Deepbrokenhighways: Road Damage Recognition System Using Convolutional Neural Networks

Sebastian Peralta-Ireijo, Bill Chavez-Arias and Willy Ugarte[a]

Universidad Peruana de Ciencias Aplicadas, Lima, Peru

Keywords: Computer Vision, Pothole Detection, Convolutional Neural Network, YOLO, MobileNet.

Abstract: Road damage, such as potholes and cracks, represent a constant nuisance to drivers as they could potentially cause accidents and damages. Current pothole detection in Peru, is mostly manually operated and hardly ever use image processing technology. To combat this we propose a mobile application capable of real-time road damage detection and spatial mapping across a city. Three models are going to be trained and evaluated (Yolov5, Yolov8 and MobileNet v2) on a novel dataset which contains images from Lima, Peru. Meanwhile, the viability of crack detection through bounding box method will be put to the test, each model will be trained once with cracks annotations and without. The YOLOv5 model was the one with the best results, as it showed the best mAP50 across all of out experiments. It got 99.0% and 98.3% mAP50 with the dataset without crack and with crack annotations, correspondingly.

1 INTRODUCTION

Currently, many third world countries struggle to maintain their roads pothole and crack free. Studies in Peru like (Dávila Estrada, 2022) showed that around eight to ten potholes are present on average, over a one kilometer road range. Furthermore the lifespan of these roads should be close to twenty five years, nevertheless, the real span is about four to six years. As written in (Ministerio de Transportes y Comunicaciones, 2015), current government road exploration procedures don't include a detailed detection of road damage. This turns the damage mapping into a lengthy and manual process, if needed. That being said, nowadays few companies devoted to roadwork like (Tenorio Construcciones y Soluciones, 2022) use software to map the damage present in a road before any work is done. That motivates us to tackle the problem of slow detection and mapping of road damage.

We aim to develop an application, capable of integrating real time detection and mapping of road damage. Aiding in the optimization of road damage detection for organizations which carry out road maintenance. This will allow people to benefit from automation via Deep Learning models, as well as obtaining a detection process free of human subjectivity. Thus, making a contribution to the optimization of road maintenance projects. While there are solutions that deal with pothole and crack detection, their method or end product doesn't line up with our objectives. For example, vibration based models like (Egaji et al., 2021) are successful in the damage detection, but lack the capabilities of providing a visual feedback of the damage and require the use of a vehicle and sensors. Furthermore other solutions that involve computer vision stop at the experimental phase, maximizing the proposed model detection capability through the use of innovative image processing like in (Aparna et al., 2022). Yet, their image dataset features do not match most of the approaches, since it involves heatmap imaging and specialised equipment.

Some solutions that involve the integration of road damage detection model and a application like (Dong et al., 2022) and (Patra et al., 2021). However, in (Dong et al., 2022) the UI design and usage wasn't friendly to the average person, as results were shown on a console like interface. Also it required the use of a small vehicle to carry out the detection which limits it's capabilities to a specific viewpoint. In (Patra et al., 2021) the solution limits itself to the exclusive detection of pothole as a generalized label. Meaning that their model is binary, either a part of an image is a pothole or not, simplifying the training of the model. Also, road crack detection was left out of the training of the model leaving various improvement points for

[a] https://orcid.org/0000-0002-7510-618X

Peralta-Ireijo, S., Chavez-Arias, B. and Ugarte, W.
Deepbrokenhighways: Road Damage Recognition System Using Convolutional Neural Networks.
DOI: 10.5220/0012685600003690

In *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024) - Volume 1*, pages 739-746
ISBN: 978-989-758-692-7; ISSN: 2184-4992

us to explore. Our approach intends to use a convolutional network architecture that is optimized for real time detection and a user friendly mobile application to go alongside it.

The key components of our approach are a YoloV5 model trained with some of the state of the art road damage datasets and meant for Bounding Box detection. This data will be expanded upon by the creation of our own dataset containing images from our specific context, Lima and Callao, Peru. That being said, using the mobile development framework Flutter, we will develop an app that takes advantage of the model for road damage detection (potholes of different sizes and depth and road cracks) and damage mapping using one of the available modules.

