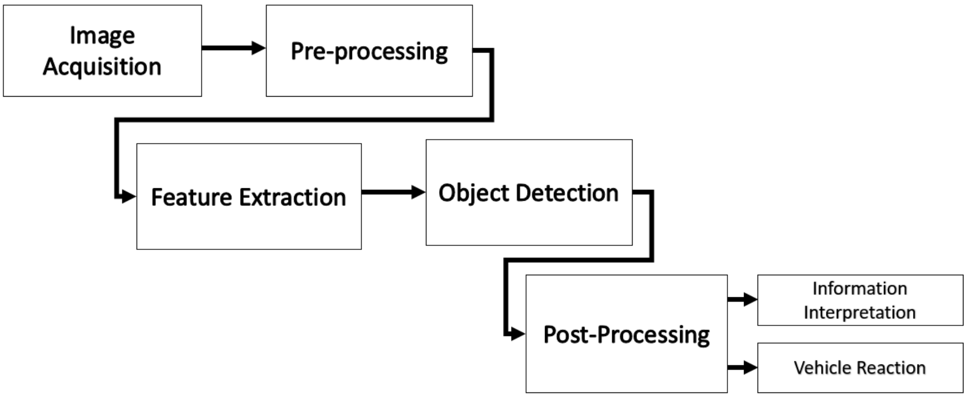


Figure 1. Road surface characterisation flow.

The deep learning technique implemented for the road anomaly detection model is YOLO v4 [40]. YOLO stands for You Only Look Once. This implies that the algorithm only requires one forward propagation pass through the neural network to carry out its prediction task, unlike other models that generate regional proposals. The YOLO algorithm carries out classification and generation of a bounding box simultaneously. The model provides identified object classes, a bounding box around the item, and confidence ratings for each identified item [41]. YOLO has been shown to be more accurate and quicker than other object identification models like R-CNN [18]. Furthermore, in comparison to other DL algorithms, the approach offers modest processing needs and even a 'tiny' variant for implementation on embedded devices. Tiny-YOLO v4 was implemented in this study to minimise the computational and memory demands of the model. In addition, YOLO v4 provides a more accurate and faster advanced detector over other available alternatives. YOLO v4 provides more accuracy in detecting partially occluded or small objects and has been trained on a larger and more diverse dataset than its counterparts such as YOLO v7 [40,42].

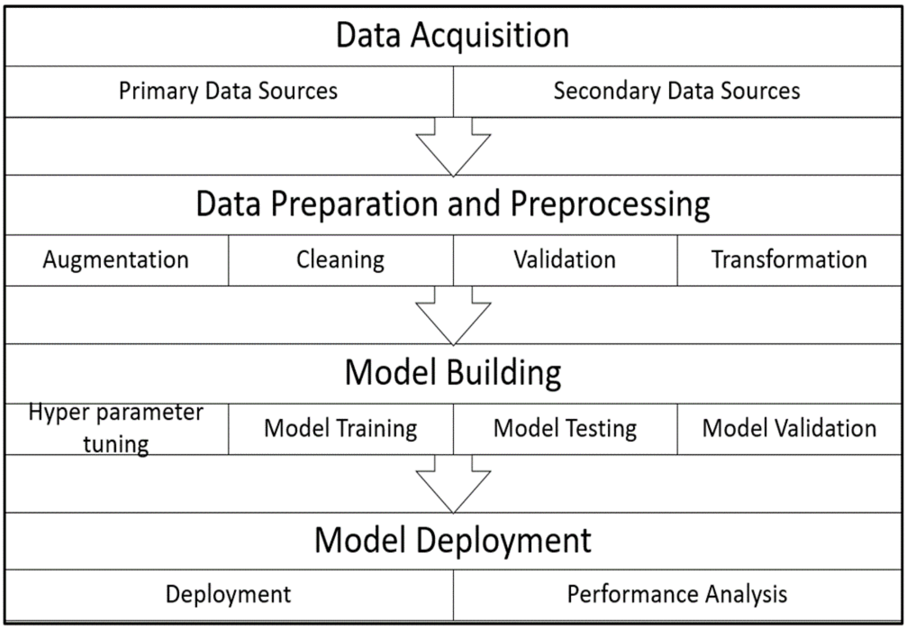


Figure 2. Road surface characterisation model pipeline.

The YOLO v4 architecture is divided into three key elements. As indicated in Figure 3, these elements include the backbone, the neck, and the detection head. CSP-Darknet53 (Cross Stage Partial Darknet53) is used as the backbone. This model is distinguished by its better resolution of input and bigger fields of reception, both of which aid in visualising whole items and detecting minute things in an image. The PANet (Path Aggregation Network) serves as the means of aggregating parameters from the various levels of the backbone contained in the neck. The section detection head forecasts the bounding boxes, categorisation, and score. The YOLO v3 model is used to carry this out.

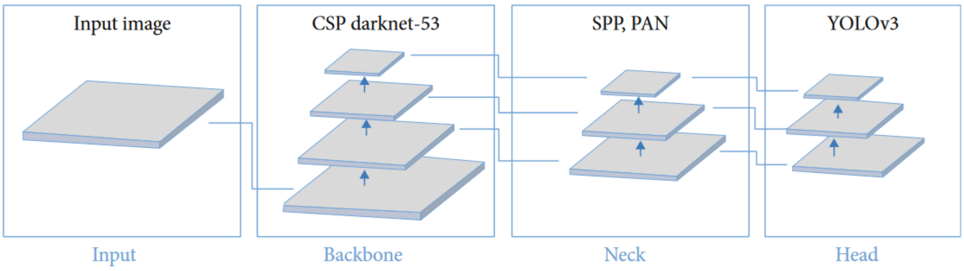


Figure 3. YOLO v4 structure [41].

Tiny-YOLO v4 was implemented in this study to minimise the computational and memory demands of the model. The architecture of this model is presented in Figure 4. Table 1 shows details of the Tiny-YOLO v4 architecture and how it compares to the traditional YOLO architecture.

