sustainability

Article

Pavement Distress Identification Based on Computer Vision and Controller Area Network (CAN) Sensor Models

Cuthbert Ruseruka [1,*], Judith Mwakalonge [1], Gurcan Comert [2], Saidi Siuhi [1], Frank Ngeni [1] and Kristin Major [2]

[1] Department of Engineering, South Carolina State University, Orangeburg, SC 29117, USA; jmwakalo@scsu.edu (J.M.); ssiuhi@scsu.edu (S.S.); fngeni@scsu.edu (F.N.)

[2] Computer Science, Physics, and Engineering Department, Benedict College, 1600 Harden St, Columbia, SC 29204, USA; gurcan.comert@benedict.edu (G.C.); kristin.major50@my.benedict.edu (K.M.)

* Correspondence: cruseruk@scsu.edu

Abstract: Recent technological developments have attracted the use of machine learning technologies and sensors in various pavement maintenance and rehabilitation studies. To avoid excessive road damages, which cause high road maintenance costs, reduced mobility, vehicle damages, and safety concerns, the periodic maintenance of roads is necessary. As part of maintenance works, road pavement conditions should be monitored continuously. This monitoring is possible using modern distress detection methods that are simple to use, comparatively cheap, less labor-intensive, faster, safer, and able to provide data on a real-time basis. This paper proposed and developed two models: computer vision and sensor-based. The computer vision model was developed using the You Only Look Once (YOLOv5) algorithm for detecting and classifying pavement distresses into nine classes. The sensor-based model combined eight Controller Area Network (CAN) bus sensors available in most new vehicles to predict pavement distress. This research employed an extreme gradient boosting model (XGBoost) to train the sensor-based model. The results showed that the model achieved 98.42% and 97.99% area under the curve (AUC) metrics for training and validation datasets, respectively. The computer vision model attained an accuracy of 81.28% and an F1-score of 76.40%, which agree with past studies. The results indicated that both computer vision and sensor-based models proved highly efficient in predicting pavement distress and can be used to complement each other. Overall, computer vision and sensor-based tools provide cheap and practical road condition monitoring compared to traditional manual instruments.

Keywords: pavement maintenance; XGBoost; CAN sensors in roads condition; YOLOv5; sensor-based model; pavement condition monitoring; Deep Learning models for road condition monitoring

check for updates

Citation: Ruseruka, C.; Mwakalonge, J.; Comert, G.; Siuhi, S.; Ngeni, F.; Major, K. Pavement Distress Identification Based on Computer Vision and Controller Area Network (CAN) Sensor Models. *Sustainability* **2023**, *15*, 6438. https://doi.org/10.3390/su15086438

Academic Editors: Lev Khazanovich and Sushobhan Sen

Received: 9 February 2023
Revised: 28 March 2023
Accepted: 30 March 2023
Published: 10 April 2023

1. Introduction

Road condition monitoring involves routinely surveying the road surface, identifying roadway deficiencies, and proposing corrective priorities. It continuously monitors the road to ensure that it provides a safe and smooth riding experience to the passenger and causes less damage to the vehicles [1]. A timely and well-planned road condition assessment can reduce roadway maintenance and operational costs. For instance, the total maintenance costs of paved roads are estimated at 2–3% of initial investment costs [2]. However, delays in maintenance cause the costs to increase with time [3].

Sahin et al. [4] pointed out five steps for road maintenance and rehabilitation (M&R): network inventory, condition assessment, needs analysis, project prioritization, and impact analysis. The road conditions monitoring process as a part of M&R ensures that road distresses are identified and addressed prevent further deterioration. Feldman et al. [5] classified the current road conditions monitoring processes into manual and automated categories.

Manual road conditions monitoring involves qualified personnel using traditional survey forms and walking along the roads to visually check, measure, and record the observed distress [6]. This method is labor-intensive, time-consuming, costly, and creates safety concerns for the surveyors who perform it during the daytime when traffic flows [7]. Sometimes, either one or more lanes must be closed to improve this method's safety. However, this brings about another shortcoming, the disruption of traffic flow.

Automated methods that have been used to date involve using special vehicles equipped with special sensors to detect and capture road defects [5]. These methods have advantages over manual techniques, such as being less dependent on human labor and rapid operation, ensuring timely reporting of defects and improved safety. However, these methods are expensive and cost authorities a lot of money [8].

Researchers have recently conducted several studies using machine learning (ML) approaches to provide less expensive and highly efficient road condition monitoring approaches [9]. These approaches involve various ML models developed either to only detect or to detect and classify pavement defects into multiple categories. The developed models are either computer vision-based or sensor-based (vibration-based).

The identification of pavement distresses using machine learning (ML) has the potential to contribute to sustainability in the construction and maintenance of pavements. ML algorithms can effectively and efficiently detect and classify pavement distresses, leading to targeted and prioritized maintenance, preventive maintenance, and improved quality control. This can reduce the overall cost of and resources needed for maintenance and the extension of pavement lifespan, thereby reducing the need for reconstruction and the environmental impact of raw material extraction. Furthermore, extending pavement lifespan through effective maintenance can contribute to reducing carbon emissions associated with pavement construction and maintenance.

This paper aims to prepare a computer vision model for detecting and classifying pavement distresses based on Deep Learning (DL) and compare its performance to the sensor-based model. The paper also aims to combine eight Controller Area Network (CAN) bus sensors to develop a sensor-based model that predicts the presence of pavement defects once vehicles equipped with these sensors are driven over said defects. The two models can be used by road authorities to automate the process of collecting road condition data. The sensors' model can complement the vision-based model in adverse weather conditions, where computer vision is ineffective.

The remainder of this paper is organized as follows: Section 2 provides a summary of recent studies that applied machine learning approaches to prepare models for road condition monitoring, Section 3 discusses the source of data used, model selection, model training, and results, and Section 4 presents the conclusion and conclusion remarks.

2. Literature Review

In literature, various studies have conducted research to detect and classify pavement distress using different DL methods based on computer vision and vibrations (sensors). Computer vision-based models involve the use of images in model training. For instance, Wang et al. [10] used 5000 images to develop a convolutional neural network (CNN) model for detecting cracks in asphalt pavements. In this study, the trained model achieved an accuracy of 96.32 and 94.29% on training and testing data, respectively. Similarly, Kim et al. [11] developed an AlexNet CNN model trained for crack detection with images scraped from the internet and achieved precision and recall values greater than 90%. The model also detected cracks from real-time video with 81% and 88% recall and precision, respectively. CrackNet CNN was employed by Zhang et al. [12] in developing a model for automated pixel-level pavement crack detection using 1800 three-dimensional (3D) images. The testing of 200 3D images showed that the model achieved 90.13%, 87.63%, and 88.86% values in precision, recall, and F1-score, respectively. Also, Zhang et al. [13] developed a model for automated pixel-level crack detection on 3D asphalt pavement surfaces using CrackNet-R recurrent neural network (RNN). The model was trained on 3000 3D images and tested on 500 3D images. The testing results showed that the model achieved 88.89% precision, 95.00% recall, and 91.84% F1-score.

Also, Maeda et al. [14] developed a vision-based DL model based on a Single Shot MultiBox Detector (SSD) algorithm using 9053 road damage images captured using a smartphone installed on a car dashboard. The model achieved recall and precision values of more than 71% and 77%, respectively. Similarly, Maeda et al. [15] investigated road damage detection using artificial images in DL models developed by generative models, such as a generative adversarial network (GAN). The results showed that the F1-score of the model improved by 2% and 5% when the proportion of original images was small and large, respectively.

Other researchers developed sensor-based models using various sensors to predict the presence of defects on road surfaces. Aleadelat et al. [16] used smartphone accelerometers to determine International Roughness Index (IRI) and achieved an adjusted R^2 of 0.8. Souza et al. [17] used sensor data collected from smartphone accelerometers and complexity invariant distance to develop an ML model. The model achieved a classification accuracy of 80% to 98%. Similarly, Christodoulou et al. [18] used vibration sensors to detect pavement patch defects using smartphone images. The results showed that the vibration-based approaches were efficient; however, they failed to cover the entire roadway and could not detect non-vibration-induced defects. To address this observed shortcoming, a vision model was developed instead. Also, Sandamal et al. [19] used onboard diagnostic devices and smartphone sensors to develop a low-cost road condition monitoring system for detecting road potholes. The system confirmed smartphone sensor data to be effective in the prediction of potholes.

In her study, Pomoni [20] explored the use of smart tires in vehicles to detect the tire–road friction. The review of 105 references revealed how different sensors can be embedded in vehicle tires and assist in the detection of road surface conditions to enhance driver comfort.

Ameddah et al. [1] developed a model using smartphone sensors based on k-means clustering algorithms and achieved 88.67% accuracy in real-time road pavement monitoring. In another study, Ahmed et al. [21] used Traffic Speed Deflectometer (TSD) data to predict pavement structural conditions using Random Forest, XGBoost, and logistic regression models. The models achieved 65%, 69%, and 57% accuracy, respectively. Lekshmipathy et al. [22] compared the performance of vibration-based and vision-based approaches for automated distress detection using ML. This study employed a vibration-based method using a smartphone accelerometer and gyroscope as well as a vision-based method using video processing. The developed models achieved 80% and 84% accuracy for the vibration-based and vision-based models, respectively. Results were validated manually on-site and revealed that the first approach is sufficient for routine monitoring purposes while the latter is more appropriate for detailed analysis.

In summary, the literature review revealed that no past studies had used CAN bus sensors to predict pavement distress. This study aimed at preparing a pavement detection model based on CAN bus sensors and compares the performance of said model with the computer vision-based model. The vision-based model is prepared based on the YOLOv5 algorithm.