Our main contributions are as follows:

- We elaborate a dataset with the characteristics of the context in which our application will be utilized.

- We developed a mobile application that is able to detect in real time types of potholes and cracks using convolutional networks.

- We test the viability of road cracks detection with a bouding box method through a quantitative analysis of different models based on metrics (ex. Average Precision and mAP50).

This paper is distributed into the following sections: First, we review related work on road damage detection in Section II. Then, we discuss relevant concepts and theories related to the background of our research and describe in more detail our main contribution in Section III. Furthermore, we will explain the procedures performed, and the experiments conducted in this work in Section IV. At the end, we will show the main conclusions of the project and indicate some recommendations for future work in Section V.

2 RELATED WORKS

The topic of pothole and damage segmentation has been around for several years now. In recent years many methods and techniques have been proposed with the goal of getting the best results. Those articles differ greatly in terms of the objective and final product. The following articles demonstrate studies and development of Deep Learning models of interest for our project. Some even mention a model integration with an application, which is relevant as well. That being said we will discuss the insights they provide, the limitations and the differences in comparison to our intentions.

In (Park et al., 2021) the authors perform a study comparing the performance of various YOLO models versions under multiple metrics as precision, recall and mAP. The ultimate goal being to find the optimum YOLO algorithm for pothole detection in real time. Results showed that the YOLOv4-tiny was the best for pothole detection. However, we also will evaluate other real time models such as SSD and MobileNet. Moreover, we seek to produce a model which is retrained for Peruvian roads. Instead in (Park et al., 2021) the dataset is mostly web-scraped from Google, meaning the model isn't trained for a specific context and results could vary greatly depending on the origin of the image.

In (Rateke and von Wangenheim, 2021) the authors propose a model based on ResNet pre-trained in ImageNet that segmentates and differentiates the types of damage between speed-bump, crack and pothole. A transfer learning process was carried out on two models with different hyper-parameters. The model with the best configuration was called r34-DW, has an accuracy of 90% in eleven out of thirteen classes. However due to the high number of labels, two classes have a low accuracy (below 72%). In contrast, we aim to use more general labels, since having various highly specific labels hinders the model performance.

In (Moscoso Thompson et al., 2022) the authors developed 6 models for evaluation on the SHREC 2022 track, which were compared to the Baseline-DeepLav3 model. The models were trained to perform semantic segmentation on the road surface, detecting potholes and cracks. From the proposed models the HCMUS-CPS-DLU-Net model had the best overall results, however these models aren't able to differentiate between different types and severity of the damage. As we seek to provide a more detailed report of the road damages, it is important to be able to identify the type and/or severity of the damage.

In (Dong et al., 2021) the authors propose a mobile damage segmentation model, which is developed to detect multiple damages in the same image, using an optimizer called DIoU-NMS. The model that performs the segmentation uses a reference object, this object allows the model to estimate the area, width, height and ratio of the damage. Even though this method has an accuracy of 93.4%, for our purposes, it is not convenient to carry a reference object and depend on having a clear shot with it to use the model. In our proposal, an accurate detection of the type and severity of the damage is of higher priority than the estimate of its dimensions.

In (Patra et al., 2021) a system called *PotSpot* was implemented. Through an Android Application, a CNN model and a Maps API (Application Programming Interface), are integrated to be able to perform the pothole detection and mapping. The damage detection is performed exclusively on potholes, moreover pothole detection is binary, *pothole* or *no pothole*, simplifying the training of the model. That being said despite this project providing an end-to-end result, it lacks the level of detail we aim to achieve. The end result fails to show the user the characteristics of the damage (e.g. type, severity, image of the damage, etc).

3 REAL-TIME POTHOLE DETECTION WITH MOBILE DEVICES

3.1 Preliminary Concepts

In this section, the main concepts used in our work are presented. We aim to train a real-time object detection model, via Computer Vision methods and a Bounding Box labeled dataset to be able to identify objects of interest within our training images.

