

Article

Deep Learning-Based Road Pavement Inspection by Integrating Visual Information and IMU

Chen-Chiung Hsieh ^{1,*} , Han-Wen Jia ¹, Wei-Hsin Huang ² and Mei-Hua Hsieh ³¹ Department of Computer Science and Engineering, Tatung University, Taipei 104, Taiwan² The Graduate Institute of Design Science, Tatung University, Taipei 104, Taiwan; wshuang@gm.ttu.edu.tw³ Department of Product Design, School of Arts and Design, Sanming University, Sanming 365004, China

* Correspondence: cchsiehl@gm.ttu.edu.tw; Tel.: +886-2-21822928 (ext. 6571)

Abstract: This study proposes a deep learning method for pavement defect detection, focusing on identifying potholes and cracks. A dataset comprising 10,828 images is collected, with 8662 allocated for training, 1083 for validation, and 1083 for testing. Vehicle attitude data are categorized based on three-axis acceleration and attitude change, with 6656 (64%) for training, 1664 (16%) for validation, and 2080 (20%) for testing. The Nvidia Jetson Nano serves as the vehicle-embedded system, transmitting IMU-acquired vehicle data and GoPro-captured images over a 5G network to the server. The server recognizes two damage categories, low-risk and high-risk, storing results in MongoDB. Severe damage triggers immediate alerts to maintenance personnel, while less severe issues are recorded for scheduled maintenance. The method selects YOLOv7 among various object detection models for pavement defect detection, achieving a mAP of 93.3%, a recall rate of 87.8%, a precision of 93.2%, and a processing speed of 30–40 FPS. Bi-LSTM is then chosen for vehicle vibration data processing, yielding 77% mAP, 94.9% recall rate, and 89.8% precision. Integration of the visual and vibration results, along with vehicle speed and travel distance, results in a final recall rate of 90.2% and precision of 83.7% after field testing.

Keywords: image recognition; deep learning; pavement inspection; intelligent inspection

Citation: Hsieh, C.-C.; Jia, H.-W.; Huang, W.-H.; Hsieh, M.-H. Deep Learning-Based Road Pavement Inspection by Integrating Visual Information and IMU. *Information* **2024**, *15*, 239. <https://doi.org/10.3390/info15040239>

Academic Editors: Nikolaos Mitianoudis and Ilias Theodorakopoulos

Received: 2 March 2024

Revised: 17 April 2024

Accepted: 18 April 2024

Published: 20 April 2024



Copyright: © 2024 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

Land transportation encompasses railroad and highway systems, with highways as vital infrastructure in Taiwan, which is crucial for national economic development. Taiwan's highway network includes national, provincial, city, county, district, and village highways, totaling 21,757 km [1]. From January to August 2011, the monthly average vehicle count reached 2826.45 million per kilometer [2], showing highway transportation's colossal capacity and importance. Highways, fixed vehicle movement, and parking facilities [3] aim to deliver safe, fast, reliable, convenient, and high-volume service conditions, relying on robust pavement structures for safe travel. Pavements between tires and roadbeds bear vehicular traffic loads and environmental stressors. They distribute these loads layer by layer, mitigating external forces to the natural soil layer below. Pavement design influences vehicle speed, comfort, safety, and operating costs. Modern highway and urban road pavements should cater to daily traffic volumes, minimizing maintenance expenses, enhancing driving efficiency, and improving comfort and safety.

As pavement health monitoring plays a crucial role in pavement management systems, this issue has been a hotspot of transportation research since the middle of the 20th century. For this reason, governments allocate a significant budget annually to provide the necessary facilities and equipment to find a high-speed, accurate, safe, and automatic detection method. According to the surveys [4,5], the main steps of automatic pavement assessment include data acquisition, data processing, and pavement interpretation. The traditional way of pavement assessment is a visual inspection that human experts can conduct as the easiest method. Other methods have been surveyed in [5] for detecting the isolation of different

kinds of distress in pavement surface images. Whether these approaches are supervised, unsupervised, or semi-supervised, various techniques can be grouped into five methods: (1) Statistical Method, (2) Physical Method, (3) Filtering Method, (4) Model-Method, and (5) Hybrid Method.

Motivated by the progress achieved by deep learning (DL), researchers have developed extensive crack segmentation models based on DL methods with significantly different levels of accuracy [6]. Although many of the models provide satisfying detection performance, why these models work still needs to be determined. The objective of [6] is to survey recent advances in automated DL crack recognition and provide evidence for their underlying working mechanism. They first reviewed 54 DL crack recognition methods to summarize critical factors in these models. Then, a performance evaluation of fourteen famous semantic segmentation models is conducted using the quantitative metrics: F-1 score and mIoU.

Although many DL-based research studies are given in [6] at this stage studying various automated highway inspections, there are still areas for speed improvement [7] while maintaining some accuracy rate. Some misjudgments not only do not reduce human resources and improve the efficiency of inspection and maintenance, but on the contrary, they also result in the need for many personnel to screen the identification results. This study adopts YOLOv7 [8] to solve the problem of workforce inspecting efficiency and improve the whole process throughput, which balances accuracy and speed as the core detection function. In addition to reducing false alarms, IMU is adopted as a filter, and therefore, an integration mechanism is investigated to fuse visual information and car altitude data. The GoPro camera and inertial measurement unit are deployed to reduce hardware costs for road pavement inspection. The data are transmitted via a 5G mobile network and then handed over to the server for recognition and storage. When the seriousness is detected to the extent that it affects traffic safety, an alarm will be issued immediately to notify the maintenance personnel to go to the emergency repairs, thus accomplishing the goal of automated inspections.

This research aims to create a system that can replace human inspection, reduce the labor required for inspection, and improve inspection efficiency. The automated inspection system, from image acquisition, image processing, and image interpretation to inspection results storage, can be fully automated, and the data can be managed in a unified way to reduce the risk of omission caused by the workforce. This study develops an efficient front-end system on an Nvidia Jetson Nano to capture pothole images and vehicle attitude data and transmit them to the back-end server via a 5G mobile network to achieve the above goal. An efficient back-end system is also developed by utilizing parallel threads for pavement categorization and result storage to Mongo Database, which are then queried by the maintenance personnel for review and repair.

Although the road pavement damage dataset is more accessible than the railroad defect dataset, the relative environment is more complex when considering different kinds of roads. The primary goal of this study is to avoid obstructing other vehicles, which may make it impossible to recognize the defects effectively. Roads, unlike railroads, are more complex environments with defects that have been filled in the past, and these repaired defects are visually very similar to pothole defects. Discriminating between these two categories is the second objective of this study, which requires a large amount of data for training. Considering road vehicles and weather factors, this study uses data enhancement to increase the dataset's diversity and strengthen the recognition effect.

While collecting data on pavement damage, several categories were found to be particularly difficult to define, with the category of pavement cracks being the most serious. The main reason for this is that cracks usually occur regionally, i.e., for one crack, there are two to three similar cracks around it, and the ends of the cracks are connected, which leads to confusion when labeling the data, which in turn leads to errors in the training set and the test set during training, as shown in Figure 1. However, these errors could be corrected by post-processing all the captured frames. Overall, the main contribution of this

study is to develop an efficient pavement defect detection system that includes parallel thread implementation on both the front and back end. By utilizing YOLOv7 and Bi-LSTM as the detection engine, the execution speed of the proposed back-end system achieves 30–40 FPS on a PC with CPU Intel i9-12900K, GPU Nvidia RTX 3090, and RAM 64 GB. A fusion mechanism is also proposed to integrate the visual and vibration results for false alarm filtering.

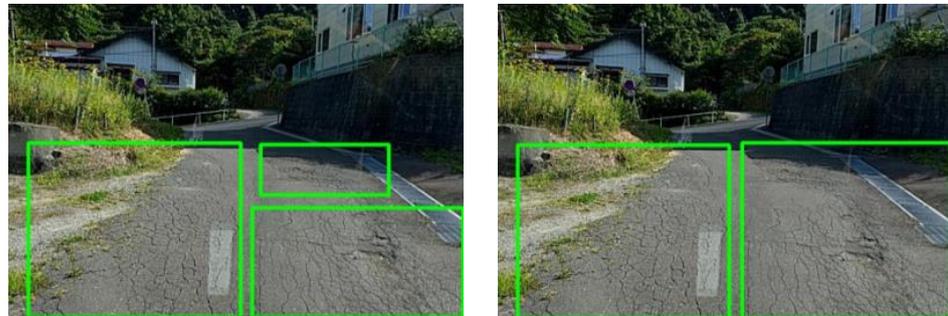


Figure 1. There are different ways to annotate the same picture.

2. Related Works

2.1. Road Pothole Inspection

The most mature techniques for road inspection are using laser 3D modeling [9], ultrasound [10], and radar [11]. Due to the high hardware cost, many domestic and international studies are trying the object detection method in computer vision for road inspection. According to Aparna et al. [12], an infrared camera is used to capture the image of potholes on the road surface, and after object recognition, the temperature difference between the inside and outside of the pothole is used to detect whether it is a real pothole or not. After testing various neural network models, it is concluded that an accuracy rate of 95% can be achieved when using the neural network ResNet 152. However, the model of this neural network needs to be bigger, and the recognition speed of the system could be faster.