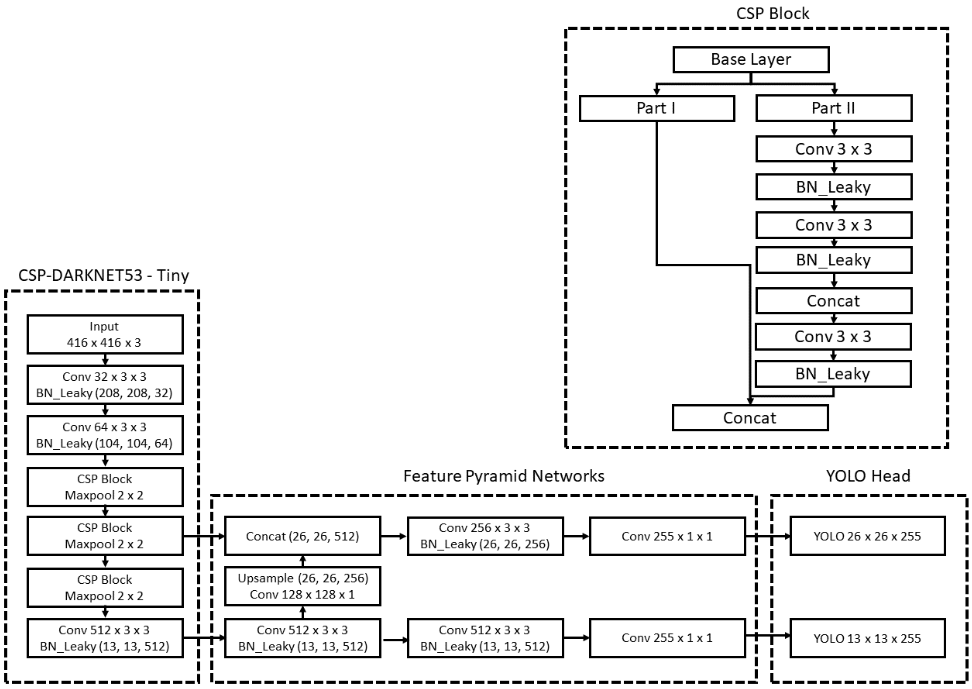


Figure 4. Tiny-YOLO v4 architecture.

Table 1. Comparison between YOLO v4 and YOLO v4-tiny.

Feature/Component	YOLO v4	YOLO v4-tiny
Backbone Network	CSPDarknet53	CSPDarknet53-tiny
Main Module	ResBlock	CSPBlock
Activation Function	Mish	LeakyReLU
Feature Fusion Technique	SPP & PAN	Feature Pyramid Network
Prediction Scales	Multiple	13 × 13, 26 × 26
Input Size	416 × 416	416 × 416

In assessing Table 1, which contrasts the YOLO v4 and YOLO v4-tiny architectures, several notable distinctions emerge. Firstly, while both variants adopt the CSPDarknet lineage for their backbone network, YOLO v4-tiny streamlines its implementation with a more lightweight CSPDarknet53-tiny variant. This trend towards simplification in YOLO v4-tiny is further evident in its employment of the CSPBlock as its primary module, diverging from YOLO v4's use of ResBlock. Additionally, YOLO v4-tiny opts for the LeakyReLU activation function, a deviation from YOLO v4's more complex Mish function. While YOLO v4 employs a more intricate fusion technique combining Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN), YOLO v4-tiny leans into the efficiency of the Feature Pyramid Network. Furthermore, the prediction scale for the tiny variant is more constrained, focusing on 13 × 13 and 26 × 26 grids. Yet, intriguingly, both architectures maintain an identical input size of 416 × 416, showcasing their compatibility in processing similar image resolutions. This comparative analysis underscores the strategic design choices behind YOLO v4-tiny, aiming for real-time efficiency while balancing performance trade-offs.

3.1.2. Data Acquisition

Given the scarcity of online benchmark road anomaly datasets and the limited publicly available datasets for pothole identification [43], 576 images of road anomalies were obtained manually from primary and secondary sources. The primary data sources were locations around Federal University of Technology Minna campus and contained potholes, cracks, lanes, and speed bumps. Additionally, the secondary data source included images obtained online containing the desired road anomalies. In addition, negative samples, which are road images containing no anomalies, were used to improve the detection performance of the system. The variability in the data orientation would make the model robust and invariant to scale as they reflect real-world scenarios that object detection models will encounter, leading to more robust and generalisable models [44,45]. Table 2 shows a decomposition of the data classes obtained with their corresponding number of instances. Figure 5 shows examples of the images used in the study.

Table 2. Data classes and instances.

Class	Instance	Percentage
Potholes	144	25%
Cracks	144	25%
Speed Bumps	144	25%
Negative Samples	144	25%
Total	576	100%

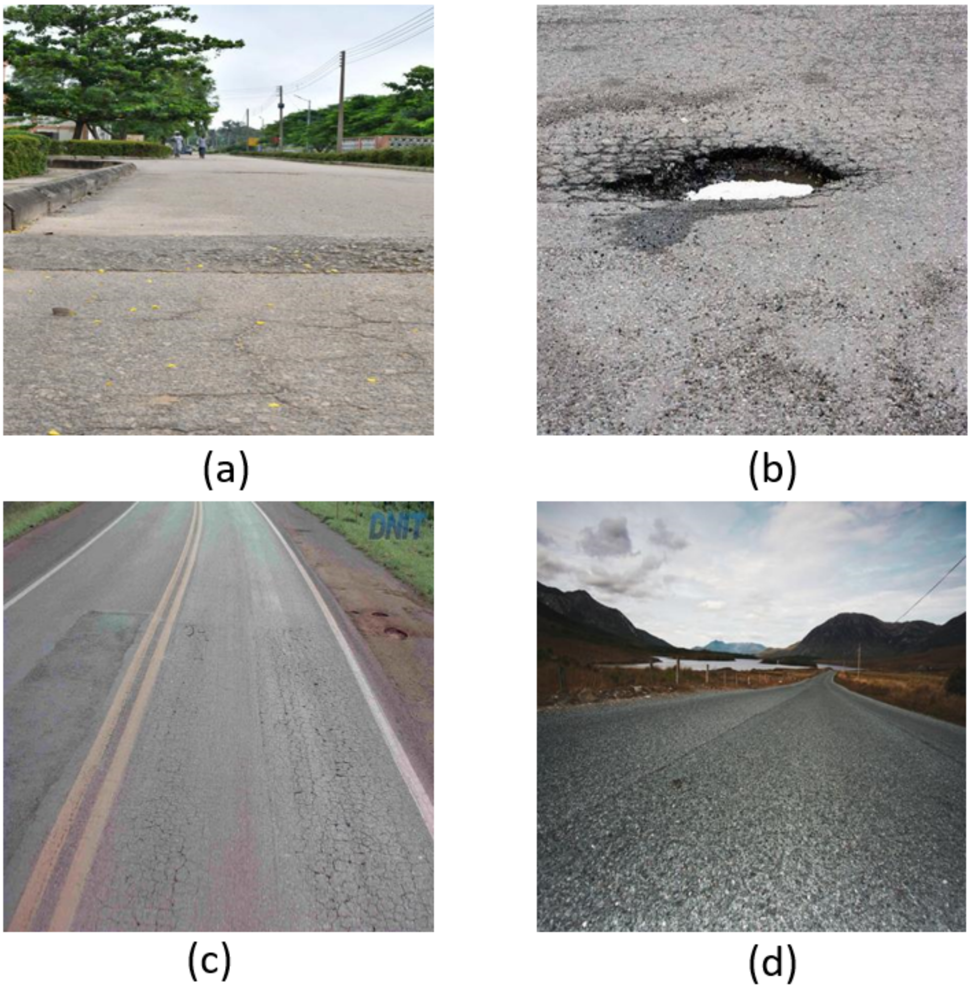


Figure 5. Road data instances: (**a**) unmarked speed bump, (**b**) pothole, (**c**) crack, (**d**) negative sample.