3. Methodology

This section explains the methods used in data collection, processing, model selection, model preparation, analysis, and evaluation of results. These methods are summarized in Figure 1 and discussed in more detail in the following subsections.

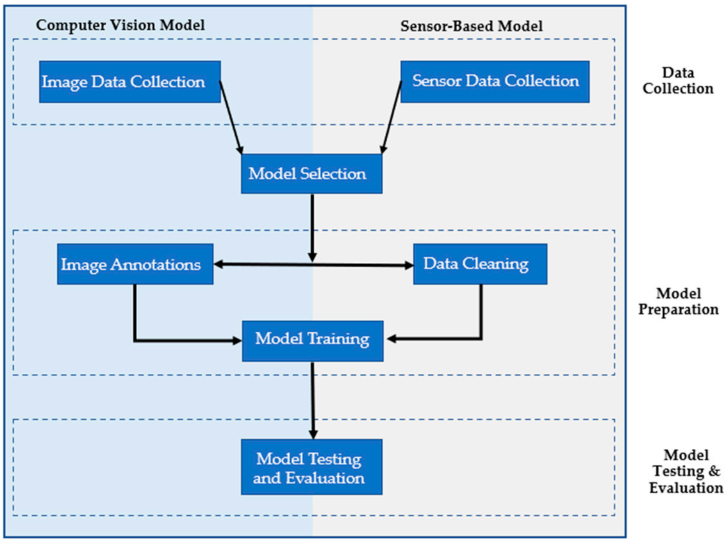


Figure 1. Flow chart of research methodology.

3.1. Data Collection

3.1.1. Vision-Based Data

Image datasets used in this study were collected from various sources on the internet to train the vision-based model. The datasets contained images of road pavement surfaces collected from multiple countries, including the United States, Japan, India, and the Czech Republic. The German Asphalt Pavement Distress (GAPs) dataset, which includes a total of 1969 gray-valued images [23], and the CRACK500 dataset, which consists of 500 Red–Green–Blue (RGB) images of pavement cracks approximately 2000 × 1500 pixels in size that were collected on the main campus of Temple University using cell phones were used in this study [24]. In addition, the Road Damage dataset was used, which consists of 9053 labeled road images 600 × 600 pixels in size, was acquired from a smartphone camera installed on the dashboard of a car [14]. This paper randomly selected 3500 images from these datasets using an excel spreadsheet with the "RAND" command. The final dataset was obtained after the excel spreadsheet was randomized three times. This dataset was then divided into 80% and 20% ratios for model training and validation, respectively, and 350 images (10% of the image dataset) were added to the training dataset as background images to reduce the effect of False Positives (FPs) [25].

3.1.2. Sensor-Based Data

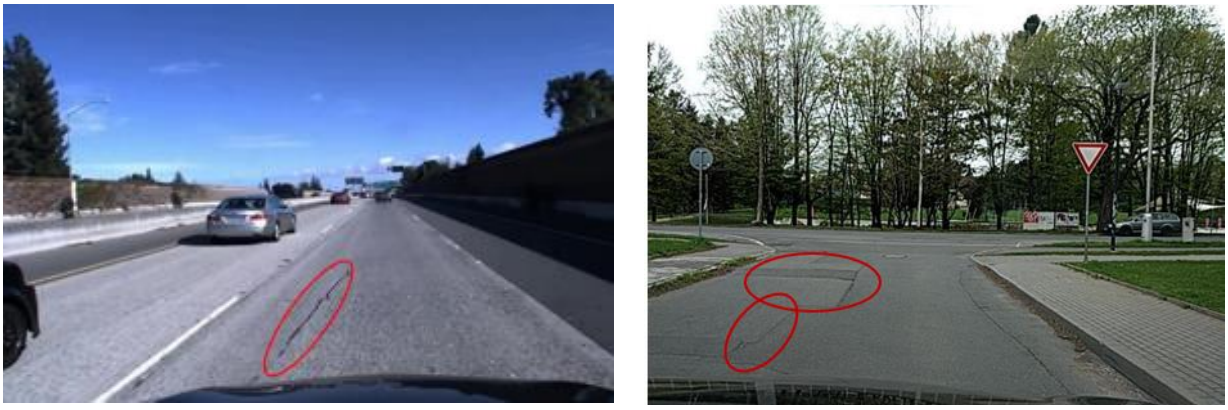
The sensor-based dataset was collected from the American Honda Motor Co., Inc. It was extracted from a data collection called 'Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning' by Ramanishka et al. [26]. The dataset includes 104 hours of actual human driving in the San Francisco Bay Area collected using an instrumented vehicle equipped with different sensors with driving speeds ranging from 0 mph to 120 mph. The dataset comprises video and sensor readings recorded on various road sections for all types of roads (based on functionality).

Videos

This study selected five videos from the dataset to represent all speeds ranging from 0 mph to 120 mph and all road classes. These videos were used to generate frames (images) for model testing. Figure 2 below shows sample images with different types of road surface distress.

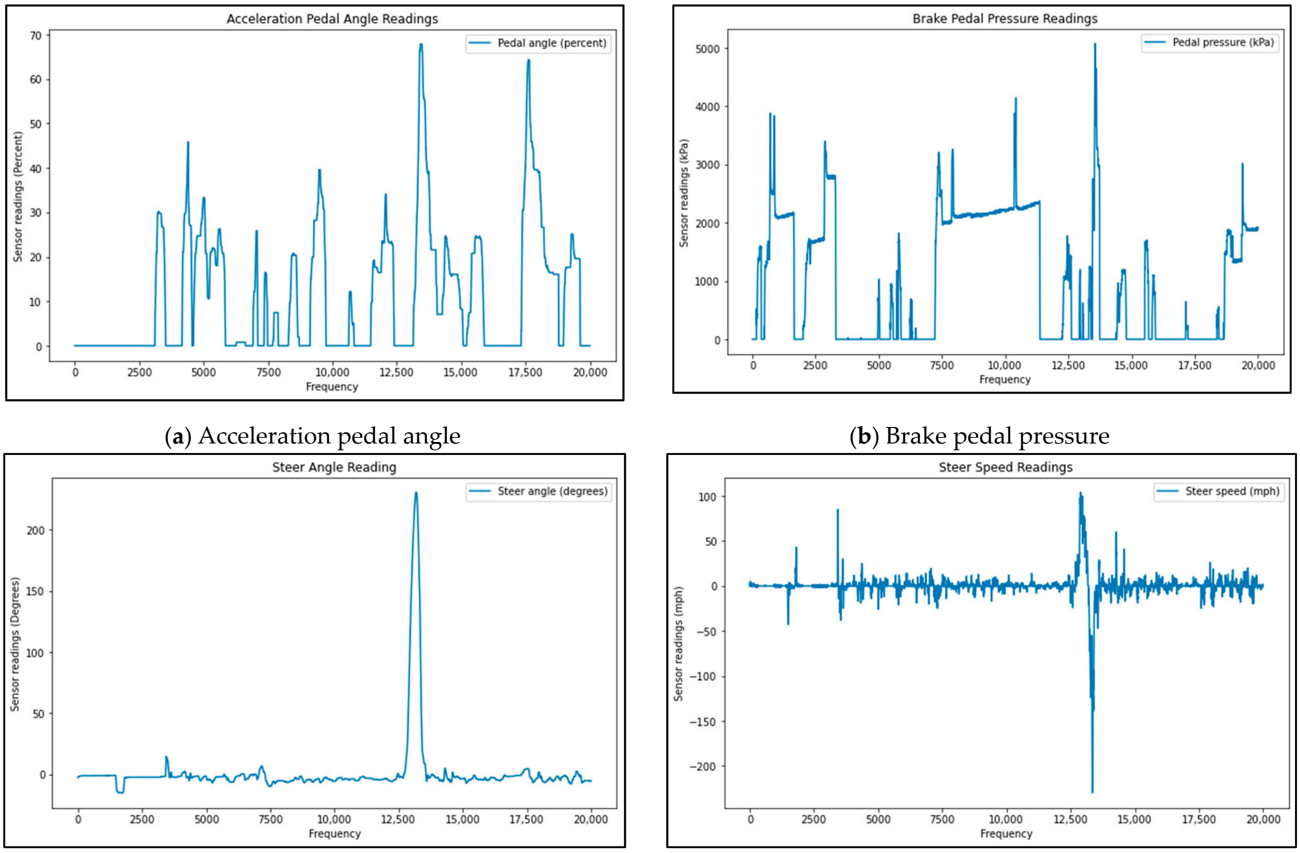
Sensors

For every video, there was a set of nine sensor readings recorded. The sensor recordings include iso-time stamp, real-time kinematic (RTK) position, real-time kinematic (RTK) track, acceleration pedal angle, brake pedal, turn signal (left turn & right turn), steer (steer angle & steer speed), speed, and yaw. These readings were recorded using Controller Area Network (CAN) bus sensors. The first three readings were not used in this study since they are not affected by the road surface condition. Figure 3 presents sample plots for the remaining sensor readings. The plots are from the first 20,000 (out of 265,000 readings associated with the five sample videos) readings (y-axis) plotted against the frequencies (x-axis).