The use of ultrasonic waves for road pothole detection was proposed by Shenu et al. [13]. Because water absorbs fewer sound waves in the 400 nm to 450 nm range, this system determines potholes by detecting whether more waves in the 400 nm to 450 nm range are reflected from the pothole. However, this method is limited because water must accumulate in the pothole. So, if the ground is dry during the inspection, the system will not be able to work. Nienaber et al. [14] used Canny edge detection for pothole detection through conventional image processing and obtained a recognition recall rate of 74.4% and an accuracy rate of 81.8%. However, more than this recognition rate is needed. Based on the opinions collected in the past, the inspectors would like to have a recognition accuracy of at least 90% to be sufficient.

Varadharajan et al. [15] mentioned that the background noise is first removed using semantic segmentation, and the cracked regions are separated from the non-cracked areas using the SLIC Superpixel Algorithm [16]. In the final classification stage, MISVM [17] is used to classify the separated cracked regions and determine the authenticity of the cracked areas. The final accuracy and recall rates still need to be higher than those of manual visual inspection (50%, 70%), which are only 40% and 64%. In [18], JICA and the Ministry of Transportation of the Republic of Tajikistan mentioned that the use of lasers for detecting road pavement smoothness was introduced into the International Roughness Index (IRI) to assess the condition of the road pavement. As shown in Figure 2, the classification of damage categories according to the index range was formalized for real-world scenarios.



Figure 2. Examples of the percentage of cracks [18] are (a) new pavement, 0% cracks; (b) partial cracks, 1–50%; (c) partially connected cracks, 50–70%; and (d) dense and mixed cracks, 70–100%.

2.2. Object Detection Neural Networks

This study uses a visual neural network to detect road pavement defects and determine whether the road pavement is damaged. Because the server needs real-time streaming for maintenance personnel to view the status, this study emphasizes using a one-stage object detection neural network. Compared with YOLOv3 [19], YOLOv4 [20] significantly improves the detection accuracy of the model while guaranteeing the recognition speed. It reduces the hardware usage requirement, obtaining 43.5% average precision (AP) and 65.7% average precision at IOU = 0.5 (AP50) on the MS COCO dataset and 10% and 12% improvement in AP and FPS, respectively. YOLOv4 is twice as fast as EfficientDet [21] with the same recognition capability.

The most crucial feature of YOLOv6 [22] is that the recognition speed is greatly improved with higher mAP than YOLOv5 [23]. YOLOv6-nano can reach 35.0% AP accuracy on COCO and 1242 FPS recognition speed on T4. YOLOv6-s can reach 43.1% AP accuracy on COCO and 520 FPS. YOLOv6 adopts a hardware-friendly backbone network design and introduces the RepVGG style structure [24], which can generate more branches during training. In the actual deployment, the structure can be equivalently fused into a single 3×3 convolutional quintic structure, which can more effectively utilize the computing power of GPUs and significantly increase the computing speed of neural networks. In addition to increasing the speed of neural network operation, YOLOv6 adds Decoupled Head [22] and two new 3×3 convolutional layers to maintain the accuracy of YOLOv6.

YOLOv7 [8] not only outperforms all YOLO series, transformer-based [25], convolutional-based, etc., models in terms of accuracy, such as the most popular YOLOR [26], PPYOLOE [27], YOLOX [28] or Scaled-YOLOv4 [29], but also achieves excellent results in terms of speed. The authors of YOLOv7 mentioned four architectures for model improvement in their paper, namely, VoVNet, CSPVoVNet, ELAN, and E-ELAN. Among them, CSPVoVNet is a combination of CSPNet and VoVNet, and the design of the architectures not only considers the number of parameters, the amount of computation, and the computational density but also analyzes the gradient paths so that the weights of different layers can learn more diverse features, making the inference faster and more accurate. In addition, YOLOv7 uses a new architecture, E-ELAN, which is based on ELAN with expand, shuffle, merge, and cardinality methods, which maintains the original gradient path and enhances the model's learning ability.

In addition to model optimization, YOLOv7 also optimizes the training process. YOLOv7 uses Model Re-Parameterization, which is divided into Model-Level and Module-Level. There are two approaches to Model-Level Re-Parameterization. One is to train multiple models with different training data and then perform weighted averaging on these models. The other is to weigh the weights of different iterations in the training

process. Module-Level parameterization splits the module into other branches during the training and then aggregates these branches into a single module during the inference computation. The most significant difference between YOLOv7 and others is the Dynamic Label Assignment strategy (DLA). The auxiliary mechanism to train the shallow network weights can significantly improve the model performance. The Lead Head has a relatively strong learning ability. By allowing the shallow Auxiliary Head to directly learn the information that the Lead Head has already learned, the Lead Head will be able to focus more on understanding the information that has not yet been discovered.

The rapid advancement of artificial intelligence enables YOLO-based deep learning techniques [30] to be applied to pavement damage detection from various images. Some researchers chose to equip a vehicle platform with a standard camera to acquire pavement images from the vehicle's front view. Single-stage target detection algorithms YOLOv4-Tiny [31], Scaled-YOLOv4 [32], and YOLOv5 [30,33,34] were applied in pavement damage detection using road images captured from the front view of the vehicle, and they all achieved high accuracy. In summary, these studies demonstrated the effectiveness of using vehicle-mounted platforms with cameras to acquire road images and the application of deep learning approaches. However, the labeled sample dataset in deep learning significantly affects the model performance and the training and testing model accuracy.

2.3. One-Dimensional Neural Networks

This study used a 1-D neural network to determine vehicle body sway. When damage is recognized in the front image, the three-axis acceleration and body attitude in inertial navigation are used to determine whether the vehicle is moving to distinguish between stains and damage. This data are time-dependent, so a 1-D neural network is used. The most significant difference between a Recurrent Neural Network (RNN) [35] and a traditional neural network is that RNN will bring the previous output into the hidden layer of the next time and train together. This also means each output is directly related to the last output. Because each output is correlated with the previous output, it makes RNN very suitable for training events associated with each other. However, the drawbacks of RNN are also quite noticeable. In addition to spending much time on training, as the results of the previous output will be added every time, the weight of the past results will be lower and lower, which gradually loses the past information, and makes it impossible to deal with the problems of long sequences.

Long-Short Term Memory (LSTM) [35,36] is improved from RNN. The primary purpose of LSTM is to solve the shortcoming of RNN, which is that the later the input is, the higher the impact; the earlier the input is, the lower the effect. Valves are used to determine the importance of inputs and decide whether to memorize the inputs and whether they can be output to the output layer to solve this problem. Bi-LSTM [37,38] was proposed by Huang et al., which consists of two LSTMs; one is responsible for processing the positive-ordered inputs, and the other handles the inverse-ordered data. This approach can avoid the loss of past data due to the uniform order of inputs in RNN and LSTM.

3. Proposed Approach

3.1. System Architecture

In this section, we will introduce the system framework for pavement defect inspection and the hardware components used in the system, such as NVidia Jetson Nano, a GoPro camera, inertial navigation, and a 5G communication module, as shown in Figure 3.

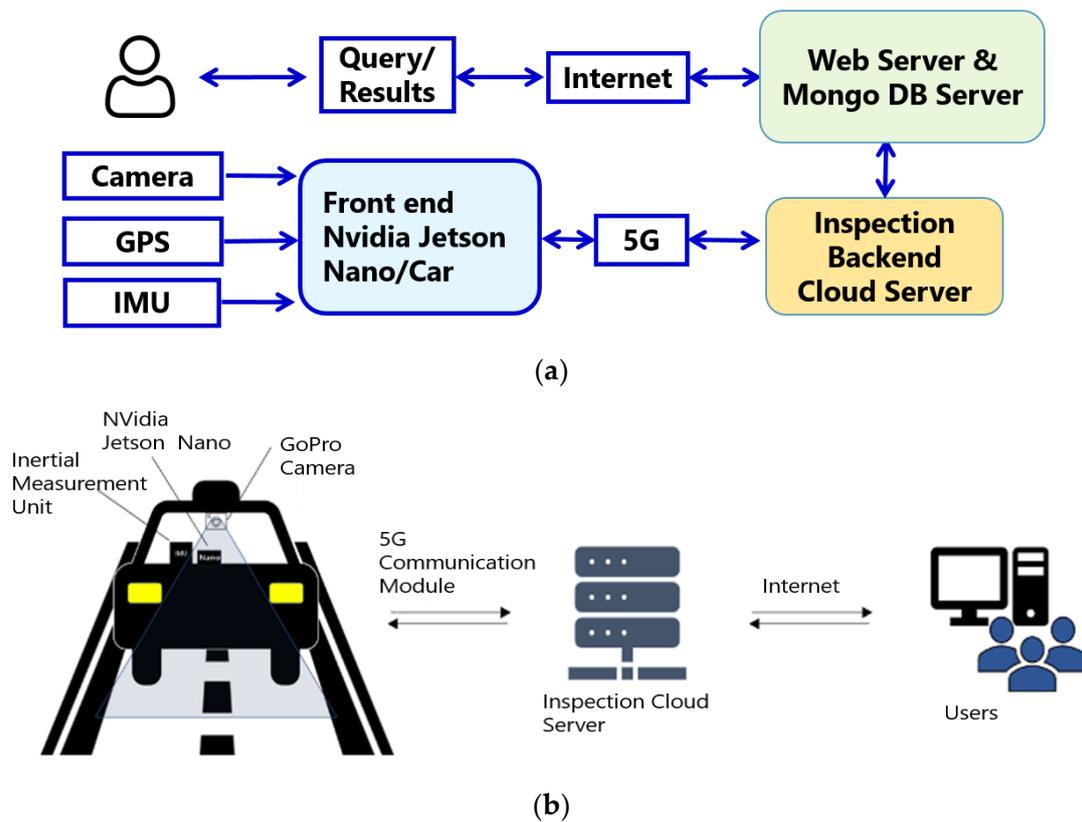


Figure 3. (a) Proposed system framework and (b) installation diagram.

