

# Determination of the Degradation Index by Detection of Pavement Distress with Transfer Learning and Image Processing

Crespin Prudence Yabi<sup>1</sup>, Emile Dehounhessi<sup>1</sup>, Edjrossè Fructueux Gildas Godonou<sup>1</sup>, Mohamed Gibigaye<sup>2</sup>, Eric Alamou<sup>1</sup>

<sup>1</sup>Laboratory of Studies and Tests in Civil Engineering (L2EGC), National University of Sciences, Technologies, Engineering and Mathematics, Abomey, Benin

<sup>2</sup>Laboratory of Applied Energetic and Mechanic (LEMA), University of Abomey-Calavi, Abomey-Calavi, Benin

Email: yabi.crespin@unstim.bj, emiledehounhessi@gmail.com, gildasgodonou@yahoo.fr, mohamed.gibigaye@uac.bj

**How to cite this paper:** Yabi, C.P., Dehounhessi, E., Godonou, E.F.G., Gibigaye, M. and Alamou, E. (2025) Determination of the Degradation Index by Detection of Pavement Distress with Transfer Learning and Image Processing. *Engineering*, 17, 680-705.

<https://doi.org/10.4236/eng.2025.1712038>

**Received:** November 16, 2025

**Accepted:** December 22, 2025

**Published:** December 25, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

A road network is essential to a country's transportation and socio-economic development. Its maintenance requires regular monitoring to guide maintenance decisions. Artificial intelligence now enables the automatic detection of damage, but monitoring a roadway does not end with detection. It also involves estimating the severity and extent of damage and determining the *Is* index. Therefore, this study allowed the development of a digital tool based on artificial intelligence and image processing for complete monitoring. This consists of detecting the roadway damages, then estimating their severity and extent to calculate the index *Is*. Four databases were designed from videos of damaged roads collected on different roads in Benin. These data were used in transfer learning to train the YOLOv9 and Roboflow 3.0 Object Detection models. A script was developed to estimate the extent of the degradations, setting a collection speed of 10 km/h, a picking height of 1.20 m and a viewing angle equal to 45°, covering the entire width of the roadway. Another script determines the index *Is* by estimating the cracking and deformation indices, with possible corrections depending on the repairs present. The best model obtained, ROCNN4, results from training Roboflow 3.0 Object Detection with the fourth base. It detects 19 classes of degradations with a precision P of 90.8%, a mAP of 91.8% and a recall R of 89.5%. These results pave the way for better road maintenance planning by providing managers with a reliable and automated decision-support tool. They thus help optimize intervention costs and improve the durability of the road network.

---

## Keywords

Road Monitoring, Artificial Intelligence, Image Processing, Extent of Damages, Degradation Index

---

## 1. Introduction

The movement of people and goods is one of the most important factors in socio-economic development globally and within specific countries. Consequently, numerous communication infrastructure projects are being undertaken globally daily to enhance the mobility of people and their goods. Roads are a prominent feature of these infrastructures. Road represents the most widespread type of communication infrastructure throughout the world [1].

Although it is one of the main factors of socio-economic development, the road is faced with certain spatio-temporal factors that influence its behaviour throughout its lifetime [2]. Among these factors, we can distinguish the quality of construction, the different characteristics of traffic loads, environmental conditions, and material properties such as elasticity, viscosity, plasticity, and resistance. The diversity of these factors explains the multitude of types of degradation that can appear on the surface of the roadway [3]. These degradations can go from the low stage to the major stage and become a source of road accidents [4].

The National Interministerial Road Safety Observatory in France informs that in 2023, approximately 30% of fatal accidents were directly linked to road surface damage. Poor road surface conditions also lead to increased fuel consumption and CO<sub>2</sub> emissions, noise emissions, and an increase in vehicle maintenance costs. To alleviate all these consequences of damage on the road and allow the roadway to fully perform its function, maintenance of the roadway is necessary. The execution of this task involves an examination of the roadway, as is done in Benin with the VIZIROAD equipment [5], which automatically collects visual data, georeferenced images, and profile measurements to characterize surface degradations. But this equipment has certain limitations, such as the reliance on manpower and qualified personnel for data collection and interpretation. It can also prove costly to operate over very long distances [6]. The many limitations posed by pavement surveying explain why the search for more efficient tools has become a major concern. As part of its public policy support activities, CEREMA has developed the Aigle-RN vehicle [7], which allows the monitoring of a road surface. Several other road monitoring systems such as IRCAN [8], LCMS [9] have been developed to perform this work.

However, with the advent of artificial intelligence, the simplification of the road monitoring task has been considered. Thanks to its ability to process a massive volume of data, artificial intelligence will automate the road monitoring process while reducing costs and time. Several researchers have worked on the automatic classification of damage using artificial intelligence.

With the advent of artificial intelligence, intelligent classification of damage has been successful in several studies thanks to machine learning. The goal in these studies is to use a database containing 2D images of the roadway to train develop a model that will later be able to detect and classify damage. Ullah *et al.* (2019) [10] developed a model to classify longitudinal and transverse cracks with an accuracy of 84.88%. Dhieb *et al.* (2022) [11] started from 32000 images to train the model Inception-V3, which classifies longitudinal, diagonal, and transverse cracks with respectively 96%, 94%, and 96%. Using the “transfer learning” technique, Mirbod and Shoar [12] were able to design a crack classification model reaching an accuracy of 98.01%. To facilitate the detection of cracks by its model, CrackN-T 89,2% accurate, Bahrami [13] exploited an image processing technique including histogram equalization. This technique allowed him to improve the contrast of images before submitting them to the model for training. A. Franggidae [3] designed a CNNBN that accurately detects potholes on a 2D image at 96%. All these models utilize convolutional neural networks (CNNs), which, according to Y. Gong [14], distinguish them from other types of neural networks by several key characteristics, mainly related to their architecture and their ability to process data efficiently, such as images. In 2025, Yabi [14] worked on the automatic classification of road damage and developed an artificial intelligence model capable of classifying four types of damage: longitudinal cracks, alligator cracks, potholes, and repairs.

However, these previously listed models with their significant performance, stop at the classification of degradations and do not consider their severity and extent. Other researchers like N. Shatnawi [15] have oriented their work towards the estimation of the depth of ruts to determine their severity. They had to opt for the use of laser scanners, which, moreover, are extremely expensive and require technical expertise for operation and data exploitation. A. Azam [16] developed a smartphone to estimate the depth of ruts, but also notes in the work that the performance of this tool remains to be improved a lot.

The development of a more accurate artificial intelligence model capable of simultaneously estimating the severity and extent of damage based on 2D images remains a very challenging job. The present study, therefore, aims to develop a better-performing digital tool that will detect and classify damage of the flexible pavement structure, but also to estimate with a reliable method the seriousness and extent of these different degradations to determine the degradation index, *I<sub>s</sub>*. This tool will be based on artificial intelligence and image processing techniques to efficiently automate the process of road monitoring to reduce costs related to auscultation investigations and to enable easy decision-making regarding road maintenance.

## 2. Research Methodology

The methodology adopted during this work is based on 6 essential points, including data collection, data pre-processing, database design, development of detec-

tion models, calculation of the extent of degradation, determination of the degradation index  $I_s$ , evaluation of the operation of the best model obtained, then scripts for determining the extent and the index  $I_s$  and finally the development of the mobile application.

## 2.1. Data Collection

The data provided to models for training constitutes images of degraded asphalt pavements. To obtain this data, videos of sections of damaged pavements were collected on several roads in Benin, including:

- Abomey—Klouekanmè
- Porto Novo—Seme
- Bohicon—Dassa

To collect high-quality video, a professional Rebel T8i camera was used. During the collection, a vehicle was kept facilitating movement along the axes, then a centimeter and a ruler were used to take measurements.

The first step in the collection process consisted of assessing the severity of the damage using a ruler and a centimeter at each section of the damaged road surface. These tools allowed us to note the dimensions of the ruts, potholes, peelings, and ridges, including their depth, height, length, and width, as indicated by the VIZIR method for each type of damage. For damage such as cracks and alligator cracks, we did not need to measure their severity, as this can be done visually on the images.

The various measurements taken were then recorded in a table, which served not only to document the observations in the field but also to facilitate the identification of degradations in the collected videos to provide their correct level of severity to the models.

The second step of our data collection method consisted of capturing videos of the damaged road sections. This task was entrusted to a professional photographer, equipped with high-quality equipment. The recorded videos cover the entirety of the identified damaged areas. They were captured from a height of approximately 1.2 m and from a 45° viewing angle, which makes the various pathologies clearly visible.

## 2.2. Data Pre-Processing

When developing a detection model with deep neural networks, it is important to properly prepare the data before using it. This preparation step, called data pre-processing, is essential to ensure good data quality. In our study, five pre-processing operations were performed to prepare our data for use in model training.

