applied sciences

MDPI

Article

Productivity Assessment of the Yolo V5 Model in Detecting Road Surface Damages

Son Vu Hong Pham and Khoi Van Tien Nguyen *

Construction Engineering & Management Department, Ho Chi Minh City University of Technology (HCMUT), Vietnam National University—HCMC, Ho Chi Minh City 700000, Vietnam; pvhson@hcmut.edu.vn
* Correspondence: 2398004@hcmut.edu.vn

Abstract: Artificial intelligence models are currently being proposed for application in improving performance in addressing contemporary management and production issues. With the goal of automating the detection of road surface defects in transportation infrastructure management to make it more convenient, this research harnesses the advancements of the latest artificial intelligence models. Notably, new technology is used in this study to develop software that can automatically detect road surface damage, which shall lead to better results compared to previous models. This study evaluates and compares machine learning models using the same dataset for model training and performance assessment consisting of 9053 images from previous research. Furthermore, to demonstrate practicality and superior performance over previous image recognition models, mAP (mean average precision) and processing speed, which are recognized as a measure of effectiveness, are employed to assess the performance of the machine learning object recognition software models. The results of this research reveal the potential of the new technology, YOLO V5 (2023), as a high-performance model for object detection in technical transportation infrastructure images. Another significant outcome of the research is the development of an improved software named RTI-IMS, which can apply automation features and accurately detect road surface damages, thereby aiding more effective management and monitoring of sustainable road infrastructure.

Keywords: road maintenance; road damage detection; Yolo V5; intelligent management system; construction management

check for updates

Citation: Pham, S.V.H.; Nguyen, K.V.T. Productivity Assessment of the Yolo V5 Model in Detecting Road Surface Damages. *Appl. Sci.* **2023**, *13*, 12445. https://doi.org/10.3390/app132212445

Academic Editor: José António Correia

Received: 21 September 2023
Revised: 6 November 2023
Accepted: 9 November 2023
Published: 17 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Transportation infrastructure, particularly highway networks, has been rapidly developed to meet transportation demands and connect economic centers. Koch et al. [1] reported that while construction is a necessary condition, the sufficient conditions for any road to safely serve traffic participants include management, operation, and maintenance. Hence, researching and developing a precise automated inspection tool to support road maintenance agencies in effectively and swiftly addressing road deterioration/damage should be a top priority. For each road segment, Cao et al. [2] emphasized that after construction and some years of use, various factors contribute to road damage, such as overloading, weather conditions, or low-quality materials. Ouma and Hahn [3] elaborated on the impact of road damage, leading to regrettable traffic accidents for traffic participants, reduced road operational efficiency, and ultimately increased road repair costs due to severe and unpreventable degradation. Gavilán et al. [4] reported that timely and proper road maintenance can reduce costs by more than 20% compared to maintenance carried out after severe road damage occurs. Maintaining roads in optimal conditions is a significant challenge faced by road management agencies, and therefore, the development and application of modern technology platforms to support road maintenance agencies in efficiently and swiftly addressing road damage is a top research priority. In the context of road management, He et al. [5] emphasized the significance of road defect detection for

road maintenance and management. Cao, Tran, Nguyen and Chang [2] stated that, to meet the demand for tools to assist in assessing road damage detection, multiple efforts have been made to evaluate the condition of asphalt roads. Various research studies have focused on road quality management models with different levels of comprehensiveness, such as manual assessments, the use of UAVs for damage detection, or laser scanning-based methods. Koch, Jog and Brilakis [1] also noted that manual road assessment is highly inefficient and susceptible to subjectivity and the biases of the technicians responsible for these inspections. Additionally, Outay et al. [6] presented a technique related to the application of UAVs in transportation, specifically for traffic monitoring and highway infrastructure management. Although their research proves effective in identifying road damage, the results regarding damage localization are often neglected, and the technical demands of UAV technology are challenging for most countries to meet. Previous methods for road defect detection have relied on vibration sensors, accelerometers, or gyroscopes, as discussed by Gunawan [7], Harikrishnan and Gopi [8]. These methods have contributed to automating the detection of defects on road surfaces, but their installation has been cumbersome and prone to significant interference due to the vehicles' movement speed, leading to inefficiencies. Furthermore, laser scanning-based methods have been proposed for road surface damage detection by Zhu et al. [9]. Ground-based laser scanning methods aimed to expedite road condition assessments and make them safer and more systematic. However, these methods exhibited unstable results and low accuracy, and the continuous operation of laser scanning over several hours often resulted in decreased accuracy. Hence, the reliability of the task needs improvement, and the operational duration must be extended. Given these limitations, road management agencies and researchers have increasingly turned to automatic machine learning-based road damage/defect identification and classification techniques to address road management challenges effectively.

In recent years, improvements in image processing techniques (IPT) have seen significant advancements in automated road surface crack detection models [10]. These models involve precise and rapid analysis of two-dimensional (2D) road surface images to extract and assess relevant features for detecting the presence of cracks and tagging appropriate images for further analysis. Pioneering models in this area were proposed by [11,12]. However, it was noted that the results of these machine learning models were negatively impacted by incorrect feature extraction, often due to manual extraction. According to Chow et al. [13], recent models based on deep learning have been identified as the most effective method for detecting objects like cracks in concrete structures and road damage. When considering road damage detection models based on machine learning techniques, several other notable studies have been proposed. These include road damage detection using Fully Convolutional Neural Networks (FCNN) or methods for detecting potholes based on videos recorded with smartphones and integrated diagnostics, reported Singh and Singh [14]. However, these methods have seen limited practical application due to their theoretical limitations and lack of practicality. Shim et al. [15] proposed a new neural network structure as well as training and prediction methods. Maeda et al. [16] and Arya et al. [17] conducted research and proposed a new deep neural network algorithm to detect road damage conditions, aiming to establish a safe road environment. Their study focused on images containing various types of road damage. Indeed, despite the improvements, crack detection based on image processing can still be prone to errors due to various factors, including poor road surface lighting and non-uniform pavement material structures. Sinharay et al. [18] report highlights that although these proposed algorithms have been compared to deep learning models, it seems that this tool has not been widely adopted by traffic management agencies due to its theoretical nature and lack of practicality.

As reported by Fang et al. [19], research directions in improving the management, operation, and maintenance of transportation infrastructure are currently focusing on enhancing productivity, quality, efficiency, and cost-effectiveness. Applying the scientific achievements of the Fourth Industrial Revolution to the management, operation, and

maintenance of transportation infrastructure is an open and appropriate approach aligned with the current infrastructure and future trends, as reported by Harris et al. [20]. In 2020, Cao, Tran, Nguyen and Chang [2] reported an important study on the performance of deep learning models for road damage detection using various dashcam image resources. This research is significant not only in advancing machine learning-based detection models but also in evaluating the performance of each model to compare their effectiveness. However, Arya, Maeda, Ghosh, Toshniwal and Sekimoto [17] also reported that software developers have been working on developing and updating upgraded versions of previous software. These subsequent versions not only enhance performance but also introduce many new features. Cao, Tran, Nguyen and Chang [2] noted that for technology developers, research on product upgrades and improvements is a survival race, for example, with technologies like CNN, R-CNN, and Fast R-CNN.

In recent years, Yolo V5 has emerged as an efficient and practical software for infrastructure object detection [21]. Therefore, this research applied the Yolo V5 model to develop a comprehensive software called RTI-IMS, which can automatically and accurately detect road surface damage. This study incorporates numerous previous achievements of scientists worldwide while eliminating relative limitations to create a more refined software. The RTI IMS software was trained using a dataset of 9053 images published Maeda, Sekimoto, Seto, Kashiyama and Omata [16]. The research also developed a tripled variation using the Mosaic method to enhance object image recognition capabilities. Furthermore, to demonstrate the superiority of the software developed in this study, it evaluated the results of RTI-IMS compared to technical infrastructure object detection platforms in previous research conducted by [2]. Among the various evaluation metrics for machine learning image recognition models, two of the most crucial assessment criteria are accuracy and processing speed. These criteria are proposed to compare RTI-IMS with previous models for road damage detection. This research provides results as evidence of the application of new technology in current traffic network management practices.

Furthermore, the main function of this software, which is road damage detection, has been thoroughly developed in the research to provide accurate data to traffic managers.

The structure of this paper is organized as follows: Part 1 is the introduction; Part 2 provides a literature review of road damage detection; Part 3 is used to present the Yolo V5 network architecture and introduce other detection models used in previous studies; Part 4 compares the operation process of Yolo V5 with previously studied detection models; Part 5 is dedicated to evaluating the performance of object detection models; and Part 6 is for the conclusion and remarks.

2. Literature Review of Road Damage Detection

Among the studies published on road damage detection and classification using machine learning and deep learning methods, Cao, Tran, Nguyen and Chang [2] reported that they could be categorized based on the type of road images used. Many researchers have successfully applied deep learning models to classify damage in manually captured road surface images with relatively accurate results. For instance, Zhang et al. [22] established a convolutional neural network model using 500 images to determine whether there are cracks in road images. Zhang et al. [23] and Malygin et al. [24] built a CrackNet architecture without a pooling layer to classify whether there are cracks on asphalt concrete road surfaces with an accuracy of 90.1%. Meanwhile, the model by Cumbajin et al. [25] included 1200 collected road surface images, and a dataset containing corresponding labels of defects was established Hatır et al. [26]. Based on this dataset, the performance of Mask R-CNN was compared with faster region-based convolutional neural networks (Faster R-CNN). Meanwhile, the testing time for models using Feature Pyramid Networks (FPN) was lower than other models, achieving 0.21 s per frame (SPF). However, most of these models had relatively small training datasets, and further improvements are needed to distinguish a variety of road surface crack types.

