Pothole and patch detection on asphalt pavement using deep convolutional neural network

Aline Calheiros Espíndola[1,2], Gabriel Tavares de Melo Freitas[2,3], Ernesto Ferreira Nobre Júnior[2]

[1]*Technology Center, Federal University of Alagoas*
Av. Lourival Melo Mota, S/N, Tabuleiro do Martins, Zip Code: 57072-970, Maceió, Alagoas, Brazil
[2]*Dept. of Transport Engineering, Federal University of Ceará*
Av. Mister Hull, s/n – Pici, Zip Code: 60455-760, Fortaleza, Ceará, Brazil
[3]*Dept. of Computer Engineering, Federal Institute of Ceará*
Rod. Pres. Juscelino Kubitschek - BR020 - Km 209, Zip Code: 63870-000, Boa Viagem, Ceará, Brazil
aline.espindola@ctec.ufal.br, aline@det.ufc.br, gabriel.tavares@ifce.edu.br, nobre@ufc.br

Abstract. The main obstacles to the widespread use of the PMS are the high financial and time costs for carrying out on-site assessments and the difficulty of processing and analyzing the data to generate the diagnoses of the current condition of the pavement. With technological advancement, some techniques such as computer vision, image processing, and machine learning can automatically extract the information of the pavements' condition. The present study proposes the exclusive use of images from cameras attached to a vehicle, simple collection and reduced cost, and extraction of information on pavement defects using a CNN. The research developed object detection models with YOLO architecture to identify potholes and patches. It was analyzed the metrics impact of the image size (224x224, 320x320, 416x416 pixels) and number of iterations for Yolo version 3 and 4. As expected, the increasing image size resulted in improved metrics results and the expansion of the iterations led to an improvement in the IoU. The CNN that presented the best overall performance, combining all the metrics, was based on Yolov3, with an image size of 416x416 and 6000 iterations training, in which it obtained an F1-score of 79.00%, an average IoU of 64.59%, and mAP@0.50 of 73.85%.

Keywords: Pavement Defect, Convolutional Neural Network, Yolo.

1 Introduction

The pavement management system (PMS) has been used for years to support managers in making decisions to maintain and extend the useful life of pavements and plan interventions and the necessary resources in the short, medium, and long term. Pavement management aims to effectively apply financial resources, providing comfortable, safe, and economical pavements to users. In other words, it allows for the best allocation of resources, following a prioritization system through the assessment of the pavement, minimizing maintenance and rehabilitation costs. However, the acquisition of information and data is the bottleneck for implementing and using the PMS. The pavement assessment can be manually or automatically. The manual approach is falling out of favor due to its subjectivity, high financial costs, time, and human resources. Manual inspection is becoming impractical due to the required scale and frequency [1]. Depending on the size of the road network, the execution of emergency services, such as stopgap operations (fill the potholes), is made difficult due to the lack of information such as the number of potholes to estimate costs and location to plan the execution of interventions.

The evolution of science and technology, together with reducing the cost of electronic component manufacturing, enables the investigation of new techniques and tools that objectively and automatically perform detection. Due to the above, universities and research centers collaborate to obtain clear images of roads and detect any road deterioration. At present, different techniques aim to detect the deterioration of road surfaces, for example, using a laser, vibration sensors, and images. The development of techniques based on image processing has prompted research into machine learning methods capable of detecting different types of deterioration of road surfaces. [2]

In all the investigations that offered a solution to detection through pattern recognition techniques, an algorithm was constructed that is capable of image recognition and the elimination of the problems that traditional deep learning solutions entail—the You Only Look Once (YOLO) algorithm. YOLO is an algorithm that works in real-time and computationally limited devices since only a single forward propagation through the neural network is necessary to determine a prediction. In this sense, the research aims to develop a pothole and patch detector, capable of identifying pavement defects in the highways from images obtained from evaluation vehicles through deep learning solutions and the You Only Look Once (YOLO) algorithm.