- Image extraction

Frame extraction is the process by which individual frames are retrieved and saved from our various collected video sequences. To achieve this, we developed a Python script called “Extractor” in the PyCharm development environment. This script enables us to extract individual frames from our videos using the

OpenCV library and automatically save them as PNG files on our computer.

- Selective sorting of extracted images

Selective sorting is the step that enables us to eliminate redundant photos, ensuring a diverse range of images in our databases. Similarly, blurry images or those whose sharpness was affected by shaking or vibration were also eliminated to make the models focus on the essential features.

- Resizing images

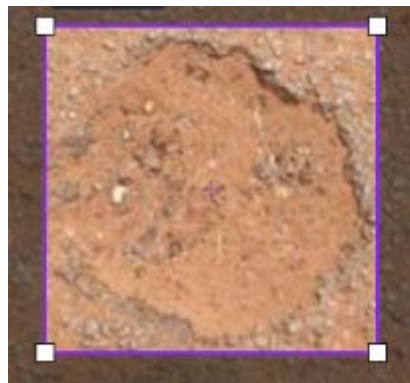
Image resizing is the process by which the size of our images is reduced to speed up the training process and reduce memory and computing power requirements. The images were reduced by 1/4 of their length and width without any loss of information. This task was accomplished using a program named “Resizer”, developed in Python language in the PyCharm development environment. When the script was run, the dimensions of our images had changed from (1920p × 1080p) to (480p × 270p).

- Pixel Normalization

To avoid overflow and underflow issues, we normalized the pixels in our various images. Using the min-max normalization method, pixel values are now between 0 and 1. This was made possible by a Python script called “Normalizer”, developed in the PyCharm development environment.

- Image annotation

Image annotation is the step that allowed us to label all the degradations located on the normalized images to provide to our models for training. For this, we utilized the Roboflow platform, which features two tools: the “Bounding Box” tool and the “Polygon” tool. During the annotation, we directly specify the severity of each degradation with numbers 1, 2, and 3, indicating the severity levels 1, 2, and 3, respectively. Severity levels 1, 2 and 3 correspond respectively to low, medium and high levels of distress. For example, in the case of rutting, level 1 indicates a shallow rut depth ( $h < 2$  cm), level 2 a moderate rut depth ( $2 \text{ cm} \leq h \leq 4$  cm), and level 3 a deep rut ( $h > 4$  cm). **Figure 1** shows the annotation of a pothole of severity 2 with the bounding box while **Figure 2** shows the annotation of some distress with polygon tool.



**Figure 1.** Annotation of a severity 2 pothole with a bounding box.



**Figure 2.** Annotation of a 3-beam and a 3-rut with the polygon tool.

### 2.3. Database Conception

Four databases were established to achieve the objectives of our study. The components of each database are presented in **Table 1**.

**Table 1.** Databases.

Databases	Total Number of images	Image size	Number of annotated degradations	Number of degradation classes
Dataset_831	831	1090*1080	2576	18
Dataset_3720	3720	1090*1080	7634	20
Dataset_4770	4770	1090*1080	12052	19
Dataset_4770A	4770	1090*1080	12756	19

Ultimately, nineteen (19) classes of degradation were able to find a place in the latest database. These classes of degradation are: Longitudinal crack 1, 2, and 3; Alligator crack 1, 2, and 3; Pothole 1 and 2, Raveling, peeling, rut 1, 2, and 3, Upheaval 3, edge break 3, Stripping, Traverse crack 1, repair, trace of accident.

#### 2.3.1. Subdivision of Databases

To ensure a good distribution, the first two databases are divided into 70% for training, 20% for validation, and 10% for testing. As for the other databases, we opted for a distribution of 80% allocated to training, 10% for validation, and 10% for testing, to optimize the learning process.

After the databases were designed, they were exported in Yolov9 and COCO formats. The Yolov9 format allowed us to train the Yolov9 model on other platforms. While the COCO format allowed us to reintegrate the databases converted to grayscale into the Roboflow platform to train the Roboflow 3.0 Object Detection model.

#### 2.3.2. Application of Image Processing Technique: Conversion of Databases to Grayscale

After exporting our databases, they were all converted to grayscale to create

other databases containing only grayscale images using a Python script called “Gray\_Convertor”, written in the Pycharm IDE and which uses the CvtColor method with the BGR2\_GRAY filter from the OpenCV library, as shown in **Figure 3** and **Figure 4**.



**Figure 3.** Pothole image converted to grayscale with BGR2\_GRAY filter.



**Figure 4.** Image of rut converted to grayscale with the BGR2\_GRAY filter.

## **2.4. Development of Detection Models**

During this work, the development of the databases was followed by the development stage of our detection models capable of identifying the degradations that can be listed on the asphalt pavements with their level of severity. To do this, we used the transfer learning technique through the exploitation of two models, including the Yolov9 model and the Roboflow Object Detection model. YOLOv9 and the Roboflow 3.0 model were selected because of their optimal balance between speed and accuracy, as well as their strong performance in transfer learning. These characteristics make them particularly well-suited for road environments, where scenes must be analyzed quickly and with a high level of reliability.

### **2.4.1. YOLOv9 Model**

YOLOv9 model training tasks were run on the Kaggle platform using the Python

language and a T4×2 GPU accelerator. For optimal training of the YOLOv9 model, several hyperparameters were defined. The values of these hyperparameters per training session are summarized in **Table 2**.

**Table 2.** Hyperparameters.

Training	Batch size	Epochs	Initial Learning Rate	Optimizer	Final learning rate
1	16	150	0.01	Adam	0.01
2	32	300	0.01	Adam	0.01
3	32	300	0.01	Adam	0.01
4	32	300	0.01	Adam	0.01

The different training sessions carried out with the YOLOv9 model were evaluated using the confusion matrix and three essential metrics, including:

- **mAP accuracy**
- **Precision P**
- **The R reminder**

#### 2.4.2. Roboflow Object Detection

We trained version 3.0 of the Roboflow Object Detection model on the Roboflow platform. Hyperparameters were already configured by default on the platform to promote optimal model training. Depending on the configuration, two types of this model were available on the platform: Roboflow Object Detection Fast and Roboflow Object Detection Accurate. We opted for the Fast type while using the free version and for the accurate type after upgrading to the paid version. Both options include the use of 300 epochs for training and COCO checkpoints.

To evaluate the performance of our models trained on the Roboflow platform for the object detection case, we used three metrics, including:

- **mAP accuracy**
- **the Precision P**
- **the Recall R**

#### 2.5. Determination of the Extent of Damage

According to the VIZIR method, after obtaining the severity of the degradations, their extent is determined to proceed with the calculation of the indices. The extent of road degradation represents the surface area occupied by the latter in relation to the surface area of the road section examined.

Degradation detection on images is performed by models using bounding boxes. During each prediction, several pieces of information are provided by the model including: the coordinates of the center of the bounding box, the length and width of the box, the confidence score related to the presence of degradation in the box, the name of the detected degradation and the class ID.

To calculate the extent, it was important to establish certain essential parame-

ters to be respected during the collection to obtain the best expected results. After field experiments. The parameters established include a constant collection speed of 10 km/h, a camera viewing angle set at 45° and a shooting height of 1.20 m so that the filmed video covers the entire width of the roadway. These values were selected following field experiments during which several combinations of travel speeds, camera heights and inclination angles were tested. The parameters 10 km/h, 1.20 m and 45° proved to offer the best compromise between image stability, surface detail clarity and adequate coverage of the full pavement width. Respecting these parameters will allow for good quality videos in which the different degradations are more visible.

The video we submit to our model is only a succession of image frames that it analyzes, but calculating the extent at each image frame will result in considering the same degradation several times. For this, our curiosity was to try to know the number of image frames from which we leave the visual field of an image, which allowed us to know at what point in the video the extent will be calculated. After having measured on the ground that the width of an image taken while respecting the previously mentioned parameters occupies 5.5m of the length of the road, we combined this information with the speed set at 10 km/h and an image stream from the camera set to 60 images/second to find the number of image frames  $n_{if} = n_{if} = 119$ .

Therefore, for each of the predictions made after 119 frames of images, our method was to retrieve all the information from the model (the detected degradations with their severity level) and then to store them in a sort of dictionary with a Python script developed in PyCharm and called "Extent\_Calculator". From this dictionary, the rest of the script uses the length and height of each bounding box to calculate the surface  $S_i$  occupied by the degradation on the image.

Let  $E$  be the extent of degradation.

$E = S_i/S_T \times 100$  with  $S_T$  the total area of an image in *pixel*<sup>2</sup> (1080\*1920 in our case).

Extent\_Calculator, therefore, calculates the extent taking this information into account and associates the value found with the degradation in the dictionary.