(**a**) Longitudinal cracks (**b**) Longitudinal cracks and patches

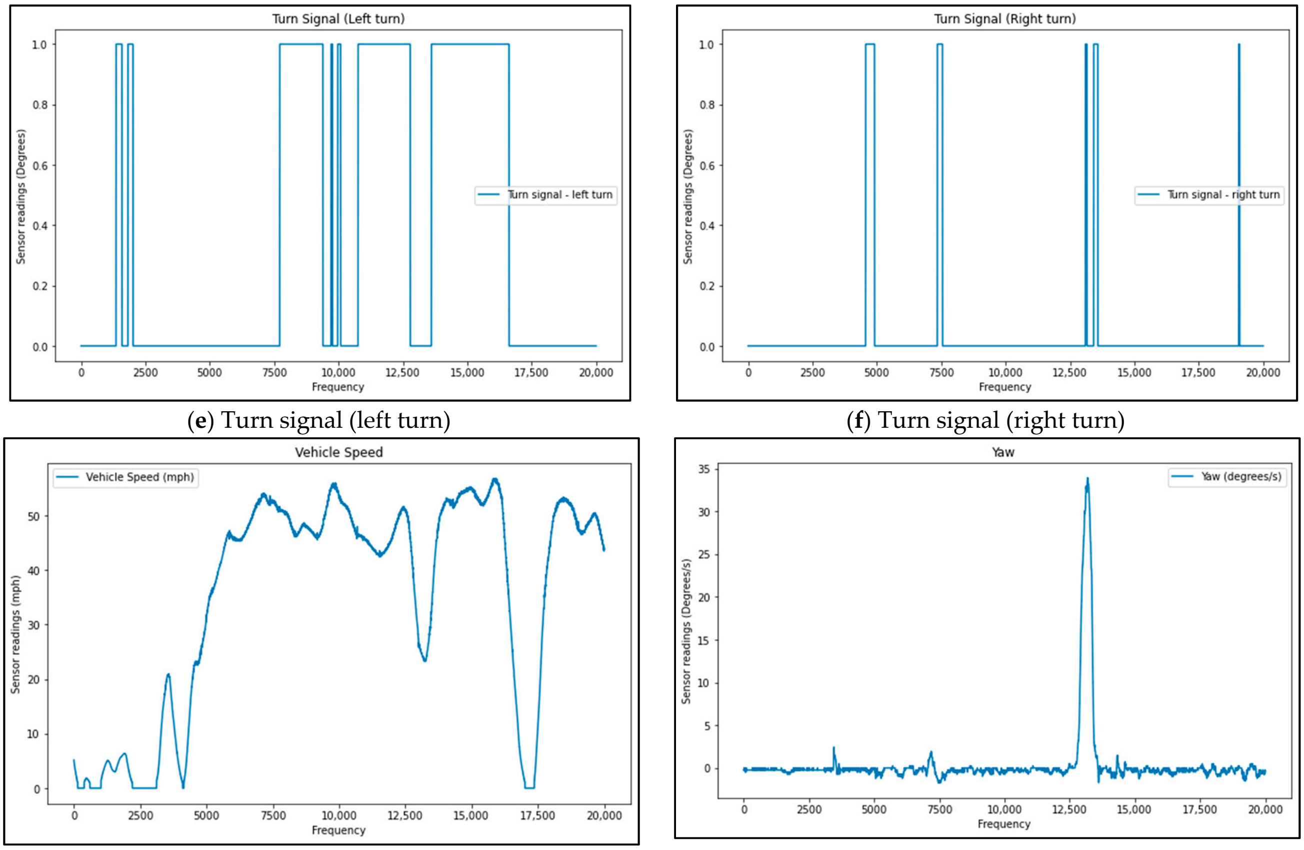
Figure 2. Distress on frames extracted from videos.



(**a**) Acceleration pedal angle (**b**) Brake pedal pressure

(**c**) Steer angle (degrees) (**d**) Steer speed (mph)

Figure 3. *Cont.*



(**e**) Turn signal (left turn)

(**f**) Turn signal (right turn)

(**g**) Speed (mph)

(**h**) Yaw (degrees/s)

Figure 3. Plots to show sensor readings.

The videos that were recorded simultaneously with the sensor readings in the above Figure 3 were analyzed to identify the distresses on the road along the vehicle path. The distresses of interest in this study were those located only within the right-of-way where vehicles typically travel. This is because the rest of the distresses did not affect the ride if the vehicle tire did not pass over them. A value of 1 was assigned for observed distresses, and a 0 for places with no distresses. Figure 4 below shows a sample distress distribution along the road.

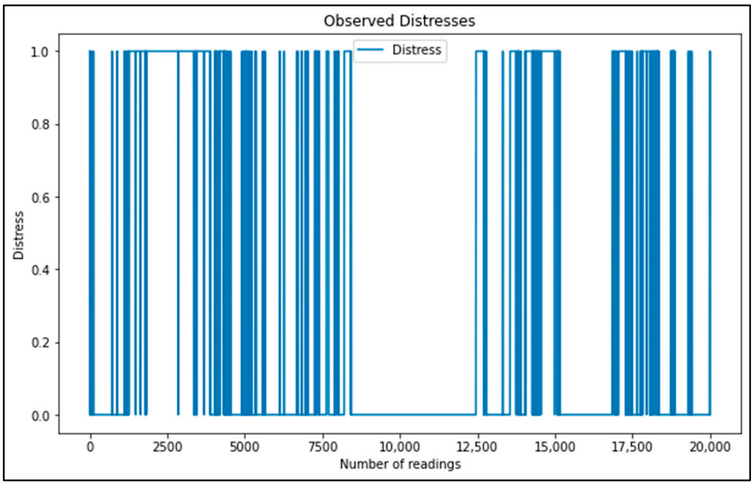


Figure 4. Presentation of observed distresses.

Figure 4. Presentation of observed distresses.

3.2. Model Selection

3.2.1. Vision-Based Model

A computer vision model was developed using the YOLOv5 algorithm. It was selected because of its advantages [27]. These advantages include ease of exporting to other file formats (exportability), its high accuracy value, ease to use, small memory requirements of about 88% compared to YOLOv4 (27 MB vs. 244 MB), and its high speed (about 180% faster than YOLOv4, 140 FPS vs. 50 FPS).

Yolov5 Architecture

Figure 5 presents the architecture of YOLOv5. The model consists of three main parts: The Backbone, Neck (PAnet), and Head (YOLO Layer). These parts play different roles in the model. The Backbone extracts vital features from an input image by reducing the spatial resolution of the input image and increasing its feature (channel) resolution. YOLOv5 uses Cross Stage Partial Network (CSP-Darknet53) as a backbone. The CSP extracts beneficial characteristics from an input image and passes them to the model neck. The model neck creates feature pyramids that aid the model simplification during object scaling. This simplification helps recognize the same object in various sizes and scales. Feature pyramids help assist the models in performing efficiently on previously unseen data. The final detection step is carried out in the Model Head, which uses anchor boxes to construct final output vectors with class probabilities, abjectness scores, and bounding boxes. It is used to perform the last stage of operations.

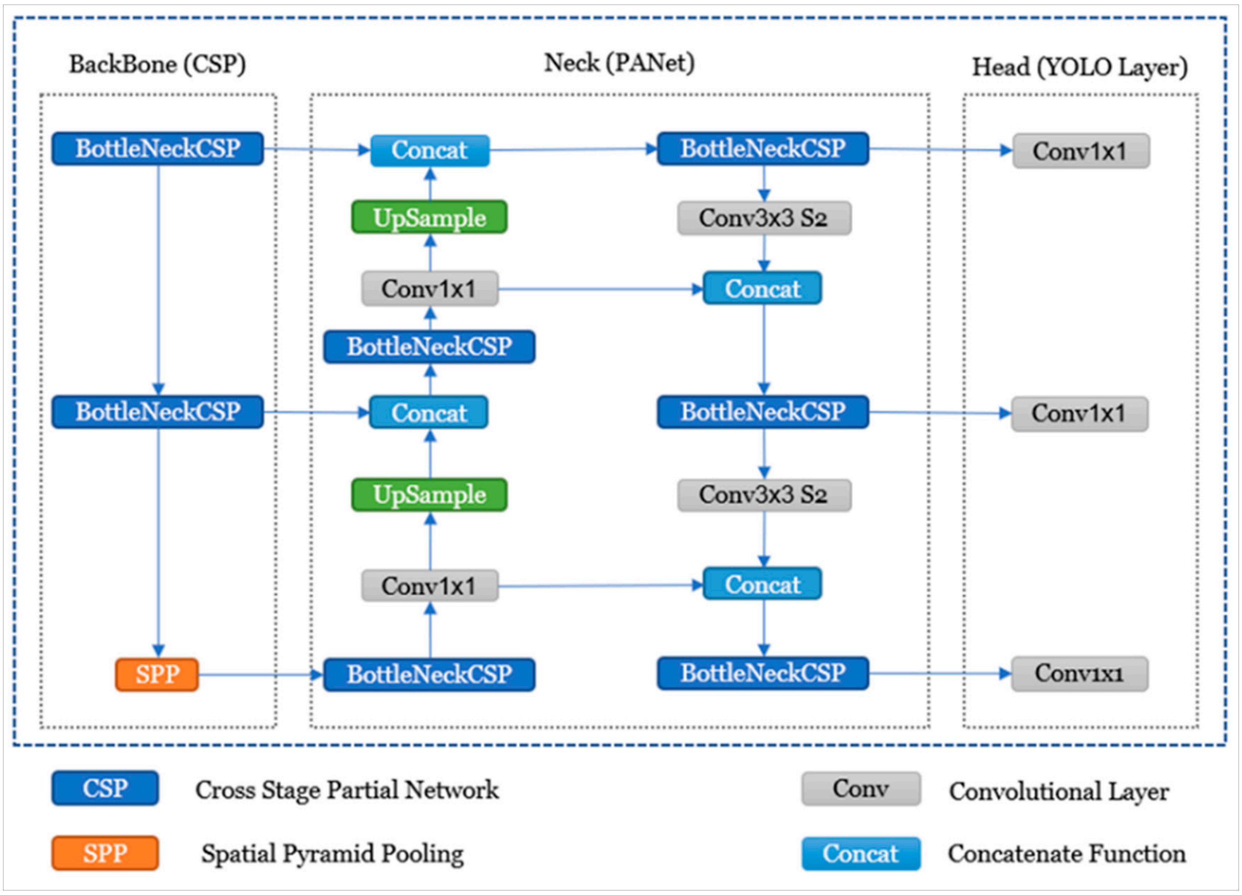


Figure 5. YOLOv5 Network Architecture.