3.1.1. Road Pavement Defect Detection

The pavement damage recognition system can be divided into front-end information collection, transmission, and back-end detection and storage. The front-end information collection consists of Nvidia Jetson Nano, a GoPro camera, and an inertial measurement unit. The Nvidia Jetson Nano collects the front image, vehicle attitude, 3-axis acceleration, and localization information and integrates all the data. The socket protocol transmits the data over a 5G mobile network. After receiving the images and other information, the back-end detection and storage system uses YOLOv7 to detect objects. After detecting the pavement damage, the pothole's location in the image is calculated. Then, Bi-LSTM is used to recognize the vibration information of the vehicle body at the corresponding distance. Finally, the recognition results are written into MongoDB, which is convenient for maintenance personnel to view.

Because the system will be applied to the general road, it needs to be installed on a vehicle with a speed of 50 km per hour for detection. In addition, to ensure the accuracy of the inertial measurement unit, the vehicle information needs to be collected about once in 0.5 m, which is converted to a transmission speed of about 30 FPS. The system uses Python as the development language and parallel processing to improve system performance. The front-end information collection and transmission consists of four threads: image acquisition, inertial measurement information, information integration, and socket transmission. The image acquisition thread and the inertial measurement information thread receive and transfer data to the information integration thread. In the information integration thread, the images are first compressed, and the valuable parts of the inertial measurement information are selected and transferred to the socket transmission thread to be recognized and stored in the back-end detection and storage system, as shown in Figure 4, which shows the activity diagram of the front-end information collection and transmission.

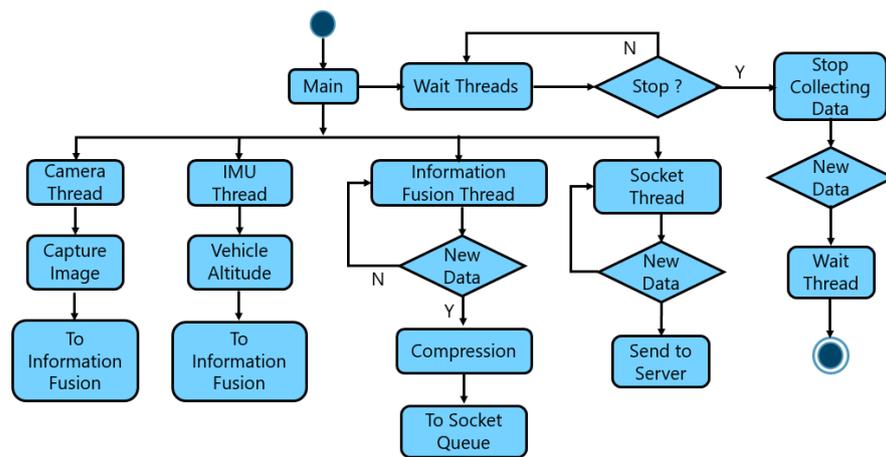


Figure 4. Proposed software system flowchart of the front end.

The back-end recognition and storage system consists of five threads: the image receiving thread, image recognition thread, vehicle attitude recognition thread, database thread, and file writing thread. The image-receiving thread is responsible for receiving images, transcoding them, and transferring them to the image recognition thread for the first stage of recognition. After the recognition, the data are sent to the Bi-LSTM thread for the second recognition stage. After both recognition phases are completed, the data will be transferred to the file writing thread and database thread for writing to the file and database, respectively. Finally, the file writing and database threads will compare whether the data can be deleted. If it cannot be deleted, the executing thread will notify the other thread by marking the variables, as shown in Figure 5’s activity diagram of the back-end recognition and storage system.

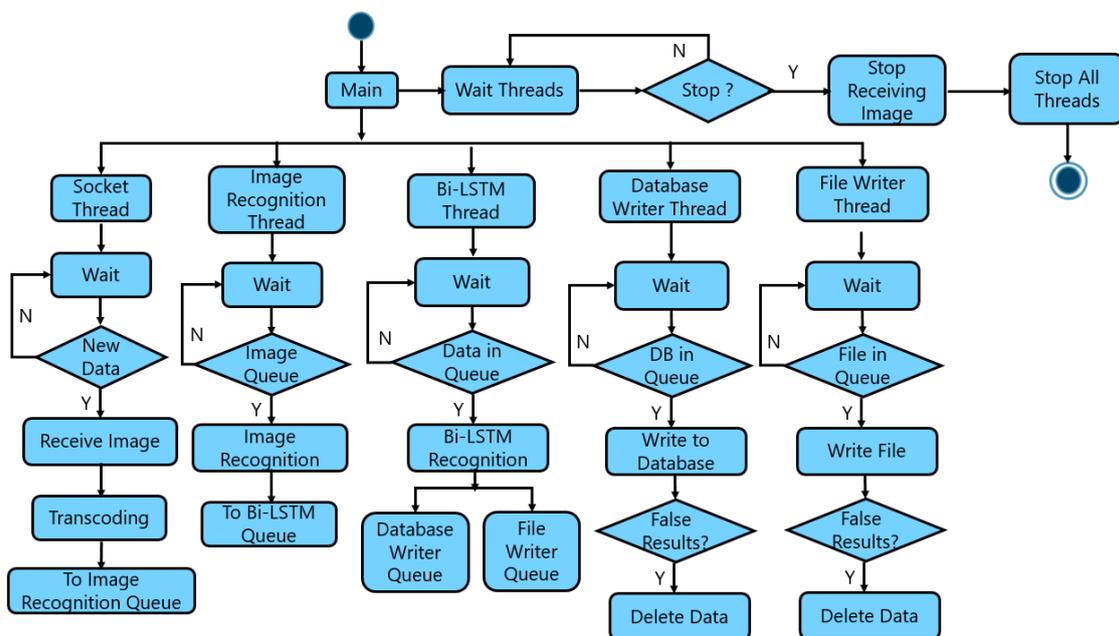


Figure 5. Proposed software system flowchart of the back-end.

The system is designed to organize and review the data with a cloud site for reviewers to check. The cloud site contains real-time streaming and inspection results, allowing users to view images and inspection results while inspecting. The inspection results in the cloud site can provide follow-up inspectors to check what areas need to be inspected and arrange the inspection schedule.

3.1.2. GoPro Camera

This study used a GoPro camera [39] for image acquisition. Using GoPro to capture images can reduce the blurring of images caused by the moving vehicle's vibration, allowing for better training modeling and recognition results. Moreover, GoPro can support up to 4K/120 FPS recording, which can provide high-resolution images for image recognition, which is very suitable for use. In addition, because this study mainly focuses on image recognition and is supplemented by vehicle vibration for inspection, although the infrared camera is evident in photographing potholes and cracks, it was not selected due to the inconsistency of the usage context. Because the depth of the cracks could not be determined by standard cameras, a GoPro with better image quality and anti-shock capability was chosen as the capture camera for this study.

3.1.3. GPS and Inertial Measurement Unit

The inertial navigation hardware equipment used in this study was provided by Sinostar [40]. It contains a GPS and an inertial measurement unit (IMU), which can be connected to the computer via USB, and the readout software is self-developed and can obtain information including GPRMC [41] and AHRS [42] in the format of Table 1's GPRMC data format and Table 2's AHRS data format. However, the IMU sensor needs to be calibrated before usage. Users must place the IMU box on a flat, stable surface before the front passenger's seat. Make sure the device is level and not moving during calibration. This process may take several minutes to complete. Detailed steps may vary according to the vendor's user manual and thus omitted.

Table 1. GPRMC data format [41].

Field No.	Structure	Description
<1>	UTC	Hhmmss (hour, minute, second)
<2>	Position status	A = data valid, V = data invalid
<3>	Latitude	ddmm.mmmm (degree, minute)
<4>	Latitude direction	N = North, S = South
<5>	Longitude	ddmm.mmmm (degree, minute)
<6>	Longitude direction	E = East, W = West
<7>	Speed over ground, knots	000.0~999.9 knots
<8>	Track made good, degrees	000.0~359.9 degree
<9>	Date	ddmmyy (day, month, year)
<10>	Magnetic variation, degrees	000.0~180.0 degree
<11>	Magnetic variation direction E/W	E (East) or W (West)
<12>	Positioning system mode indicator	A = autonomous, D = differential, E = estimated, N = data not valid

Table 2. AHRS data format [42].

Field No.	Parameter	Format
<1><2><3>	x, y, z (Attitude)	xxx.xxx
<4><5><6>	x, y, z acceleration (GPS definition)	xxx.xxx
<7><8><9>	Angular rate	xxx.xxx

3.1.4. Nvidia Jetson Nano

In this study, Nvidia Jetson Nano is used as the front-end information collection and transmission system platform, which has the advantages of low power consumption, small size, system stability, . . . , etc. A Nvidia Jetson Nano consumes only 10 W of power, and it can be supplied by the power supply device in the car and run stably. In addition, a Nvidia Jetson Nano provides four USB 3.0 ports for cameras, inertial navigation modules, and a 5G communication module and supports up to 4K @ 60 fps (H.264/H.265) encoding

and decoding for video processing. Based on the above advantages, a Nvidia Jetson Nano performs excellently in front-end information collection and transmission.