Definition 1 (Computer Vision(Baek and Chung, 2020)): Computer vision is a technology that extracts useful information by inputting visual data into a computer and analyzing it. The goal of computer vision is to extract interesting information from pattern recognition, statistical learning, and projection geometry through object detection, segmentation, and recognition in images.

Definition 1 (Convolutional Neural Network (Baek and Chung, 2020))**.** *CNN aims to reduce the complexity of the model and extract significant features by applying a convolution operation. In visual data, object detection is a technique to find a candidate region for a detection target to recognize a specific target and to predict the type and location of the object (bounding box).*

Definition 2 (Vision-based Pothole Detection (Kim et al., 2022))**.** *A vision-based method uses images or videos as input data and determines the presence of potholes on the road surface by applying image-processing and deep-learning technology. ...suitable for determining the number and approximate shape of potholes.*

Example 1 (Vision-based Pothole Detection)**.** *Is shown in Figure 1.*

Definition 3 (Global Positioning System (Wu et al., 2020))**.** *GPS, which is used to record location. As mentioned in (Pandey et al., 2022) and (Egaji et al., 2021), location data means latitude and longitude coordinate information.*

Figure 1: Example of vision-based Pothole Detection via Bounding Box labeling in (Park et al., 2021).

Example 2 (Global Positioning System)**.** *Is shown in Figure 2.*

Figure 2: Example of GPS data usage for pothole mapping in (Patra et al., 2021).

3.2 Method

In this section, the main contributions proposed in this project will be detailed.

3.2.1 Peruvian Road Damage Dataset

The first contribution of this work is the creation of a dataset containing information of potholes on the roads of Lima, Peru. This is due to the fact that, currently, there is no publicly available road damage dataset of Peru. The dataset without augmentation has 618 images and three labels (Cracks, Pothole and Severe Pothole), the labels for the potholes were chosen as follows: the most critical potholes were labeled as Severe while medium sized and small potholes were labeled as Pothole.

3.2.2 Real-Time Pothole Severity and Crack Detection Model

The second contribution of this work is the implementation of a model for real-time detection of pothole severity. We will use the pre-trained Yolov4 model, this model will be trained further using our data augmented Peruvian Road Damage Dataset (Figure 4). The architecture of the Yolov5 model is composed of 3 fundamental parts; the first is the Backbone, where Darknet or CSPDarknet53 is generally used to extract features in different dimensions from the input. The second is the Neck, where FPN or SPP modules are generally used to fuse the features extracted from the backbone at different scales (sizes) and finally the Head, which is responsible for performing detection at multiple scales (sizes).

3.2.3 Road Damage Detection and Mapping System

The third contribution of this work is a mobile application Fig. 3, developed with the React Native framework, with road damage detecting and mapping functionalities. Using the model we developed, damage in roads will be detected in real-time through a smartphone's camera. As damage is detected the user will be capable of taking a picture of the detection and choose to register (upload) the damage to a Cloud Firestore Database. The damage registration data includes: the picture, latitude and longitude of where the picture was taken and the detected label. Once damage is detected and registered, users will be able to see damage detected throughout the city through a map implemented into the application.

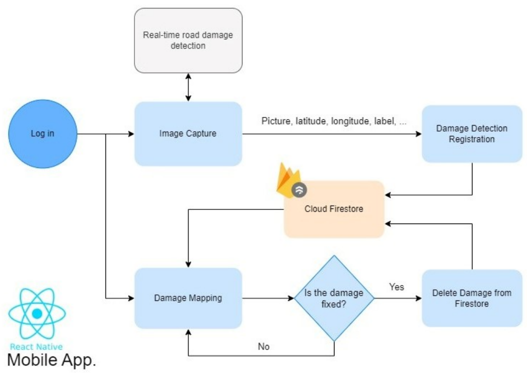


Figure 3: System Flow Diagram.

4 EXPERIMENTS

In this section, we will discuss the experiments that were performed for our project. Detailing the configuration and resources needed to replicate said experiments. Moreover, the results produced by these experiments will also be discussed and interpreted.