3.1.3. Data Preparation and Pre-processing

For successful training of an object detection model such as YOLO v4, the following parameters are required from an image:

- **Class**: This is the category where the object belongs. The class values start from 0 to n−1 number of objects.
- **xmin**: This refers to the central value on the x-axis where the boundary box originates. This parameter takes a value of 0 to 1.
- **ymin**: This refers to the central value on the y-axis where the boundary box originates. This parameter takes a value of 0 to 1.
- **w**: This is the boundary box width. This parameter takes a value of 0 to 1.
- **h**: This is the boundary box height. This parameter takes a value of 0 to 1.

These parameters are acquired by labelling each image by drawing bounding boxes around the anomalies and assigning the appropriate class label to the boxes. This process was carried out using the LabelImg software [46]. This application is an open source tool that can be used to draw and save bounding box parameters of each image. These parameters are saved to a .txt file. In addition, the images were resized to a size of 360 by 240. This was performed to enhance faster training on the GPU without compromising the information required from the image.

Given the small quantity of the gathered dataset, data augmentation techniques were used to improve the volume and variety of the data. The augmentation procedures were selected based on their ability to increase not only the volume of the data but also its variability in terms of light intensity and blurriness. The augmentation procedures utilised reflect the influence of external disturbance factors such as blurred and low quality images, as well as image contrasts affected by lighting conditions. The procedures used for augmentation were as follows:

1. **Blurring**: This involves reducing the intensity of sharp transitions between pixel values. This is achieved using a convolution of the original image with a mean filter kernel, as shown in Equation (1).

$$y(i,j) = \sum_i \sum_j \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} x(i,j) \tag{1}$$

The parameter *y(i, j)* is the blurred image, *x(i, j)* is the original image, and $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ is the mean filter kernel.

2. **Contrast adjustment**: In this case, the contrast of the image was enhanced by altering the hue, saturation, and brightness values. This is achieved using Equation (2).

$$c(i,j) = truncate(CCF * i(i,j) + BAV) \tag{2}$$

In Equation (2), *c(i, j)* is the contrast adjusted image, *CCF* is the contrast correction factor, *i(i, j)* is the original image, and *BAV* is the brightness adjustment value. The truncate function ensures the pixel intensity values stay within 0 to 255.

3. **Noise addition**: Gaussian noise was added to the images. This was achieved by randomly altering the intensities of pixels with values ranging from 0 to 1, normally distributed. This process is presented in Equation (3).

$$f(i,j) = i(i,j) + n(i,j) \tag{3}$$

where *f(i, j)* is the new image, *i(i, j)* is the original image, and *n(i, j)* is the noise which has a Gaussian random distribution with zero mean.

4. **Mosaic**: This is a data augmentation technique combines four images into a single image. In this augmentation technique, a cropped image is covered with a rectangle region of other images. The labels are adjusted accordingly in this operation to reflect the altered image. This technique allows for detection of objects outside their usual context.

A total of 2880 images were obtained from the augmentation process. This significantly increased the number of images from the previous 576 images. In addition, the 2880 images were divided into training, testing, and validation sets after randomly shuffling the images. The shuffling process ensured that the model had reduced bias, improved generalisation, better performance, and minimal overfitting.

3.1.4. Model Building

The road anomaly detection model was developed on the Google Colab platform with Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM. The characterisation model was developed using transfer learning on a pre-trained model. Instead of initiating the learning process from scratch, one begins with models that have already been trained on vast datasets, absorbing a wealth of knowledge. This pre-existing knowledge serves as a foundation, and the learning then becomes more about adapting this base knowledge to a new, specific context or dataset. Through this adaptation, a significant reduction in computational cost and training time can be achieved. By effectively retraining only the final layers of the model, it becomes attuned to the nuances of the new dataset. This process allows for the rapid development of highly effective models, even when the available data for the specific task might be limited. After data augmentation, 2880 images were obtained and used for the training. A training, validation, and testing ratio of 70:20:10 was adopted for this study. The hyper-parameters selected for the model training are presented in Table 3.

Table 3. Selected hyper-parameters for YOLO v4 model.

Parameter	Value
Network Input Size	416 × 416
Total Images	2880
Number of Classes	3
Batch Size	64
Subdivisions	16
Maximum Batches	10,500
Learning Rate	0.001
Activation	Leaky ReLU
Burn in	1000
Training–Validation–Testing Split	70:20:10

The training of the neural network is carried out with a momentum of 0.9 and a weight decay of 0.0005. The optimisation of the network's weights is achieved using stochastic gradient descent (SGD). Given the loss function L, the weight update rule for each epoch is as follows:

$$w_{t+1} = w_t - \eta \nabla L \tag{4}$$

where η is the learning rate, w represents the weights, and ∇L is the gradient of the loss function with respect to the weights.

The primary activation function used in the network's layers is Leaky ReLU, which is represented mathematically as follows:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \tag{5}$$

where α is a small constant, typically set around 0.01, though it might vary depending on the specific implementation.

3.1.5. Model Performance Evaluation

The generated model's performance was measured in terms of precision, recall, average precision (AP), and mean average precision (mAP). The metrics are commonly used in evaluating object identification algorithms and may be used to evaluate the model's performance on various datasets.

Precision is a measure of the accuracy of the model in detecting road anomalies. This metric is evaluated using Equation (6) [47].

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The parameter TP represents the true positives, and FP represents the false positives.

Recall is a measure of the model performance in detecting all anomalies in the images. This metric is evaluated using Equation (7) [47].

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

In Equation (7), the parameter FN represents false negatives. Because there are a high number of instances that should not be recognised as objects, the true negative (TN) metric does not apply in object detection activities.

The parameter AP represents the area under the precision–recall (PR) curve, which is a plot showing precision as a function of recall. The mAP is the average of all classes' AP scores. The AP and mAP are shown in Equations (8) and (9), respectively [47].

$$AP@w = \int_0^1 p(r)dr \tag{8}$$

$$mAP@w = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{9}$$

The AP and mAP values are evaluated with a confidence threshold, ω; this is a proportion of the intersection between the ground truth and prediction area to the corresponding union of the ground truth and prediction area. A commonly used ω value is 0.5.

The Intersection over Union (IoU) is used in computing the total loss at each batch. The Complete IoU (CIOU) loss function is utilised in the model development and the function is presented in Equations (10) [43] and (11) [40].

$$IOU = \frac{A \cup B}{A \cap B} \tag{10}$$

$$CIOU = S(B, B^{gt}) + D(B, B^{gt}) + V(B, B^{gt}) \tag{11}$$

$S(B, B^{gt})$ is the overlap region between the projected and ground truth bounding boxes, $D(B, B^{gt})$ is the normalised IoU loss between the expected and ground truth bounding boxes' centers, while $V(B, B^{gt})$ is the aspect ratio consistency. All these parameters are normalised to have values between 0 and 1, thus, making them invariant to the regression scale.