When using Nvidia Jetson Nano for the first time, we need to use SDK Manager to install the system, and we can see the Linux screen after the system is installed. Because Nvidia Jetson Nano does not support the Anaconda management environment, and to save storage space, this study uses the Linux built-in Python environment to write the program. Because we used a GoPro with a capture box to acquire images, we used the OpenCV package to control the camera. Because the customary navigation module provided by Nvidia uses USB as the interface, we used the PySerial package to connect to the serial port.

3.1.5. Cloud Server

This study uses a high-performance host as the cloud server and Docker to manage the recognition program and MongoDB. Through the built-in function of Docker, all the functions will be started automatically at boot time. When the program starts automatically, it first loads the YOLOv7 neural network, and after YOLOv7 is loaded, it sends signals to the main program and loads the Bi-LSTM, and when all the threads are ready, the system waits for the front-end system to connect and transmit. Because the system manages the database and the recognition system as two containers, connecting the two containers via a virtual network is necessary before connecting to the database. Then, the recognition system connects to the database via the IP of the virtual network.

3.2. Dataset

The datasets used in this study are mainly categorized into two types: front-end image and vehicle attitude. The first is the front image dataset, consisting of web-based, self-recorded, and retouched datasets. At the initial stage of dataset production, most datasets are open source on the internet and are manually adjusted to remove the datasets that do not meet the usage context. As more and more front-end images were recorded using a vehicle recorder, it was found that pavement cracks usually occurred regionally and were connected at the end, as shown in Figure 6, leading to confusion when defining whether a crack was one or two. Therefore, images that meet the above criteria are retouched to ensure no confusion during validation testing.



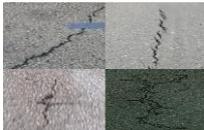
Figure 6. Dataset processing. (a) The original image before processing. (b) The modified image after processing.

The vehicle attitude dataset was recorded in-house using an inertial navigation module mounted in front of the vehicle's windshield and collected simultaneously with the video dataset. The vehicle speed must be kept below 54 km/h during the process to ensure that more than two vibrations per meter are obtained.

3.2.1. Front View Image Dataset

The dataset has two types of roadway pavement defects: potholes and pavement cracks. Potholes are visible in the roadway pavement, and pavement cracks are visible in the roadway pavement, as categorized in Table 3.

Table 3. Road pavement defects.

No.	Type	Sample Image	Definition
1	Potholes		Visible potholes in road pavement
2	Cracks		Visible cracks in road pavement

3.2.2. Vehicle Vibration Dataset

Vehicle vibration data includes vehicle three-axis acceleration and vehicle attitude. Every five strokes is a frame, and Bi-LSTM is used for the recognition. The dataset is categorized into two types: road pavement uneven—low risk and road pavement uneven—high risk, as shown in Table 4 and Figure 7. Each subgraph in Figure 8 contains nine points, each of which is the current acceleration value of the vehicle in that direction. The result would be high risk when at least two criteria are met. Each criterion measures the variation of the accelerations. All the thresholds are defined according to our experiences from experiments.

Table 4. Pavement defects by vehicle attitude.

No.	Type	Sample Image (Time Domain)	Definition (at Least Two Items Are Met)
1	Road pavement uneven—low risk	As shown in Figure 7a are the acceleration (ax, ay, az) in (x, y, z).	<ul style="list-style-type: none"> • $\max(ax)-\min(ax) < 1208 \text{ mm/s}^2$ • $\max(ay)-\min(ay) < 1168 \text{ mm/s}^2$ • $\max(az)-\min(az) < 472 \text{ mm/s}^2$
2	road pavement uneven—high-risk	As shown in Figure 7b are the acceleration (ax, ay, az) in (x, y, z).	<ul style="list-style-type: none"> • $\max(ax)-\min(ax) \geq 1208 \text{ mm/s}^2$ • $\max(ay)-\min(ay) \geq 1168 \text{ mm/s}^2$ • $\max(az)-\min(az) \geq 472 \text{ mm/s}^2$

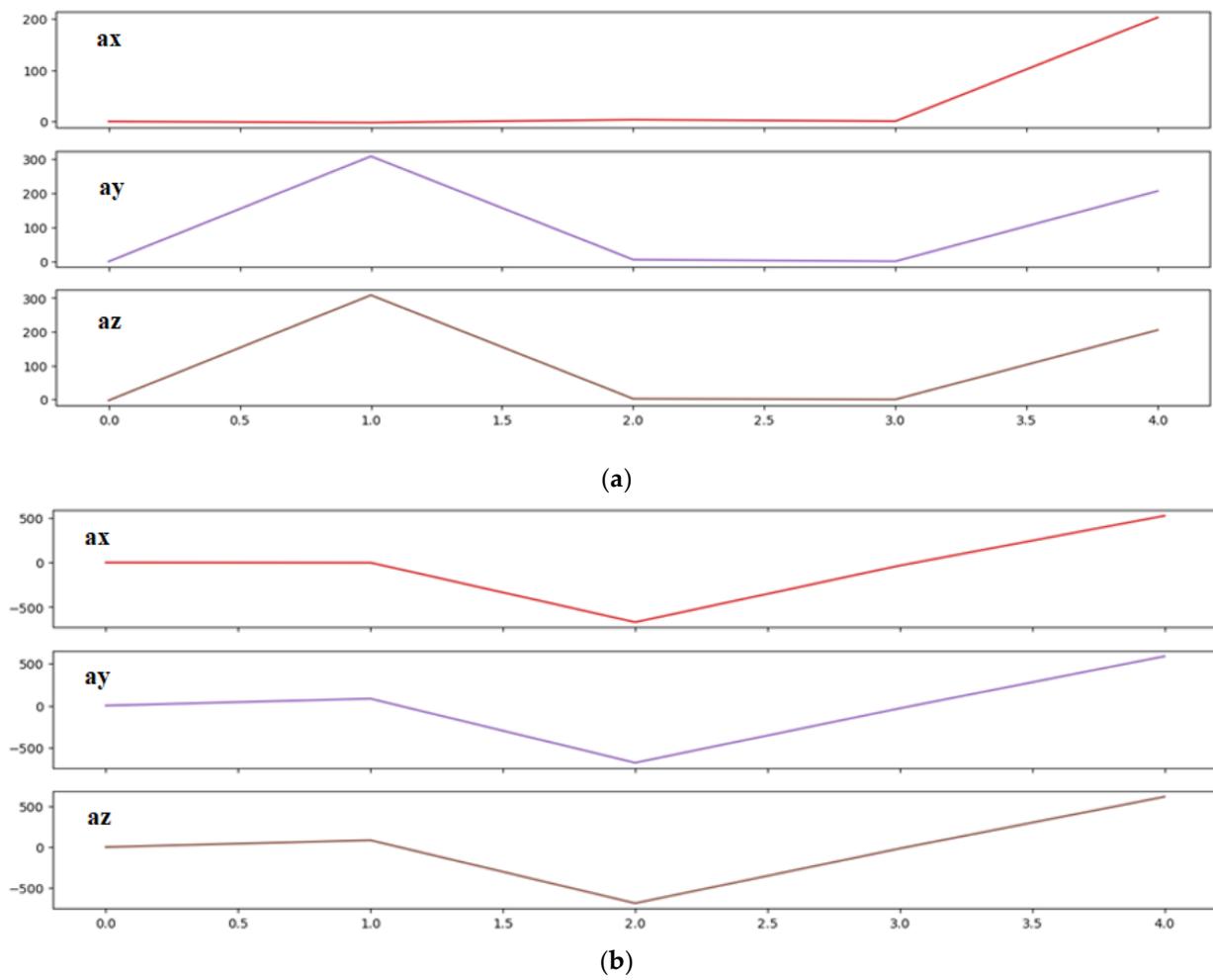


Figure 7. Accelerations (a_x , a_y , a_z) in the x , y , and z directions versus time domain. (a) Road pavement uneven—low risk. (b) Road pavement uneven—high risk.

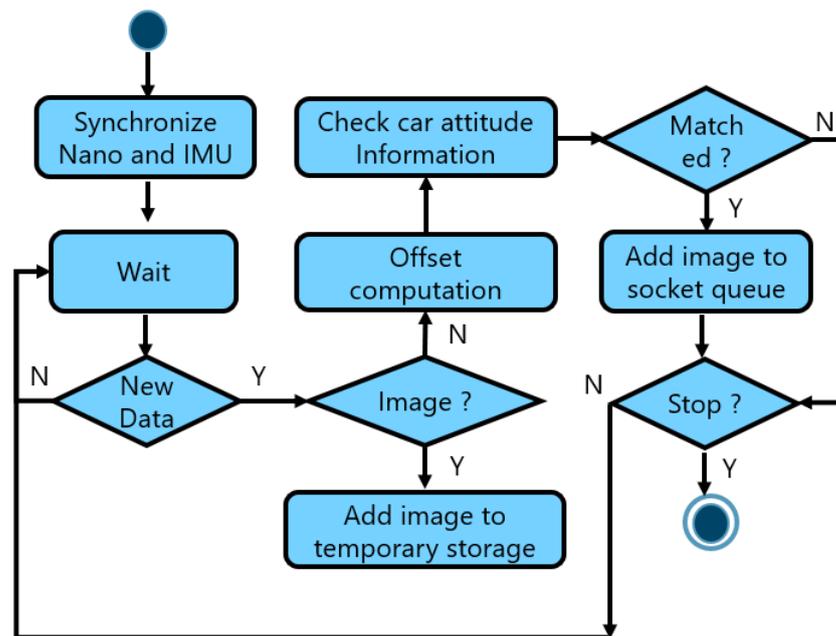


Figure 8. Information fusion threaded activity diagram.