## 2.6. Determination of the Degradation Index $I_s$

The VIZIR method consists of a standardized visual inspection of the pavement, during which the section to be surveyed is examined to identify all existing distresses. Each distress is classified according to its nature (cracking, rutting, raveling, stripping, etc.), its severity level (low, medium, or high), and its extent, which is estimated either in area or in length, depending on the type. The VIZIR method categorizes distress extent into three classes: 0 - 10%, 10% - 50%, and over 50%. The information regarding extent and severity is then used to estimate the degradation index, which provides an overall rating of the condition of the inspected pavement. The degradation index  $I_s$  is determined by combining three parameters: the cracking index ( $I_f$ ), the deformation index ( $I_d$ ), and a correction factor

related to the presence of repairs on the pavement surface. The resulting degradation index always ranges from 1 to 7, and qualifies the pavement condition over the surveyed section.

Values 1 and 2 of the index  $I_s$  indicate a pavement in good condition, requiring little to no immediate maintenance.

Values 3 and 4 correspond to a fair or moderately deteriorated condition, for which maintenance works should be scheduled regardless of other considerations.

Values 5, 6, and 7 indicate a poor pavement condition requiring major maintenance or rehabilitation works. According to the VIZIR method of road monitoring, the degradation index results from the combination of three parameters, including the cracking index, the deformation index, and the correction related to the presence of repairs on the roadway. All these parameters are determined through a combination of the extent and severity of the degradations as ruts, subsidence, ridges and waners are used to have  $I_d$  and repairs are used to carry out a possible correction. This situation meant tears were not considered (pothole, alopecia, feathering, stripping) in the estimation of the degradation index  $I_s$ .

After every 119 frames of images, the output of the Extent calculator provides us with the extent and name (already including the severity) of each degradation predicted by the model, so the task of determining the indices was relatively easy. To achieve this, we developed a Python code that exploits information in the Extent Calculator dictionary and combines the severity and extent of longitudinal cracks, transverse cracks, and crazing to determine the value of the cracking index  $I_c$ . The same process is performed in the case of deformation to find the value of the deformation index  $I_d$ .

The value of  $I_d$  obtained is then associated with that of  $I_r$  to output the value of the road degradation index in the absence of repairs. If the model has detected repairs, the various information retrieved in the output dictionary of Extent Calculator allows the script to correct the value of the degradation index previously determined.

The calculation of the index  $I_s$  is carried out for each image on which the extents are calculated, at an interval of 119 image frames and therefore on a maximum length of 5.5 m each time. According to the VIZIR method, three types of steps can be adopted for calculation of the index  $I_s$ . We can distinguish:

- the 500 m step, used in the case of a study of road maintenance management support systems, which represents a global study.
- The 200 m step, used in the case of a route maintenance project.
- The 50 m step, used for the test sections.

In the case, 500 m corresponds approximately to 91 images of calculation of the extents and index  $I_s$ , 200 m corresponds to 37 images, and 50 m corresponds to 9 images. We therefore proceeded to an agglomeration of the indices calculated at each 5.5 m to have a value at each of these steps on the route according to what the user would have chosen. An interpretation is then given to the values found according to the VIZIR method interpretations for decision-making regarding the

maintenance of the different sections.

### 2.7. Evaluation of the Operation of ROCNN4, the Extent Calculator, and the *Is* Calculation Script

To evaluate the operation of the process “Detection—Calculation of extent—Calculation of *Is*” carried out by ROCNN4, Extent\_Calculator and the script intended for calculating the surface degradation index *Is*, a new data collection respecting the parameters (video taking the entire width of the road at a constant speed of 10 km/h, a viewing angle of 45° and a height of 1 m 20) was made on two sections of 200 m of road in Benin. The first section is located on the Carrefour Zè-Zè Centre section and the second on the Sèhouè-Massi section. We also carried out manual surveys on these two sections to compare the results of the automatic method with the realities on the ground.

## 3. Model Training Performance

### 3.1. Training the Models with the First Database (Dataset\_831)

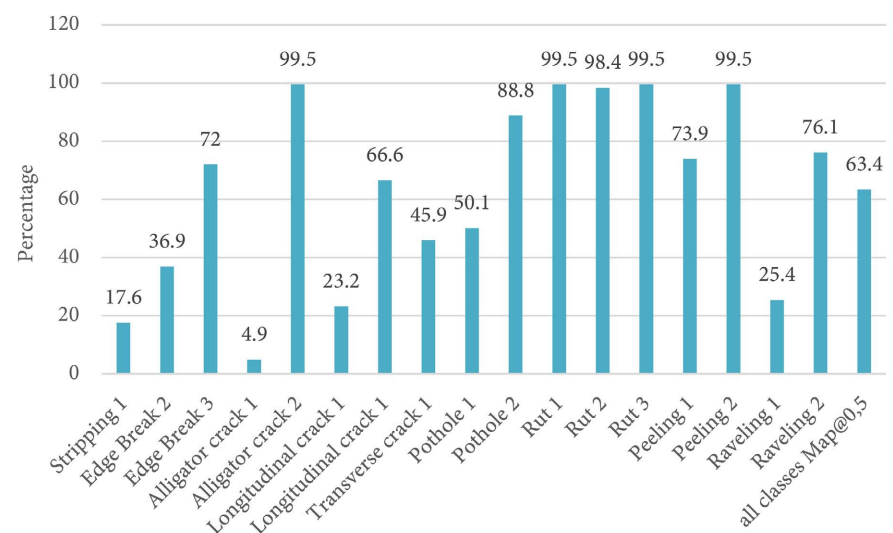
#### 3.1.1. YOLOv9

Dataset 1 was the database that started it all. After training the YOLOv9 model on this database, a model called YOCNN1 is obtained with the following performance in Table 3.

**Table 3.** Performance of YOLOv9 on Dataset\_831.

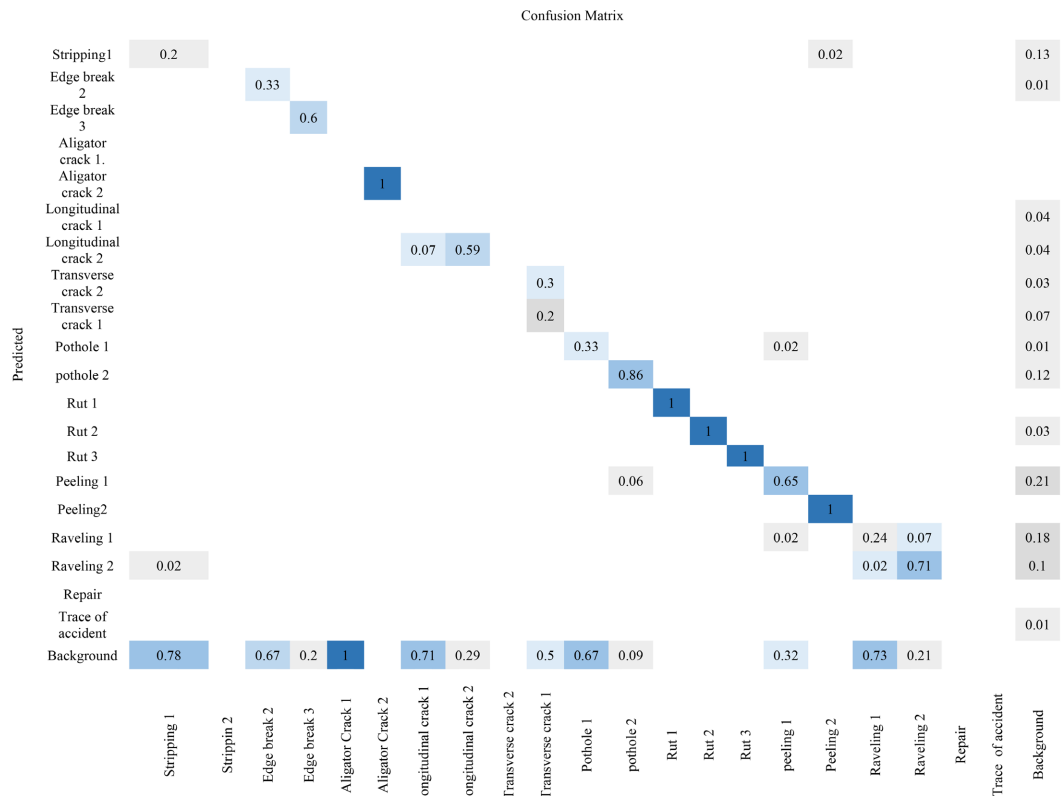
Accuracy P (%)	mAP50 (%)	Recall R (%)
62.9	63.4	63.6

Figure 5 shows the mAp50 accuracy of each degradation class by YOCNN1 on Dataset\_831.



**Figure 5.** mAp50 accuracy of each degradation class by YOCNN1 on Dataset\_831.

This Figure shows very good detection performance of all rut severities, the second severity of alligator, and the second severity of peeling up to 99%. The confusion matrix is as follows in **Figure 6**.