4.1 Experimental Protocol

In this section, the details about the configuration, required resources and applications used to perform the experiments. The experiments were developed on the Google Colaboratory Platform with a GPU Runtime. This includes a Intel(R) Xeon(R) CPU @ 2.30GHz The GPU we utilised for our experiments was a Tesla T4, alongside 12 GB of RAM. Our code was developed on Python 3.9.16. Data augmentation related work relies mainly on the Albumentations library.

For the implementation of the Yolo models, we used the Ultralytics open-source project that facilitates the training of Yolo models. It can be found at https://github.com/ultralytics/yolov5. Some of the libraries required for these experiments include opencv-python>=4.1.1, Pillow>=7.1.2 and torch>=1.7.0 as well as others detailed on the requirements file of the ultralytics repository.

For the MobileNet v2 model, we'll use the TF2Lite Model Detection API. For this experiment packages required for the Model Detection API as well as the tensorflow=2.8.0 library are required.

The experiments consist of training the two models just mentioned with our Peruvian Road Damage Dataset alongside images from widely used Road Damage Dataset 2020 (RDD20). Moreover, to test the viability of crack detection using a Bounding Box method each model will be trained twice, with and without including the crack damage annotations. The Dataset used for these experiments contains 325 images from Peru and 293 from the RDD20 (Arya et al., 2021), which includes images from India and Japan. After Data Augmentation, which involves 3 different types of transformations, the final dataset for experimenting has 4994 images. A example of the performed transformations is shown in (Figure 4).

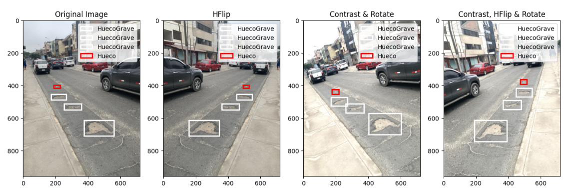


Figure 4: Data Augmentation performed on Dataset (Hueco = Pothole; HuecoGrave = SeverePothole).

4.2 Results

In this subsection the experiments carried out and the corresponding results obtained in each one will be detailed and discussed.

4.2.1 YOLOv5

Training with YOLOv5 for the dataset without crack annotations lasted about 4 hours with 21 minutes, training a total of 350 epochs. On the other hand, training with crack annotations lasted 4 hours with 31 minutes, training a total of 350 few. The training was performed with the pre-trained Yolov5s weights and with a batch size of 16 for both the datasets, with crack annotations (see Figure 5a) and without crack annotations (see Figure 5b).

Although good results were obtained, the training could not be more extensive due to the limited resources used, being a maximum of 350 epochs as previously mentioned. However, the loss obtained in the last epochs of the training is minimal. This means that the model is highly optimized and training beyond this point would not cause a drastic change in results.

4.2.2 YOLOv8

Training with YOLOv8 for the dataset without crack annotations lasted about 4 hours with 45 minutes, training a total of 220 epochs. On the other hand, training with crack annotations lasted 2 hours with 25 minutes, training a total of 110 epochs. The training was performed with the pre-trained Yolov8n weights and with a batch size of 16 for both the datasets, with crack annotations and without crack annotations.

Due to the limited GPU resources the Google Colab platform offers, it was not possible to perform a more exhaustive training; however, when observing Fig. 6a and Fig. 6b, it is evident that as more training epochs are added, the loss will continue to decrease. It is worth noting that the epochs in YOLOv8 training are fewer, this is due to YOLOv8 having higher computational resource requirement than YOLOv5.

4.2.3 SSD MobileNet V2

The training for this model with the dataset with both datasets lasted 35000 epochs for approximately 4 hours. When performing detection on test images the minimum confidence was set at 0.15%, with the purpose of filtering very low quality, inaccurate and redundant overlapping detections. Moreover, the observed total loss during training had a constant downward tendency, however the MobileNetV2 model trained with no crack annotations still showed pronounced signs of a decrease in loss as shown in Fig. 7a. That being said there's a potential improvement to be reached if trained for even longer (more epochs). Meanwhile the MobileNetV2 trained with cracks annotations showed signs of stabilizing as shown in Fig. 7b.