3.3. Integration of Visual and Vibration Information

In this study, two types of information, visual and vibration, are used to recognize the damage of roadway pavement. Because the visual information is the image of the front of the vehicle, and there is a time difference between the actual vehicle body passing through the damage, an information integration thread is designed in the front-end information collection and transmission system to integrate the two types of information. In the information fusion thread, as shown in Figure 8, the system will compare the visual and vibration information by time and vehicle speed. The speed and time of the vibration information can be obtained by the inertial measurement unit, as shown in Figure 9, while the image acquisition time is based on the computer time. The two times are compared when the information fusion thread is activated to prevent the conversion error caused by the time difference to avoid the time difference between the computer and the inertial navigation hardware.

```

$AHRS,-1.628,-0.306,89.583,-104,42,-422,0,0,0
$GPRMC,065544.00,A,2504.0951,N,12131.3354,E,2.1,0.0,2805-80,0.0,E,N*1a
$AHRS,-1.629,-0.307,89.580,-112,44,-420,0,0,0
$GPRMC,065544.00,A,2504.0951,N,12131.3354,E,2.1,0.0,2805-80,0.0,E,N*1a
$AHRS,-1.629,-0.308,89.578,-110,34,-428,0,0,0
$GPRMC,065544.00,A,2504.0951,N,12131.3354,E,2.1,0.0,2805-80,0.0,E,N*1a
$AHRS,-1.629,-0.309,89.574,-102,27,-427,0,0,0
    
```

Figure 9. Information obtained by GPRMC.

Because the system integrates both visual and vibration information through moving distance, as in Equation (1), it is necessary to measure the relative distance of the front wheels from the markers in the image, and the measurement results are shown in Figure 10, which shows the relative position of the front wheels from the image. The numbers in the figure are the meters from the front wheel, and the data source is the actual measurement. Each image contains vibration information to cover the area within 18 m in front of the vehicle to facilitate the cloud server in comparing the pothole location and vehicle vibration data in the images. In Equation (1), v is the vehicle speed that GPRMC can obtain, and t is the time fixed to 0.5 because the sampling frequency is 30 Hz.

$$S = vt \tag{1}$$

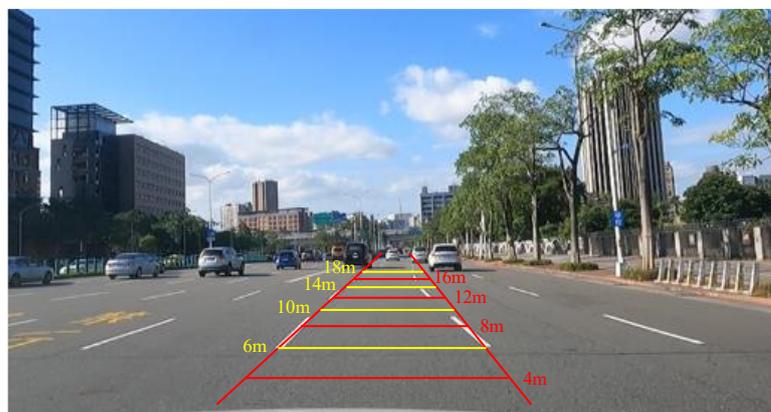


Figure 10. The relative position of the front wheels to the markers in our image.

The integrated information consists of images, file names, and vehicle vibration information, and the detailed data structure is shown in Figure 11. The file names are named according to the current time the information is acquired. The vehicle vibration information contains several pieces of information to cover the distance captured in all the images. The example in the figure comprises three pieces of information about the vehicle body vibration, which are arranged in the order of occurrence. When confirmed that the image has moved more than 18 m since it was captured, it is placed in the transmission thread and waits to be sent to the cloud server. After the cloud server receives the information and performs the first identification stage to confirm the damage category, the system will use Equations (2) and (3) to calculate the Bounding Box center point coordinates to determine how many meters the damage point is located. The displacement distance is then calculated based on the vehicle speed V_i in the GPRMC of the body vibration information, and the displacement distance is used to determine which body vibration information the pothole coordinates belong to, as shown in Equation (4). Then, we could evaluate i value in Equation (4) and locate its position.

$$x_{Center\ point} = \frac{x_0 + \left(\frac{bounding_box_width}{2}\right)}{2} \quad (2)$$

$$y_{Center\ point} = \frac{y_0 + \left(\frac{bounding_box_height}{2}\right)}{2} \quad (3)$$

$$\begin{cases} vehicle_vibration_i, & \text{if } y_{Center\ point} \leq \sum_{i=0}^m V_i \times \Delta T \\ m = m + 1, & \text{else} \end{cases} \quad (4)$$

```
Informantion:{
  Image:image
  Image_name:'2023.08.13.10.14.32.738.jpg'
  vehicle_vibration:[
    v1:{
      '$GPRMC':'',
      '$AHRS':''
    },
    v2:{
      '$GPRMC':'',
      '$AHRS':''
    },
    v3:{
      '$GPRMC':'',
      '$AHRS':''
    },
  ]
}
```

Figure 11. Integrated information format.

Take Figure 12 as an example; there are three pieces of body vibration information in the figure, and $i = 0 \sim m - 1$, where m is the number of vehicle vibration data. The system will use Equation (4) for the second identification stage. Calculate each piece of body vibration information individually and check whether the vehicle passes through the damaged position. Figure 13 depicts a pothole was detected between 6 m and 8 m.

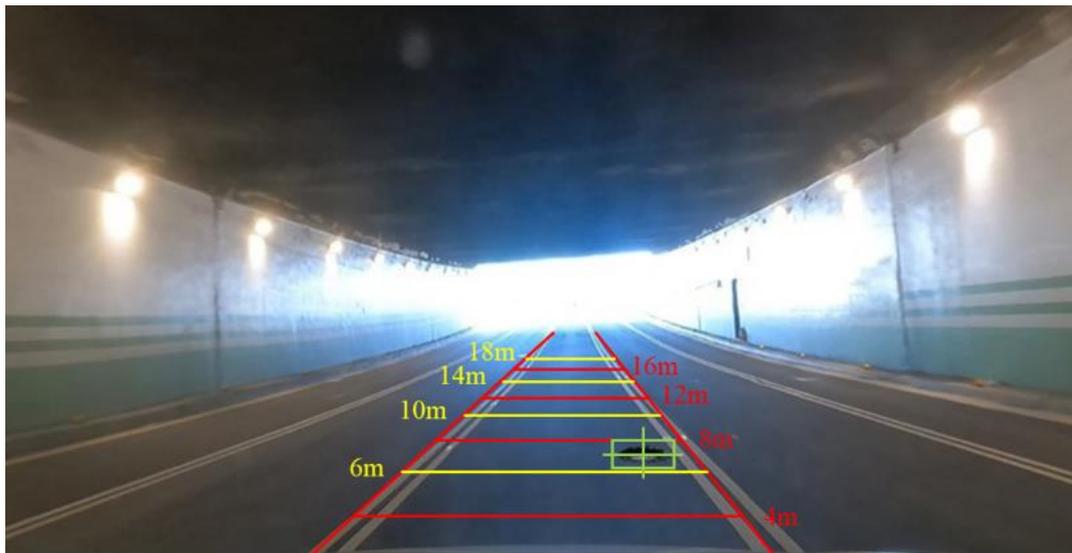


Figure 12. Schematic diagram of a pothole and the markings.

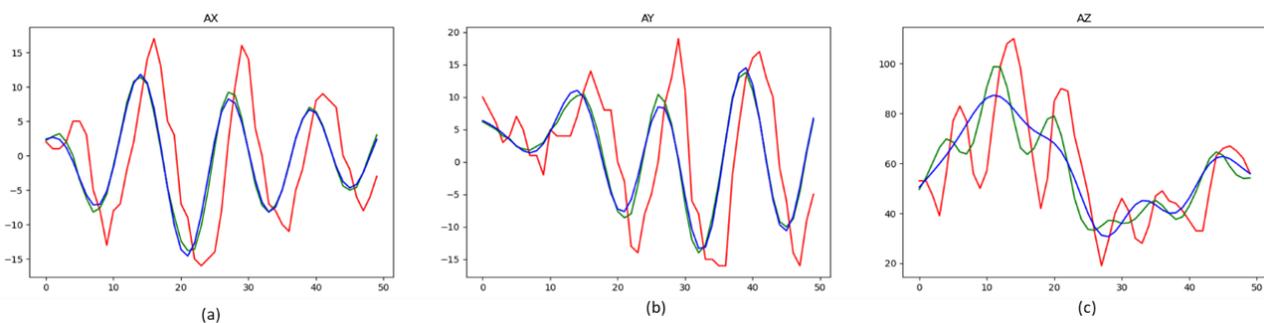


Figure 13. Comparison of the original (red curve) acceleration signals (a_x , a_y , a_z) after 1st stage pre-processing (green curve) and 2nd stage pre-processing (blue curve). (a) Acceleration signal of X. (b) Acceleration signal Y. (c) Acceleration signal Z.