The second group focusing on road damage classification and detection primarily aims to establish models for image-based detection using cameras mounted on vehicles, such as dashcams and smartphones Kyriakou et al. [27]. Jo and Ryu [28] developed a novel pothole detection algorithm that operates within an embedded computing environment. It processes video feeds from dash cameras, enabling real-time and accurate detection of potholes. Chun and Ryu [29] applied Fully Convolutional Networks (FCN) with semi-supervised learning to detect road damage on images captured by dashboard cameras. They used image segmentation and brightness adjustment to enhance the results. Kalfarisi et al. [30] integrated Faster R-CNN with Structured Random Feature Enhancement Detection (SRFED) to detect cracks on structural surfaces. In a recent study, Maeda, Sekimoto, Seto, Kashiyama and Omata [16] proposed the use of SSD Inception V2 and SSD MobileNet to detect and classify road damage into eight categories, achieving an accuracy rate of 77% and a recall rate of 71%. While these results may not meet modern dataset standards, they have led to a decent average precision (AP) score. However, the datasets used in these studies were collected through dash cameras, making it challenging to quantitatively determine the actual size and area of damages. Thus, future work should address three key aspects: image acquisition perspective, dataset augmentation, and the exploration of more algorithms to further improve road damage detection efficiency.

The Fourth Industrial Revolution has introduced various technological tools that have significantly enhanced multiple fields. Zhang, Guo, Wu, Tian, Tang and Guo [22] reported that one of the prominent machine learning software platforms is Yolo V5 (2023) Zhang, Guo, Wu, Tian, Tang and Guo [22]. Yolo (You Only Look Once) is known for its efficiency in machine learning-based object detection, as stated by [22]. The Yolo V5 software, being the fifth generation of Yolo, offers an effective tool for developing software that can detect objects in images related to technical infrastructure Zhang, Yang, Zhang and Zhu [23]. This advancement in Yolo V5 allows the development of software with superior accuracy and detection speed compared to earlier platforms. Scientific research in the field of information technology always demands novelty and staying updated with trends. One of the important requirements for researchers is to establish the basis for comparing research results, often relying on evaluations from prior studies. To demonstrate the optimization of the software developed in this study, it introduced the best evaluation criteria, focusing on model detection performance/accuracy and model detection speed. Alongside successfully developing a model by incorporating scientific achievements from previous studies, this research also highlighted the model's superior advantages and its effective response to traffic infrastructure management needs.

3. Presenting the Yolo V5 Network Architecture Model and Introducing Other Detection Models

3.1. Overview of Road Surface Damage Image Detection Models

In recent years, improvements in image processing techniques (ITPs) have spurred significant improvements in automatic pavement crack detection models. In these models, precise and rapid analyses of two-dimensional (2D) images of road pavement are conducted to extract and assess useful features for yes/no crack detection and to tag appropriate images for further analyses Zhang et al. [31]. Therefore, this study was designed to comprehensively assess the performance and capability of the recently developed deep learning-based state-of-the-art detection models to detect various types of damage to asphalt roads, including longitudinal cracks, horizontal cracks, potholes, and alligators Jiang et al. [32]. The models' performance is assessed based on results obtained from implementing detection tasks on many image sources with different sizes Cha et al. [33,34].

This part of the study will delineate the primary constituents of object detection software: the model for detecting objects and the foundational structures of deep learning employed in extracting features from images. Initially, this research will showcase two models for object detection: the Faster RCNN, a two-stage model, and the SSD model (Single Shot Multi-Box Detection), a one-stage model. Subsequently, it will introduce four

crucial deep learning structures used to extract features—MobileNet, Inception, ResNet, and Inception ResNet, Google LLC, Mountain View, CA, USA. Ultimately, the investigation will cover eight detection models formed by amalgamating the object detection model with the feature extraction architectures (Figure 1). The aim is to identify objects in images and subsequently contrast the performance of these models.

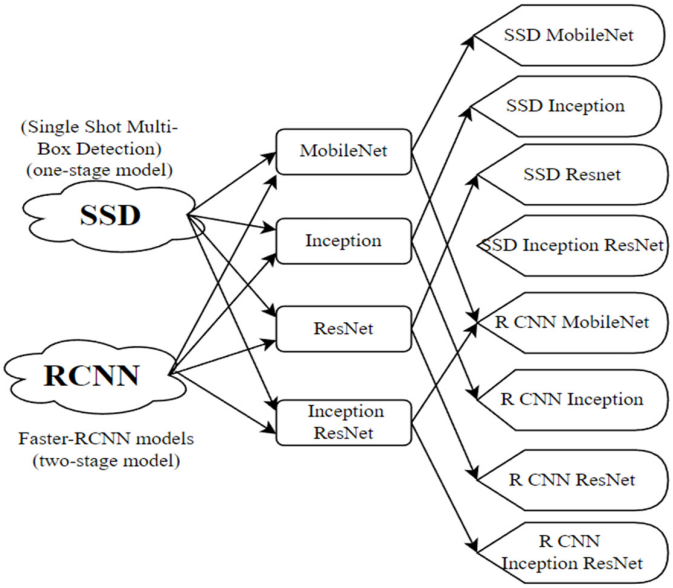


Figure 1. The formation methods of the 8 detection models.

3.1.1. Two Object Detection Models Are Faster RCNN and SSD in the Latest Object Detection in Images Technology Today

This study briefly introduces the working process of the two newest object detection models, including Faster-RCNN models (two-stage model) and SSD models (Single Shot Multi-Box Detection) (one-stage model). In general, Faster RCNN is a two-stage model, in which the first stage is to generate region proposals using the RPN network and the second stage is to classify and improve the position. Next, SSD is a one-stage model that predicts bounding boxes and classifies directly on grid cells on specific feature maps [8,35].

Faster RCNN (two-stage model): The workflow of Faster R-CNN (Faster Region-based Convolutional Neural Network) includes two main modules: Region Proposal Network (RPN) and R-CNN (Region-based Convolutional Neural Network) (Figure 2).

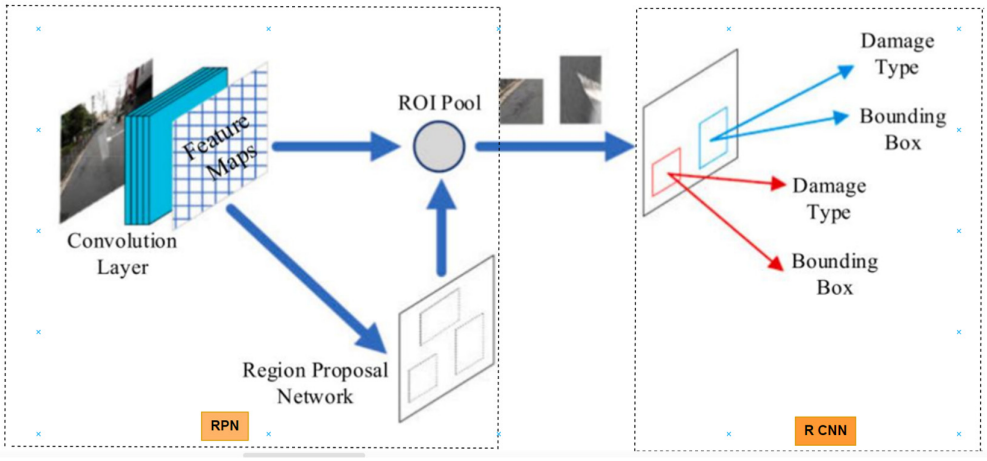


Figure 2. Working process of Faster R-CNN.

Region Proposal Network (RPN): RPN is a convolutional neural network used to recommend candidate regions that are likely to contain objects in an image. RPN predicts candidate bounding boxes and calculates confidence scores for each box. Boxes with high confidence scores are suggested as candidate regions to contain objects. RPN uses loss functions to optimize the correct bounding box prediction and ensure the diversity of candidate regions.

R-CNN (Region-based Convolutional Neural Network): R-CNN is a convolutional neural network used to classify and accurately predict the position of objects in each candidate region proposed by RPNs. R-CNN extracts features from each candidate region and passes them through fully connected layers to classify objects and adjust the position of bounding boxes. R-CNN also uses loss functions to optimize classification and position prediction.

3.1.2. SSD Model (Single Shot Multi-Box Detection) (One-Stage Model)

The Single Stage Detection (SSD) model, initially introduced by Liu [36], stands as a well-regarded swift model with commendable accuracy. The operational procedure of SSD (Single Shot Multi-Box Detection) is depicted in Figure 3, as detailed below.

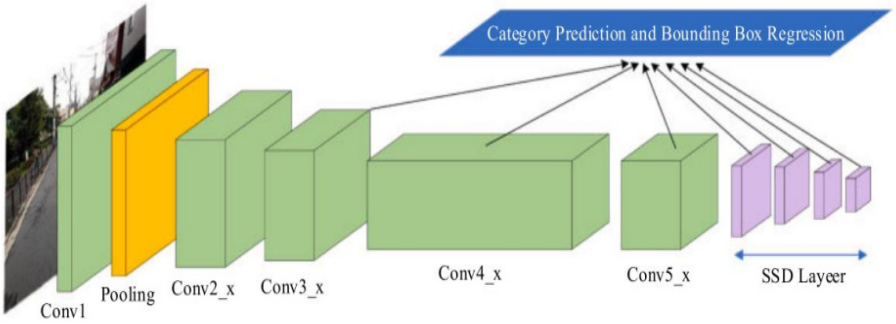


Figure 3. Working process of SSD (Single Shot Multi-Box Detection).

Single Shot Multibox Detector (SSD) is an object detection model in computer vision. The highlight of the SSD is its fast and efficient object detection in near real time.

SSDs use Convolutional Neural Networks (CNNs) to perform both main tasks: object detection and prediction of bounding boxes containing objects. This model performs both of these tasks in a single shot, making the detection process fast and efficient. SSD uses convolution and detection layers to predict the positions and layers of objects in the image. It generates a set of bounding boxes and predicts the probability that belongs to each layer for each of these bounding boxes. Then, bounding boxes with a reliable probability are namely selected as the region proposals containing the object.

3.2. Basic Architectures of Deep Learning

The basic architectures of deep learning include MobileNet, Inception, ResNet, and Inception ResNet; all four of these architectures have proven to be effective in many tasks in deep learning, including object detection, image classification, and image segmentation.

3.2.1. MobileNet

MobileNet is a lightweight and optimized neural network architecture for resource-constrained devices, such as mobile phones. It uses the Depthwise Separable Convolution technique to reduce the number of parameters and computations. This architecture enables MobileNet to achieve high performance in feature extraction with fast speed and low consumption of computing resources. Within MobileNet, the notion of depthwise separable convolution is presented, breaking down the standard convolution into depthwise convolution and 1×1 convolution. This approach is employed to construct streamlined deep neural networks. This technique aids in diminishing computational demands and parameter counts. Unlike traditional convolution, which merges input channels into a

solitary pixel, depth-per-form convolution operates individually. It allocates distinct weight sets to each image unit. Channels can be processed with color filtering as well as edge and other feature detectors (as depicted in Figure 4).