**Figure 6.** YOLOv9 confusion matrix on Dataset\_831.

This matrix allowed us to understand that the model makes more false predictions at the level of degradation, presenting a low performance, and makes several background errors. These background errors are related to the predictions that the model makes at the level of the background of our images, where we left unannotated degradations.

### 3.1.2. Results of Training Roboflow 3.0 Object Detection on Dataset\_831

After training this model with Dataset\_831, a ROCNN1 name model is obtained with the following performance in **Table 4**.

**Table 4.** Performance of Roboflow 3.0 Object Detection on Dataset\_831.

Accuracy P (%)	mAP50 (%)	Recall R (%)
76.8	67.7	58.1

This model shows a 13.9% performance improvement in accuracy compared to the YOCNN1 model. This proves it's slightly more interesting learning and it's good identification of certain degradation classes in **Figure 7**.

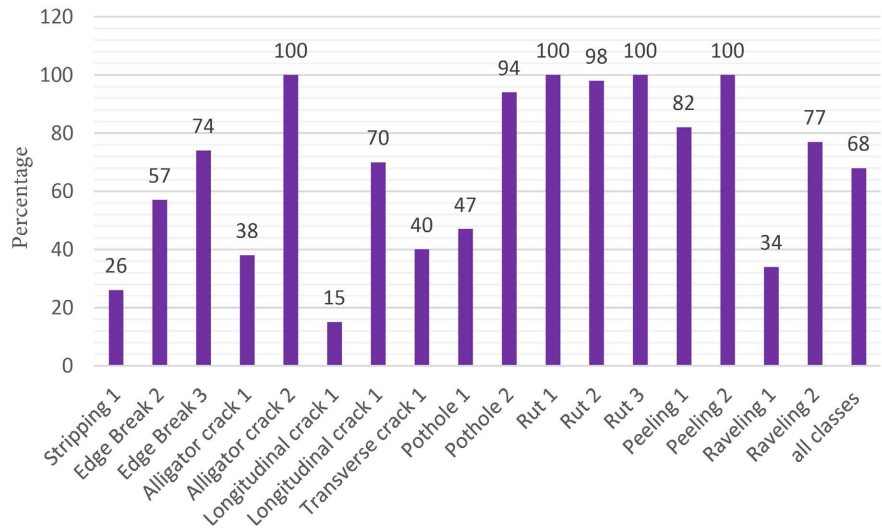


Figure 7. mAP50 accuracy of each degradation class by ROCNN1 on Dataset\_831.

### 3.1.3. YOLOv9 with Dataset\_831 Converted to Grayscale

After converting the first database to grayscale, we were able to use it to train the YOLOv9 model. The performance results are as follows in Table 5 and Figure 8.

Table 5. Performance of YOLOv9 on Dataset\_831 Converted to Grayscale.

Accuracy P (%)	mAP50 (%)	Recall R (%)
66.7	63.7	59.5

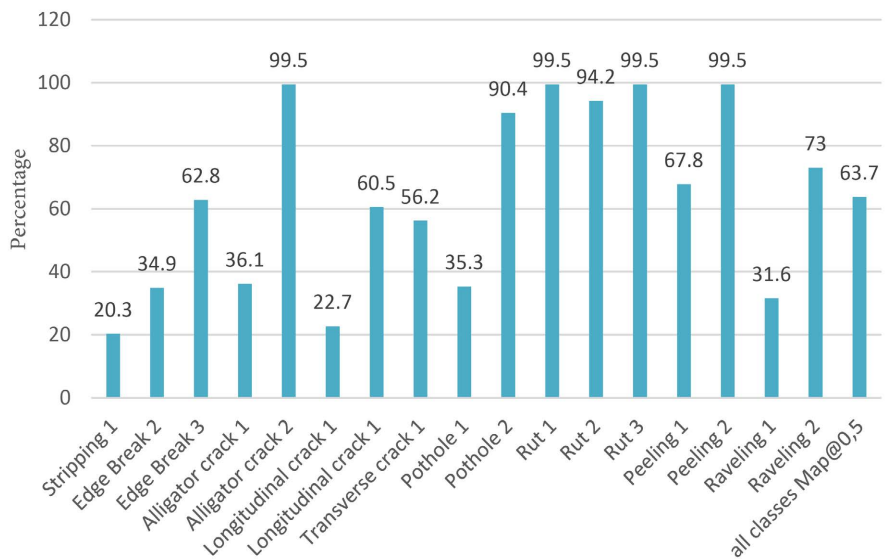


Figure 8. mAp50 accuracy with Dataset\_831 converted to Grayscale.

With the conversion to grayscale, we observe a 4% increase in P accuracy as well as a performance improvement in some degradation classes such as transverse crack 1, stripping 1, alligator crack 1, pothole 2, and raveling 1.

### 3.2. Training the Models with the Second Database (Dataset\_3720)

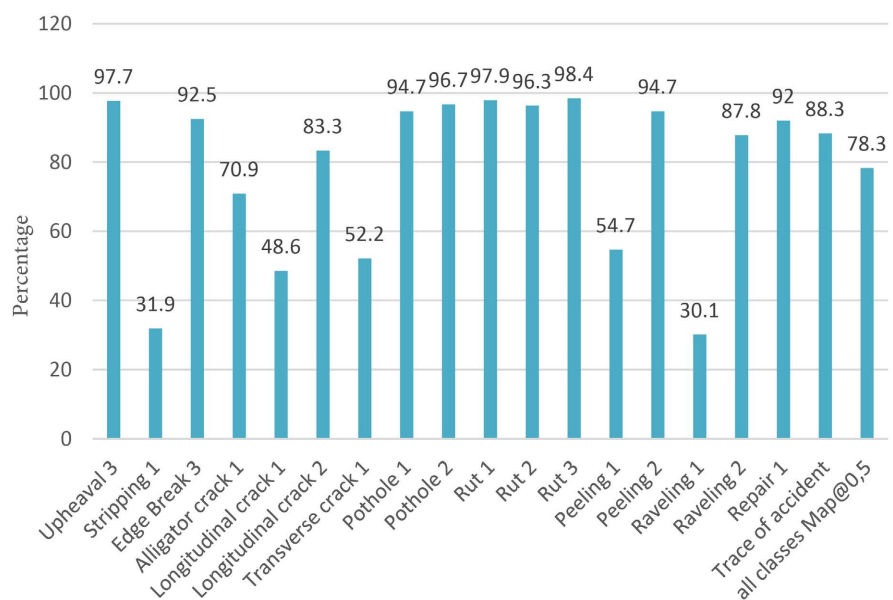
#### 3.2.1. Result of Training with YOLOv9

Training YOLOv9 on this second database allowed us to obtain a model called YOCNN2 with the following performances in **Table 6**.

**Table 6.** Performance of YOLOv9 on Dataset\_3720.

Accuracy P (%)	mAP50 (%)	Recall R (%)
77.9	78.3	76.5

The performance at each degradation class level is observed in **Figure 9**.



**Figure 9.** mAp50 accuracy of each degradation class by YOCNN2 on Dataset\_3720.

There is a slight improvement in performance across the different classes, but it is not as significant as expected. The confusion matrix is as follows in **Figure 10**.

From this matrix, we observe a decrease in the bad predictions made by the model, but the background errors persist.

#### 3.2.2. Results of Training with Roboflow 3.0 Object Detection

ROCNN2 is the model that results from training Roboflow 3.0 Object Detection on Dataset\_3720 with different performances located in **Table 7** and **Figure 11**.

This model achieved performance above 80% in all classes except alligator crack 1, longitudinal crack 1 and transverse crack 1. The addition of the trace of accident to the used classes allowed the model to better learn to identify longitudinal cracks 1, but not yet to perform as expected.

#### 3.2.3. Results of YOLOv9 with Dataset\_3720 Converted to Grayscale

Grayscale conversion of Dataset\_3720 gives the following performance with

YOLOv9 in Table 8.