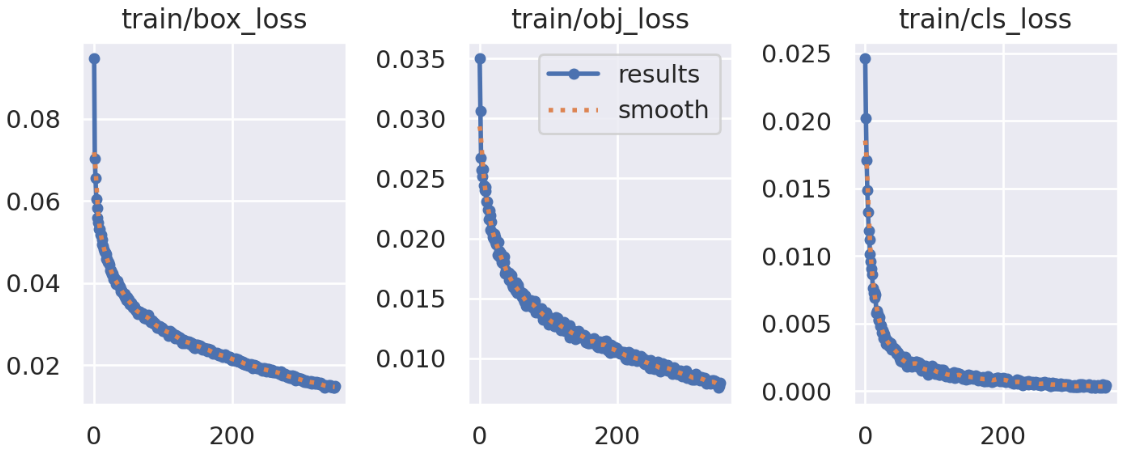
4.3 Discussion

In this subsection, the results obtained in the previous section are detailed and discussed.

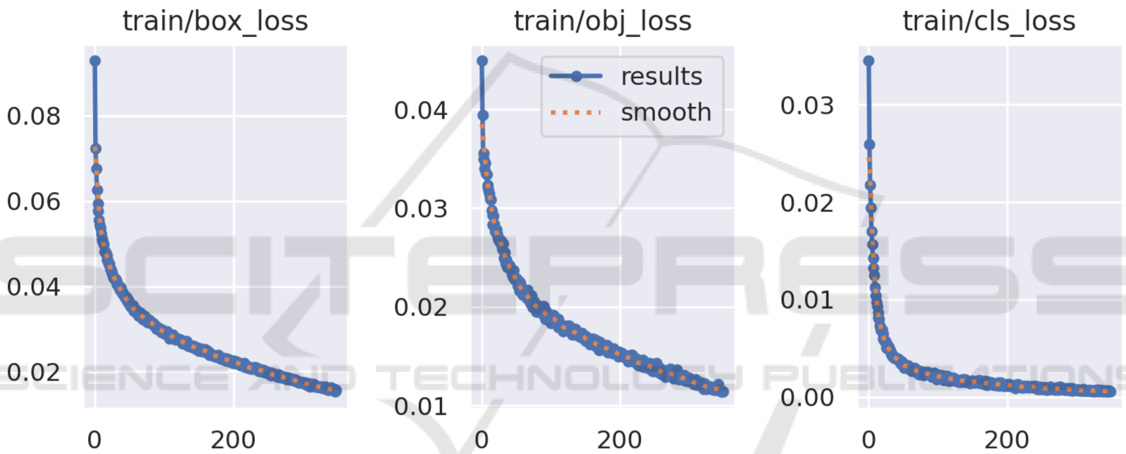
Models trained without crack annotations, manifest excellent pothole detection capabilities as shown in Table 1. The three models included in the experiments show a pattern, which is a higher Average Precision for the Severe Pothole class than the Pothole class. This is due to severe potholes, being a more critical damage, it possess more notorious characteristics and therefore is more distinguishable. On the other hand, our ”regular” pothole class, which contains smaller and less dangerous potholes, being a less significant damage in the road it involves a higher difficulty of detection. That being said, within the batch of models trained without crack annotations, YoloV5 stands out as the best. As it has the top mean average precision (mAP50), that being 99.0%.