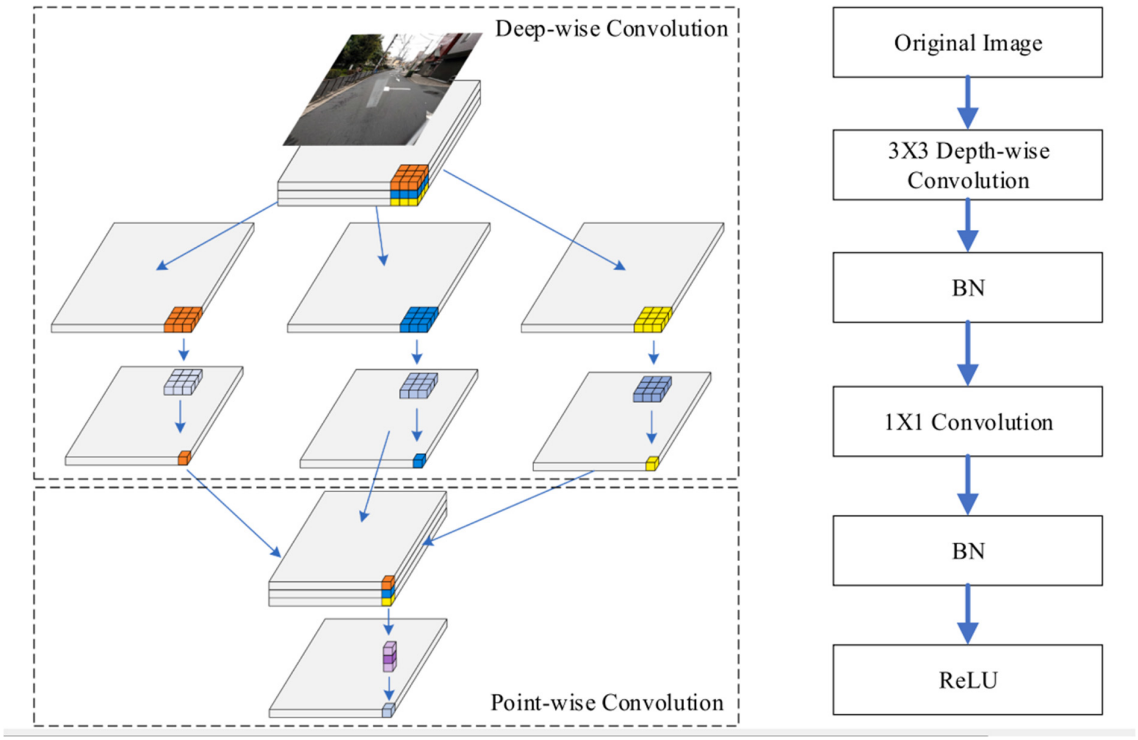


Figure 4. Architectural process of MobileNet.

3.2.2. Inception

The Inception architecture (also known as GoogLeNet, Google LLC, Mountain View, CA, USA) focuses on building Inception modules, in which convolution steps with different dimensions are performed in parallel and combine information from different dimensions of the feature map. This allows the model to extract features from spatial information at different scales and create a very deep network with relatively low complexity. The Inception series of feature extraction tools emphasizes scalability. One of the notable strengths of this approach is the integration of normalization within the model's architecture, with normalization being applied to each training sub-batch. Each time batch normalization is activated, only two extra parameters are introduced, preserving the network's representational capacity. Batch normalization allows for the effective use of much higher learning rates, simultaneously streamlining the initialization process. Through the integration of increased learning rates, the omission of Dropout, and adjustments rooted in batch normalization, Inception-V2 attains a heightened level of advancement compared to its previous version, all while requiring only a fraction of the training iterations previously deemed essential. Notably, Inception-V2 significantly outperformed the most renowned system on the ImageNet dataset by a considerable margin during the ILSVRC 2014 competition.

3.2.3. ResNet

ResNet is a neural network architecture (Figure 5) capable of training extremely deep models without the problem of depth loss. It introduces the concept of residual blocks, where "skip connections" allow information to go directly from previous layers to later layers. This helps to avoid information loss and makes it easier for the model to converge during training. The Residual Network (ResNet) architecture introduced by He, Zhang, Ren and Sun [21] is a very popular deep neural network model in the field of computer vision. ResNet networks are mainly used to solve the "vanishing gradient" problem during deep

network training. The ResNet architecture uses Residual Blocks with branch connections to avoid vanishing gradients and to allow the construction of deep neural networks without problems related to network depth. This has contributed to ResNet's growth and success in many applications in the field of computer vision.

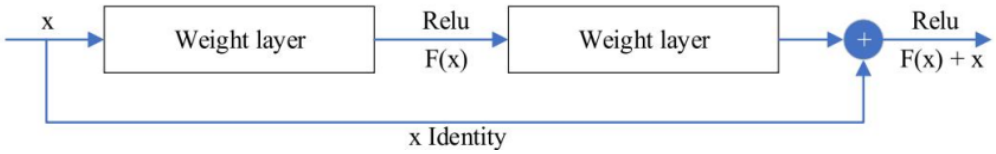


Figure 5. Architectural process of Residual Network (ResNet).

3.2.4. Inception ResNet

It is a combination of two network architectures, Inception and ResNet (Figure 6). It combines the advantages of both models to create a powerful and efficient deep neural network. The Inception-ResNet architecture achieves higher accuracy compared to its predecessors primarily due to its utilization of a validation accuracy assessment based on an image cropping classification criterion. The working process of Inception ResNet is similar to that of normal ResNet. It consists of stacking Inception Residual Blocks and convolution layers combined with pooling layers and activation layers. The weights in the network are updated through training and using optimization algorithms such as Gradient Descent to minimize the loss function.

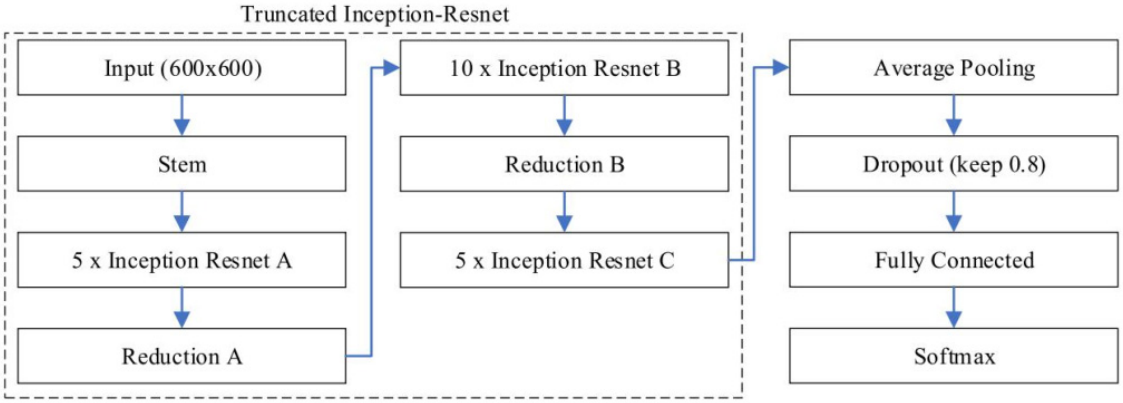


Figure 6. Architectural process of Inception ResNet.

3.3. Yolo Introduction to Yolo Network Architecture Model

3.3.1. Introduction to Yolo (Yolo V5)

Yolo (You Only Look Once) is a famous neural network architecture in the field of fast and accurate object detection in images and videos. Yolo V5 is the fifth (05) generation version of Yolo, developed by Ultralytics. Yolo V5 has a number of outstanding features such as:

Single shot detection: Yolo V5 is a one-stage object detection model, capable of detecting and classifying objects in a single run over the entire image without the need to create region pre-proposals like two-stage models.

Backbone architecture: Yolo V5 uses a convolutional neural network (CNN) architecture as the "backbone" for feature extraction. The Yolo V5 instance uses EfficientNet architecture as the backbone, making the model capable of learning complex features and achieving good performance.

Scale variants: Yolo V5 is designed with different versions (s, m, l, x), corresponding to small, medium, large, and extra-large model sizes. The availability of multiple versions allows the user to customize the model size to meet the requirements of speed and accuracy.

Multi-scale inference: Yolo V5 supports applying different scaling levels to the input image during the prediction process. This makes the model capable of detecting objects at different size ratios and achieving better performance on small or large objects.

Platform support: Yolo V5 supports multiple platforms, including PyTorch and ONNX, making it easy for users to deploy and integrate into their projects.

Yolo V5 has achieved high performance and fast speed in object detection in images and videos, and it has been widely used in practical applications such as autonomous vehicles, security monitoring, and detection of technical infrastructure objects in construction.

3.3.2. Yolo V5's Structure

Yolo's structure consists of four main parts: input, backbone, neck, and head. Below is a detailed description of each part.

Input: This is the first part of the Yolo model and is responsible for input processing. Normally, the input image will be divided into (grid) blocks and converted into tensors to be fed into the neural network.

Backbone: This is the most important part of the model; it performs feature extraction from the image through a neural network architecture. In Yolo V5, EfficientNet architecture is often used as the backbone to learn complex features from images.

Neck: Neck is the middle part of the model and is responsible for combining features from previous layers to create common features, which improves the ability to detect and localize objects. In Yolo, the neck is usually not used, and features are passed directly from the backbone to the next part.

Head: This is the final part of the model and performs object prediction on the image. Head includes two main components, Dense Prediction and Sparse Prediction; specifically,

Dense Prediction: Dense Prediction predicts detailed information about the position, size, and class of objects in each grid cell of an image. Through several convolutional layers and up-sampling techniques, it produces highly detailed and dense predictions.

Sparse Prediction: Sparse Prediction uses convolutional layers with larger kernels to generate sparse predictions for larger regions in the image. These predictions help in detecting larger objects that may span multiple grid cells.

Combining both Dense Prediction and Sparse Prediction, Yolo V5 is capable of detecting and classifying objects on images quickly and accurately.