2 Literature Review

Deep learning methods aim at learning hierarchies of features with features from higher levels of the hierarchy that are formed by a combination of lower-level features. Learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output directly from data, without depending on human-crafted features [3]. Deep learning technologies have achieved success in various computer vision tasks such as image classification, object detection, and image segmentation.

According Russakovsky et al. (2015) image classification, Figure 1 (A), involves predicting the class of one or more objects in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. Object detection, Figure 1 (B), combines these two tasks: localizes and classifies one or more objects in an image. Segmentation assigns a label or a score to each pixel in the image, Figure 1 (C).

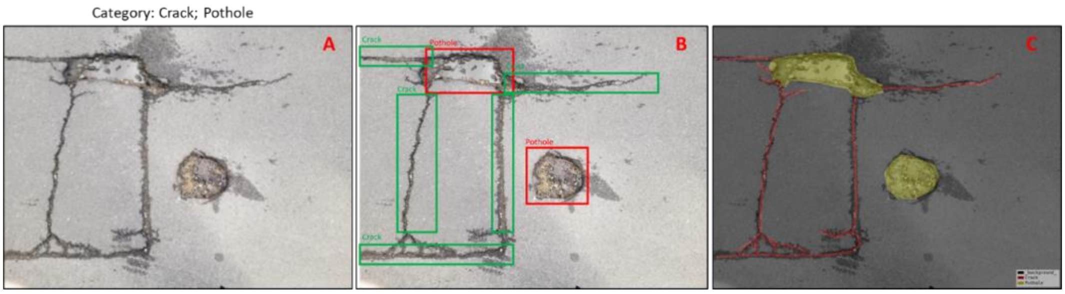


Figure 1. Example of cracks and potholes identifications by the methods (A) Multi-label Classification, (B) Detection and (C) Segmentation.

Methods for object detection generally fall into either machine learning-based approaches or deep learning-based approaches. For Machine Learning approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification. On the other hand, deep learning techniques can do end-to-end object detection without specifically defining features and are typically based on convolutional neural networks (CNN).

The main object detection based on deep learning architecture are R-CNN [5], Fast R-CNN [6], Faster R-CNN [7], cascade R-CNN [8], Single Shot MultiBox Detector (SSD) [9], You Only Look Once (YOLOv4) [10], [11], [12], and Retina-Net [13].

Several researches have been developed for each computer vision task, within the scope of pavement assessment. How the present work is focused on using object detection as an approach to identify and count the defects in the pavement, so Table 1 presents research developed using object detection as an identification method.

Table 1. Research is on pavement assessment using deep learning to object detection.

Defetcs	**Research's**
Pothole	[2], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] and [29]
Crack	[2], [14], [15], [16], [17], [18], [19], [20], [21], [29] and [30]
Other	[15], [14], [16], [17], [18], [19], [29], [31] and [32]

Cracks and potholes were the distress more researched, and the primary architecture model used was YOLO, as shown in Table 2.

Table 2. The deep learning approaches used for pavement defect detection

Architecture		Research's
YOLO	V1	[18]
	V2	[14], [17], [22], [23], [26] and [29]
	V3	[15], [19], [21], [24] and [28]
	V4	[2] and [20]
Faster R-CNN		[2], [16], [17], [30], [21], [24], [29] and [32]
SSD		[17], [30], [21], [24], [27], [28] and [31]
Other		[17], [20], [22], [24], [25] and [27]

YOLO has been evolving in its different versions, including YOLOv3 and YOLOv4, the latest stable versions. The difference between the versions is that YOLOv3 uses characteristic pyramid networks (FPN) for object detection, whereas YOLOv4 uses PANet as a method of aggregating parameters for different detection levels together with an increase in average precision (AP) and frames per second (FPS), a feature that makes the accuracy of YOLOv4 much higher than that of YOLOv3. [2]

3 Methods

3.1 Dataset

The datasets images are from two state highways located in Ceará state, Brazil. All highways have hot mix asphalt as the surface with granulometric composition "C" range by the National Department of Transport Infrastructure [33]. The right of way (ROW) video images with 1920x1080 were captured by a survey vehicle by Garmin VIRB Ultra30 Action camera. Defects found in the pavement's images are, potholes and patches in large numbers of images and a few cracks, due to the amount of crack on the highway available being low would not have enough images for the model training, so it was decided to create the detector with the pothole and patch labels.