Distress Classification

The computer vision model's purpose was to detect and classify pavement distress into nine groups. The groups include Fatigue/Alligator, Block Cracks, Transverse Cracks, Longitudinal Wheel Path Cracks, Longitudinal Non-Wheel Path Cracks, Edge, Joint, and reflective Cracks, Patches, Potholes, Raveling, Shoving, and Rutting. The classification is based on the Distress Identification manual by the United States Department of Transportation [6].

3.2.2. Sensor-Based Model

This study uses Deep Learning (DL) to predict the presence of distress using sensors. The expected output is either 1 (distress present) or 0 (distress absent). Therefore, a classification model (type of supervised learning) is selected.

Several classification models have been used to date [28]. In this study, the XGBoost model has been selected because of its advantages over other approaches. These advantages include ease of use, high computational efficiency, and high model accuracy compared to algorithms like Random Forest (RF) and logistic regression [21]. This model was used to train on all eight sensors combined to predict the distresses.

XGBoost Model Architecture

XGBoost stands for Extreme Gradient Boosting. It is a popular boosting algorithm for regression and classification purposes. It uses successive iterations to improve the errors of base estimators by taking multiple weak learners. In this model, decision tree classifiers are used as base estimators. If data is not complicated, XGBoost creates an ensemble of linear models. It can also create an ensemble of a gradient-boosted tree (gbtree), which utilizes a decision tree as a base estimator. It first establishes a base model that predicts the target variable, and subsequent models are trained to fit the residuals from the previous steps. The XGBoost algorithm uses decision trees in a sequential form. This algorithm assigns weights to all the independent variables, which are then fed into the decision tree, which predicts results. The weight of variables predicted incorrectly by the tree is increased, and the variables are then fed to the second decision tree. These individual classifiers then ensemble to give a stronger and more precise prediction model. Figure 6 shows the flowchart of this model.

(i) Decision Tree

A decision tree is a building block of an XGBoost model. It has a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. It is commonly used for classification and regression models. A tree can be seen as a piecewise constant approximation. The output of an XGBoost model does not depend on a single decision tree since every decision tree has a high variance. The best method to improve the outcome is to combine several trees; when multiple trees are combined and perfectly trained on sample data, the overall (resultant) variance is low. In the case of a classification problem, the final output is taken using the majority voting classifier.

(ii) Boosting

In the boosting technique, weak classifiers are used to build a robust classifier, achieved by building weak classifiers in series. The first step involves creating the model from training data. The second step consists in making the second model that tries to correct the errors in the first model. The process continues until the complete dataset is predicted or the maximum number of models is added, as shown in Figure 5.

To assess the prediction performance, the Loss (L) is calculated using Equation (1), where y_i stands for the actual value of data and p_i stands for the corresponding predicted value. The overall loss of the algorithm is shown in Equation (2).

$$L(y_i, p_i) = \frac{1}{2}(y_i - p_i)^2 \tag{1}$$

$$\textstyle\sum_{i=1}^{n} L(y_i, p_i) = \frac{1}{2}(y_i - p_i)^2 \tag{2}$$

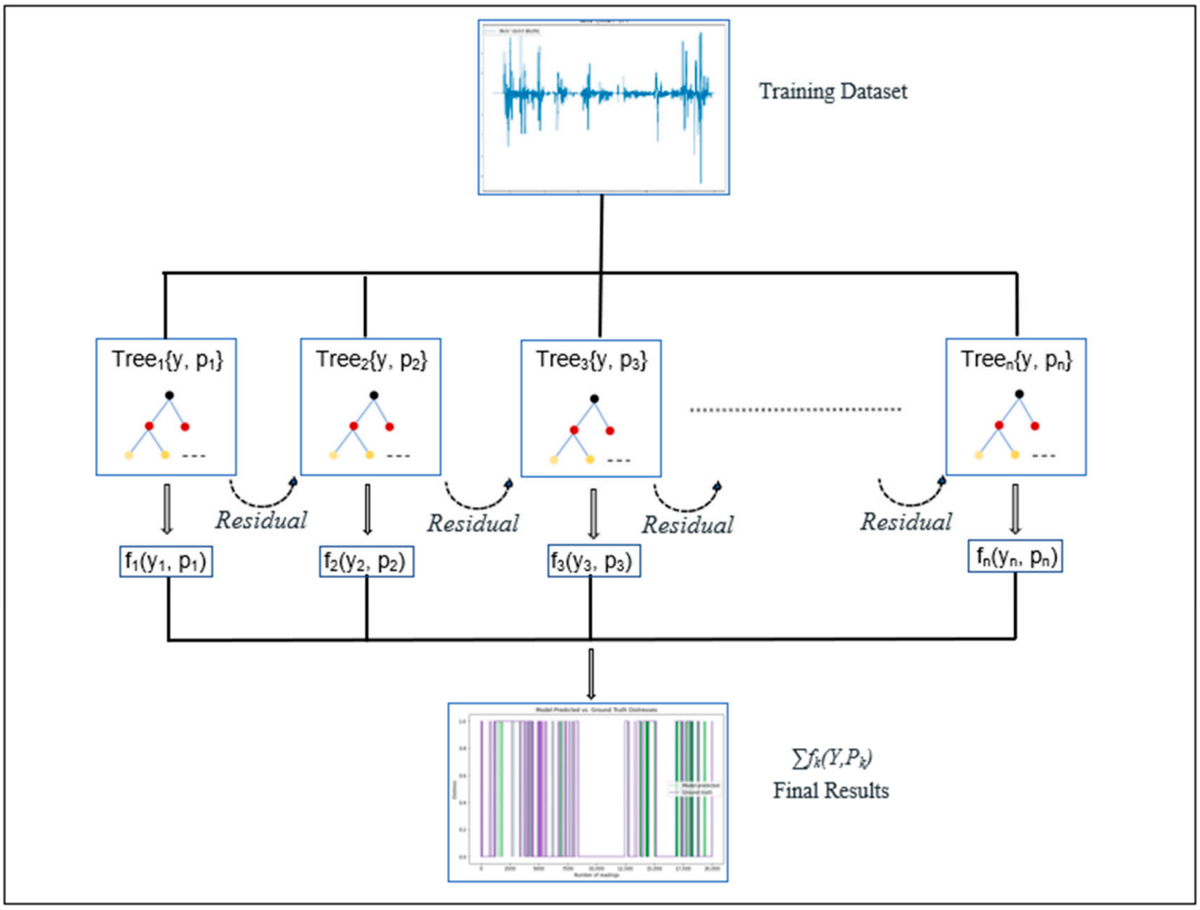


Figure 6. XGBoost model flowchart.

3.3. Model Training

3.3.1. Vision-Based Model

The model was trained on the Google Colaboratory (Google Colab) environment. The training parameters were fine-tuned to achieve desirable results. Table 1 shows the final values obtained from the training.

Table 1. Training parameters.

S/N	Parameter	Value
1	Batch Size	40
2	Epochs	150
3	Learning Rate	0.01
4	Optimizer	SGD = 1×10^{-2}
5	Anchor Sizes	Dynamic

3.3.2. Sensor-Based Model

The XGBoost model was trained on a Windows 10 Pro with NVIDIA GEFORCE GTX GPU, AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz, and 16 GB RAM using TensorFlow. All eight sensor readings were combined into a single excel spreadsheet file for training and to check their influences on the prediction of distress made by the model. Table 2 shows the hyperparameters used in the initial training. The initial training aimed to obtain the optimum training parameters. When completed, hyperparameter tuning was done using GridSearchCV. This optimum value was achieved by setting initial hyperparameter lists, as shown in Table 2. These values and their ranges are provided by

XGBoost developers [29]. After initial training, the outputs were plotted to demonstrate the estimated model's optimum number of trees and tree depth.

Table 2. (a) Initial training hyperparameters. (b) Hyperparameter sets.

S/N	Parameter	Value
	(a)	
1	learning rate	0.1
2	max depth	3
3	n_estimators	5000
4	subsample	0.5
5	colsample_bytree	0.5
	(b)	
1	learning_rate_list	[0.02, 0.05, 0.1]
2	max_depth_list	[2, 3, 5]
3	n_estimators_list	[1000, 2000, 3000]

Figure 7 shows plots of the results of the model performance. The table indicates that the model performs best at a maximum depth of 5 and higher values of trees.