4. The Working Process of Yolo V5 in RTI IMS Software Development to Automatically Detect Road Surface Damage

4.1. Application of Yolo V5 Model to Develop RTI IMS Software to Automatically Detect Road Surface Damage

In this study, the Yolo V5 platform model was used to develop RTI-IMS software that can automatically detect road surface damage. To accomplish the above purpose, the training data for the RTI IMS software use a dataset of 9053 images provided Maeda, Sekimoto, Seto, Kashiyama and Omata [16]. The dataset used to train the machine learning data model is the road surface damage images in the studies in Taiwan, Japan, and India mentioned above. In addition, in order to make the RTI-IMS software improve inference without affecting inference speed, this study uses the Mosaic method in the Yolo V5 platform to help the training model not be overfit with input data in order to reduce many false predictions and make the model quality better on test data. This Mosaic method converts the input data to triple the variation, resulting in a more diverse and intelligent dataset.

4.2. Training and Detection Process of Yolo V5 in RTI IMS Software

4.2.1. YOLO V5's Object Training Algorithm

The object training process of YOLO V5 requires a large enough dataset and computational resources to train a model with good performance. YOLO V5's object training algorithm includes the following steps:

Prepare data: First, it is necessary to prepare the training data. The training data must contain the input image and the bounding boxes containing the object to be detected. These bounding boxes are contained in the label files corresponding to each image.

Build configuration file: Next, it is necessary to create a configuration file to configure the training parameters. The configuration file defines the model architecture and training parameters, such as learning rate, epochs, batch size, and other parameters.

Conduct training: Based on the data and configuration file, we proceed to train the YOLO V5 model. The training process consists of passing the corresponding images and bounding boxes through the neural network, calculating the model's predictions, comparing them with the actual labels, and calculating the loss. This process will be repeated over many epochs to improve the model performance.

Evaluate and adjust: After training, the model needs to be evaluated to measure performance. This is usually done by measuring metrics such as accuracy, recall, and F1. Based on the evaluation results, we can adjust the training parameters to improve the model's performance.

Test and deploy: Finally, the model is tested on independent test data to ensure generality and accuracy. The model can then be deployed and used in real applications.

4.2.2. YOLO V5's Object Training Process in RTI IMS Software

Similar to the process of training machine learning to automatically detect objects in other platform-based models, RTI-IMS software uses a dataset of 9053 images provided by Maeda, Sekimoto, Seto, Kashiyama and Omata [16]. The RTI IMS software in this study was also built based on the sequence diagram of the steps and processes from receiving input data to the process of machine learning and model completion to detect objects that are road surface damages on the test set. The training data diagram of the RTI IMS software is shown in Figure 7.

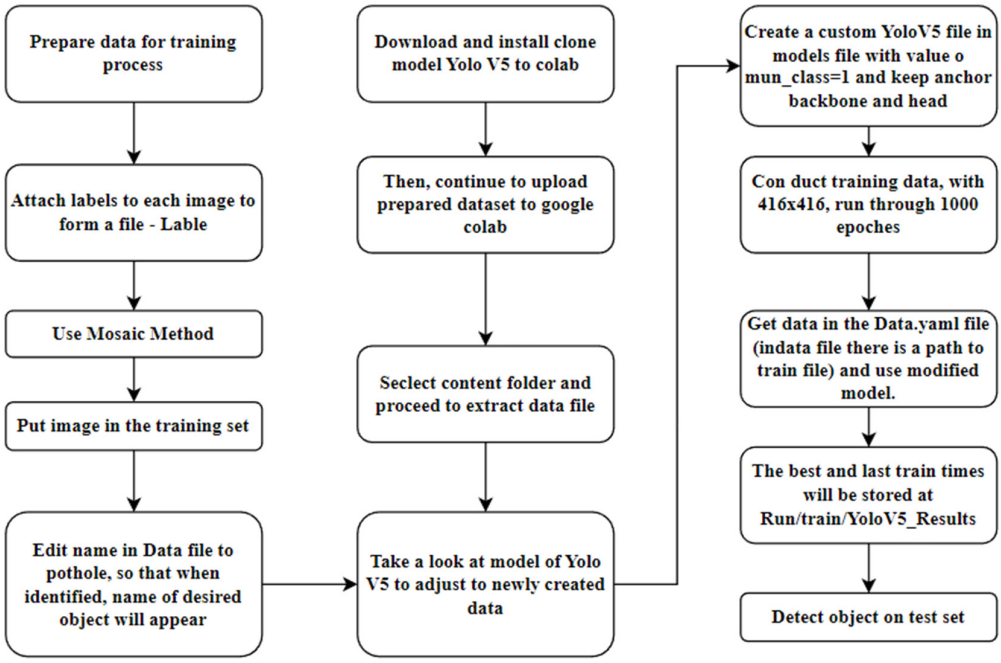


Figure 7. Training data diagram of RTI IMS software.

5. Performance Evaluation of Object Detection Models

5.1. Performance Evaluation Criteria of Detection Models

There exist various fundamental criteria for assessing the effectiveness of object detection models within the foundational frameworks of deep learning. This study evaluates two important criteria including:

Assessment of model efficacy based on mean average precision (mAP) and average recall (AR) values: Model accuracy measures the ability to correctly classify and detect objects. This is an important criterion for evaluating the performance of road surface damage image detection models.

In order to calculate AP (average precision), we need to compute precision and recall.

$$\text{Precision}: \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{1}$$

$$\text{Recall}: \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{2}$$

With:

TP (True Positive): The model correctly identifies it as a pothole, and the image indeed contains a pothole.
FP (False Positive): The model incorrectly identifies it as a pothole, but the actual label is not a pothole (could be a license plate, for example).
TN (True Negative): The model correctly identifies it as not a pothole, and the image truly does not contain a pothole (car, manhole cover, traffic light, etc.).
FN (False Negative): The model incorrectly identifies it as not a pothole, but in reality, it is a pothole.

The calculation of TP, TN, FP, and FN involves a parameter called IoU (Intersection over Union), which is the ratio of the overlapping area of two bounding boxes to the union area of the two bounding boxes. If the IoU threshold is 0.5 and the IoU value for a prediction is 0.7, then we classify the prediction as correct (TP). On the other hand, if the IoU is 0.3, we classify it as a false prediction (FP).

Intersection and Union:

$$IoU = \frac{intersect\ area}{union\ area} \tag{3}$$

The AP value represents the area beneath the precision–recall curve, derived from the detections produced by the model. This threshold defines what the model designates as True Positives (TP) based on its class predictions. AP is determined as the mean value across all categories. The mean average precision (mAP) score is then calculated by averaging the AP values across all classes or various Intersection over Union (IoU) thresholds.

$$\text{mAP} = \frac{1}{n}\sum_{i=1:n}(AP_i) \tag{4}$$

$$\text{AP} = \frac{1}{m}\sum_{r\in\{0,0.1,\ldots,1\}}\rho interp(r) \tag{5}$$

$$\rho interp(r) = \max_{\tilde{r}:\tilde{r}\geq r}\rho\left(\tilde{r}\right)$$
$$\text{Within}: \rho\left(\tilde{\text{r}}\right) \text{ is precision at recall } \tilde{r} \text{ with } \tilde{r} \geq \text{r} \tag{6}$$

"APi" is the average precision value for the *i*th layer. We divide recall from 0–1 into 11 equally spaced intervals [0, 0.1, 0.2,. . ., 1] and take the average precision at these 11 points (m = 11 in this study).

The obtained recall and precision values can be used to plot the precision–recall (PR) curve for each individual class. AP (average precision) corresponds to the area under the PR curve. A larger area under the curve signifies higher precision and recall, which in turn implies that the model has better quality.

Speed: The model's processing speed measures its ability to perform object detection quickly on an image or video. For real-time applications, speed is an important factor (Figure 8).



Figure 8. The model's running speed during testing with a resolution of 640 × 640.

In the context of object detection, evaluation addresses two distinct aspects: (1) recognizing whether an object is present or absent in an image (classification), and (2) locating the position of an object (localization, involving regression).

Moreover, the datasets encompass multiple layers characterized by unequal distribution. Hence, relying solely on accuracy as a metric could lead to skewed inaccuracies. Therefore, in this study, the central yardstick for comparing the efficacy of different models was the mean average precision (mAP). This metric entails calculating the average of average precision (AP) values for an information retrieval system across a set of n feature layers. The mAP provides a way to assess the propensity for misclassification, employing the "feature score" or model score assigned to each bounding box in the context of object detection.

5.2. Data Collection and Classification

The dataset employed for assessing model performance should include all instances of each object detection layer under uniform circumstances. As previously noted, the data used for both training the model and evaluating its performance encompass 9053 images sourced from a prior study by Maeda, Sekimoto, Seto, Kashiyama and Omata [16]. These 600 × 600-pixel images are captured at one-second intervals by a dashboard camera affixed to a vehicle's dashboard while traveling at a speed of 40 km/h.

Given that different kinds of cracks demand distinct repair methods, accurate crack classification becomes crucial for effectively allocating maintenance resources. In this research, cracks were redefined into four categories (as outlined in Table 1). Category I, which consists of 1378 occurrences, encompasses lateral cracks caused by temperature variations, as well as asphalt binder hardening/cracking resulting from underlying cracks, and the deterioration of construction joints. Category II, comprising 6557 instances, encompasses longitudinal fatigue cracks (frequently resulting from overloaded trucks) and poorly executed joints [37]. Category III covers alligator cracks resulting from diverse fatigue factors and unstable asphalt bases. Lastly, Category IV, with a total of 409 instances, is related to potholes, separations, bumps, and ruttings. These are characterized by profound depressions arising from extensive water damage or unaddressed cracks. Remarkably, damage in Categories IV and V can severely jeopardize vehicle safety and rapidly lead to extensive road degradation. Observing Table 1, it is evident that the instances in Category IV were considerably fewer than those in other categories. A sum of 440 images was manually extracted and labeled to augment the dataset with an additional 730 instances of Category IV damage. In total, 9493 road damage images were amalgamated for the training and validation of the eight detection models and Yolo V5 as part of this study [38,39].

5.3. Performance Evaluation Using a Single Data Source

5.3.1. Model Performance Based on Rising mAP and AR Values

The mAP (mean average precision) value is a measurement used to evaluate the performance of object detection models. It is often used to compare and evaluate the accuracy of different models.