4. Experimental Results and Analysis

In this section, the gyroscope signals are pre-processed and analyzed, and the training results of the latest neural networks on the same dataset are verified. We compare the differences between different neural networks and verify the feasibility of a pavement damage recognition system.

4.1. Three-Axis Acceleration and Vehicle Attitude Analysis

In this study, in addition to pothole identification using images, a gyroscope was used to collect the three-axis acceleration and attitude of the vehicle. After pre-processing the signals, the study analyzed whether the three-axis acceleration information conforms to the normal distribution. After analysis and consideration, it was decided to use a dataset other than two times the standard deviation as the negative sample. The vehicle attitude and the three-axis acceleration were recorded in this study, and the vehicle speed was kept at no more than 54 km/h to ensure that the collected data could cover the driving path completely.

4.1.1. Vehicle Information Pre-Processing

The vibration signals collected by the gyroscope can be categorized into signal and noise; signal includes standard and crack signals, and noise includes road and system noise. This study uses the moving average for the first stage of vibration signal processing to solve

the effect of high-frequency signals on pothole identification in collecting vehicle vibration information. Considering that five samples are one frame for subsequent recognition, it is decided to use a filter with a window size of 5 for smoothing to avoid over-smoothing to filter out the signal features. Then, a finite impulse response filter (FIR) is used as the second stage of signal processing to retain the low frequency and suppress the high-frequency features based on the characteristics of the FIR filter. The result of the two-stage signal pre-processing is shown in Figure 13.

4.1.2. Vehicle Vibration Information Analysis

After analyzing the data, it was confirmed that the three-axis acceleration data belonged to the normal distribution as in Figure 14. There was no linear correlation between the three-axis acceleration and the vehicle attitude. Therefore, the data outside of two standard deviations, as shown in Table 5, was chosen as the negative sample (5%) and within two standard deviations as the positive sample (95%).

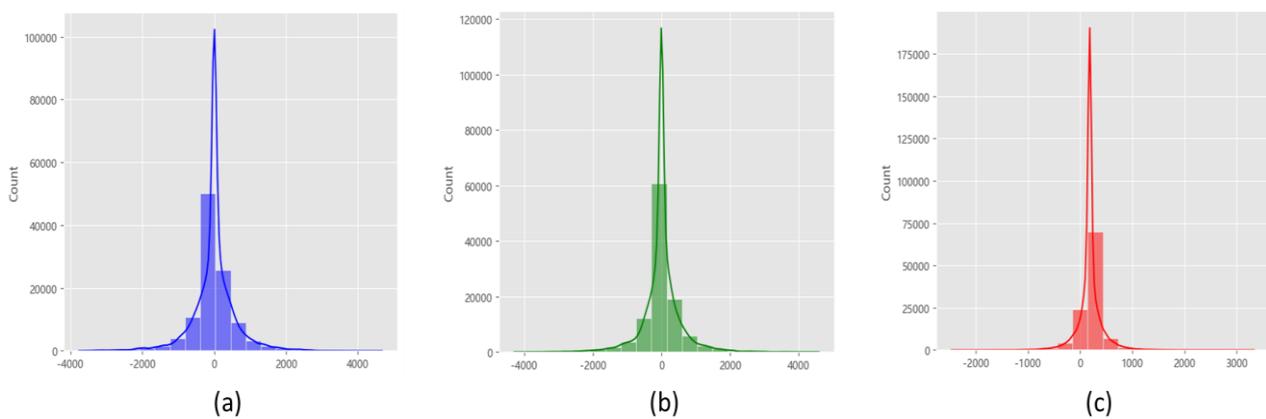


Figure 14. (a) *x*-axis acceleration distribution. (b) *y*-axis acceleration distribution. (c) *z*-axis acceleration distribution.

Table 5. The standard deviation of three-axis acceleration.

Name	Std (mm/s ²)
Ax	603.58
Ay	584.28
Az	236.27

After the above pre-processing, this study filters again, with the condition that the acceleration of both axes must be two times larger than the standard deviation to be regarded as road pavement unevenness. Because 5 data are one frame, to avoid being unable to fit the data in the subsequent training, the number of abnormal signals in the same frame must be greater than 3/5 of the number of signals. Some results of the categorization are shown in Figure 15.

4.2. Visual Neural Network Training

In this study, the training environment is a PC with a CPU Intel i9-12900K, GPU Nvidia RTX 3090, and RAM 64 GB. The object recognition neural network and the 1-D signal neural network were trained in the experimental process. The object recognition neural network consists of three types: YOLOv4, YOLOR, and YOLOv7. Precision, recall rate, and FPS are important indicators, so the confusion matrix must be derived first, which consists of true positive, false positive, true negative, and false negative. The reason for considering precision and recall rate in this study is to ensure recognition accuracy when the damage category can be determined. Precision represents the percentage of positive among

optimistic predictions ($TP/TP+FP$); the higher it is, the less misjudgment it represents. The recall rate represents the percentage of positives captured ($TP/TP+FN$); the higher the rate, the fewer missing cases are.

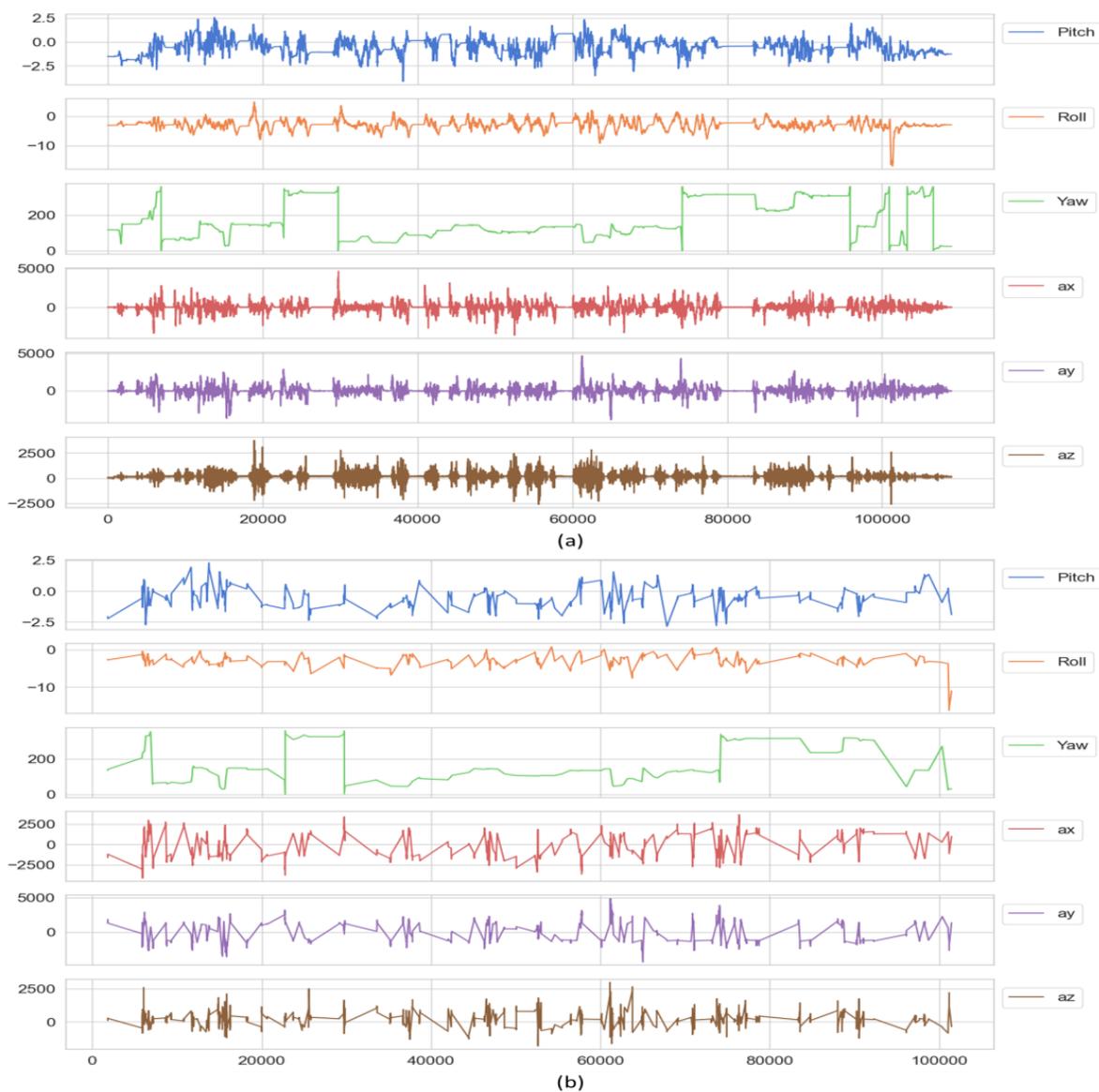


Figure 15. Pitch, yaw, roll, ax, ay, and az data versus time domain for (a) road pavement uneven—low risk. (b) Road pavement uneven—high risk.

The neural networks used in this training are YOLOv4/YOLOv4-tiny/YOLOR/YOLOv7, which take 8/6/4/8 h with 7500/12,000/1003/2000 iterations and a total of 882/10,828/10,828/10,828 images. The training results are shown in Tables 6 and 7. The main reason for the excellent performance of YOLOR compared with YOLOv4 is that YOLOR resizes image data during the training process. Because the datasets collected in this study are from different periods, and most of them are open-source datasets after post-processing, the size of the images varies, and the data augmentation during the training process of YOLOR can help the neural network to perform better.