3.2. RA-SLAM

3.2.1. Overview

The modified V-SLAM technique developed is called RA-SLAM (Road Anomaly SLAM). This technique was built based on ORB-SLAM, which is an open source visual SLAM technique that is suitable for monocular, stereo, and RGB-D cameras [48]. This choice of process is based on its high accuracy and precision [18]. ORB-SLAM comprises three primary elements: Tracking, Local Mapping, and Loop Closure. Within the Tracking phase, the system locates the camera, pulls out key points, and decides when to introduce a fresh keyframe, all by leveraging ORB features from the captured images. The Local Mapping phase employs these keyframes to recreate the environment around the camera's position. The Loop Closure phase, meanwhile, scouts for loops within the keyframes to refine the resulting map. These ORB-SLAM stages focus specifically on tracking, creating maps, and identifying loops. To incorporate object detection into this framework, this research turned to deep learning methods. This integration with the V-SLAM technique is presented in Figure 6.

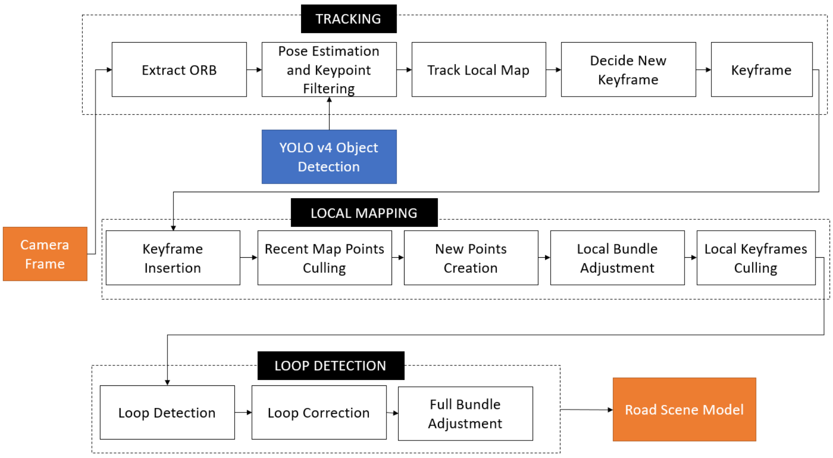


Figure 6. Modified V-SLAM technique.

From Figure 6, it can be observed that the YOLO v4 technique is instrumental in the keypoint selection (Tracking component). The technique detects objects in the scene and relays the information to the keypoint aggregation module of the tracking component. This in turn provides information on the detected objects in the scene.

3.2.2. Object Detection and Keypoint Aggregation

Algorithm 1 presents the keypoint aggregation process using YOLO v4. The procedure initiates with the YOLO method pinpointing objects in the view, subsequently producing bounding outlines. The system then assesses if these bounding outlines lie within the targeted region, adjusting the outline dimensions if needed. These bounding outline coordinates serve as the focal points for drawing out ORB characteristics near the detected object. The ORB points found around the object are then combined with the points recognised in the entire picture. Such an approach guarantees that the characteristics of the spotted objects are incorporated during the localisation and mapping stages.

Algorithm 1 KEYPOINT AGGREGATION ALGORITHM

```
Require: ORB features of main image, $F_m$
Require: YOLO bounding box list of detected objects, $BB$
Ensure: Filtered Keypoints, $F_{\text{new}}$
keypoint_list ← 0
$F_{\text{all}} \leftarrow 0$
for $i = 1$ to size( $BB$ ) do
    resize $BB(i)$ to fit $F_m$ dimension
    extract ORB features, $F_b$ from $BB(i)$
    keypoint_list ← keypoint_list + $F_b$
end for
$F_{\text{new}} \leftarrow F_m +$ keypoint_list
```

4. Results

4.1. Road Anomaly Detection

The training process for the road surface characterisation model concluded after 3.6 h. Figures 7 and 8 show the performance of model based the loss function (Equation (11)) and the mAP@0.5 (Equation (9)). The curves were obtained by plotting the loss value and mAP@0.5 against the batch processed. The plots highlight the learning ability of the model as it learns incrementally from the training dataset. It can be observed from the figures that as the number of batches processed increases, the loss value reduces while the mAP@0.5 increases. It can be observed from the figures that the training loss reduces from 0.56 at batch 500 to 0.14 at batch 10,500. On the other hand, the mAP@0.5 increases from 0% at batch 500 to 95.34% a batch 10,500.

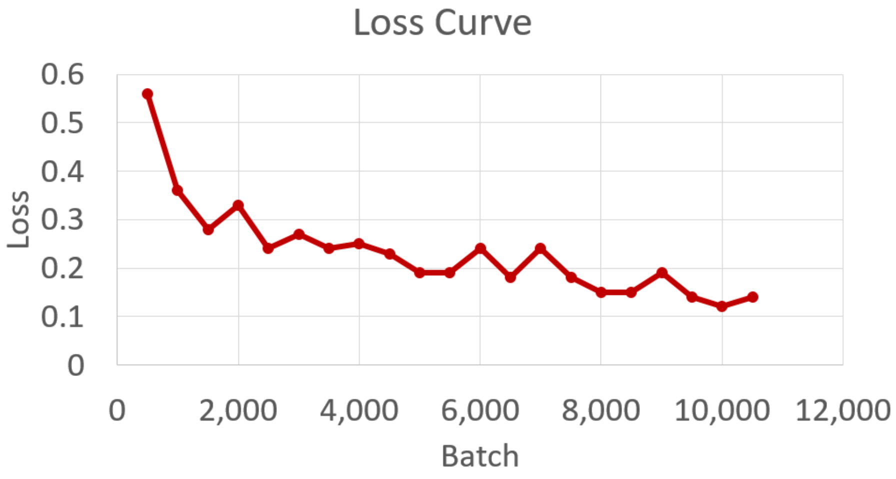


Figure 7. Loss curve.

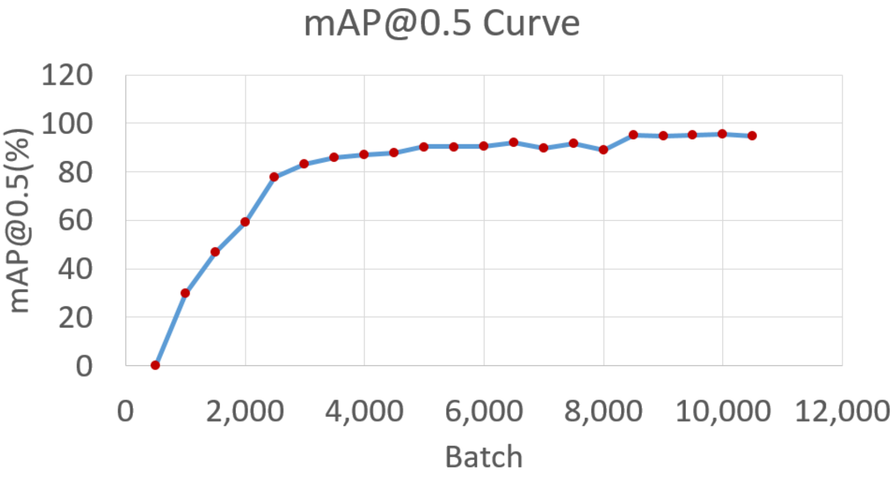


Figure 8. mAP@0.5 curve.