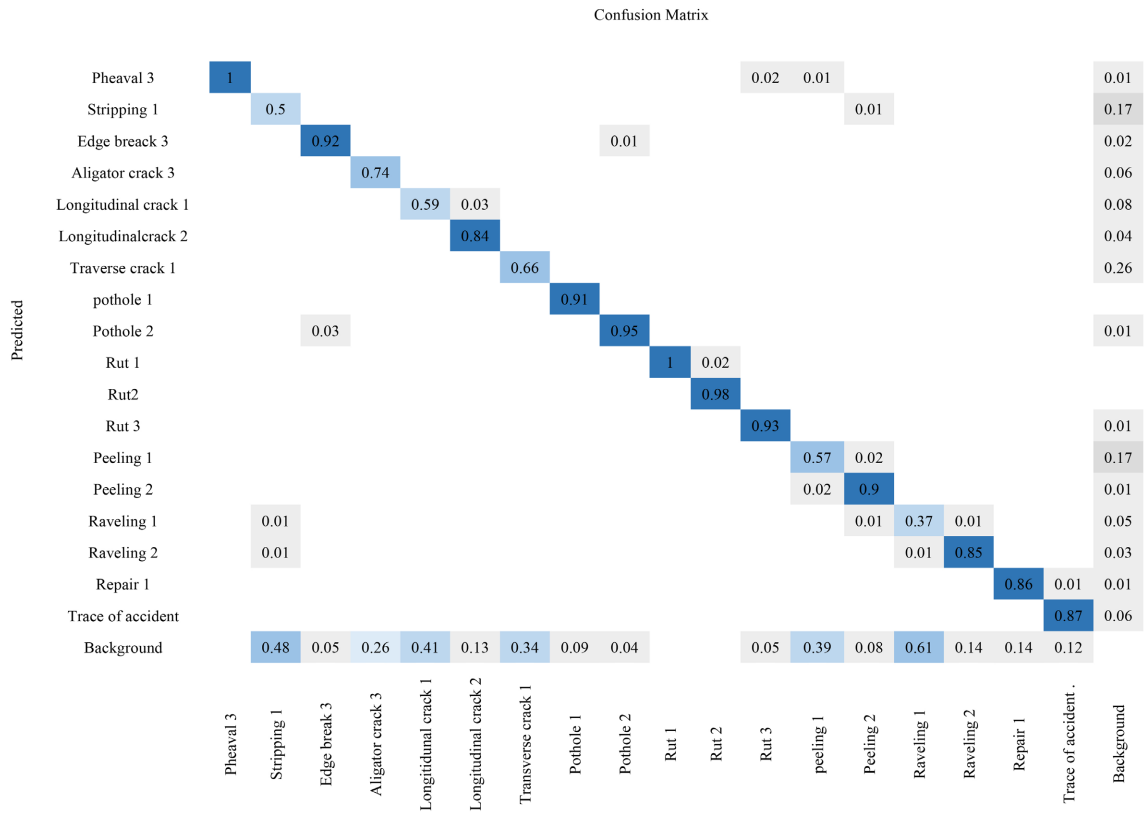


Figure 10. YOLOv9 confusion matrix on Dataset\_3720.

Table 7. Performance of Roboflow 3.0 object detection on Dataset\_3720.

Accuracy P (%)	mAP50 (%)	Recall R (%)
82.3	87.9	86.2

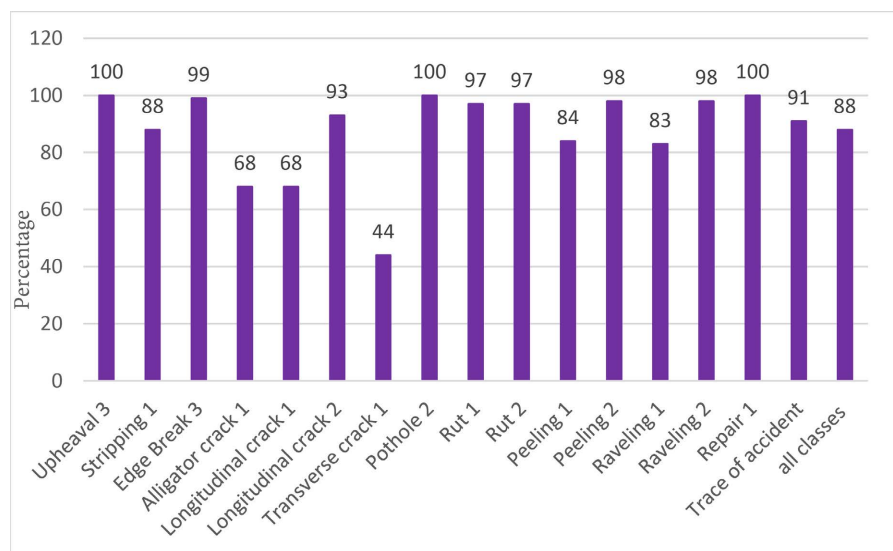
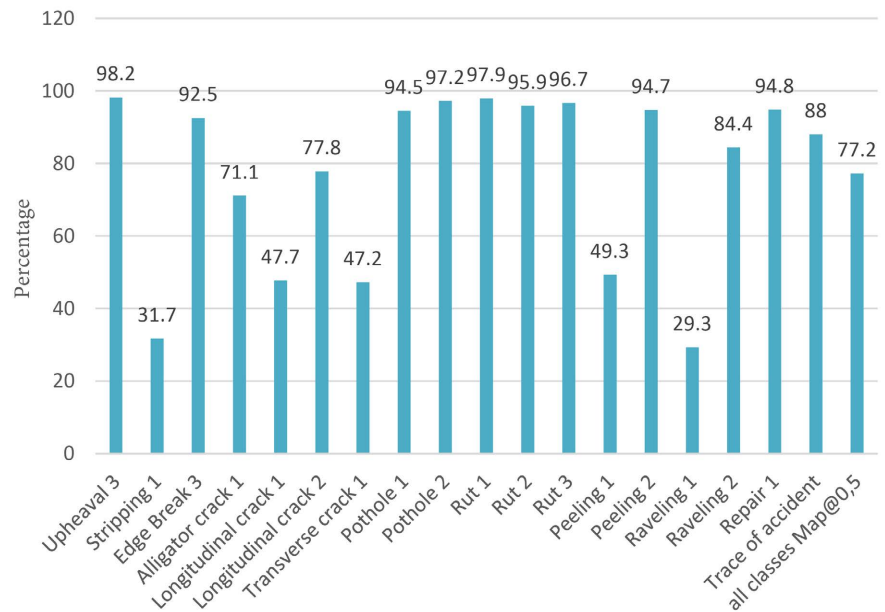


Figure 11. mAP50 accuracy of each degradation class by ROCNN2 on Dataset\_3720.

**Table 8.** Performance of YOLOv9 on Dataset\_3720 converted to Grayscale.

Accuracy P (%)	mAP50 (%)	Recall R (%)
79.8	77.2	77.5

The performance at each degradation class level is observed in **Figure 12**.

**Figure 12.** mAp50 accuracy with Dataset\_3720 converted to Grayscale.

A 2% improvement in overall accuracy is seen with this grayscale conversion. This translates into slight performance improvements across the various degradation classes.

### 3.3. Training the Models with Dataset\_4770

#### 3.3.1. YOLOv9 Model with Dataset\_4770

After training this model on Dataset\_4770, we obtain a model called YOCNN3 with the following performances in **Table 9**.

**Table 9.** Performance of YOLOv9 on Dataset\_4770.

Accuracy P (%)	mAP50 (%)	Recall R (%)
83.2	85.8	82.8

The performance at each degradation class level is observed in **Figure 13**.

This model consistently shows better performance on deformations and averages on other classes, reducing the overall performance to a mAp accuracy of 83.2%. The confusion matrix resulting from model training is shown in **Figure 14**.

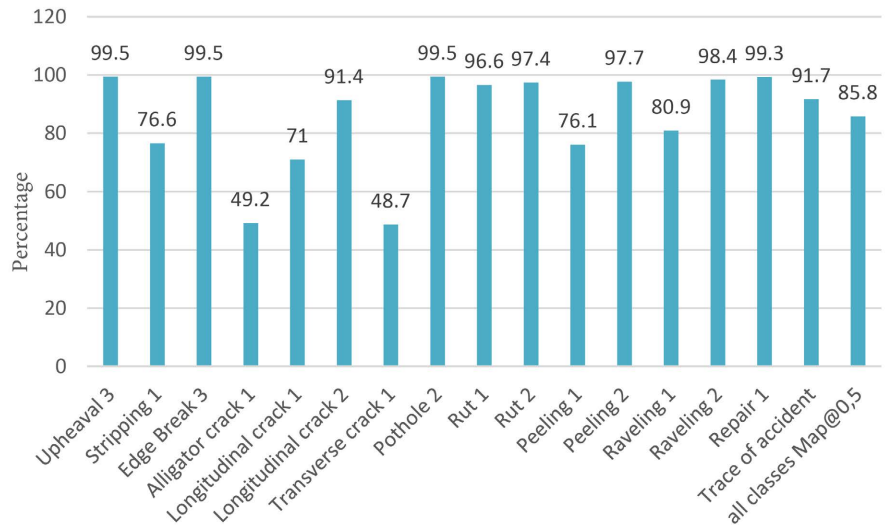


Figure 13. mAp50 accuracy of each degradation class by YOCNN3 on Dataset\_4770.