Table 6. Performance comparison of the trained different YOLO versions.

Models	Precision	Recall	mAP	FPS
YOLOv4	87.6%	88.6%	89%	20~22
YOLOv4-tiny	86.6%	73.8%	87%	150~200
YOLOR	79.1%	92.6%	92.5%	40~50
YOLOv7	93.2%	87.8%	93.3%	30~40

Table 7. Confusion matrixes of different versions of YOLO.

Types\Models	TP	FP	FN
	v4/v4-tiny/R/v7	v4/v4-tiny/R/v7	v4/v4-tiny/R/v7
Pothole	47/50/50/45	7/3/14/0	5/2/2/7
Crack	31/15/32/32	4/7/8/5	5/21/4/4

We adjusted the horizontal flip possibility to 0.5 while training YOLOv7 and tested it with an independent test set. During the training process, it was found that the features learned by YOLOv7 were concentrated around potholes. Therefore, whenever there is a significant height difference in the image, it will be misclassified as a pothole. The pothole image types are too similar and have considerable height differences. In addition, the horizontal and vertical flips were turned on during training. During subsequent testing, it was found that turning on the vertical flip resulted in the misinterpretation of the streetlights as cracks. Therefore, the vertical flip was turned off during data augmentation.

Table 6 shows that YOLOv7 performs best among all neural networks in terms of mAP, recall rate, and precision. Although the processing speed of YOLOv4-tiny is much higher than YOLOv7, we choose to use YOLOv7 as the object detection model because the system only needs 30 FPS.

4.3. One-Dimensional Neural Network Training

We trained three kinds of 1-D neural networks: Simple RNN, LSTM, and Bi-LSTM. Because only two types of classes are used in 1-D neural networks in this study, we used Sigmoid as the output layer of the neural networks, classified the final outputs into two classes, and used binary cross entropy as the loss function. In addition, Relu is used as the activation function of the 1-D neural network, and the training speed is greatly improved by using Relu's high-speed computational property for the convergence of stochastic gradient descent.

A total of 4360 datasets were used in the training and were divided 8:2 into training and validation sets. The neural network first used for the training is a Simple RNN, which contains 512 units and achieves the best results in 343 iterations. The results of the training were loss: 0.05 and accuracy: 0.98. The Simple RNN was tested using a test set independent of the training and validation datasets, and the results are shown in Table 8. When using the test set for validation, it was found that there was overfitting during the training process, so the batch size was readjusted, and the dropout was set to 0.2. However, because Simple RNNs forget the first half of the data, it is easy to misjudge the signal when it occurs in the middle of the back of a frame. Other neural networks are also tested to solve the above problem.

Train the Long Short Term Memory neural network, which contains 512 units. Five hundred iterations were performed, and the best result was obtained at 442 iterations. The results of the training are loss: 0.07 and accuracy: 0.98. Because reducing LR (learning rate) on the plateau is set, and from the training results, it can be seen that when it is impossible to minimize the loss, adjusting the LR downward can effectively improve the learning results. The test results are shown in Tables 8 and 9. LSTM improves the inference speed by 25% compared with Simple RNN and has higher accuracy in recognizing abnormal signals

in the middle and late segments because LSTM can control whether to save the parameters through the gate.

Table 8. Performance comparison of different 1-D neural networks.

Metrics	Precision	Recall	Accuracy	FPS
Types\Models	RNN/LSTM/Bi-LSTM	RNN/LSTM/Bi-LSTM	RNN/LSTM/Bi-LSTM	RNN/LSTM/Bi-LSTM
Pavement uneven—low risk	99%/98.6%/99.1%	99.3%/99.4%/99.6%	98.9%/98.1%/98.7%	
Pavement uneven—high risk	77.2%/71.4%/80.4%	82.9%/85.3%/90.2%	66.6%/63.6%/74%	
Average	88.1%/85%/89.75%	91.1%/92.35%/94.9%	82.75%/80.85%/86.35%	25/33.3/40

Table 9. Confusion matrix for different 1-D neural networks.

	TP	FP	FN
Types\Models	RNN/LSTM/Bi-LSTM	RNN/LSTM/Bi-LSTM	RNN/LSTM/Bi-LSTM
Pavement uneven-low risk	1039/1034/1040	7/6/4	10/14/9
Pavement uneven-high risk	34/35/37	10/14/9	7/6/4

The neural network Bi-LSTM is trained with 512 units, 200 iterations, and the best result is obtained at 175 iterations. The training set results are loss: 0.06 and accuracy: 0.99. Compared to Simple RNN and LSTM, Bi-LSTM requires less than half of the number of iterations to achieve the same training results, and the inference speed is improved by 38% compared with Simple RNN and 17% compared with LSTM. The results after testing are shown in Tables 8 and 9. Bi-LSTM can learn the features faster compared with LSTM, and due to the possibility of inference in the reverse direction, it is more capable of extracting and remembering the features of abnormal vibrations compared with Simple RNN, which will forget the past inputs and the unidirectional inference of LSTM.

Table 8 shows that Bi-LSTM has a vast improvement compared with Simple RNN and LSTM, and the inference speed is increased by 38% compared with Simple RNN and 17% compared with LSTM. Therefore, Bi-LSTM was chosen as the RNN in this study's pavement vibration recognition system.

4.4. Field Test Results

This study uses the image as the primary method and the vehicle vibration information as the supplementary method to detect pavement damage. When the cloud server receives the image, it will recognize it first, and the first recognition stage will be carried out first. Suppose the damaged object is identified in the first stage. In that case, the system will determine the distance between the damaged location and the vehicle according to the center point coordinates of the damaged object. Then, the vehicle speed determines the corresponding vibration information, and Bi-LSTM is used to recognize the information. The information is judged damaged when the second recognition stage results in an uneven road surface-high risk.

The vehicle used in this study was a Toyota Corolla Crossover. A GoPro was used to capture the front image of the car. An inertial navigation module provided by Thinkstar was used to measure the acceleration and attitude of the vehicle. Then, an Nvidia Jetson Nano was used as the computer equipment for the information collection and transmission.

In the first test, the data from Banqiao to Shulin were recorded on 23 November 2022 for the field test, and the images and vehicle body information were transmitted via a 5G mobile network. The pictures and vehicle vibration information were recorded simultaneously, and the recall was 83.3%, and precision was 84.0%. We also tested the stability and recognition speed of the system. After 2 h of testing, the average transmission

and recognition speed was 54 FPS, which could transmit all the data from the front end to the back end. The system can handle speeds up to 54 km per hour, and the inspector can view the recognition results and real-time information via real-time streaming.

The second test was conducted on 28 November 2022 on the urban road from Banqiao District, New Taipei City, to Daan District, Taipei City. The results are shown in Table 10, including pavement image pothole detection, vehicle attitude information for pavement damage detection, and integrated pavement damage detection results. Although the precision of integrated detection is 75.6%, the recall is 86%, which means the most actual pavement defects would be picked out.

Table 10. The second field test results of the proposed pavement damage detection system.

		Recall	Precision
Pavement image	Pothole	95.0%	92.8%
	Crack	92.5%	91.9%
Vehicle attitude	Uneven—low risk	99.4%	99.1%
	Uneven—high risk	93.0%	82.3%
Integrated detection	Pothole—high risk	94.4%	91.9%
	Crack—high risk	86.0%	75.6%

4.5. Comparison with State-of-the-Art

There are four known road pavement inspection approaches: ultrasonic [10], LiDAR [11], thermal [12], and camera [9]. The advantages of the first three approaches are that they have higher accuracy and can measure the depth of potholes and cracks. Our vehicle's IMU information recognition has lower accuracy but costs much less than the others. However, the IMU information is highly related to the road pavement where the vehicle's wheel is contacted. The accuracy would be higher if vehicle wheels could cover most road pavements. There is a vehicle designed to have many wheels installed across a whole lane. Then, the vibration would be measured to achieve pavement defect detection. A comparison of road pavement inspection approaches is shown in Table 11. Some closely related YOLO base damage detection models from images are also listed. Because only F1 scores are given in [32,33], Equation (5) is used to describe it as the harmonic mean of the precision and recall. The two metrics contribute equally to the score, ensuring that the F1 metric correctly indicates the reliability of a model. Based on the performance index in terms of precision and recall, our YOLOv7+Bi-LSTM has excellent performance, while FPS may not have the advantage. It demonstrated the feasibility of the proposed methods.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Table 11. Comparison of road pavement inspection approaches.