It was select 360 images to label, with the aid of the *labelImg* software, the process of labelling the defects in each image was performed, as shown in Figure 2. In general, the images had more than one defect, and the select images produced a total of one thousand labels, 500 for each class, pothole, and patch. During the CNN training, 80% of the images are used for the training set and 20% for the validation set, each chosen randomly.

Figure 2 - *Labeling* of defects in the pavement

3.2 Proposed Procedure

Training is a procedure involving mathematical optimization, involving a cost function, called loss function, so that the neural network's weights and biases can be automatically tuned. In this process, the training data with expected labels are presented to the optimization procedure to find a global minimum loss function.

CNN training has massive computational power consumption. Therefore, it is necessary to use GPUs that can process data in a parallel, the other alternatives are field-programmable gate arrays or application-specific integrated circuits. For this purpose, the computing resources of Google Colab are used to conduct CNN training. Due to the nature of Google Colab, each training is performed on a randomly allocated GPU, for the models two GPU were used, Tesla P100-PCIE-16GB and Tesla V100-SXM2-16GB.

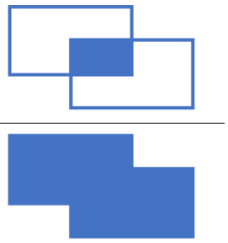
The models are based on transfer learning, and the backbones used for training were CSPDarknet53 to Yolov4 and Darknet-53 to Yolov3, images size tested were 224x224, 320x320, and 416x416 pixels, the batch size of 64 and subdivisions of 16, and binary cross-entropy with logistic activation (sigmoid) as the loss function for the class predictions. For all models developed were used the same optimization parameters, momentum used was 0.9, decay of 0.0005 and learning rate (LR) 0.001.

The number of interactions recommended by the Yolo developer is 2000 iterations for each class, with a minimum number of 6000 iterations [10]. During the development of this research, the progressive expansion of the iterations number of training was analyzed, 1000 of range, with a minimum value of 1000 to a maximum value of 8000 (double the recommended by developer [10]).

Five different metrics were used, intersect over union (IOU), precision, recall and average precision and mean average precision (mAP).

Intersect Over Union (IOU) can be computed as Area of Intersection divided over Area of Union of two boxes, so IOU must be ≥0 and ≤1. When predicting bounding boxes, we need the find the IOU between the predicted bounding box and the ground truth box to be ~1.

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \qquad (1)$$



Precision can define precision as the ratio of true positive - TP (true predictions) and the total number of predicted positives (total predictions). The formula is given as such:

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

Recall which is the ratio of true positive (true predictions) and the total of ground truth positives (total number). The formula is given as such:

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

Average Precision (AP) and Mean Average Precision (mAP): A brief definition for the average precision is the area under the precision-recall curve. AP combines both precision and recall together. It takes a value between 0 and 1 (higher is better). To get AP =1 need both the precision and recall being equal to 1. The mAP is the mean of the AP calculated for all the classes.

4 Results and Discussion

As expected, it was increasing the size of the images resulted in improved metrics results, in general, the size images of 416x416 pixels had the best results of accuracy, mainly for the Pothole. However, when testing sizes larger than 416x416, training became significantly time-consuming and was discontinued. Yolov3 models were more stable mAP results throughout training. In contrast, Yolov4 had a large variation of mAP and loss values, as shown in Figure 3.