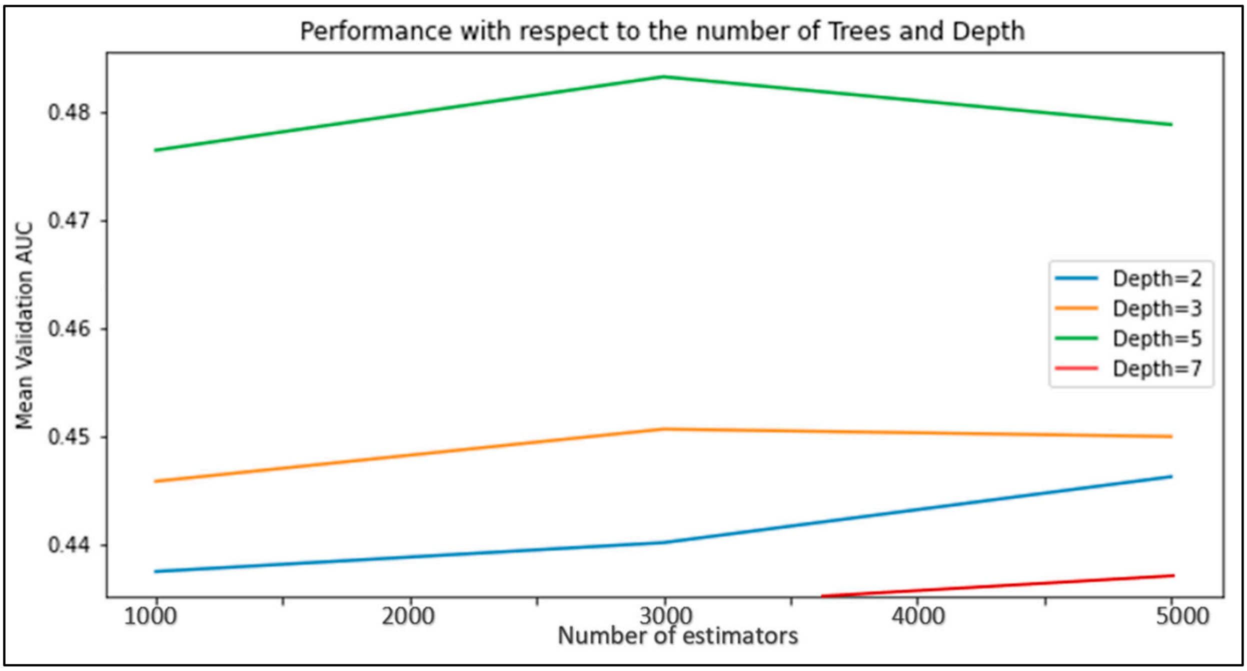


Figure 7. Model Performance at a different number of trees and tree depths.

Figure 8 shows the model performance at various learning rates, which shows that the performance is optimum at 0.05. Therefore, the optimal training parameters are learning rate = 0.05, maximum depth = 5, and the number of estimators (n_estimators/ number of trees) = 3000, subsample = 0.5, and colsample_bytree = 0.5.

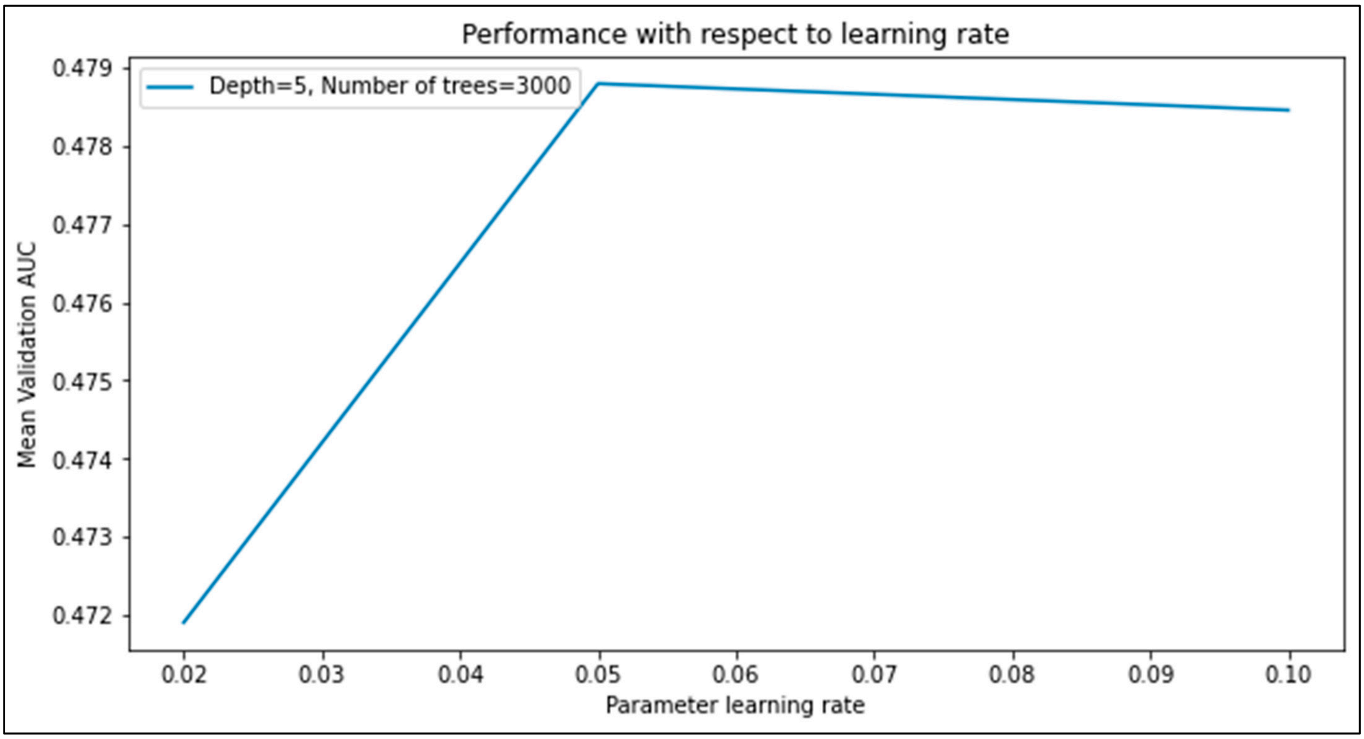


Figure 8. Model performance at different learning rates.

3.4. Performance Metrics

3.4.1. Vision-Based Model

The performance of the vision-based model developed using the YOLOv5 algorithm is assessed based on precision, recall, average Mean Precision (*mAP*), and F1-score. Equations (3)–(6) show that the four metrics are measures of True Positives (TP), False Positives (FP), and False Negatives (FN). FP is the measure of how the model makes wrong predictions, FN measures how the model misses the detections, and TP is the measure of correct detections done by the model. AP_k stands for the average precision of class k, and n stands for the total number of classes.

The F1-score is the harmonic mean of precision and recall. It is a good performance measure for imbalanced data since it considers how data is distributed [30]. Equation (8) shows F1-score computation, where P stands for Precision and R stands for recall.

$$Precision\ =\ \frac{True\ Positives}{True\ Positives\ +\ False\ Positives} \tag{3}$$

$$Recall\ =\ \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives} \tag{4}$$

$$mean\ Average\ Precision\ (mAP) = \frac{1}{n}\sum\nolimits_{k=1}^{k=n} AP_k \tag{5}$$

$$F1score = \frac{1}{\frac{1}{2}(\frac{1}{P} + \frac{1}{R})} = \frac{2 * P * R}{(P + R)} \tag{6}$$

3.4.2. Sensor-Based Model

Metrics

The performance of an XGBoost model is assessed using a Pearson correlation, accuracy, and F1-score. The Pearson correlation coefficient ($\rho_{r,p}$) between two arrays (R,P) is defined as the covariance between array R and array P divided by the product of their respective standard deviations (σ_r,σ_p). Its value ranges from −1 to 1, where a value −1 means perfect negative correlation, 0 means no correlation, and 1 means a perfect positive linear relationship between the two values. Equation (7) shows the mathematical expression of the Pearson correlation [31].

$$\rho_{r,p} = \frac{cov(R,P)}{\sigma_r\sigma_p} = \frac{E((R - {}_{\mu r})(P - {}_{\mu p}))}{\sigma_r\sigma_p} \tag{7}$$

where: ${}_{\mu r}$ stands for the mean of R—array, ${}_{\mu p}$ stands for the mean of P—array, and E((R − ${}_{\mu r}$)(P − ${}_{\mu p}$)) represents the average value of (R,P) that is expected in a long sequence of repeated trials of the random experiment.

Accuracy is the ratio of all correct predictions to the total number of predictions. It is obtained as a ratio of the sum of True Positives (TP) and True Negatives (TN) to the total number of predictions sample size, as shown by Equation (8). The F1-score is presented in Equation (6).

$$Accuracy = \frac{True\ Posives\ +\ True\ Negatives}{Total\ Sample\ Size} \tag{8}$$

Feature Importance Assessment

The sensor-based model was prepared using a combination of 8 different sensors. All sensors contributed to the final model results and performance. Feature Importance Assessment (FIA) was done to assess the extent to which individual sensors contribute to the final results. Figure 9 presents the results of this analysis. The results show that the prediction made by the model is mainly influenced by the steering angle at 26.70% and influenced the least by steer speed at 5.30%.