On the side of the models trained with crack annotations, present an average precision over 90% in all classes as shown in Tab. 1. The only exception to this is the MobileNetV2, which shows a considerable amount of drop off in average precision for the pothole class in comparison to its counterpart trained with no crack annotations. This is caused due to having more classes to predict makes the model more likely to confuse one for another. Especially when the class objects are similar in appearance as it is the case here. Another thing to consider is that road cracks come in various shapes and types, which also has an impact on the pothole detection. YoloV5 is again identified as the best model as it obtained the best metrics in comparison to Yolov8 and MobileNetV2. Also, the deficiencies observed from the MobileNetV2 model don't apply. YoloV5 provides a excellent average precision that is 99.3% for potholes and severe potholes. While also having a 96.2% average precision for cracks. An acceptable drop off in average precision, in comparison to the other classes, considering how irregular and varied the appearance of road cracks can be, making it the toughest class to label and detect.

Lastly, through these experiments we have found that crack detection via a Bounding Box method to be viable. This being due to the competent average pre-



(a) YOLOv5 training loss without crack annots.



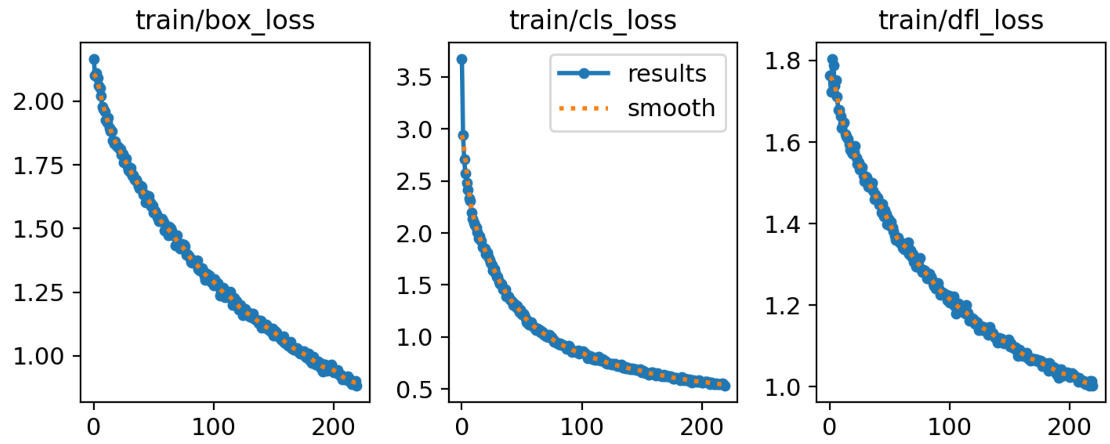
(b) YOLOv5 training loss with crack annots.

Figure 5: Comparison of YOLOv5 training (a) without crack annots and (b) with crack annots.

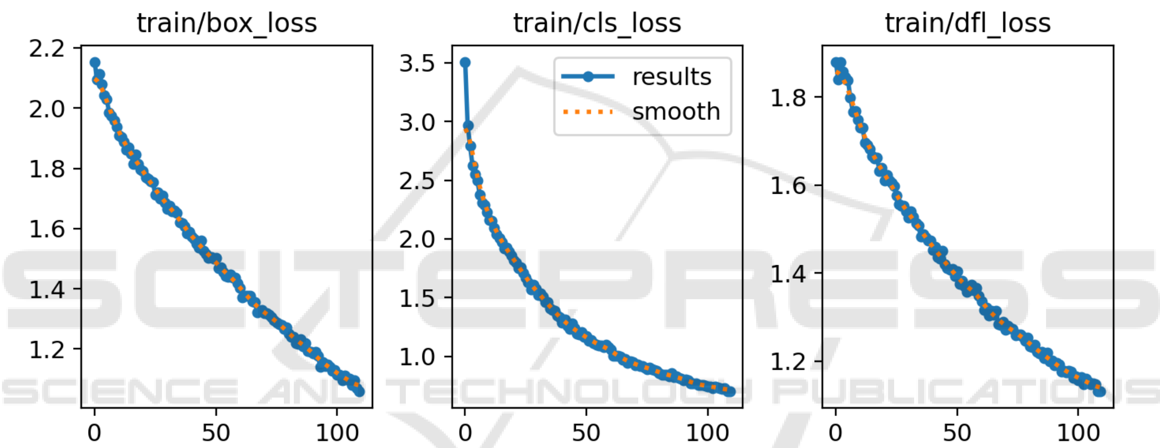
Table 1: Av. Precision at 0.5 IoU.