In the object detection problem, mAP measures the ability to properly classify and accurately locate objects. The model is evaluated based on calculating the accuracy of the proposed bounding boxes compared to the actual bounding boxes in the test dataset.

Table 1. Explanation of varieties of road damage and occurrences.

Damage Type		Instance Number	Details
Type I	Lateral crack	1378	Wheel mark part Construction joint part
Type II	Longitudinal crack	6557	Equal interval Construction joint part
Type III	Alligator crack	2541	Fatigue causes Unstable asphalt bases
Type IV	Corruption	409 (original) 730 (Added)	Rutting, bump pothole, separation
Type V	Blurring	4550	Crosswalk blur While line blur

The mAP value is calculated by calculating the average precision of a series of precision–recall values for different thresholds. A high mAP value indicates that the model has good object detection and positioning capabilities.

Comparing mAP values between models allows us to identify which model has better performance in object detection in test datasets. Models with higher mAP values are generally considered to have better performance and are preferred for use in practical applications.

In this research, all the models underwent training and validation using the dataset provided by Maeda, Sekimoto, Seto, Kashiyama and Omata [16]. A summary of the validation outcomes for the eight models and the Yolo V5 model is displayed in Tables 2 and 3 and Figures 9–12:

Table 2. Average precision (AP) scores comparison for different models.

Detection Systems	mAP (%)					
	Small	Medium	Large	@.50IoU	@.75IoU	Average
Yolo V5	80.26	87.20	89.50	80.19	38.90	40.72
SSD MobileNet-V1	0	5.32	18.45	35.81	13.2	16.47
SSD MobileNet-V2	0	6.3	21.1	38.7	16.34	18.81
SSD Inception-V2	0	6.6	22.25	40.54	16.57	19.45
SSD Lite-MobiNet-V2	0	6.57	18.93	36.58	14.43	17.1
Faster R-CNN Inception-V2	3.97	12.08	30.17	51.86	24.18	26.45
Faster R-CNN ResNet-50	10.23	10.56	30.25	51.25	23.73	26.08
Faster R-CNN ResNet-101	4.61	11.73	31.06	52.85	24.15	27.35
Faster R-CNN Inception-Resnet-V2	8.07	14.14	31.26	54.75	24.94	27.66

Table 3. Aspect ratio (AR) values for comparative models.

Detection Systems	AR@1	AR@10	mAP (%)			
			Small	Medium	Large	Average
YOLO V5	73.20%	75.61%	75.50%	80.21%	83.87%	74.52%
SSD MobileNet-V1	22.82%	34.90%	0.00%	20.24%	42.30%	37.31%
SSD MobileNet-V2	25.04%	37.81%	0.00%	23.56%	44.17%	40.51%
SSD Inception-V2	26.01%	38.66%	0.00%	23.40%	46.61%	41.44%
SSD Lite-MobiNet-V2	23.29%	35.82%	0.00%	26.25%	41.15%	38.91%
Faster R-CNN Inception-V2	33.61%	49.82%	16.00%	41.10%	58.05%	53.06%
Faster R-CNN ResNet-50	32.88%	47.98%	11.00%	36.62%	57.15%	51.33%
Faster R-CNN ResNet-101	33.78%	48.82%	19.00%	41.15%	56.99%	52.06%
Faster R-CNN Inception-Resnet-V2	34.41%	49.27%	23.67%	38.14%	56.73%	51.45%



Figure 9. mAP 0.5.

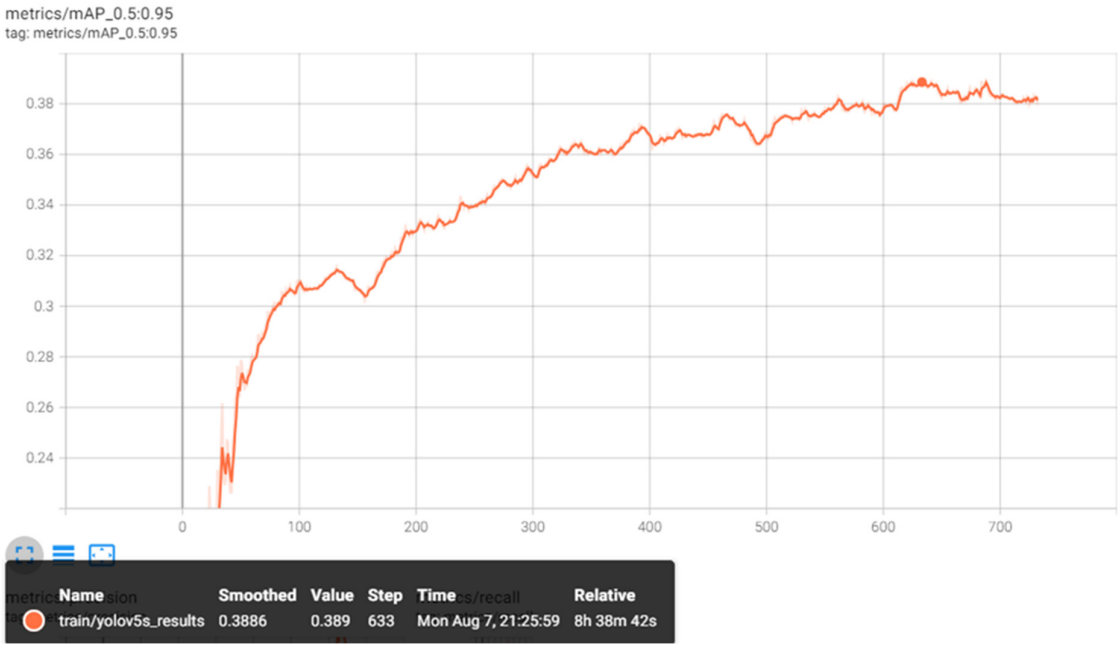


Figure 10. mAP 0.5:0.95.

In relation to the enhancement of AP values illustrated in Table 3, all complete-scale detection models implementing the Faster R-CNN detector outperformed those employing the SSD detector. However, an entirely distinct outcome was observed with Yolo V5.

Among the listed models, Faster R-CNN Inception ResNet V2 displayed the most noteworthy average mAP score at 27.66%, trailed by Faster R-CNN ResNet 50 and Faster R-CNN Inception V2, which achieved mAP values of 27.35% and 26.45%, correspondingly. In contrast, the SSD detection models consistently demonstrated lower mAP scores (20%). Interestingly, the Yolo V5 model, utilizing the SSD detector, yielded an exceptional mAP score of 73.02% and achieved the highest mean at 75.61%, which was quite surprising.

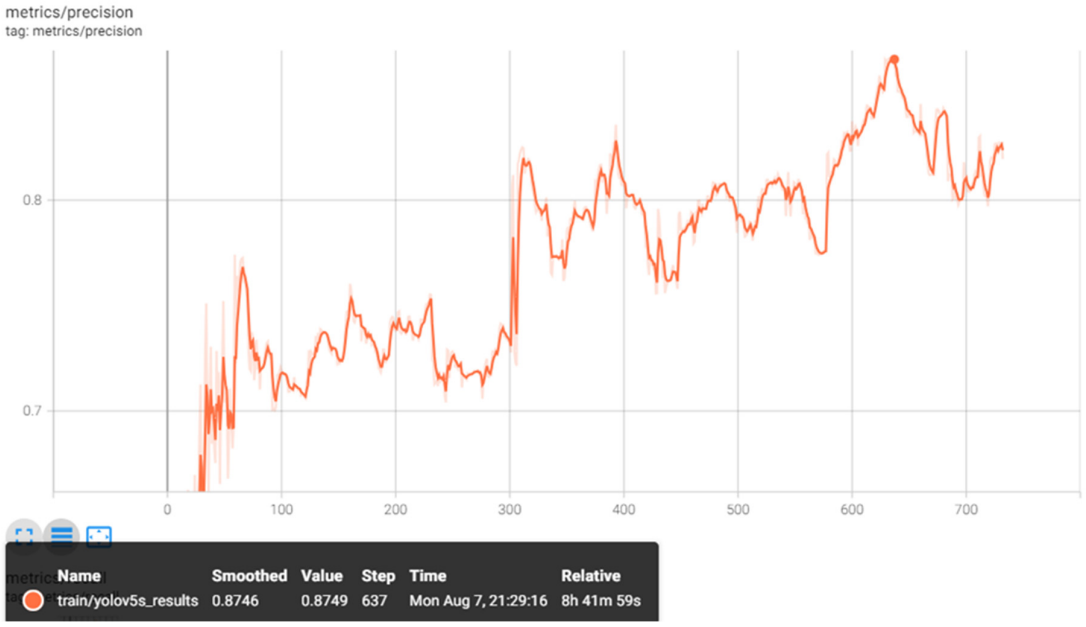


Figure 11. Precision.

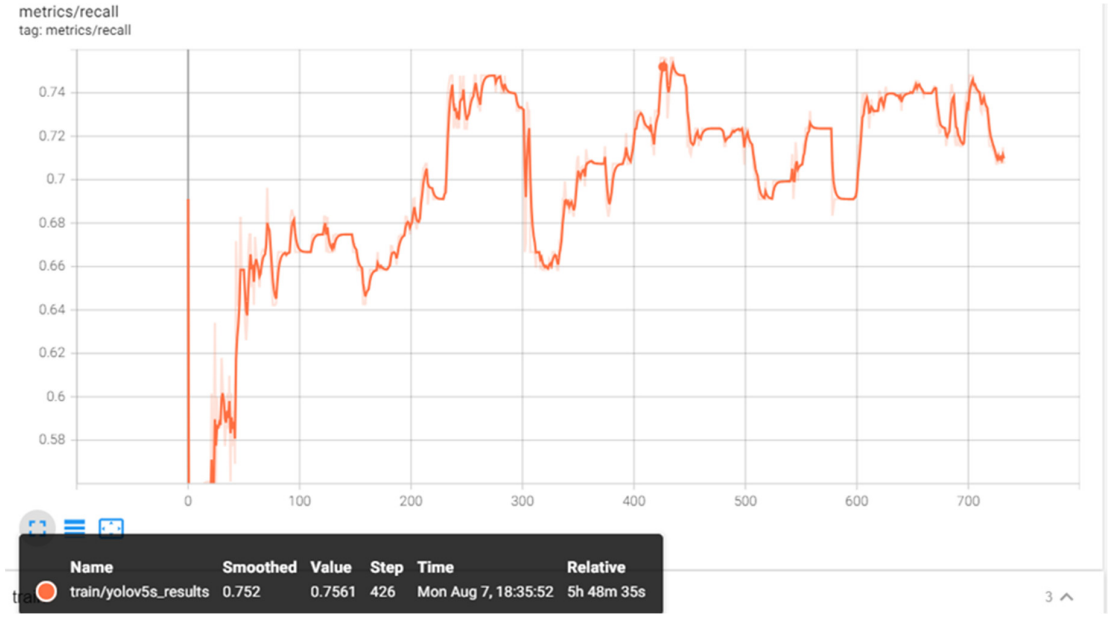


Figure 12. Recall.