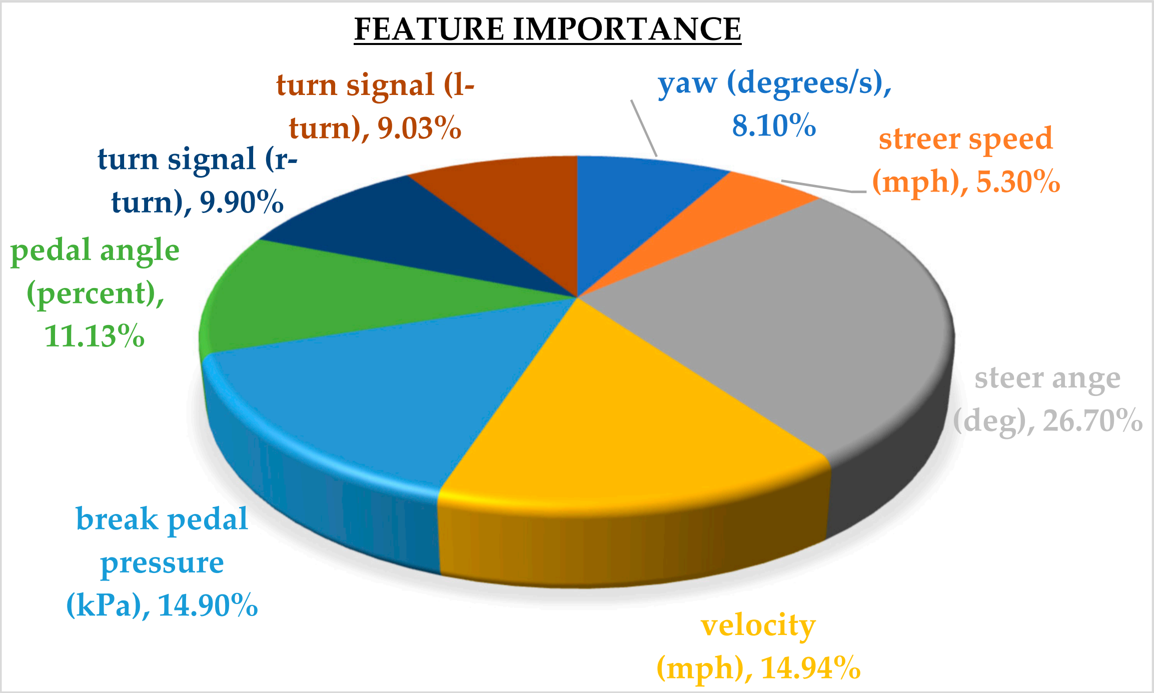


Figure 9. Feature importance to the model predictions.

3.5. Results and Analysis

3.5.1. Results of Vision-based Road Surface Detection Model

Figure 10 shows the precision–recall curves (PR curves). The curves show how the precision values vary with the increase in recall values during training. It shows that the model attained an overall mean average precision (mAP@0.5) of 93.9% in all pavement classes. All curves are close to each other and are concentrated in the upper right corner, which indicates that the model can predict and classify the distresses with high accuracy.

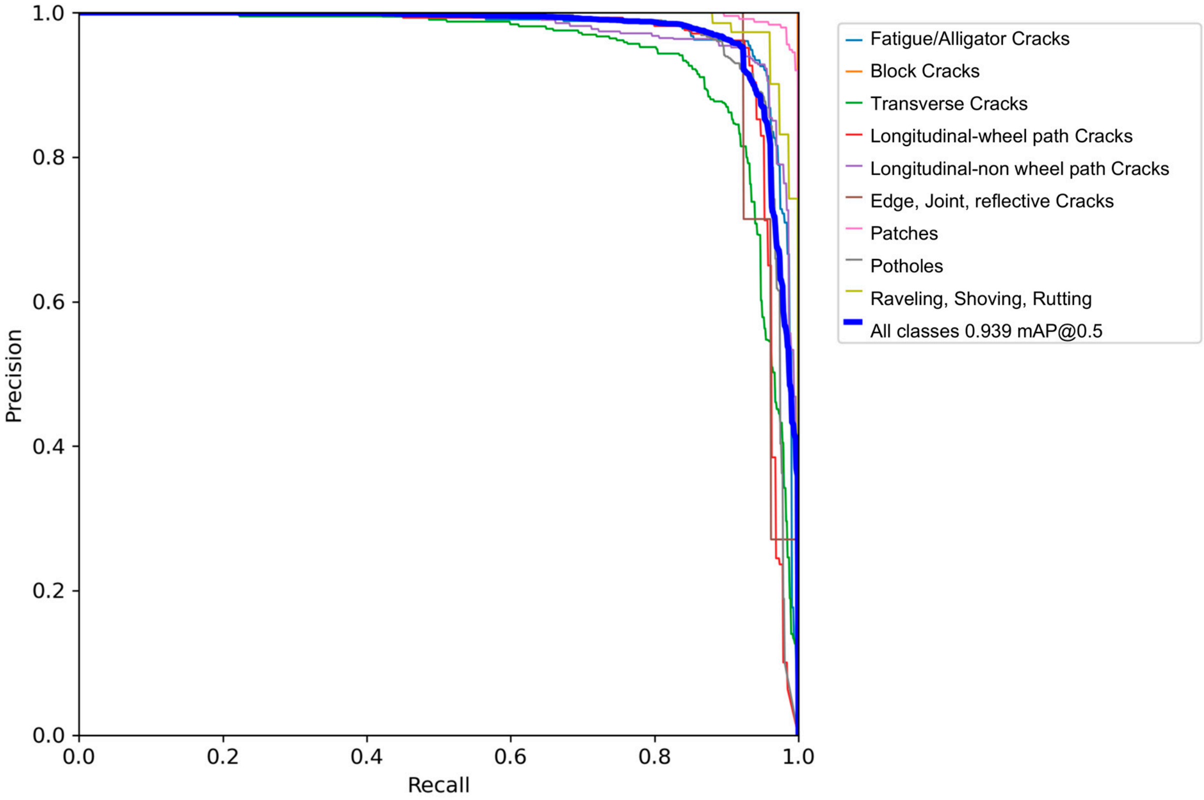


Figure 10. Precision–recall curve.

Figure 11 shows how the F1-score changes with the increase in confidence during training. The model attained an overall F1-score of 82%, indicating good accuracy under this metric [30].

3.5.2. Results of Sensor-based Road Surface Detection Model

Figure 12 shows how the values of AUCs change with the increase in the number of trees. The XGBoost model achieved an accuracy of 81.28% and an F1-score of 76.40%, implying that the model can predict distresses with high accuracy. Also, the trained model reached 98.42% AUC in the training dataset and 97.99% in the validation dataset, using the Area Under the Curve (AUC) metrics. The figure shows that the values of AUC increase sharply with the number of trees from 0 to 500, then the rate of increase decreases and becomes almost constant as the number of trees approaches 3000. This observation implies that the training process has been successful, and there is no overfitting. Also, these results show that the developed model has high prediction accuracy since the AUC value above 90% indicates high prediction accuracy, while AUC between 70% and 90% presents moderate accuracy. An AUC of less than 70% means poor prediction accuracy of the model [32].

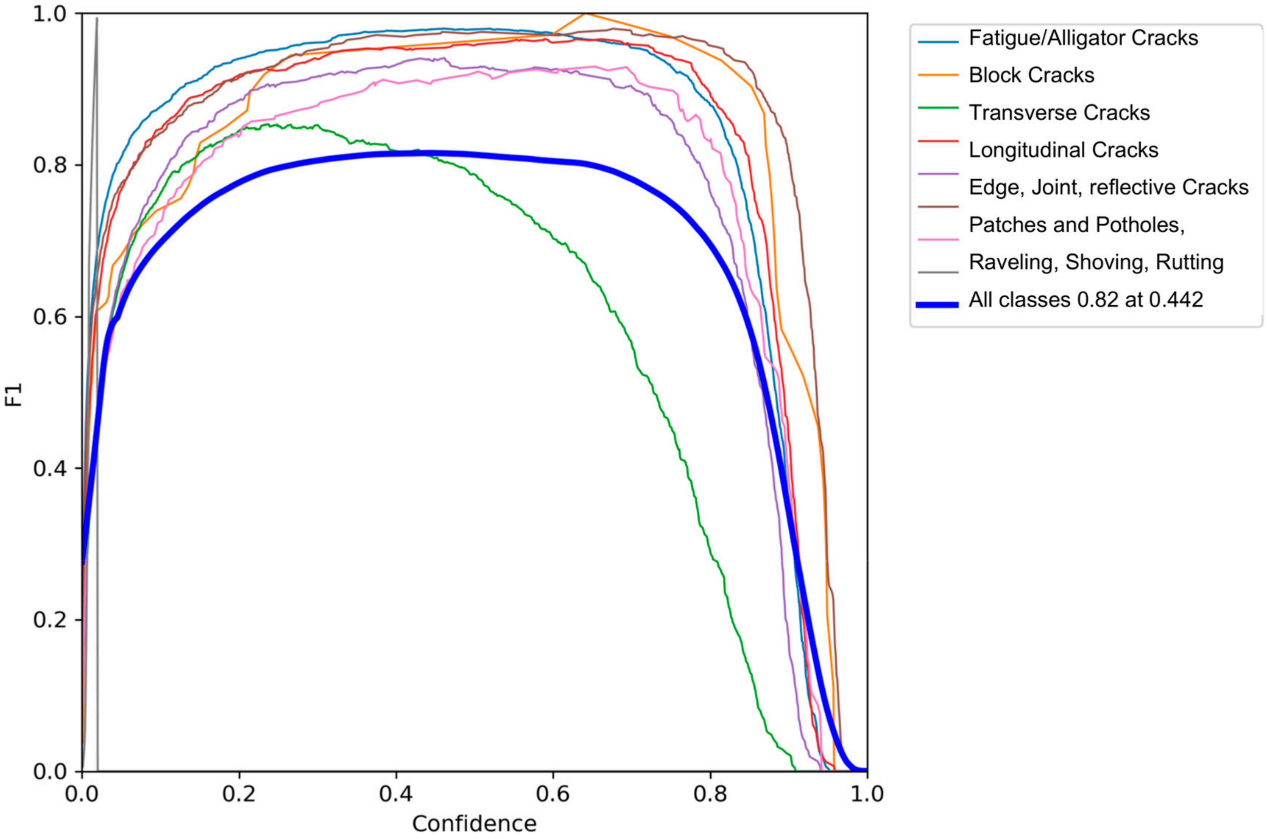


Figure 11. F1 confidence curve.