Items	Sensors	Precision/Recall	Classification	Defect Types	FPS
[9]	Line Camera	98.29%/93.86%	SVM	Ten types	NA
[10]	Ultrasonic	Error < 7%	FFT	Crack depth	NA
[11]	LiDAR	70%/73.9%	GCN	Crack	NA
[12]	Thermal imaging	Accuracy 97.08%	CNN-ResNet	Pothole	NA
[31]	Smartphone	65%/55%	YOLO4 Tiny	Four types of cracks	NA
[32]	Smartphone	F1 score = 51.9%	Scaled YOLOv4	Four types of cracks	73.8
[33]	Smartphone	F1 score = 58.4%	YOLOv5x	Four types of cracks	33.3
[34]	CCD+LED	83.38%/90.45%	YOLOv5s	Cracks/sealed cracks	67.5
[30]	CCD	78.2%/72.1%	YOLOv5s-M	Seven types	42
Ours	Camera +IMU	93.3%/86.35%	YOLOv7+Bi-LSTM	Pothole/crack	>30

5. Conclusions and Future Works

5.1. Conclusions

This study uses the vehicle body vibration information and the front image to form the pavement damage recognition system. Precision is more critical than recall to avoid misjudgment and increase inspectors' workload. Therefore, YOLOv7, which has the highest precision, was chosen as the neural network model for image recognition. As for the second stage of assisted judgment, after comparing the Simple-RNN, LSTM, and Bi-LSTM neural networks, we finally decided to use the better Bi-LSTM for road pavement damage detection.

In this study, a GoPro camera, an IMU, and a Nvidia Jetson Nano are used to collect the vehicle vibration and the front image of the vehicle, which are then transmitted to the back-end server via a 5G mobile network for recognition. Parallel processing in both the front and back end separates the asynchronous steps, and MongoDB stores the detection results to improve the system's efficiency. The cloud-based identification system also provides a real-time streaming service so that inspectors can check the current inspection results through real-time streaming to facilitate the inspectors' checking of the inspection results.

Regarding object recognition, the accuracy and recall rate of cracks cannot simultaneously reach more than 90%. The main reason is that cracks appear around potholes, and cracks often appear regionally, so when the end of the cracks connects, it is easy to lead to misjudgment of the neural network, which leads to a decrease in the accuracy and recall rate. Although the use of cameras with body vibration for road pavement inspection is not widespread at this stage, considering the increasing proportion of vehicles equipped with cameras, gyroscopes, and communication equipment, it may be possible to save inspection costs and improve the timeliness of repairs in the future by using civilian vehicles to report on road conditions.

5.2. Future Works

Currently, fewer defects are used in this study, with only cracks and potholes in the image and no risk or high risk of body vibration. In the future, the risk level should be further divided, and the completion time of repair work should be divided according to the different levels. In the future, we should further categorize the risk level and the completion time of repair work according to the different levels. We should also increase the number of categories affecting the vibration, such as maintenance hole covers, pavement patches, etc., to help inspectors filter out unnecessary information more quickly.

Due to the limitation of gyroscope hardware, the current inspection speed cannot reach the goal of highway inspection. In the future, the gyroscope can be replaced to improve the sampling rate so that the system can have more vibration signals in a frame to be analyzed. Incorporating additional features can enhance identification accuracy, expedite inspection speed, and bolster overall inspection efficiency.

Author Contributions: Methodology, C.-C.H. software, H.-W.J.; validation, C.-C.H.; data curation, H.-W.J.; writing—original draft preparation, H.-W.J.; writing—review and editing, C.-C.H.; supervision, W.-H.H. and M.-H.H.; project administration, W.-H.H.; funding acquisition, W.-H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

- National Statistics, R.O.C. (Taiwan). Statistics on the Length of Roads in China in the Past Ten Years. 20 September 2022. Available online: <https://statdb.dgbas.gov.tw/pxweb/Dialog/View.asp> (accessed on 24 December 2022).
- Freeway Bureau, R.O.C. (Taiwan). Million Vehicle Kilometer Statistics. 13 December 2022. Available online: <https://www.freeway.gov.tw/Publish.aspx?cnid=1656&p=26767> (accessed on 24 December 2022).
- Tamkang University. Pavement Engineering. 13 December 2022. Available online: <https://mail.tku.edu.tw/yinghair/lee/te/pdf> (accessed on 24 December 2022).
- Zakeri, H.; Nejad, F.M.; Fahimifar, A. Image-Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review. *Arch. Computat. Methods Eng.* **2017**, *24*, 935–977. [[CrossRef](#)]
- Kheradmandi, N.; Mehranfar, V. A Critical Review and Comparative Study on Image Segmentation-based Techniques for Pavement Crack Detection. *Constr. Build. Mater.* **2022**, *321*, 126162. [[CrossRef](#)]
- Gong, H.; Liu, L.; Liang, H.; Zhou, Y.; Cong, L. A State-of-the-Art Survey of Deep Learning Models for Automated Pavement Crack Segmentation. *Int. J. Transp. Sci. Technol.* **2024**, *13*, 44–57. [[CrossRef](#)]
- Qiu, J.Y. Research on Detection Based on Improved Mask RCNN Algorithms. Master's Thesis, Department of Information and Communication Engineering, Chaoyang University, Taiwan, China, 2020.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [[CrossRef](#)]
- 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](#)] [[PubMed](#)]
- Her, S.-C.; Lin, S.-T. Non-Destructive Evaluation of Depth of Surface Cracks Using Ultrasonic Frequency Analysis. *Sensors* **2014**, *14*, 17146–17158. [[CrossRef](#)] [[PubMed](#)]
- Feng, H.; Li, W.; Luo, Z.; Chen, Y.; Fatholahi, S.N.; Cheng, M.; Wang, C.; Junior, J.M.; Li, J. GCN-Based Pavement Crack Detection Using Mobile LiDAR Point Clouds. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11052–11061. [[CrossRef](#)]
- Aparna, Y.; Yukti, B.; Rachna, R.; Varun, G.; Naveen, A.; Aparna, A. Convolutional Neural Networks based Potholes Detection Using Thermal Imaging. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 578–588. [[CrossRef](#)]
- Shenu, P.M.; Soumya, J. Automated Detection of Dry and Water-Filled Potholes Using Multimodal Sensing System. Doctoral Dissertation, Indian Institute of Technology Hyderabad, Kandi, Indian, 2012.
- Nienaber, S.; Booyen, M.; Kroon, R. Detecting Potholes Using Simple Image Processing Techniques and Real-World Footage. In Proceedings of the 34th Annual Southern African Transport Conference, Pretoria, South Africa, 6–9 July 2015.
- Varadharajan, S.; Jose, S.; Sharma, K.; Wander, L.; Mertz, C. Vision for Road Inspection. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March 2014; pp. 115–122. [[CrossRef](#)]
- Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunk, S. *SLIC Superpixels*; EPFL Technical Report 149300; EPFL: Lausanne, Switzerland, 2010.
- Ardeshir, N.; Sanford, C.; Hsu, D. Support Vector Machines and Linear Regression Coincide with Very High-Dimensional Features. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 4907–4918.
- Japan International Cooperation Agency. Pavement Inspection Guideline. Ministry of Transport, 13 December 2016. Available online: https://openjicareport.jica.go.jp/pdf/12286001_01.pdf (accessed on 24 December 2022).
- Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
- Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2004**, arXiv:2004.10934.
- Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787. [[CrossRef](#)]
- Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
- Liu, H.; Sun, F.; Gu, J.; Deng, L. SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors* **2022**, *22*, 5817. [[CrossRef](#)] [[PubMed](#)]
- Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737. [[CrossRef](#)]
- Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872.
- Wang, C.; Yeh, I.; Liao, H. You Only Learn One Representation: Unified Network for Multiple Tasks. *J. Inf. Sci. Eng.* **2021**, *39*, 691–709.
- Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An Evolved Version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
- Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
- Wang, C.; Bochkovskiy, A.; Liao, H.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13024–13033.

30. Ren, M.; Zhang, X.F.; Chen, X.; Zhou, B.; Feng, Z. YOLOv5s-M: A Deep Learning Network Model for Road Pavement Damage Detection from Urban Street-View Imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *120*, 103335. [[CrossRef](#)]
31. Zhou, Y.; Wei, Y.; Chen, J. Improved YOLOv4-Tiny Lightweight Country Road Pavement Damage Detection Algorithm. In Proceedings of the 2022 2nd International Conference on Algorithms, High-Performance Computing and Artificial Intelligence (AHPCAI), Guangzhou, China, 21–23 October 2022; pp. 160–163. [[CrossRef](#)]
32. Fassmeyer, P.; Kortmann, F.; Drews, P.; Funk, B. Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 27–30 September 2021; IEEE: New York, NY, USA, 2021; pp. 1–7.
33. Jeong, D. Road Damage Detection Using YOLO with Smartphone Images. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: New York, NY, USA, 2020; pp. 5559–5562.
34. Yang, N.; Li, Y.; Ma, R. An Efficient Method for Detecting Asphalt Pavement Cracks and Sealed Cracks Based on a Deep Data-Driven Model. *Appl. Sci.* **2022**, *12*, 10089. [[CrossRef](#)]
35. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *arXiv* **2018**, arXiv:1808.03314. [[CrossRef](#)]
36. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* **2019**, arXiv:1909.09586.
37. Schuster, A.M.; Paliwal, K.K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
38. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
39. goprotaiwancsl.com.tw. GoPro. Available online: <https://www.goprotaiwancsl.com.tw/> (accessed on 24 December 2022).
40. Newstar. Newstar AHRS Series Attitude and Heading Chip AH8. Available online: <https://www.facebook.com/watch/?v=290400158530629> (accessed on 29 February 2024).
41. Docs.novatel.com. GPRMC. 23 July 2023. Available online: <https://docs.novatel.com/OEM7/Content/Logs/GPRMC.htm> (accessed on 29 February 2024).
42. Mathworks. Attitude and Heading Reference System. 2023. Available online: <https://www.mathworks.com/help/nav/ref/ahrs.html> (accessed on 29 February 2024).

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.