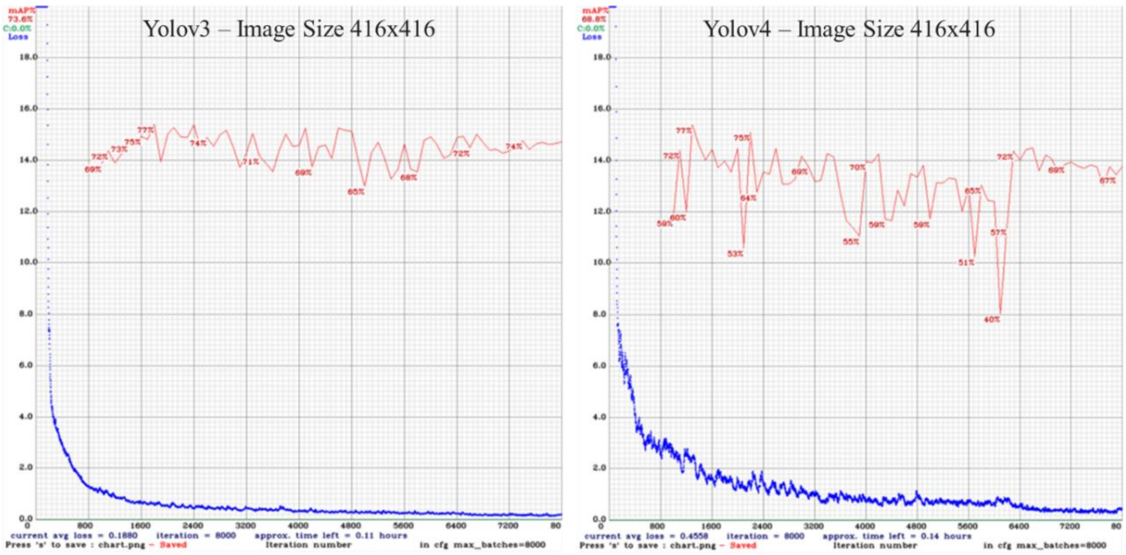


Figure 3 – Loss & mAP chart during training Yolo v3 and v4 - 416x416 pixels models

The Table 3 presents iteration the each model configuration had the best overall performance, combining all the metrics. The CNN that presents the best results was based on Yolov3, with an image size of 416x416 and 6000 iterations, in which it obtained an F1-score of 79.00%, an average IoU of 65.9%, and mAP@0.50 of 74.5%.

Of the 10 best models, 9 were from Yolov3; it shows that the changes made to the Yolo architecture did not bring real gains for the potholes and patches detectors on this dataset. And the dimensions of 416x416 and 224x224 are predominant in the best models, with the dimension of 416x416 present in 5 and 224x224 in 4 of the best models.

Table 3. Ranking of 10 best models

#	Model	AP		Precision	Recall	F1-score	Average IoU	mAP@ 0.50
		Pothole	Patch					
1	**Yolov3 - 416x416 - 6000**	**80,2%**	**68,8%**	**89,0%**	**71,0%**	**79,0%**	**65,9%**	**74,5%**
2	Yolov3 - 224x224 - 2000	77,2%	76,5%	83,0%	71,0%	77,0%	61,3%	76,9%
3	Yolov3 - 224x224 - 3000	78,1%	74,9%	81,0%	74,0%	78,0%	60,3%	76,5%
4	Yolov3 - 224x224 - 8000	79,7%	70,0%	83,0%	74,0%	78,0%	62,1%	74,9%
5	Yolov3 - 416x416 - 7000	74,2%	69,9%	87,0%	70,0%	77,0%	65,8%	72,0%
6	Yolov3 - 224x224 - 7000	76,5%	69,8%	86,0%	69,0%	76,0%	64,3%	73,1%
7	Yolov4 - 320x320 - 3000	80,7%	72,5%	78,0%	74,0%	76,0%	56,8%	76,6%
8	Yolov3 - 416x416 - 8000	77,8%	69,4%	82,0%	72,0%	77,0%	61,9%	73,6%
9	Yolov3 - 416x416 - 3000	81,6%	63,6%	81,0%	73,0%	77,0%	61,5%	72,6%
10	Yolov3 - 416x416 - 2000	79,0%	71,1%	76,0%	75,0%	76,0%	56,7%	75,0%

Table 4 presents the five best results for the defects separately. It is possible to confirm that the pothole defect is easier to identify and presented AP values around 6% higher than the AP for the patch. When analyzing the defects individually, the best performance model to pothole was Yolov4, 8000 iterations, and 416x416 pixels with an AP of 82.2%, and for the patch, it was Yolov3, 2000 iterations and 224x224 pixels with an AP of 76.5%.