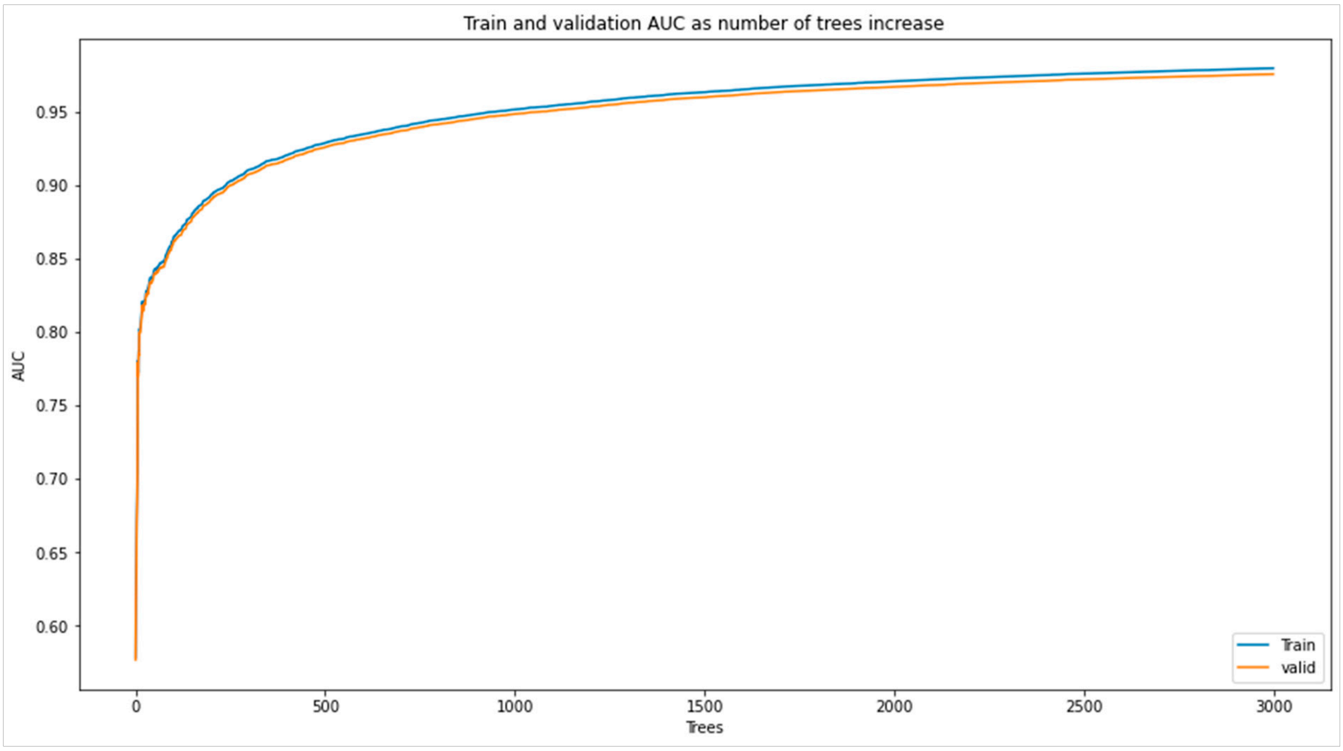


Figure 12. Train and validation AUC versus the number of trees.

3.6. Score the Test Data

This paper used 20,000 arrays of sensor-based data for testing purposes to see how the model detects pavement distresses. The distresses detected by the computer vision-based models and the sensor-based models were used to make separate arrays, and Pearson correlation analysis was done using the Pandas library in Python. Figure 13 shows the confusion matrix for actual versus predicted distresses. The matrix shows a correlation of 83% between the distresses detected by the computer vision-based model and those detected by the sensor-based model.

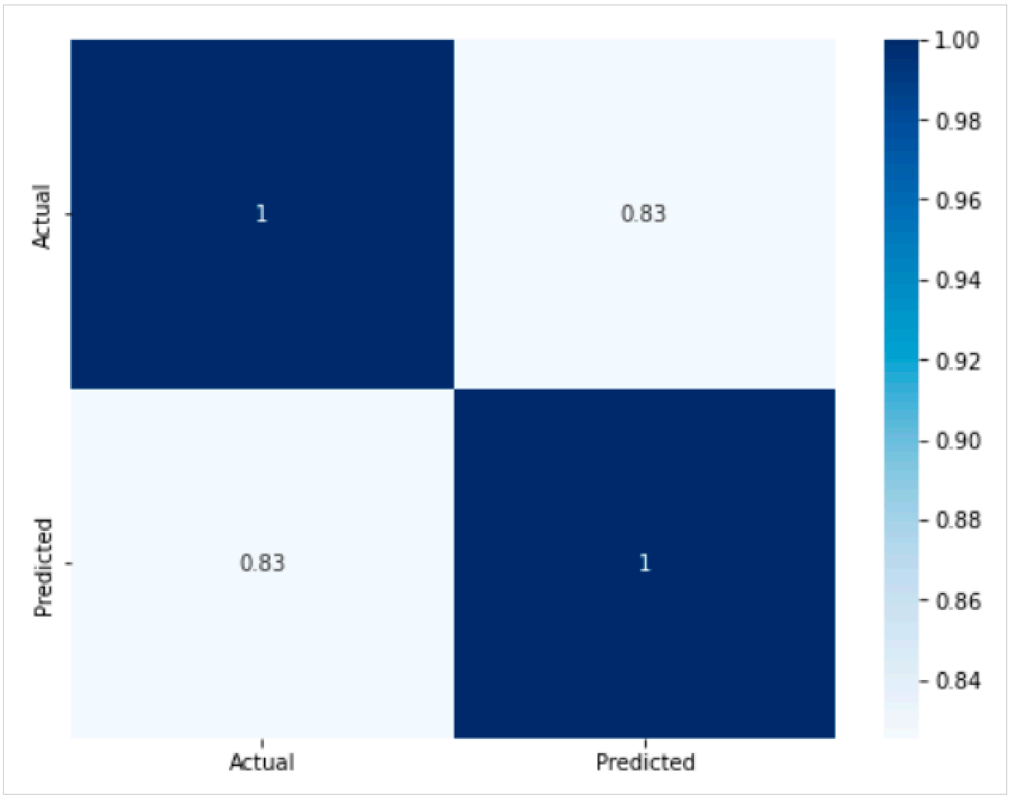


Figure 13. Confusion matrix for Actual versus Predicted distresses.

Figure 14 below shows the relationship between the observed distresses (ground truth) and the model-predicted distresses (model-predicted). The overlaps between the two lines indicate that the model prediction was in agreement with the actual condition on-site. This plot shows that the model can predict most of the distresses and can be used with high accuracy.

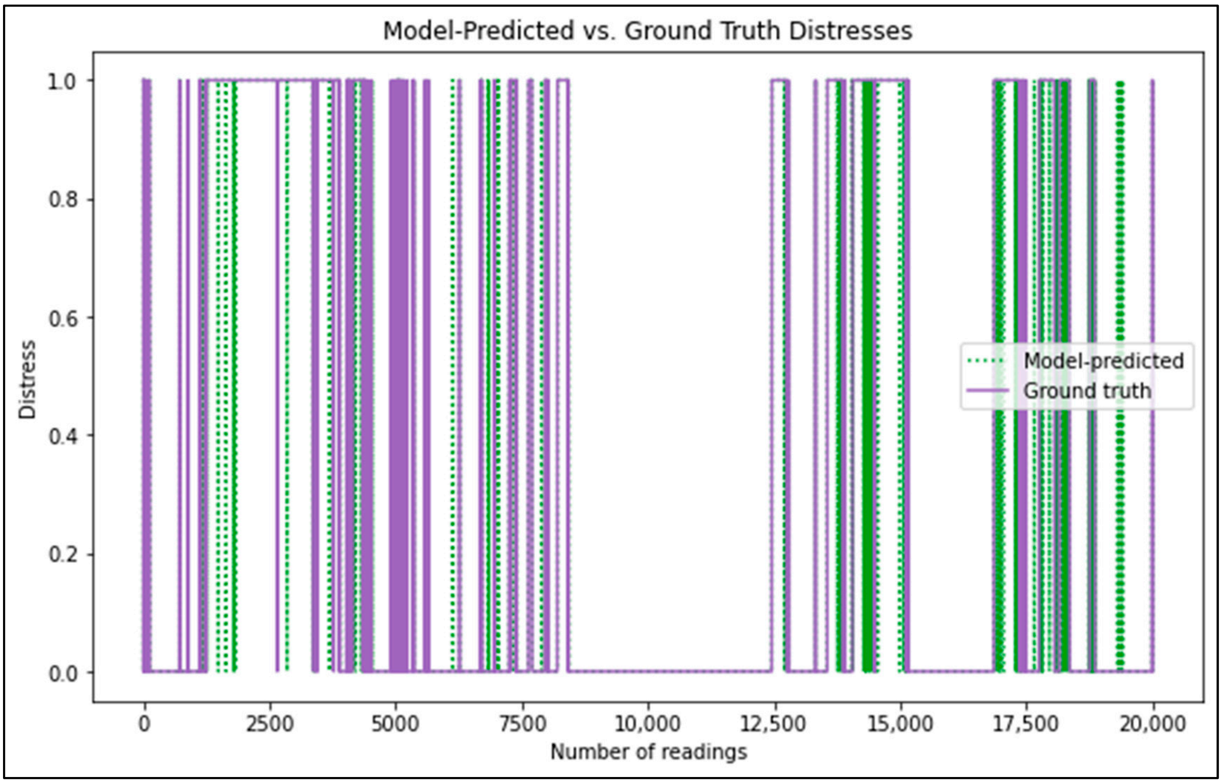


Figure 14. Relationship between predicted and actual distresses observed.