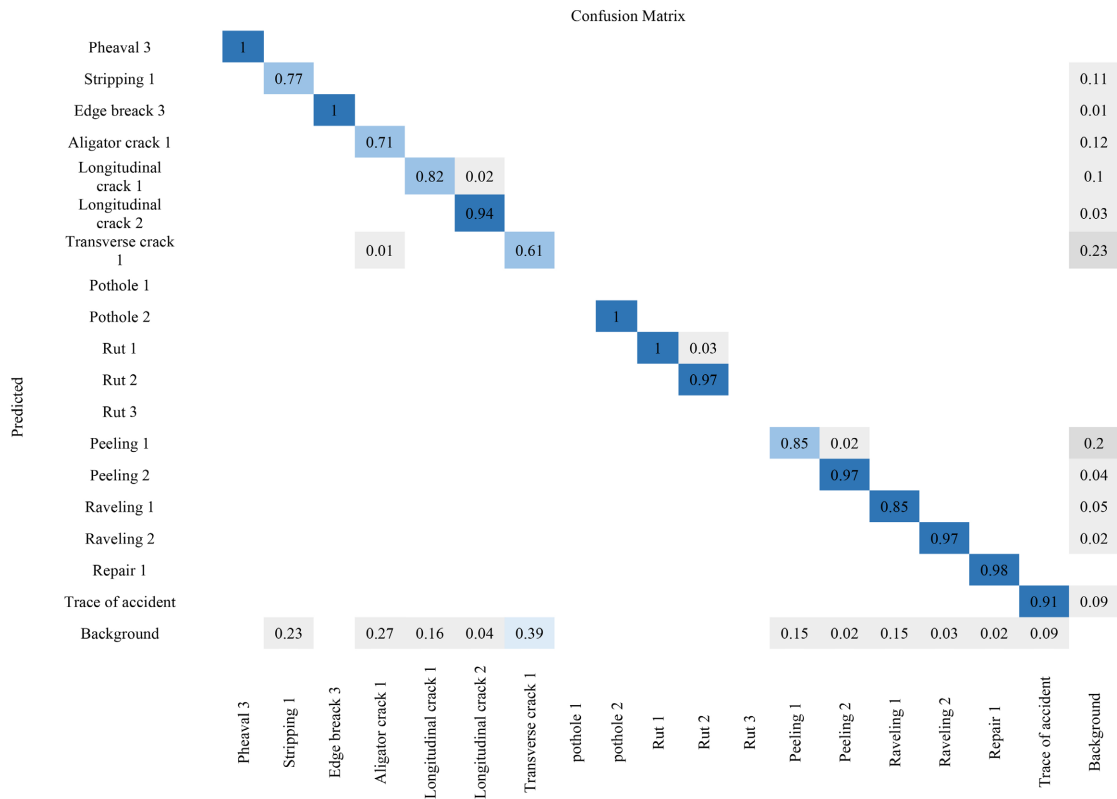


Figure 14. mAp50 YOLOv9 confusion matrix on Dataset\_4770.

The matrix shown in Figure 14 concerns the model’s ability to identify degradations with fewer false predictions. Similarly, background errors have decreased, but not enough to help us achieve our goal.

### 3.3.2. Roboflow 3.0 Object Detection Model with Dataset\_4770

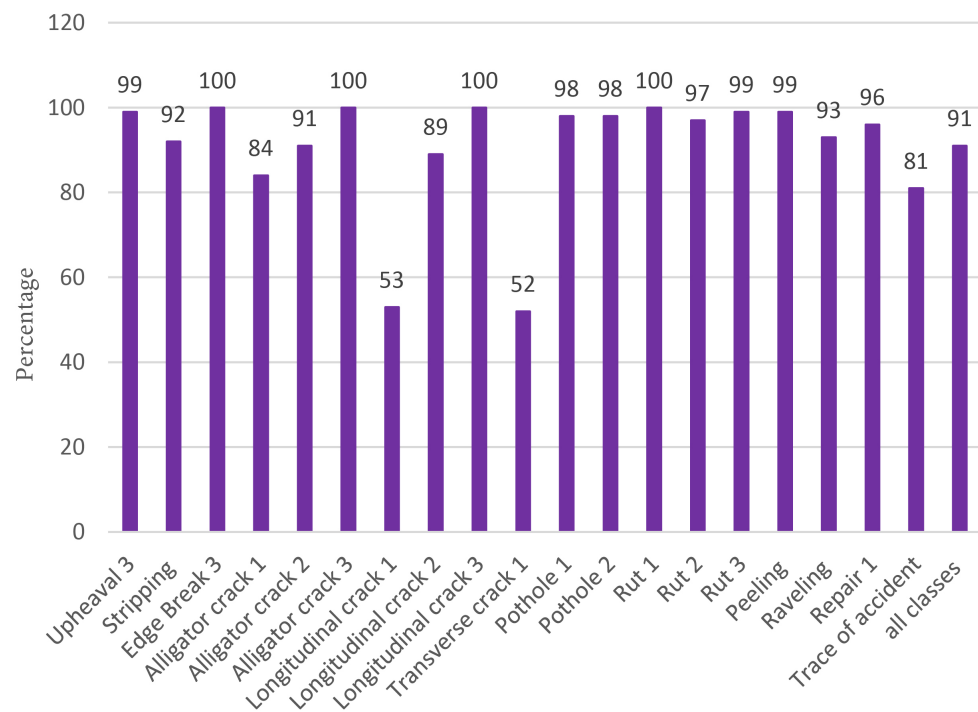
The ROCNN3 model, resulting from training Roboflow 3.0 Object Detection with

Dataset\_4770, exhibits the following performance, as shown in **Table 10**.

**Table 10.** Performance of Roboflow 3.0 object detection on Dataset\_4770.

Accuracy P (%)	mAP50 (%)	Recall R (%)
87.4	90.5	81.2

The different detection performances of each degradation class can be seen in **Figure 15**. **Figure 15** shows better learning of the model, which allows it to reach 90% at the mAP level. All deformations and tears present very good performances. Alligator cracks and cracks are also well identified by the model, except for longitudinal cracks 1 and transverse cracks 1, at which the model remains average. The confusion matrix resulting from model training is shown in **Figure 16**.



**Figure 15.** mAP50 accuracy of each degradation class by ROCNN3 on Dataset\_4770.

This matrix shows that background errors are high at the level of track accident traces then longitudinal cracks 1 and transverse cracks 1. The model also predicts more false negatives at the level of these degradations, which justifies the average performance observed.

With a Roboflow feature that uses the confusion matrix to identify images that pose an annotation problem or an error on the part of the model, we were able to see that the model predicts longitudinal 1 and transverse 1 cracks found inside the alligator crack. It was therefore necessary to review all our annotations at the level of these two types of degradation to improve Dataset\_4770 for future training.

GROUND TRUTH	Pheaval 3	Stripping	Edge break 3	Alligator crack 1	Alligator crack 2	Alligator crack 3	Longitudinal crack 1	Longitudinal crack 2	Longitudinal crack 3	Transverse crack 1	pothole 1	Pothole 2	Rut 1	Rut 2	Rut 3	Peeling	Raveling	Repair 1	Trace of accident	False positive	
	Pheaval 3	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stripping	0	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	1	0	224
Edge break 3	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
Alligator crack 1	0	0	0	31	19	0	0	3	0	0	0	0	0	0	0	0	0	0	0	2	53
Alligator crack 2	0	0	0	0	161	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	111
Alligator crack 3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Longitudinal crack 1	0	0	0	0	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0	1	89
Longitudinal crack 2	0	0	0	0	0	0	0	61	1	0	0	0	0	0	0	0	0	0	0	1	39
Longitudinal crack 3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Transverse crack 1	0	0	0	0	1	0	1	0	0	59	0	0	0	0	0	0	0	0	0	0	181
Pothole 1	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	3
Pothole 2	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	39
Rut 1	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	46
Rut 2	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	70
Rut 3	8	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	42
Peeling	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	1	0	0	35
Raveling	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	31	0	0	0	155
Repair 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	0	0	27
Trace of accident	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	15	15	191
False positive	7	11	5	0	21	2	36	6	0	60	3	6	0	0	0	6	7	16	70	0	0

Figure 16. Confusion matrix of Roboflow 3.0 object detection on Dataset\_4770.

### 3.3.3. Roboflow 3.0 Object Detection with Dataset\_4770 Converted to Grayscale

From the third database, we discovered that exporting our databases in COCO format will allow us to reintegrate the grayscale-converted versions of these databases into the Roboflow platform to train the Roboflow 3.0 Object Detection model. Dataset\_4770 has therefore undergone this process and gives the following performances in Table 11 after training.

Table 11. Performance of Roboflow 3.0 object detection on Dataset\_4770 converted to Grayscale.

Accuracy P (%)	mAP50 (%)	Recall R (%)
83.8	89.5	85.8

The performances obtained at the level of each degradation class can be seen in Figure 17.

This new database converted to grayscale increases the recall R by more than 4%, but the precision P and the mAP decrease. This explains the oscillation observed within the different classes, where some show an improvement and others a decrease. However, all the metrics are better compared to those obtained during the training of Dataset\_4770 with YOLOv9.