Table 4 provides a comprehensive overview of the performance results for all models concerning each damage type, evaluated using an Intersection over Union (IoU) threshold of 0.5. The outcomes reveal that Faster R-CNN Inception ResNet V2 stands out as the most proficient model in identifying three out of the five damage types (I, II, and III), showcasing AP values of 71.64%, 45.28%, and 70.77%, respectively. Faster R-CNN Inception V2 excels in detecting type V (blurring), achieving an AP value of 76.59%. Additionally, it matches the performance of Faster R-CNN Inception ResNet V2 in detecting type I, with an AP of 71.29%. Conversely, Faster R-CNN ResNet 50 achieves the highest AP value (12.23%) for identifying type IV damage. In contrast, the Yolo V5 model displays exceptional performance across all types of road surface damage, a feat rarely accomplished by other models. Notably, Yolo V5 SSD exhibits relatively lower performance, especially in the detection of type IV, with a value of 70.65%.

Table 4. Detailed performance results of the models for each damage type (IoU = 0.5).

Damage Type	Faster-RCNN				SSD				Yolo V5
	Resnet 50	Resnet 101	Inception V2	Inception Resnet V2	Resnet 50	Resnet 101	Inception V2	Inception Resnet V2	
Type I	68.2	68.5	71.3	71.6	49.8	49.8	50.8	53.5	79.76
Type II	34.6	41.4	37.6	45.3	11.0	14.3	17.2	9.5	75.23
Type III	67.4	69.0	62.8	70.8	50.4	55.3	55.0	47.4	75.85
Type IV	12.2	10.8	11.1	11.6	5.5	5.4	12.0	5.7	70.65
Type V	73.8	74.6	76.6	74.5	62.4	68.7	67.7	66.7	77.04

5.3.2. Model Processing Speed

In the field of computer science, the computational complexity of an algorithm signifies the resources it demands for its execution, notably encompassing time and memory requirements. As demonstrated in Table 5, the most intricate model in terms of computational complexity is Faster R-CNN ResNet 101, which involves the deep extraction of distinct features for identifying road damage using 101 convolutional layers. This model boasts around 62.4 million parameters. Subsequently, Faster R-CNN Inception ResNet V2, which combines Inception and ResNet for road damage feature extraction, comprises 59.4 million parameters.

Table 5. Processing time results and parameters of detection system.

Model	Detection Model	Parameters	Test		
			600 × 600	1920 × 1080	3068 × 2760
Model 1	SSD MobileNet-V1	5.60 M	2.10 s	3.70 s	10.50 s
Model 2	SSD MobileNet-V2	4.70 M	2.0 s	3.80 s	10.30 s
Model 3	SSD Inception V2	3.20 M	3.40 s	5.40 s	11.60 s
Model 4	Faster R-CNN Inception-Resnet V2	59.4 M	17.8 s	21.5 s	24.3 s
Model 5	SSD Lite-MobileNet V2	13.70 M	1.80 s	3.90 s	9.80 s
Model 6	Faster R-CNN Inception V2	43.10 M	4.80 s	6.30 s	11.60 s
Model 7	Faster R-CNN Resnet50	43.30 M	7.10 s	9.80 s	14.60 s
Model 8	Faster R-CNN Resnet101	62.40 M	10.80 s	13.90 s	18.00 s
Model 9	Yolo V5	7.2 M	0.5 s	1.2 s	3.2

In contrast, models that utilize MobileNet as the foundational architecture exhibit considerably fewer parameters. Among these, the SSD Lite Mobile V2 model stands out as the least complex, with a mere 3.2 million parameters. The Yolo V5 SSD is an architecture with low parameters at 7.3 M, and the execution time excels in all tests including 0.12 s, 1.2 s, and 3.2 s. Fundamentally, the operational speed of a detection model hinges on an incorporated backbone architecture that employs numerous convolutional layers to identify distinctive attributes of target objects. It is important to highlight that dashcams are commonly configured to record videos and images in high resolution. Consequently, the computation duration for road damage detection should primarily be assessed with respect to 1920 × 1080 (full HD) resolution.

Considering that both precision in road damage detection and processing speed are pivotal for effective road maintenance, the Yolo V5 model emerges as a recommended choice for this purpose.

5.3.3. Results Obtained on Actual Testing by Webcam Detecting RTI IMS

During the process of training input image data for the RTI IMS software, the image data were carefully selected and categorized to detect various types of road damage. Figure 13 illustrates the results of identifying different types of road damage by the RTI IMS software.

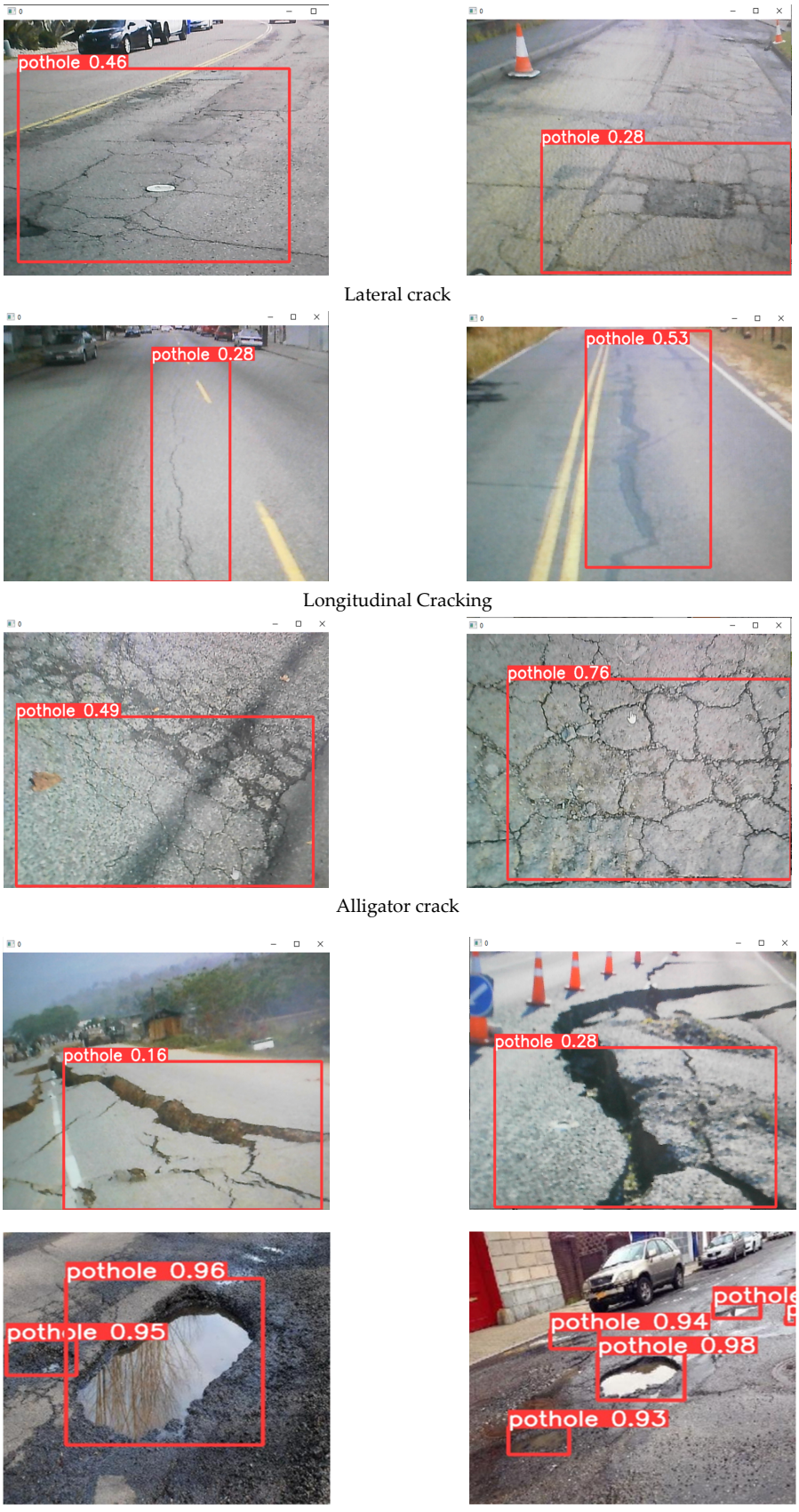


Figure 13. Results obtained when identifying images with potholes.

6. Results and Discussion

6.1. Results

During the operation and use phase, the quality of a road is essential not only for ensuring efficient transportation and the smooth flow of goods but also for the safety of all traffic participants/road users [40,41]. Even small signs of damage, no matter how minor, can potentially lead to serious accidents. Detecting these damages quickly is vital as it ensures both the safety of traffic participants and helps in the more effective allocation of maintenance and repair budgets. The most significant outcome of this research is the successful development of a software called RTI-IMS, which can autonomously and accurately detect road damages promptly.

Scientific research in the field of information technology always demands freshness and staying up-to-date with trends. This is a crucial requirement for researchers [42,43]. However, the basis for comparing research results often relies on the evaluations of previous studies. To demonstrate the excellence of the software developed in this research, this study established optimal evaluation criteria, primarily focusing on model detection performance/accuracy and model detection speed. As a result, alongside the successful development of a model building on the achievements of previous scientific research, this study also showcases the superior advantages of the model and its capability to meet the needs of transportation infrastructure managers.