Table 4 presents the performance evaluation results of the training process. The table shows the correctly predicted classes (TP) and the incorrectly predicted classes (FP). Additionally, the table shows the AP values, which is the area under the precision–recall (PR) curve. These parameters provide an indication of how well the model trained. The AP values are evaluated using Equation (6). It can be observed from the table that the Speed Bump, Pothole, and Crack classes had average precisions of 99.15%, 97.65%, and 87.71% respectively. These values indicate that the model exhibited a good performance on the training data. The Crack class showed the lowest AP value, and the Speed Bump class had the highest performance of the three classes. The relatively low performance of the crack class can be attributed to difficulty of the model in detecting the class based on its size, shape color, or background. Considering that the dataset was balanced, it was not as a result of imbalanced class distribution [45].

Table 4. Training evaluation results.

Class ID	Road Anomaly	TP	FP	AP (%)
0	Speed Bump	143	3	99.15
1	Pothole	139	12	97.65
2	Crack	125	33	87.71

In Table 5, the performance evaluation results of the testing process is presented. The table indicates TP and FP instances, as well as the AP values. These parameters provide an indication of how well the model performed on the test dataset after training. The table shows that the Speed Bump, Pothole, and Crack classes had average precisions of 99.89%, 99.55%, and 91.55% respectively. These values indicate that the performance of the model on the testing data was good, with the Crack class having the lowest and the Speed Bump class having the highest performance of the three classes.

Table 5. Testing evaluation results.

Class ID	Road Anomaly	TP	FP	AP (%)
0	Speed Bump	66	5	99.89
1	Pothole	72	4	99.55
2	Crack	74	17	91.55

The overall performance of the model during training and testing was evaluated and compared. The results of this performance are presented in Figure 9. The figure shows a comparison in terms of the precision, recall, F1-score, average IoU, and mAP@0.5. The figure highlights the performance of the model during training, where the model learns, as compared to during testing, where the model has learned. On one hand, the training performance exhibited a precision of 89%, a recall of 94%, an F1-score of 91%, an average IoU of 68.47%, and a mAP of 95.34%. On the other hand, the testing performance exhibited a precision of 89%, a recall of 96%, an F1-score of 92%, an average IoU of 68.72%, and a mAP of 97%. It can be observed from the figure that the testing performance of the model was slightly better than the training performance in all the metrics considered. This low difference between the performances also indicates that the model did not over-fit.

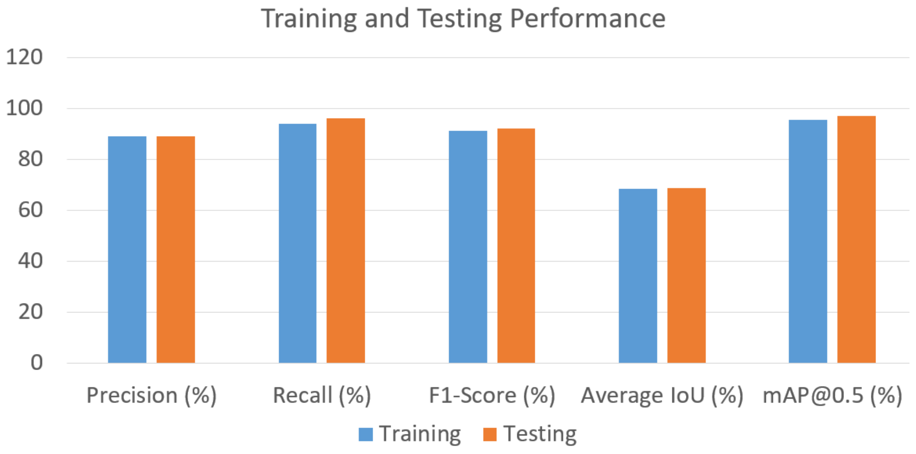


Figure 9. Training and testing performance.

Figure 10 presents the qualitative performance evaluation of the model in terms of its ability to detect road anomalies. The figure was obtained by testing the detector on random images which contained the specified anomalies. From the figure, it can be observed that the model was able to identify and localise the anomalies appropriately. This implies that based on the bounding box co-ordinates, the model can provide information about the position of the anomalies with respect to the scene.

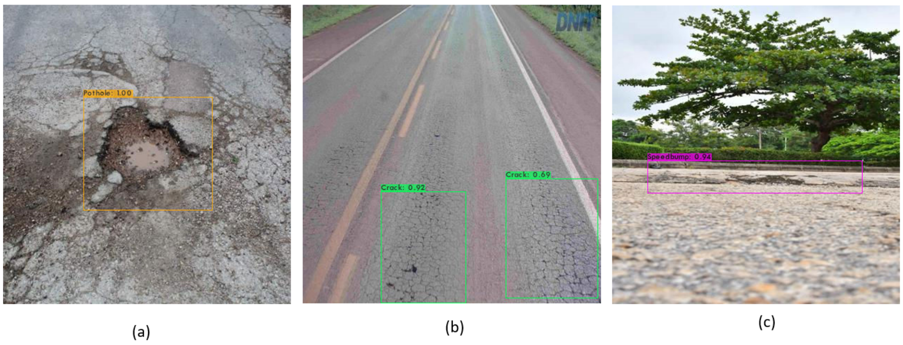


Figure 10. Qualitative performance evaluation: (**a**) detected pothole, (**b**) detected cracks, (**c**) detected speed bumps.

The performance of the model was compared to similar models in recent existing literature. The comparison was made in terms of precision, recall, F1-score, and mAP@0.5. Table 6 presents the comparative analysis of the different road anomaly detection models.

Table 6. Testing evaluation results.

Ref.	**Anomaly Type**	**Technique**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**mAP@0.5**
[49]	Potholes	YOLO v3	88.00	60.00	75.53	-
[49]	Potholes	YOLO v4	88.00	71.00	81.82	-
[50]	Potholes	YOLO	83.45	-	-	-
[51]	Potholes	Faster R-CNN	-	-	13.7	41.5
[51]	Potholes	SSD	-	-	7.6	18.5
[51]	Potholes	YOLO v3	-	-	42.0	34.7
[47]	Potholes	MobileNet v	42.0	56.0	47.9	47.4
[47]	Potholes	YOLO v1	82.0	69.0	74.0	79.55
[47]	Potholes	YOLO v2	81.0	76.0	78.0	81.21
[47]	Potholes	YOLO v3	77.0	78.0	78.0	83.60
[47]	Potholes	Tiny YOLO v4	76.0	75.0	76.0	80.04
[47]	Potholes	YOLO v4	81.0	83.0	82.0	85.48
[47]	Potholes	YOLO v5	**93.0**	83.0	87.0	95.00
[52]	Potholes	Faster R-CNN	78.0	73.0	-	-
Ours	Crack, Pothole, Speed Bump	Tiny YOLO v4	89.0	**94.0**	**91.0**	**95.34**

Table 6 presents a comparison between the developed model and existing road anomaly detection schemes in the literature. The table was obtained by reviewing recent work in the area of road anomaly detection and identifying the performance metrics in each case. Table 5 compares the precision, recall, F1-score, and mAP@0.5 values for the different models, thus providing insights into the more effective models. From the table, it can be observed that the model in [47] exhibited the highest precision, with 93%. Our developed model showed the highest recall, F-1 score, and mAP@0.5, with values of 94%, 91%, and 95.34% respectively. Furthermore, in comparison to other models that can identify only one type of anomaly, our model can identify three anomalies accurately. This implies that among the models compared, our model is the most effective in road surface characterisation and road anomaly detection based on the parameters measured.