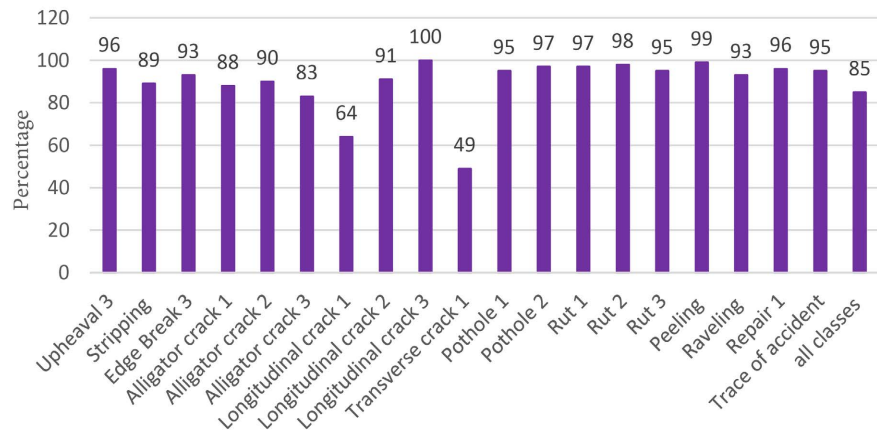


Figure 17. mAP50 accuracy with Dataset\_4770 converted to Grayscale.

### 3.4. Training the Models with the Dataset\_4770A

#### 3.4.1. Roboflow 3.0 Object Detection Models with the Dataset\_4770A

Dataset\_4770A is used to train the Roboflow 3.0 Object Detection model to obtain ROCNN4 with the performances in Table 12.

Table 12. Performance of Roboflow 3.0 object detection on Dataset\_4770A.

Accuracy P (%)	mAP50 (%)	Recall R (%)
90.8	91.8	89.5

Details of the training and the different performances of the degradation classes can be seen in Figure 18.

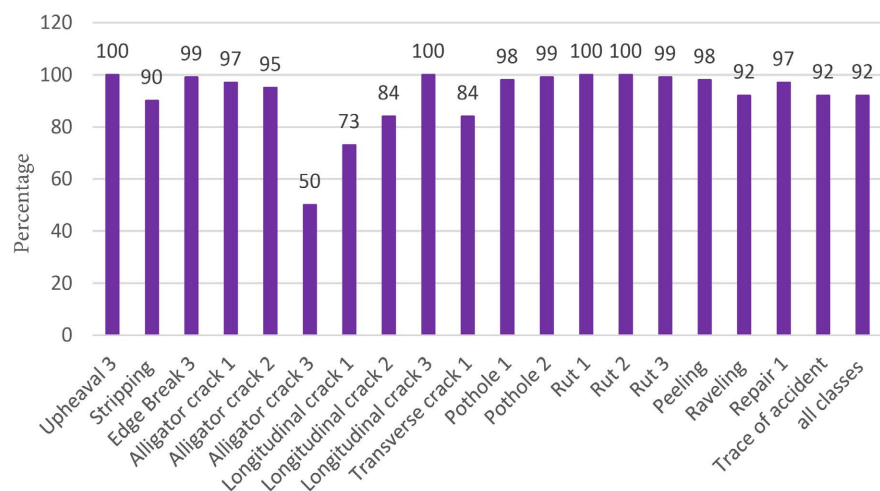
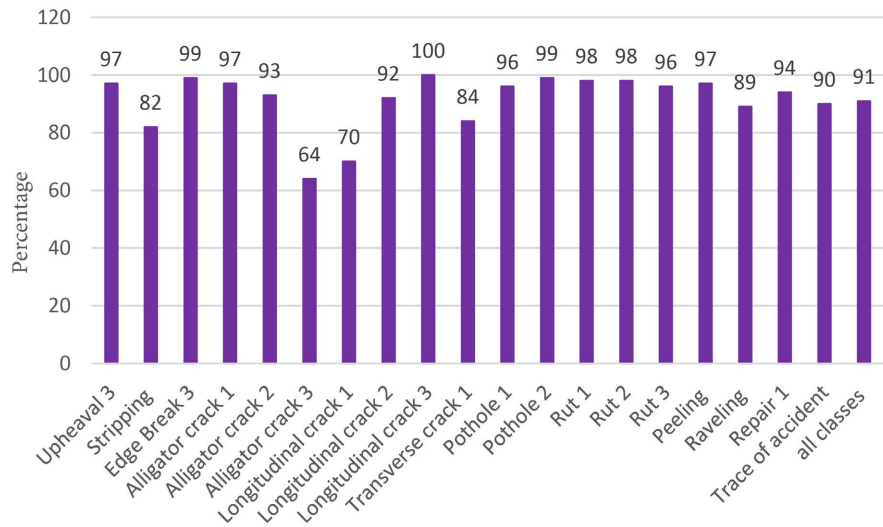


Figure 18. mAP50 accuracy of each degradation class by ROCNN4 on Dataset\_4770A.

Except for alligator crack 3, this model learned to recognize each of the degradations very well, which allows it to have an overall accuracy stabilized above 90% during training. The confusion matrix resulting from this training is as follows in

**Figure 19.**



**Figure 19.** mAp50 accuracy with Dataset\_4770A converted to Grayscale.

This matrix shows the model’s ability to make correct predictions while minimizing errors across all classes.

### 3.4.2. Roboflow 3.0 Object Detection with Dataset\_4770A Converted to Grayscale

The grayscale converted version of Dataset\_4770A trained with the Roboflow 3.0 Object Detection model shows the following performance in **Table 13**, in **Figure 19** and **Figure 20**.

**Table 13.** Roboflow 3.0 object detection on Dataset\_4770A converted to Grayscale.

Accuracy P (%)	mAP50 (%)	Recall R (%)
86.5	91.2	86.3

This grayscale conversion keeps mAP accuracy above 90%. Other metrics decrease.

### 3.5. Comparison of Model Accuracy on the Four Databases

**Figure 21** shows the performance of the four datasets after training. It is noted that accuracy grows with the number of images used for training. After the conversion to grayscale (performed in the PyCharm IDE), which offers superior performance to that of YOLOv9 trained with images coming directly from the collections, the Roboflow 3.0 Object Detection model is the one that demonstrates the best accuracy with its training on the unconverted Dataset\_4770A.

The best performing model that we obtained during this work, therefore, represents the ROCNN4 model, which manages to detect 19 classes of degradation with a precision P = 90.8%.

	Pheaval 3	Stripping	Edge breack 3	Aligator crack 1	Aligator crack 2	Aligator crack 3	Longitudinal crack 1	Longitudinal crack 2	Longitudinal crack 3	Transverse crack 1	pothole 1	Pothole 2	Rut 1	Rut 2	Rut 3	Peeling	Raveling	Repair 1	Trace of accident	False positive	
Pheaval 3	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	67
Stripping	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	231
Edge breack 3	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32
Aligator crack 1	0	0	0	19	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51
Aligator crack 2	0	0	0	0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	159
Aligator crack 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Longitudinal crack 1	0	0	0	0	5	0	28	0	0	0	0	0	0	0	0	0	0	0	0	1	108
Longitudinal crack 2	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	70
Longitudinal crack 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Transverse crack 1	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	105
Pothole 1	0	0	0	0	0	0	0	0	0	0	43	0	0	0	0	0	0	0	0	0	20
Pothole 2	0	0	0	0	0	0	0	0	0	0	0	74	0	0	0	0	0	0	0	0	44
Rut 1	0	0	0	0	0	0	0	0	0	0	0	0	29	7	0	0	0	0	0	0	15
Rut 2	0	0	0	0	0	0	0	0	0	0	0	0	2	33	0	0	0	0	0	0	22
Rut 3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	24
Peeling	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	39
Raveling	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	154
Repair 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	0	47
Trace of accident	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	59	477
False positive	3	0	3	1	3	0	9	0	0	1	1	2	0	2	0	2	0	4	11	0	0

Figure 20. Roboflow 3.0 object detection confusion matrix on Dataset\_4770A.

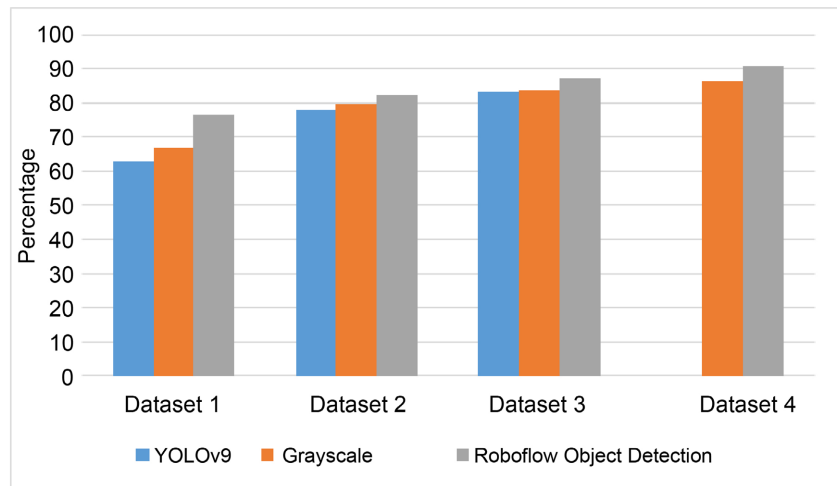


Figure 21. Comparison of model accuracy on the four datasets.