To demonstrate the superiority of the model in the road infrastructure sector compared to previous research, the two most important evaluation criteria for a machine learning detection model, accuracy and processing speed, have been proposed to compare RTI-IMS with previous road damage detection models, showcasing superior results. In terms of improving the average precision (AP) value, all detection models using the RTI-IMS detector achieved a higher mean average precision (mAP) value compared to models using other detectors, specifically achieving an mAP of 80.19% at an Intersection over Union (IoU) of 0.5%. RTI-IMS was developed based on the Yolo V5 software with SSD detection, achieving superior mAP results as shown in Table 2, Figure 8.

The AP (average precision) values of comparative models are related to the mean average precision (mAP) values of machine learning detection models. These values represent the performance accurately, and they were used to compare the results of the RTI-IMS model with existing models in previous research. Typically, Fast-CNN (F-CNN) is used to develop models for Fast CNN-based machine learning detection to demonstrate superiority (50~65%). However, with the triple fusion method applied in this research, the RTI-IMS developed on the Yolo V5 (2023) platform showed the opposite results. Table 3 shows a mAP value of 74.52%, which demonstrates its superiority compared to average values of around 50%.

For each type of road damage, the research also calculated detailed performance results of the models for each type of damage (IoU = 0.5). The mAP values show that Faster R-CNN Inception Resnet V2 is the most effective model for detecting three out of five types of damage (I, II, and III) with respective AP values of 71.64%, 45.28%, and 70.77%. Faster R-CNN Inception V2 is the best model for detecting type V damage (blurry) with an AP value of 76.59%. However, with the upgraded 2023 version of YOLO V5 applied in the RTI-IMS, the software demonstrates superiority in all mAP values, with values of 79.76%, 75.23%, 75.85%, 70.65%, and 77.04% for each respective type of damage (Table 4), making it suitable for effectively detecting all types of road damage. With IoU machine learning recognition performance indicators (0.5) and the average being $\geq$ 80%, it has proved to have relatively high image detection performance (Figures 9 and 10). These images are used to compare the performance of an RTI-IMS model containing full instances of each object detection class on the same conditions.

The results in this study have been carefully evaluated based on the speed of detection criteria. Thanks to the integration of Mosaic's triple version in this research, the machine learning image data were not only tripled but also helped reduce the processing load on computers. With other processing speed results, effective object detection times

were 2.0 s, 3.7 s, and 9.8 s for images with SSD MobileNet-V1, SSD MobileNet-V2, and SSD Inception Resnet. While in the RTI IMS, with a parameter of 7.2 M, the processing and object detection times were 0.5 s, 1.2 s, and 3.2 s for images with increasing resolutions. However, this is not overly necessary since most dashcam videos typically use resolutions up to a maximum of 640 × 640 (Table 5). This also confirms the results that the object detection speed is almost in real time for the model in this study. In today's world, where computer tools are no longer huge barriers, new models like Yolo V5 are worth exploring to efficiently serve management tasks. The RTI-IMS software developed in this research has become a highly practical application tool that can provide immediate assistance to road infrastructure management agencies for detecting road damage and facilitating road maintenance work.

The comparison results in this study provide an objective view of the novelty and high applicability of the Yolo V5 model in detecting technical infrastructure objects [36,44]. Exploring its efficacy in facilitating management tasks for researchers and managers justifies further research and development of diverse machine learning-based image detection models. Figure 14 showcases images of road damage detected by the RTI-IMS software.

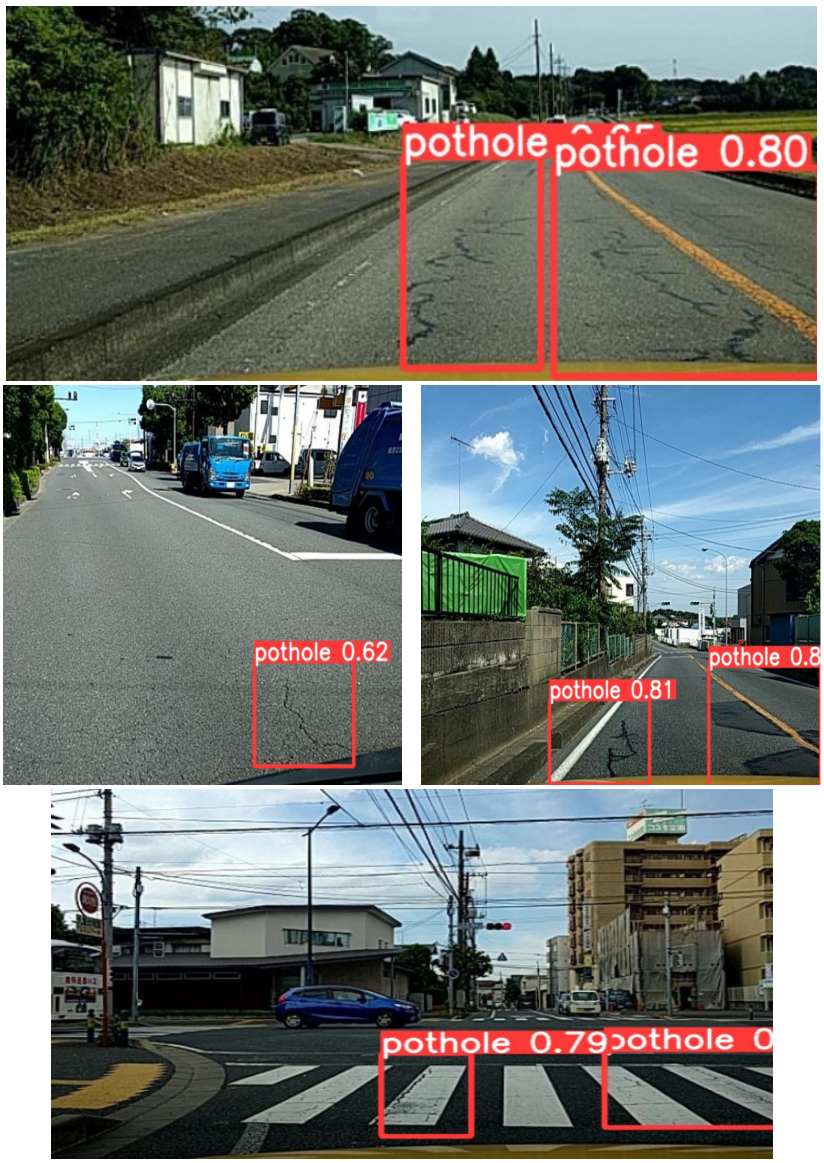


Figure 14. *Cont.*

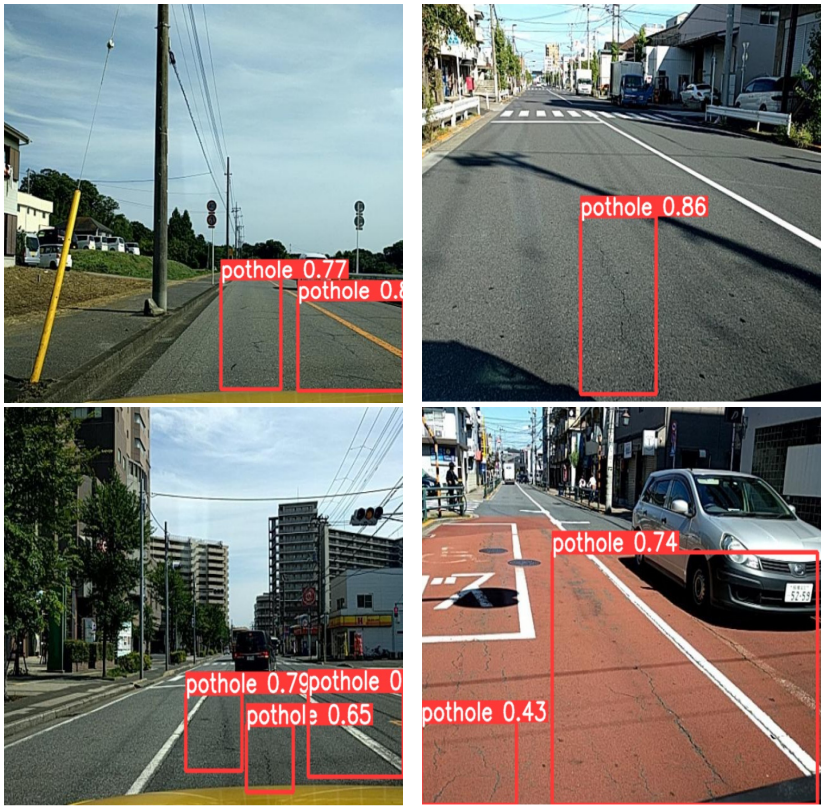


Figure 14. Compilation of road surface detection images.

6.2. Discussion

The RTI-IMS software, although already completed through training with a dataset of 9053 images from previous studies, can benefit from more extensive and diverse training datasets. The effectiveness and intelligence level of a machine learning model like RTI-IMS can be improved by training it with a larger and more comprehensive image library. At this stage of the research, the authors aimed to use the same dataset as previous studies to ensure objective comparisons. However, it is important to note that the dataset for the RTI-IMS software in this research is continuously expanding, even after the publication of this study.

In the approach used as described above, the model was tested with three local traffic management agencies in Vietnam, specifically Quang Ngai Province, Dong Nai Province, and Ho Chi Minh City. Initially, the results were positively evaluated for automating the road damage detection process for these local traffic management agencies. The research team also received feedback, including suggestions for improvement, such as expanding the data storage capacity of the software. Additionally, there were recommendations to add the capability to identify layers of accumulated dust on the road surface as opposed to just identifying the damage that needs repair.

7. Conclusions

This research on the RTI-IMS software has made significant contributions to the field of construction management, particularly in traffic infrastructure management. By integrating Industry 4.0 technologies into road infrastructure management, it has created a highly applicable system that can be readily used by current traffic infrastructure managers. This research has also contributed to enhancing the safety of individuals conducting road inspections, thus reducing traffic accidents resulting from collisions during inspection activities. This humanistic aspect of the research makes it highly valuable and deserving of publication.