	without crack damage annots			with crack damage annots			
Model	Pothole	SeverePothole	mAP50	Pothole	SeverePothole	Crack	mAP50
SSD MobileNetV2	89.6%	95.3%	92.5%	83.2%	92.2%	90.8%	88.7%
YoloV5	98.6%	99.5%	99.0%	99.3%	99.3%	96.2%	98.3%
YoloV8	98.0%	99.2%	98.6%	94.2%	98.9%	94.4%	95.8%

cision obtained of 96.2% for cracks with the YoloV5 model, alongside 99.3% average precision for pothole classes. Taking that into consideration, the model selected for implementing into the mobile application will be a YoloV5 model trained with crack annotations. It presents the best performance from our set of experiments, having a minimal decrement in mAP50 in comparison to the ”crack-less” YoloV5 model as it involves a wider range of labels to detect.



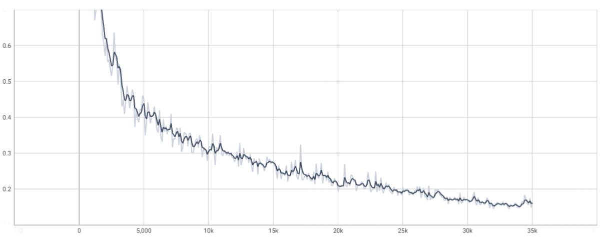
(a) YOLOv8 training loss without crack annots.



(b) YOLOv8 training loss with crack annots.

Figure 6: Comparison of YOLOv8 training loss (a) without crack annots and (b) with crack annots.

(a) Without crack annots.



(b) With crack annots.

Figure 7: Comparison of MobileNetV2 training total loss (a) without crack annots and (b) with crack annots.

5 CONCLUSIONS

In this paper, a system for detection and monitoring of road damages is devised. The scope for this paper includes the experimentation and selection of the model which is most capable for road damage detection. Moreover, road crack detection has been proven to be viable as the tested models got an average precision above 90% for that specific label. That being said we have found YOLOv5 to be the best model for road damage detection in Peru. As it reports the best mAP50 across both experiments 99.0% and 98.3%, with and without cracks correspondingly. As we have found crack detection to be viable a YoloV5 model trained with crack annotations will be the one chosen for our mobile application.

Due to the lack of data availability corresponding to our context which is Lima, Peru; pushed us into forming our own novel dataset. That being said, time was also a constraint allowing us to elaborate our experiments with 618 photos (325 from Peru and 293 from RDD20(Arya et al., 2021)), thus a data augmentation phase was required in order to be able to train a robust model with plenty of data. This process allowed us to train our models with 2497 images, in contrast to our original 618 images. Our results demonstrated the effectiveness and usefulness of data augmentation for road damage detection.

As future work, we would like to be able to use drone or satellite imaging in order to further optimize and speed up the road damage detection process. While also testing its practicality as Peru's roads are highly transited, it poses the question if cars would be a major obstacle to the visibility of potholes similarly to other topics (Rodríguez et al., 2021; Fernandez-Ramos et al., 2021; Alfaro-Paredes et al., 2021).