4.2. RA-SLAM

The performance evaluation of the RA-SLAM algorithm was carried out using the root mean square error (RMSE) and mean absolute error (MAE) between the estimated and ground truth trajectories. The RA-SLAM algorithm was implemented in MATLAB on a computer with an Intel core i7 processor operating at a clock speed of 2.2 GHz. The system had 8 GB of RAM and an NVIDIA GeForce GTX GPU with a size of 8GB, which was utilised for accelerating computations in the algorithm. The modified V-SLAM technique was tested on the KITTI odometry benchmark dataset, which consists of stereo sequences recorded from a car driving in an urban environment. The KITTI dataset is a widely used benchmark in computer vision and robotics research, providing a large-scale dataset for various tasks such as object detection, 3D object detection, stereo, optical flow, and more. The dataset provides ground truth poses and 3D point clouds for evaluation of the SLAM algorithm. Specifically, eight sequences of the dataset were used for evaluation. The modified V-SLAM technique was tested on eight sequences of the KITTI odometry benchmark dataset. The technique (RA-SLAM) was compared to the conventional ORB-SLAM, both of which were implemented in the same environment. The generated map for the sequences is presented in Figure 11.

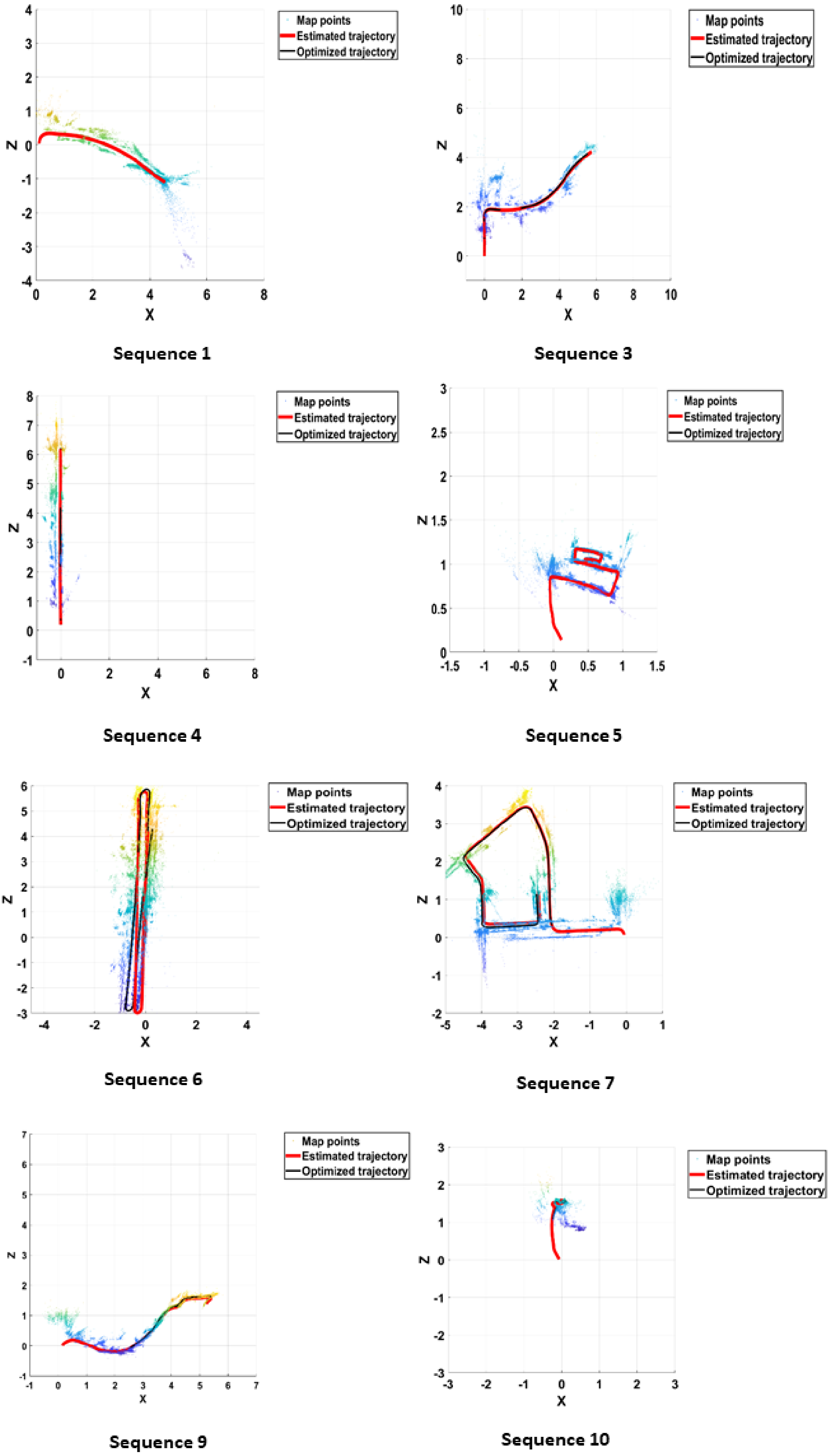
Throughout multiple test sequences, ORB-SLAM and RA-SLAM consistently showcased their proficiency in identifying and monitoring keyframes, albeit with slight variations in detection counts between the two. Evaluating their performance using metrics such as the root mean squared error (RMSE) and the mean average error (MAE), the outcomes showed a mix of results. In specific sequences, RA-SLAM surpassed ORB-SLAM in terms of RMSE and MAE, pointing to superior camera pose estimations and enhanced mapping precision. Conversely, there were instances where ORB-SLAM either matched or outperformed RA-SLAM. Table 7 shows the obtained error values in all test cases.

Table 7. Performance comparison of RA-SLAM against ORB-SLAM.

KITTI Seq. No.	No. of Frames	ORB-SLAM KFs	ORB-SLAM RMSE	ORB-SLAM MAE	RA-SLAM KFs	RA-SLAM RMSE	RA-SLAM MAE
1	1101	501	1.0000	1.0000	457	1.0000	1.0000
3	801	194	0.1154	0.1151	188	0.0963	0.0965
4	271	82	0.0000	0.0000	83	0.0000	0.0000
5	2761	801	0.2532	0.2526	806	0.2795	0.2801
6	1101	447	0.1285	0.1278	453	0.0852	0.0807
7	1101	289	0.0384	0.0382	285	0.0432	0.0433
9	1591	647	0.2405	0.2397	665	0.2357	0.2360
10	1201	440	0.2453	0.2444	441	0.1281	0.1283

1

Across the KITTI sequences compared between RA-SLAM and ORB-SLAM, the performance in terms of RMSE and MAE varied. RA-SLAM demonstrated lower RMSE and MAE values in Sequences 3, 6, 9, and 10. In Sequences 1 and 4, both methods yielded identical values, showcasing no distinguishable difference in their performances. Conversely, ORB-SLAM exhibited a slight edge, displaying lower RMSE and MAE figures in Sequences 5 and 7. This comparison highlights RA-SLAM's generally superior performance, even though the two algorithms exhibited varied efficacy across different testing sequences.