4. Conclusions

This study has proposed models for detecting pavement distress as a measure of road condition monitoring. The models comprise both computer vision and sensor-based approaches. The study used freely available data from the internet and Honda Motors Company [26]. In developing the vision model, this study used YOLOv5 algorithm that was trained on 3500 images. The model achieved 95%, 93.4%, 97.2%, and 94% values in precision, recall, mean average precision, and F1-score, respectively. The sensor-based model was developed using the XGBoost model. The model was trained on eight different CAN bus sensors combined. The model achieved 98.42% and 97.99% in training and validation using area-under-curve (AUC) metrics, compared to 83.04% achieved by Chen et al. [33] using the XGBoost model. The results obtained in this paper also fall within the high accuracy range of an AUC above 90% [32]. In comparison, AUC that lies between 70% and 90% presents moderate accuracy and an AUC of less than 0.7 indicates a poor prediction. In conclusion, the results obtained in this paper showed that different CAN bus sensors could be used to predict the presence of pavement distresses with high accuracy and therefore, they can be used to complement the vision-based model in adverse weather conditions.

Limitations and Recommendations of the Study

This paper prepared a CAN bus sensor model to predict the presence of pavement distresses without classification of the distress types. There is a need to conduct further research to employ sensors and develop models that can detect and classify pavement distresses simultaneously. In some conditions where the vision-based model does not perform well, a model pooling technique can be employed where the sensor-based model can complement the vision-based model. This pooling technique will assist in capturing the distresses during adverse weather conditions, such as rainy weather or wet road surfaces—i.e., where the poor performance of the vision-based model is observed.

Author Contributions: Conceptualization, C.R., J.M., G.C. and F.N.; data curation, C.R., G.C., F.N. and K.M.; methodology, C.R., J.M. and G.C.; project administration, J.M.; supervision, J.M.; writing—original draft preparation, C.R.; writing—review & editing, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research and The APC were funded by the U.S. Department of Education partly provides funding support for this study through the HBCU Master's Program Grant, by Grant No. P120A210048, the U.S. Department of Transportation's University Transportation Centers Program grant administered by the Transportation Program at South Carolina State University (SCSU), Tier I University Transportation Center for Connected Multimodal Mobility, and NSF Grant Nos. 1719501, 1954532, and 2131080.

Data Availability Statement: This study used image, video, and CAN bus sensors datasets. The image dataset was collected from the internet and is freely available. The other two were collected from Honda Motor Company by Ramanishka et al. [26], and are made available upon request and signing the agreement of their use.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ameddah, M.A.; Das, B.; Almhana, J. Cloud-Assisted Real-Time Road Condition Monitoring System for Vehicles. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018.
2. Engström, R. *The Roads' Role in the Freight Transport System*; 6th Transport Research Arena: Göteborg, Sweden, 2016.
3. Vaitkus, A.; Čygas, D.; Motiejūnas, A.; Pakalnis, A.; Miškinis, D. Improvement of road pavement maintenance models and technologies. *Balt. J. Road Bridg. Eng.* **2016**, *11*, 242–249. [CrossRef]
4. Sahin, H.; Narciso, P.; Hariharan, N. Developing a Five-year Maintenance and Rehabilitation (M&R) Plan for HMA and Concrete Pavement Networks. *APCBEE Procedia* **2014**, *9*, 230–234.
5. Feldman, D.R.; Pyle, T.; Lee, J. *Automated Pavement Condition Survey Manual*; California Department of Transportation: Los Angeles, CA, USA, 2015.
6. USDOT. *Distress Identification Manual for the Long-Term Pavement Performance Program*; USDOT, Federal Highway Administration: Washington, DC, USA, 2014.
7. Ranyal, E.; Sadhu, A.; Jain, K. Road Condition Monitoring Using Smart Sensing and Artificial Intelligence: A Review. *Sensors* **2022**, *22*, 3044. [CrossRef] [PubMed]
8. Majidifard, H.; Jin, P.; Adu-Gyamfi, Y.; Buttlar, W.G. Pavement Image Datasets: A New Benchmark Dataset to Classify and Densify Pavement Distresses. In Proceedings of the TRB 99th Annual Meeting, Washington, DC, USA, 12–16 January 2020.
9. Sholevar, N.; Golroo, A.; Esfahani, S.R. Machine learning techniques for pavement condition evaluation. *Autom. Constr.* **2022**, *136*, 104190. [CrossRef]
10. Wang, K.C.P.; Zhang, A.; Li, J.Q.; Fei, Y.; Chen, C.; Li, B. Deep Learning for Asphalt Pavement Cracking Recognition Using Convolutional Neural Network. *Airfield Highw. Pavements* **2017**, 166–177. [CrossRef]
11. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [CrossRef] [PubMed]
12. Zhang, A.; Wang, K.C.P.; 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]
13. Zhang, A.; Wang, K.C.P.; Fei, Y.; Liu, Y.; Chen, C.; Yang, G.; Li, J.Q.; Yang, E.; Qiu, S. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces with a Recurrent Neural Network. *Comput. Civ. Infrastruct. Eng.* **2019**, *34*, 213–229. [CrossRef]
14. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Comput. Civ. Infrastruct. Eng.* **2018**, *33*, 1127–1141. [CrossRef]
15. Maeda, H.; Kashiyama, T.; Sekimoto, Y.; Seto, T.; Omata, H. Generative adversarial network for road damage detection. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *36*, 47–60. [CrossRef]
16. Aleadelat, W.; Saha, P.; Ksaibati, K. Development of serviceability prediction model for county paved roads. *Int. J. Pavement Eng.* **2018**, *19*, 526–533. [CrossRef]
17. Souza, V.M. Asphalt pavement classification using smartphone accelerometer and Complexity Invariant Distance. *Eng. Appl. Artif. Intell.* **2018**, *74*, 198–211. [CrossRef]
18. Christodoulou, S.E.; Kyriakou, C.; Hadjidemetriou, G. Pavement Patch Defects Detection and Classification Using Smartphones, Vibration Signals and Video Images. *Adv. Comput. Strateg. Eng.* **2019**, 365–380. [CrossRef]
19. Sandamal, R.M.K.; Pasindu, H.R. Applicability of smartphone-based roughness data for rural road pavement condition evaluation. *Int. J. Pavement Eng.* **2022**, *23*, 663–672. [CrossRef]

20. Pomoni, M. Exploring Smart Tires as a Tool to Assist Safe Driving and Monitor Tire–Road Friction. *Vehicles* **2022**, *4*, 744–765. [CrossRef]
21. Ahmed, N.S.; Huynh, N.; Gassman, S.; Mullen, R.; Pierce, C.; Chen, Y. Predicting Pavement Structural Condition Using Machine Learning Methods. *Sustainability* **2022**, *14*, 8627. [CrossRef]
22. Lekshmipathy, J.; Samuel, N.M.; Velayudhan, S. Vibration vs. vision: Best approach for automated pavement distress detection. *Int. J. Pavement Res. Technol.* **2022**, *13*, 402–410. [CrossRef]
23. Eisenbach, M.; Stricker, R.; Seichter, D.; Amende, K. How to get pavement distress detection ready for deep learning? A systematic approach. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2039–2047.
24. Hamishebahar, Y.; Guan, H.; So, S.; Jo, J. A Comprehensive Review of Deep Learning-Based Crack Detection Approaches. *Appl. Sci.* **2022**, *12*, 1374. [CrossRef]
25. Gur-Arie, P.L. The Practical Guide for Object Detection with YOLOv5 Algorithm. 14 January 2023. [Online]. Available online: https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843 (accessed on 12 November 2022).
26. Ramanishka, V.; Chen, Y.-T.; Misu, T.; Saenko, K. Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7699–7707. [CrossRef]
27. Garg, A. How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection. Available online: https://www.analyticsvidhya.com/blog/2021/12/how-to-use-yolo-v5-object-detection-algorithm-for-custom-object-detection-an-example-use-case/ (accessed on 6 January 2023).
28. Justo-Silva, R.; Ferreira, A.; Flintsch, G. Review on Machine Learning Techniques for Developing Pavement Performance Prediction Models. *Sustainability* **2021**, *13*, 5248. [CrossRef]
29. XGBoost. XGBoost Parameters. XGBoost Developers. Available online: https://xgboost.readthedocs.io/en/stable/parameter.html (accessed on 26 February 2023).
30. Allwright, S. What Is a Good F1 Score and How Do I Interpret It? Available online: https://stephenallwright.com/good-f1-score/ (accessed on 28 July 2022).
31. Montgomery, D.C.; Runger, G.C. *Applied Statistics and Probability for Engineers*; Wiley: Hoboken, NJ, USA, 2018.
32. Mcdowell, I. *Measuring Health: A Guide to Rating Scales and Questionnaires*; Oxford University Press: Oxford, UK, 2006.
33. Chen, T.; Guestrin, C.E. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.

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.