Table 4. Best results for the defects separately

Pothole		Patch	
Model	**Average Precision**	**Model**	**Average Precision**
Yolov4 - 416x416 - 8000	82,2%	Yolov3 - 224x224 - 2000	76,5%
Yolov3 - 416x416 - 3000	81,6%	Yolov3 - 224x224 - 3000	74,9%
Yolov3 - 320x320 - 4000	81,3%	Yolov4 - 320x320 - 3000	72,5%
Yolov3 - 320x320 - 5000	81,0%	Yolov3 - 224x224 - 6000	71,8%
Yolov4 - 320x320 - 3000	80,7%	Yolov3 - 416x416 - 2000	71,1%

In general, the expansion of iterations led to an improvement in the IoU and F1-Score, but not necessarily the other metrics. The analysis of the mAP results did not show a specific trend regarding the number of interactions and performance gain, with good results from 2000 iterations onwards depending on the configuration of the images, such as Yolov3 - 224x224.

Figure 4 shows an example of the detection performed by the best model: Yolov3 model - 416x416 pixels trained with 6000 iterations. It was found that the model can detect defects up to a range of approximately 3-5m in front of the camera point. In Figure 4, it is possible to see that it was able to identify 4 potholes closest to the survey vehicle, however, it did not detect the farthest patches.

Figure 4. Defect detection by Yolov3 - 416x416 pixels model

As the evaluation is continuous, the survey vehicles have an odometer that can be connected to the cameras and thus program the image capture at specific distances, being recommended to obtain the images every 5m, avoiding the duplication of detection and the non-detection of defects that are far apart.

5 Conclusions

With the research, it was possible to verify that the Yolo architecture could detect the defects in the pavement in a satisfactory way. The Yolov3 - 416x416 model showing the best combine metrics results. Regarding the images dimensions, 416x416pixels and 224x224pixels had the best results of the models; both can be used, being limited by the processing time during the training. If the computer used is more powerful, the larger size can be used; otherwise, the smaller size will also bring satisfactory results. The pothole defect is easier to identify compared to the patch.

Although it was not possible to identify an ideal interaction number for the training of this dataset, it is recommended to keep the analysis range from 1000 to 8000 iterations for the next training. This work provides support for the continuation of the research; the image size and iteration range settings were defined as 416x416pixels, and the iterations range is 1000 to 8000. The research next stage will be the complete analysis of the dataset of the 10,000 labeled images, with variations in lighting throughout the day.

Acknowledgements. This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brazil (CAPES) under Grant [Finance Code001]. The authors would like to express their appreciation for CERTARE Engenharia e Consultoria Ltda in Brazil for providing the images dataset.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection," *Constr Build Mater*, vol. 157,

pp. 322–330, 2017.

[2] L. A. Silva, H. Sanchez San Blas, D. Peral García, A. Sales Mendes, and G. Villarubia González, "An Architectural Multi-Agent System for a Pavement Monitoring System with Pothole Recognition in UAV Images," *Sensors (Basel)*, vol. 20, no. 21, 2020.

[3] C. R. Ikram, "A benchmark for evaluating Deep Learning based Image Analytics," UNIVERSITY OF OSLO, 2019.

[4] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, pp. 211–252, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, vol. 1, p. 5000, 2014.

[6] R. Girshick, "Fast R-CNN," *Proc IEEE Int Conf Comput Vis*, vol. 2015 Inter, pp. 1440–1448, 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, 2017.

[8] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, vol. 2019-June, no. 2, pp. 821–830, 2019.

[9] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[10] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, no. May, 2020.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc - 30th IEEE Conf Comput Vis Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.

[13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, pp. 318–327, 2020.