Sequence 1

Sequence 3

Sequence 4

Sequence 5

Sequence 6

Sequence 7

Sequence 9

Sequence 10

Figure 11. Map generated from KITTI sequences.

5. Conclusions

In this study, a CNN-based road anomaly detection model was developed, utilising the YOLO v4 object detection model. This technique accurately identified various road features, including cracks, potholes, and speed bumps. The experimental results showed impressive precision, recall, and F1-score values of 89%, 94%, and 91%, respectively. The model's performance, underlined by a mAP@0.5 value of 95.34%, surpassed many existing techniques, especially given its ability to identify three distinct anomalies, while most counterparts detect only one. In tandem with this, a refined visual SLAM technique called RA-SLAM was developed, enhanced by the aforementioned anomaly detection algorithm. When comparing RA-SLAM against ORB-SLAM using the KITTI sequences, the former often outperformed the latter, demonstrating superior accuracy and showing greater scalability potential. Yet, there were instances where ORB-SLAM outperformed RA-SLAM, showing the continued relevance of feature-based techniques in certain contexts. Future work will focus on implementing the developed algorithm on an AV prototype to ascertain its performance in real time.

Author Contributions: Conceptualisation, J.A.B. and S.A.A.; Methodology, J.A.B. and S.A.A.; Software, J.A.B.; Validation, A.M.A. and S.A.A.; Formal analysis, J.A.B., S.A.A. and A.M.A.; Investigation, J.A.B. and S.A.A.; Resources, S.A.A. and A.M.A.; Data Curation, J.A.B.; Writing—original draft preparation, J.A.B.; Writing—review and editing, J.A.B., A.M.A. and S.A.A.; Supervision, A.M.A. and S.A.A.; Project administration, S.A.A.; Funding Acquisition, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Tertiary Education Trust Fund (TETFUND), Nigeria, under the project titled 'Novel Road Accident Monitoring and Prevention System for Nigerian Roads and Highways' (TETF/ES/DR&D-CE/NRF2020/SETI/95/VOL1) under the 2020 National Research Fund (NRF) grant cycle.

Acknowledgments: The authors acknowledge the Tertiary Education Trust Fund (TETFUND), Nigeria, for funding this research under the project titled 'Novel Road Accident Monitoring and Prevention System for Nigerian Roads and Highways' (TETF/ES/DR&D-CE/NRF2020/SETI/95/VOL1) under the 2020 National Research Fund (NRF) grant cycle.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, H.; Sun, F.; Wang, Z.; Lin, C.; Zhang, C.; Xiong, R.; Deng, J.; Zhu, X.; Xie, P.; Zhang, S.; et al. China's Battery Electric Vehicles Lead the World: Achievements in Technology System Architecture and Technological Breakthroughs. *Green Energy Intell. Transp.* **2022**, *1*, 100020. [CrossRef]
2. Hong, J.; Zhang, C.; Chu, H.; Gao, B.; Wu, H.; Wei, G.; Liu, H.; Xu, F. Gear Downshift Control of Inverse-Automatic Mechanical Transmission of Electric Vehicle. *Green Energy Intell. Transp.* **2022**, *1*, 100005. [CrossRef]
3. Zheng, S.; Zhu, X.; Xiang, Z.; Xu, L.; Zhang, L.; Lee, C.H. Technology Trends, Challenges, and Opportunities of Reduced-rare-earth PM Motor for Modern Electric Vehicles. *Green Energy Intell. Transp.* **2022**, *1*, 100012. [CrossRef]
4. Chen, J.; Zhou, Z.; Zhou, Z.; Wang, X.; Liaw, B. Impact of battery cell imbalance on electric vehicle range. *Green Energy Intell. Transp.* **2022**, *1*, 100025. [CrossRef]
5. Liu, J.; Guo, H.; Shi, W.; Dai, Q.; Zhang, J. Driver-automation Shared Steering Control Considering Driver Neuromuscular Delay Characteristics Based on Stackelberg Game. *Green Energy Intell. Transp.* **2022**, *1*, 100027. [CrossRef]
6. Zhang, H.; Liu, C.; Zhao, W. Segmented Trajectory Planning Strategy for Active Collision Avoidance System. *Green Energy Intell. Transp.* **2022**, p. 100002. [CrossRef]
7. Yasin, J.N.; Mohamed, S.A.; Haghbayan, M.H.; Heikkonen, J.; Tenhunen, H.; Plosila, J. Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. *IEEE Access* **2020**, *8*, 105139–105155. [CrossRef]
8. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [CrossRef]
9. Luo, D.; Lu, J.; Guo, G. Road anomaly detection through deep learning approaches. *IEEE Access* **2020**, *8*, 117390–117404. [CrossRef]
10. Prykhodchenko, R.; Skruch, P. Road scene classification based on street-level images and spatial data. *Array* **2022**, *15*, 100195. [CrossRef]
11. Wang, H.; Fan, R.; Sun, Y.; Liu, M. Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms. *IEEE Trans. Cybern.* **2021**, *52*, 10750–10760. [CrossRef] [PubMed]