In current research methods, the development in road management and maintenance methods are very diverse [45,46]. This study aims to achieve the most optimal goals in cost, management efficiency, and operation of transportation networks. Utilizing the proposed advanced technology to replace humans in patrolling and detecting damaged locations on road infrastructure is important for traffic management. In addition, the research has demonstrated Yolo V5 is the best version of technical infrastructure image detection programming. In terms of performance and speed, the RTI-IMS has demonstrated itself to be superior compared to existing methods. With IoU machine learning recognition performance indicators (0.5) and an average $\geq$ 80%, it has proved relatively high image detection performance. These images are used to compare the performance of an RTI-IMS model containing full instances of each object detection class on the same conditions.

The economic and social benefits of the smart transportation infrastructure network act as an important link in the operation and development of the whole country [22,31]. This research has provided an automatic system that is resilient, adaptive, and flexible to face unstable climate conditions. Previous studies on road traffic infrastructure quality management models have made many positive contributions to improving traffic infrastructure management. However, the rapid development of the technology field has provided humanity with new tools for more effective management, and RTI-IMS is the next inheritance and development in scientific research.

The ITM-IMS system has published image data on road damage on a number of routes in the transportation network, as shown in Sections 5 and 6. These road surface damage detection images were compared with data from the current construction assessment process. Therefore, these results can be provided to local traffic authorities as a basis for calculating the cost of repairing road damage on different routes.

The authors are continuously applying an expansion process to update various methods and incorporate new technologies, such as GPS, IoT, and Big Data, into subsequent stages to make the RTI-IMS software even more useful and yield more positive results. The user interface for RTI-IMS will also be subject to further research to become more user-friendly in the near future.

Author Contributions: Methodology, S.V.H.P. and K.V.T.N.; Project administration, K.V.T.N. Both two authors wrote, prepared, and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All necessary data related to the results of this research have already been presented and explained in this manuscript.

Acknowledgments: We would like to thank Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for the support of time and facilities for this study.

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

References

1. Koch, C.; Jog, G.M.; Brilakis, I. Automated pothole distress assessment using asphalt road surface video data. *J. Comput. Civ. Eng.* **2013**, *27*, 370–378. [CrossRef]
2. Cao, M.T.; Tran, Q.V.; Nguyen, N.M.; Chang, K.T. Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources. *Adv. Eng. Inform.* **2020**, *46*, 101182. [CrossRef]
3. Ouma, Y.O.; Hahn, M. Wavelet-morphology based detection of incipient linear cracks in asphalt pavements from RGB camera imagery and classification using circular Radon transform. *Adv. Eng. Inform.* **2016**, *30*, 481–499. [CrossRef]
4. Gavilán, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocaña, M.; Aliseda, P.; Yarza, P.; Amírola, A. Adaptive road crack detection system by pavement classification. *Sensors* **2011**, *11*, 9628–9657. [CrossRef]
5. He, Y.; Jin, Z.; Zhang, J.; Teng, S.; Chen, G.; Sun, X.; Cui, F. Pavement Surface Defect Detection Using Mask Region-Based Convolutional Neural Networks and Transfer Learning. *Appl. Sci.* **2022**, *12*, 7364. [CrossRef]

6. Outay, F.; Mengash, H.A.; Adnan, M. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transp. Res. Part A Policy Pract.* **2020**, *141*, 116–129. [CrossRef] [PubMed]
7. Gunawan, F. Detecting road damage by using gyroscope sensor. *ICIC Express Lett.* **2018**, *12*, 1089–1098. [CrossRef]
8. Harikrishnan, P.; Gopi, V.P. Vehicle vibration signal processing for road surface monitoring. *IEEE Sens. J.* **2017**, *17*, 5192–5197. [CrossRef]
9. Zhu, C.; Pastor, G.; Xiao, Y.; Ylajaaski, A. Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges. *IEEE Commun. Mag.* **2018**, *56*, 58–63. [CrossRef]
10. Zakeri, H.; Nejad, F.M.; Fahimifar, A. Image based techniques for crack detection, classification and quantification in asphalt pavement: A review. *Arch. Comput. Methods Eng.* **2016**, *24*, 935–977. [CrossRef]
11. Kamaliardakani, M.; Sun, L.; Ardakani, M.K. Sealed-crack detection algorithm using heuristic thresholding approach. *J. Comput. Civ. Eng.* **2014**, *30*, 04014110. [CrossRef]
12. Oliveira, H.; Correia, P.L. Automatic road crack segmentation using entropy and image dynamic thresholding. In Proceedings of the 2009 17th European Signal Processing Conference, Glasgow, UK, 24–28 August 2009; IEEE: New York, NY, USA, 2009; pp. 622–626.
13. Chow, J.K.; Su, Z.; Wu, J.; Tan, P.S.; Mao, X.; Wang, Y.H. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inf.* **2020**, *45*, 101105. [CrossRef]
14. Singh, D.; Singh, M. Internet of vehicles for smart and safe driving. In Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE), Shenzhen, China, 19–23 October 2015; IEEE: New York, NY, USA, 2015; pp. 328–329. [CrossRef]
15. Shim, S.; Kim, J.; Lee, S.W.; Cho, G.C. Road surface damage detection based on hierarchical architecture using lightweight auto-encoder network. *Autom. Constr.* **2021**, *130*, 103833. [CrossRef]
16. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road damage detection using deep neural networks with images captured through a smartphone. *arXiv* **2018**, arXiv:1801.09454.
17. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Sekimoto, Y. RDD2020: An annotated image dataset for automatic road damage detection using deep learning. *Data Brief* **2021**, *36*, 107133. [CrossRef]
18. Sinharay, A.; Bilal, S.; Pal, A.; Sinha, A. Low computational approach for road condition monitoring using smartphones. In Proceedings of the Computer Society of India (CSI) Annual Convention, Theme: Intelligent Infrastructure, Visakhapatnam, India, 13–15 December 2013; pp. 13–15.
19. Fang, W.; Luo, H.; Xu, S.; Love, P.E.; Lu, Z.; Ye, C. Automated text classification of near misses from safety reports: An improved deep learning approach. *Adv. Eng. Inform.* **2020**, *44*, 101060. [CrossRef]
20. Harris, D.K.; Alipour, M.; Acton, S.T.; Messeri, L.R.; Vaccari, A.; Barnes, L.E. The citizen engineer: Urban infrastructure monitoring via crowd-sourced data analytics. In Proceedings of the Structures Congress 2017, Denver, CO, USA, 6–8 April 2017; pp. 495–510.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-time vehicle detection based on improved yolo v5. *Sustainability* **2022**, *14*, 12274. [CrossRef]
23. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
24. Malygin, I.; Komashinsky, V.; Tsyganov, V. International experience and multimodal intelligent transportation system of Russia. In Proceedings of the 2017 Tenth International Conference Management of Large-Scale System Development (MLSD), Moscow, Russia, 2–4 October 2017; pp. 1–5.
25. Cumbajin, E.; Rodrigues, N.; Costa, P.; Miragaia, R.; Frazão, L.; Costa, N.; Fernández-Caballero, A.; Carneiro, J.; Buruberri, L.H.; Pereira, A. A Systematic Review on Deep Learning with CNNs Applied to Surface Defect Detection. *J. Imaging* **2023**, *9*, 193. [CrossRef]
26. Hatır, M.E.; İnce, İ.; Korkanç, M. Intelligent detection of deterioration in cultural stone heritage. *J. Build. Eng.* **2021**, *44*, 102690. [CrossRef]
27. Kyriakou, C.; Christodoulou, S.E.; Dimitriou, L. Smartphone-based pothole detection utilizing artificial neural networks. *J. Infrastruct. Syst.* **2019**, *25*, 04019019. [CrossRef]
28. Jo, Y.; Ryu, S. Pothole detection system using a black-box camera. *Sensors* **2015**, *15*, 29316–29331. [CrossRef]
29. Chun, C.; Ryu, S.-K. Road surface damage detection using fully convolutional neural networks and semi-supervised learning. *Sensors* **2019**, *19*, 5501. [CrossRef] [PubMed]
30. Kalfarisi, R.; Wu, Z.Y.; Soh, K. Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization. *J. Comput. Civ. Eng.* **2020**, *34*, 04020010. [CrossRef]
31. Zhang, A.; Wang, K.C.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 805–819. [CrossRef]
32. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
33. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]

34. Socaciu, C. Bioeconomy and green economy: European strategies, action plans and impact on life quality. *Bull. UASVM Food Sci. Technol.* **2014**, *71*, 1–10. [CrossRef]
35. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
37. Hoang, N.-D.; Nguyen, Q.-L.; Tien Bui, D. Image processing–based classification of asphalt pavement cracks using support vector machine optimized by artificial bee colony. *J. Comput. Civ. Eng.* **2018**, *32*, 04018037. [CrossRef]
38. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7310–7311.
39. Sun, C.; Luo, Y.; Li, J. Urban traffic infrastructure investment and air pollution: Evidence from the 83 cities in China. *J. Clean. Prod.* **2018**, *172*, 488–496. [CrossRef]
40. Tomiyama, K.; Kawamura, A.; Fujita, S.; Ishida, T. An effective surface inspection method of urban roads according to the pavement management situation of local governments. *J. Jpn. Soc. Civ. Eng. Ser. F3 (Civ. Eng. Inform.)* **2013**, *69*, I-54–I-62.
41. Zhankaziev, S. Current trends of road-traffic infrastructure development. *Transp. Res. Procedia* **2017**, *20*, 731–739. [CrossRef]
42. Hatır, E.; Korkanç, M.; Schachner, A.; Ince, I. The deep learning method applied to the detection and mapping of stone deterioration in open-air sanctuaries of the Hittite period in Anatolia. *J. Cult. Herit.* **2021**, *51*, 37–49. [CrossRef]
43. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755, Proceedings, Part V 13, 2014.
44. Azam, A.; Alshehri, A.H.; Alharthai, M.; El-Banna, M.M.; Yosri, A.M.; Beshr, A.A. Applications of Terrestrial Laser Scanner in Detecting Pavement Surface Defects. *Processes* **2023**, *11*, 1370. [CrossRef]
45. Guan, H.; Li, J.; Yu, Y.; Chapman, M.; Wang, H.; Wang, C.; Zhai, R. Iterative tensor voting for road surface crack extraction using mobile laser scanning data. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1527–1537. [CrossRef]
46. Li, S.; Cao, Y.; Cai, H. Automatic road surface-crack detection and segmentation based on steerable matched filtering and an active contour model. *J. Comput. Civ. Eng.* **2017**, *31*, 04017045. [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.