[14] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated Road Crack Detection Using Deep Convolutional Neural Networks," *Proc - 2018 IEEE Int Conf Big Data, Big Data 2018*, pp. 5212–5215, 2019.

[15] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, "A Deep Learning Approach for Road Damage Detection from Smartphone Images," *Proc - 2018 IEEE Int Conf Big Data, Big Data 2018*, pp. 5201–5204, 2019.

[16] L. Song and X. Wang, "Faster region convolutional neural network for automated pavement distress detection," *Road Mater Pavement Des*, vol. 0, no. 0, pp. 1–19, 2019.

[17] M. Al Qurishee, "Low-cost deep learning UAV and Raspberry Pi solution to real time pavement condition assessment," *Master'sThesis Univ Tennessee Chattanooga*, pp. 1–99, 2019.

[18] D. R. Sulistyaningrum, D. Oranova, R. H. Pramestya, I. Mukhlash, B. Setiyono, and E. Ahyudanari, "Pavement Distress Classification Using Deep Learning Method Based on Digital Image," vol. 193, no. Istsdc 2019, pp. 143–146, 2020.

[19] X. Lei, C. Liu, L. Li, and G. Wang, "Automated Pavement Distress Detection and Deterioration Analysis Using Street View Map," *IEEE Access*, vol. 8, pp. 76163–76172, 2020.

[20] V. Mandal, A. R. Mussah, and Y. Adu-Gyamfi, "Deep Learning Frameworks for Pavement Distress Classification: A Comparative Analysis," 2020.

[21] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *Int J Pavement Eng*, vol. 0, no. 0, pp. 1–14, 2020.

[22] L. K. Suong and K. Jangwoo, "Detection of potholes using a deep convolutional neural network," *J Univers Comput Sci*, vol. 24, no. 9, pp. 1244–1257, 2018.

[23] B. Prakash and S. N. Nayak, "ROAD POTHOLE DETECTION USING YOLOv2 MODEL," *Int Res J Mod Eng*, vol. 02, no. 08, pp. 1401–1405, 2020.

[24] P. Ping, X. Yang, and Z. Gao, "A Deep Learning Approach for Street Pothole Detection," *Proc - 2020 IEEE 6th Int Conf Big Data Comput Serv Appl BigDataService 2020*, pp. 198–204, 2020.

[25] H. Chen, M. Yao, and Q. Gu, "Pothole detection using location-aware convolutional neural networks," *Int J Mach Learn Cybern*, vol. 11, no. 4, pp. 899–911, 2020.

[26] H. Majidifard, Y. Adu-Gyamfi, and W. G. Buttlar, "Deep machine learning approach to develop a new asphalt pavement condition index," *Constr Build Mater*, vol. 247, p. 118513, 2020.

[27] S. Gupta, P. Sharma, D. Sharma, V. Gupta, and N. Sambyal, "Detection and localization of potholes in thermal images using deep neural networks," *Multimed Tools Appl*, 2020.

[28] N. Camilleri and T. Gatt, "Detecting road potholes using computer vision techniques," pp. 343–350, 2020.

[29] H. Majidifard, P. Jin, Y. Adu-Gyamfi, and W. G. Buttlar, "Pavement Image Datasets: A New Benchmark Dataset to Classify and Densify Pavement Distresses," *Transp Res Rec*, vol. 2674, no. 2, pp. 328–339, 2020.

[30] M. T. Cao, Q. V. Tran, N. M. Nguyen, and K. T. Chang, "Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources," *Adv Eng Informatics*, vol. 46, no. September, p. 101182, 2020.

[31] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone," *Comput Aided Civ Infrastruct Eng*, 2018.

[32] G. Ciaparrone, A. Serra, V. Covito, P. Finelli, C. A. Scarpato, and R. Tagliaferri, *A deep learning approach for road damage classification*, vol. 518. Springer Singapore, 2019.

[33] DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES - DNIT, "DNIT 031- ES: Pavimentos flexíveis - Concreto asfáltico - Especificação de serviço," 2006.