12. Basavaraju, A.; Du, J.; Zhou, F.; Ji, J. A machine learning approach to road surface anomaly assessment using smartphone sensors. *IEEE Sensors J.* **2019**, *20*, 2635–2647. [CrossRef]
13. Pan, Y.; Zhang, X.; Cervone, G.; Yang, L. Detection of asphalt pavement potholes and cracks based on the unmanned aerial vehicle multispectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3701–3712. [CrossRef]
14. Hu, T.; Sun, X.; Su, Y.; Guan, H.; Sun, Q.; Kelly, M.; Guo, Q. Development and performance evaluation of a very low-cost UAV-LiDAR system for forestry applications. *Remote Sens.* **2020**, *13*, 77. [CrossRef]
15. Ort, T.; Gilitschenski, I.; Rus, D. Autonomous navigation in inclement weather based on a localizing ground penetrating radar. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3267–3274. [CrossRef]
16. Yue, J.; Wen, W.; Han, J.; Hsu, L.T. LiDAR data enrichment using deep learning based on high-resolution image: An approach to achieve high-performance LiDAR SLAM using Low-cost LiDAR. *arXiv* **2020**, arXiv:2008.03694.
17. Li, F.; Chen, W.; Xu, W.; Huang, L.; Li, D.; Cai, S.; Yang, M.; Xiong, X.; Liu, Y.; Li, W. A mobile robot visual SLAM system with enhanced semantics segmentation. *IEEE Access* **2020**, *8*, 25442–25458. [CrossRef]
18. Soares, J.C.V.; Gattass, M.; Meggiolaro, M.A. Crowd-SLAM: Visual SLAM Towards Crowded Environments using Object Detection. *J. Intell. Robot. Syst.* **2021**, *102*, 50. [CrossRef]
19. Ai, Y.; Rui, T.; Lu, M.; Fu, L.; Liu, S.; Wang, S. DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with deep learning. *IEEE Access* **2020**, *8*, 162335–162342. [CrossRef]
20. Bibi, R.; Saeed, Y.; Zeb, A.; Ghazal, T.M.; Rahman, T.; Said, R.A.; Abbas, S.; Ahmad, M.; Khan, M.A. Edge AI-based automated detection and classification of road anomalies in VANET using deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*. [CrossRef]
21. Kalim, F.; Jeong, J.P.; Ilyas, M.U. CRATER: A crowd sensing application to estimate road conditions. *IEEE Access* **2016**, *4*, 8317–8326. [CrossRef]
22. Daraghmi, Y.A.; Daadoo, M. Intelligent Smartphone based system for detecting speed bumps and reducing car speed. In Proceedings of the MATEC Web of Conferences, Amsterdam, The Netherlands, 23–25 March 2016; Volume 77, p. 09006.
23. Park, Y.; Jung, J. Non-Compression Auto-Encoder for Detecting Road Surface Abnormality via Vehicle Driving Noise. In Proceedings of the 2021 IEEE 3rd International Conference on Architecture, Construction, Environment and Hydraulics (ICACEH), Miaoli, Taiwan, 24–26 December 2021; pp. 70–72.
24. Celaya-Padilla, J.M.; Galván-Tejada, C.E.; López-Monteagudo, F.E.; Alonso-González, O.; Moreno-Báez, A.; Martínez-Torteya, A.; Galván-Tejada, J.I.; Arceo-Olague, J.G.; Luna-García, H.; Gamboa-Rosales, H. Speed bump detection using accelerometric features: A genetic algorithm approach. *Sensors* **2018**, *18*, 443. [CrossRef] [PubMed]
25. Al-Shargabi, B.; Hassan, M.; Al-Rousan, T. A novel approach for the detection of road speed bumps using accelerometer sensor. *TEM J.* **2020**, *9*, 469. [CrossRef]
26. Bello-Salau, H.; Aibinu, A.; Onumanyi, A.; Onwuka, E.; Dukiya, J.; Ohize, H. New road anomaly detection and characterization algorithm for autonomous vehicles. *Appl. Comput. Inform.* **2018**, *16*, 223–239. [CrossRef]
27. Baldini, G.; Giuliani, R.; Geib, F. On the application of time frequency convolutional neural networks to road anomalies' identification with accelerometers and gyroscopes. *Sensors* **2020**, *20*, 6425. [CrossRef] [PubMed]
28. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
29. Ye, X.Y.; Hong, D.S.; Chen, H.H.; Hsiao, P.Y.; Fu, L.C. A two-stage real-time YOLOv2-based road marking detector with lightweight spatial transformation-invariant classification. *Image Vis. Comput.* **2020**, *102*, 103978. [CrossRef]
30. Liu, S.; Han, Y.; Xu, L. Recognition of road cracks based on multi-scale Retinex fused with wavelet transform. *Array* **2022**, *15*, 100193. [CrossRef]
31. Bhatia, Y.; Rai, R.; Gupta, V.; Aggarwal, N.; Akula, A. Convolutional neural networks based potholes detection using thermal imaging. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 578–588.
32. Pandey, A.K.; Iqbal, R.; Maniak, T.; Karyotis, C.; Akuma, S.; Palade, V. Convolution neural networks for pothole detection of critical road infrastructure. *Comput. Electr. Eng.* **2022**, *99*, 107725. [CrossRef]
33. Varona, B.; Monteserin, A.; Teyseyre, A. A deep learning approach to automatic road surface monitoring and pothole detection. *Pers. Ubiquitous Comput.* **2020**, *24*, 519–534. [CrossRef]
34. Zeng, F.; Wang, C.; Ge, S.S. A survey on visual navigation for artificial agents with deep reinforcement learning. *IEEE Access* **2020**, *8*, 135426–135442. [CrossRef]
35. Cheng, J.; Wang, C.; Meng, M.Q.H. Robust visual localization in dynamic environments based on sparse motion removal. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 658–669. [CrossRef]
36. Bala, J.A.; Adeshina, S.A.; Aibinu, A.M. Advances in visual simultaneous localisation and mapping techniques for autonomous vehicles: A review. *Sensors* **2022**, *22*, 8943. [CrossRef] [PubMed]
37. Cheng, J.; Zhang, H.; Meng, M.Q.H. Improving visual localization accuracy in dynamic environments based on dynamic region removal. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1585–1596. [CrossRef]
38. Yang, D.; Bi, S.; Wang, W.; Yuan, C.; Qi, X.; Cai, Y. DRE-SLAM: Dynamic RGB-D encoder SLAM for a differential-drive robot. *Remote Sens.* **2019**, *11*, 380. [CrossRef]
39. Li, D.; Yang, W.; Shi, X.; Guo, D.; Long, Q.; Qiao, F.; Wei, Q. A visual-inertial localization method for unmanned aerial vehicle in underground tunnel dynamic environments. *IEEE Access* **2020**, *8*, 76809–76822. [CrossRef]

40. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
41. Fan, Y.C.; Yelamandala, C.M.; Chen, T.W.; Huang, C.J. Real-Time Object Detection for LiDAR Based on LS-R-YOLOv4 Neural Network. *J. Sens.* **2021**, *2021*, 5576262. [CrossRef]
42. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
43. Ahmed, K.R. Smart pothole detection using deep learning based on dilated convolution. *Sensors* **2021**, *21*, 8406. [CrossRef] [PubMed]
44. Lyu, P.; Yao, C.; Wu, W.; Yan, S.; Bai, X. Multi-oriented scene text detection via corner localization and region segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7553–7563.
45. Zhao, Z.Q.; Zheng, P.; Xu, S.t.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
46. Tzutalin, D. LabelImg. 2015. Available online: https://github.com/HumanSignal/labelImg (accessed on 11 September 2023).
47. Asad, M.H.; Khaliq, S.; Yousaf, M.H.; Ullah, M.O.; Ahmad, A. Pothole Detection Using Deep Learning: A Real-Time and AI-on-the-Edge Perspective. *Adv. Civ. Eng.* **2022**, *2022*, 9221211. [CrossRef]
48. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
49. Shaghouri, A.A.; Alkhatib, R.; Berjaoui, S. Real-time pothole detection using deep learning. *arXiv* **2021**, arXiv:2107.06356.
50. Baek, J.W.; Chung, K. Pothole classification model using edge detection in road image. *Appl. Sci.* **2020**, *10*, 6662. [CrossRef]
51. Gajjar, K.; van Niekerk, T.; Wilm, T.; Mercorelli, P. Vision-Based Deep Learning Algorithm for Detecting Potholes. *J. Phys. Conf. Ser.* **2022**, *2162*, 012019. [CrossRef]
52. Hassan, S.I.; Sullivan, D.O.; Mckeever, S. Pothole Detection under Diverse Conditions using Object Detection Models. In Proceedings of the International Conference on Image Processing and Vision Engineering (IMPROVE 2021), Online, 28–30 April 2021; pp. 128–136. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.