### 3.6. Results Obtained with the Evaluation Collection

#### 3.6.1. First Portion of 200 m (Axis Carrefour Zè-Zè Centre)

After submitting the video taken on this first 200m section to the ROCNN4—Extent\_Calculator—Is Calculation Script set, we obtain the results recorded in

**Table 14.****Table 14.** Value of Index on the road section (axis Carrefour Zè-Zè Centre).

	Developed method index	Manual survey index
Portion cracking index	0	0
Portion deformation index	0	0
Degradation index $I_s$	1	1
Corrected $I_s$	1	1

The model detects only tear-type damage in this section. The extent of each degradation, as well as the  $I_s$  indices per image, is obtained, and a final value of the degradation index equal to 1 is found by aggregation. This value corresponds to good surface condition, not requiring immediate maintenance. After carrying out manual surveys of the damage in this section, we obtained the results. These survey results show that the model correctly detects most of the damage present on the first section of the road. Similarly, the absence of cracks and deformations led to an index of degradation value equal to 1. This corresponds to what the automatic method found and confirms its proper functioning.

### 3.6.2. Validation of the Developed Method on the Second Section of Sèhouè-Massi (RNIE 2)

The results of the automatic and manual methods on the second portion are shown in **Table 15**.

**Table 15.** Value of Index on the road section (Sèhouè-Massi).

	Developed method index	Manual survey index
Portion cracking index	3	3
Portion deformation index	3	2
Degradation index $I_s$	5	4
Corrected $I_s$	5	5

In addition to tearing, the model detects deformations, cracks, and crazing. All the extents as well as the  $I_s$  indices per image are obtained, then the aggregation allows for a surface degradation index value of the portion equal to 5. This value corresponds to a poor surface condition requiring major maintenance.

These surveys confirm the presence of tears, deformations, cracks, and alligator cracks on the second section of the road, which our model correctly detected. The value of the degradation index before correction gave us 4 with the manual surveys, but 5 with the automatic method. However, this value returns to the same (Corrected  $I_s = 5$ ) for both methods after correction and corresponds to a poor surface condition.

At the level of the two sections of road used, the results that we obtained with

the manual surveys and the automatic method are similar, which show the proper functioning of our model as well as the scripts for calculating the extent and the surface degradation index, *Is*.

### 3.6.3. Model Limitations and Application Constraints

Despite the encouraging performance of the system, certain limitations must be highlighted. The detection accuracy remains sensitive to variations in lighting conditions (strong brightness, backlighting, shadowed areas), as well as adverse weather conditions such as rain, humidity or the presence of standing water on the pavement. These factors can degrade image quality and reduce the model's ability to distinguish certain types of distress. In addition, the robustness of the model could be improved to better handle environments with significant dust, debris or highly irregular surface textures. These elements represent avenues for expanding the dataset and further optimizing the model.

## 4. Conclusions

This work represents an important step towards the implementation of more efficient tools to facilitate the execution of road monitoring. The aim was to be able to develop a better-performing digital tool based on artificial intelligence and an image processing technique to detect road surface damage with its severity, with a minimum of 90% accuracy and then estimate its extent with a view to determining the degradation index *Is*. To achieve these objectives, we have implemented and followed a rigorous methodological approach. The development of scripts such as Extractor, Resizer, and Normalizer permitted to extract several images from our videos collected on various roads in Benin and prepare them to constitute four databases used to train two models, including YOLOv9 and Roboflow 3.0 Object Detection. ROCNN4 remains the best-performing model resulting from this work and can detect 19 degradation classes with their severity level at an accuracy of 90.8%. A script called Extent\_Calculator was then developed to estimate the extent of the different degradations detected by the model using the prediction results and three parameters fixed for data collection, including:

- A constant collection speed of 10 km/h
- A grip height of 1.20 m
- A 45° viewing angle for the camera to film the entire width of the roadway

Using the severity and extent of the recorded degradations, our last script of this work determines the degradation index after calculating the cracking and deformation indices and then making a possible correction related to the presence of repairs on the roadway examined. This entire process was validated by a new data collection carried out on two sections of roadway, each 200 m long.

## Data Availability

The four datasets used here are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] African Union (2021) Report on the Status of Implementation of the African Road Safety Action Plan (2011-2020) towards a Post-2020 African Road Safety Strategy.
- [2] Mubarak, A.S., Ameen, Z.S. and Al-Turjman, F. (2023) Effect of Gaussian Filtered Images on Mask RCNN in Detection and Segmentation of Potholes in Smart Cities. *Mathematical Biosciences and Engineering*, **20**, 283-295. <https://doi.org/10.3934/mbe.2023013>
- [3] Rafi, F.A., Fanggidae, A. and Polly, Y.T. (2023) Asphalt Road Damage Detection System Using Canny Edge Detection. *Jurnal Komputer dan Informatika*, **11**, 85-90. <https://doi.org/10.35508/jicon.v11i1.10100>
- [4] Guo, L., Li, R. and Jiang, B. (2021) A Road Surface Damage Detection Method Using YOLOV4 with PID Optimizer, *International Journal of Innovative Computing, Information and Control*, **17**, 1763-1774.
- [5] DGSI, & DERPR (2018) Benin Road Maintenance Manual. Ministry of Infrastructure and Transport.
- [6] Kothai, R., Prabakaran, N., Srinivasa Murthy, Y.V., Reddy Cenkeramaddi, L. and Kakani, V. (2024) Pavement Distress Detection, Classification, and Analysis Using Machine Learning Algorithms: A Survey. *IEEE Access*, **12**, 126943-126960. <https://doi.org/10.1109/access.2024.3455093>
- [7] Kaddah, W. (2019) Apports de nouveaux outils de traitement d'images et de programmation pour le relevé automatique de dégradations sur chaussées. Université de Bretagne Occidentale.
- [8] Seo, H., Shi, Y. and Fu, L. (2024) Automatic Damage Detection of Pavement through Darknet Analysis of Digital, Infrared, and Multi-Spectral Dynamic Imaging Images. *Sensors*, **24**, Article 464. <https://doi.org/10.3390/s24020464>
- [9] Laurent, J. (2010) Pavemetrics. LCMS: Laser Crack Measurement System, 1-43.
- [10] Ullah, A., Zhaoyun, S., Tariq, U., Uddin, M.I., Khatoon, A. and Rizvi, S.S. (2022) Gray-Level Image Transformation of Paved Road Cracks with Metaphorical and Computational Analysis. *Mathematical Problems in Engineering*, **2022**, 1-14. <https://doi.org/10.1155/2022/8013474>
- [11] Dhieb, N., Ghazzai, H., Besbes, H. and Massoud, Y. (2019) A Very Deep Transfer Learning Model for Vehicle Damage Detection and Localization. 2019 31st *International Conference on Microelectronics (ICM)*, Cairo, 15-18 December 2019, 158-161. <https://doi.org/10.1109/icm48031.2019.9021687>
- [12] Mirbod, M. and Shoar, M. (2023) ScienceDirect Intelligent Concrete Surface Cracks Detection Using Computer Vision, Pattern Recognition, and Artificial Neural Networks. *Procedia Computer Science*, **217**, 52-61. <https://doi.org/10.1016/j.procs.2022.12.201>
- [13] Qayyum, W., Ehtisham, R., Bahrami, A., Camp, C., Mir, J. and Ahmad, A. (2023) Assessment of Convolutional Neural Network Pre-Trained Models for Detection and Orientation of Cracks. *Materials*, **16**, Article 826. <https://doi.org/10.3390/ma16020826>
- [14] Yabi, C.P., Gbehoun, G.F., Tamou, B.C.K., Alamou, E., Gibigaye, M. and Farsangi, E.N. (2025) Analysis and Classification of Distress on Flexible Pavements Using Con-

- volutional Neural Networks: A Case Study in Benin Republic. *Infrastructures*, **10**, Article 111. <https://doi.org/10.3390/infrastructures10050111>
- [15] Shatnawi, N., Obaidat, M.T. and Al-Mistarehi, B. (2021) Road Pavement Rut Detection Using Mobile and Static Terrestrial Laser Scanning. *Applied Geomatics*, **13**, 901-911. <https://doi.org/10.1007/s12518-021-00400-4>
- [16] Azam, A., Alshehri, A.H., Alharthai, M., El-banna, M.M., Yosri, A.M. and Beshr, A.A.A. (2023) Applications of Terrestrial Laser Scanner in Detecting Pavement Surface Defects. *Processes*, **11**, Article 1370. <https://doi.org/10.3390/pr11051370>