REFERENCES

Alfaro-Paredes, E., Alfaro-Carrasco, L., and Ugarte, W. (2021). Query by humming for song identification using voice isolation. In *IEA/AIE (2)*, volume 12799 of *Lecture Notes in Computer Science*, pages 323–334. Springer.

Aparna, Bhatia, Y., Rai, R., Gupta, V., Aggarwal, N., and Akula, A. (2022). Convolutional neural networks based potholes detection using thermal imaging. *J. King Saud Univ. Comput. Inf. Sci.*, 34(3):578–588.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., and Sekimoto, Y. (2021). Rdd2020: An annotated image dataset for automatic road damage detection using deep learning. *Data in Brief*, 36:107133.

Baek, J.-W. and Chung, K. (2020). Pothole classification model using edge detection in road image. *Applied Sciences*, 10:6662.

Dong, J., Li, Z., Wang, Z., Wang, N., Guo, W., Ma, D., Hu, H., and Zhong, S. (2021). Pixel-level intelligent segmentation and measurement method for pavement multiple damages based on mobile deep learning. *IEEE Access*, 9:143860–143876.

Dong, J., Wang, N., Fang, H., Wu, R., Zheng, C., Ma, D., and Hu, H. (2022). Automatic damage segmentation in pavement videos by fusing similar feature extraction siamese network (sfe-snet) and pavement damage segmentation capsule network (pds-capsnet). *Automation in Construction*, 143:104537.

Dávila Estrada, H. A. (2022). Propuesta de un concreto para pavimentos rígidos con adición de polvo de vidrio en reemplazo parcial del cemento y agregado fino, afín de reducir la contaminación producida por la construcción de la capa de rodadura en la carretera mayocc-huanta, tramo allccomachay-huanta departamento de ayacucho.

Egaji, O. A., Evans, G., Griffiths, M. G., and Islas, G. (2021). Real-time machine learning-based approach for pothole detection. *Expert Syst. Appl.*, 184:115562.

Fernandez-Ramos, O., Johnson-Yañez, D., and Ugarte, W. (2021). Reproducing arm movements based on pose estimation with robot programming by demonstration. In *ICTAI*, pages 294–298. IEEE.

Kim, Y.-M., Kim, Y.-G., Son, S.-Y., Lim, S.-Y., Choi, B.-Y., and Choi, D.-H. (2022). Review of recent automated pothole-detection methods. *Applied Sciences*, 12(11).

Ministerio de Transportes y Comunicaciones (2015). Resumen ejecutivo del inventario basico de la red vial departamental o regional.

Moscoso Thompson, E., Ranieri, A., Biasotti, S., Chicchon, M., Sipiran, I., Pham, M.-K., Nguyen-Ho, T.-L., Nguyen, H.-D., and Tran, M.-T. (2022). Shrec 2022: Pothole and crack detection in the road pavement using images and rgb-d data. *Computers & Graphics*, 107:161–171.

Pandey, A. K., Iqbal, R., Maniak, T., Karyotis, C., Akuma, S., and Palade, V. (2022). Convolution neural networks for pothole detection of critical road infrastructure. *Computers and Electrical Engineering*, 99:107725.

Park, S.-S., Tran, V.-T., and Lee, D.-E. (2021). Application of various yolo models for computer vision-based real-time pothole detection. *Applied Sciences*, 11(23).

Patra, S., Middya, A. I., and Roy, S. (2021). Potspot: Participatory sensing based monitoring system for pothole detection using deep learning. *Multim. Tools Appl.*, 80(16):25171–25195.

Rateke, T. and von Wangenheim, A. (2021). Road surface detection and differentiation considering surface damages. *Auton. Robots*, 45(2):299–312.

Rodríguez, M., Pastor, F., and Ugarte, W. (2021). Classification of fruit ripeness grades using a convolutional neural network and data augmentation. In *FRUCT*, pages 374–380. IEEE.

Tenorio Construcciones y Soluciones (2022). Mantenimiento en vías.

Wu, C., Wang, Z., Hu, S., Lepine, J., Na, X., Ainalis, D., and Stettler, M. (2020). An automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors*, 20(19).