# Enriching and maintaining road digital twins with condition information

Ching Yau (Fergus) Mok[1], Peihang (Hank) Luo[1], Weiwei Chen[1,2,*], Ioannis Brilakis[1]
1 University of Cambridge, UK
2. University College London, UK
wc349@cam.ac.uk,weiwei.chen@ucl.ac.uk

**Abstract:** Road inspections are mostly carried out manually, limiting their scalability and creating subjective human errors. Research into automating them was studied, though there are still gaps in how to efficiently replace various aspects of manual inspections, such as how to improve detection accuracy and how to integrate them into a defect progression tracking pipeline. Tackling these gaps would enable a digitalised solution for detecting and registering road defects, such as a road digital twin. This paper aims at closing these gaps by providing two unique contributions: Firstly, performing an analysis of the performances of defect detection using different data modalities (RGB images and point cloud data) under different conditions to outline the strengths and weaknesses of each modality. Secondly, to create a pipeline for using past defect information to guide detection through detecting within bounding boxes of known defects and to enable continuous tracking of defect conditions compatible with the IFC format. Mask RCNN was used for detection in the experiments. Results indicate that incorporating information from different modalities can indeed lead to more consistent and accurate detections, and past defect information can enhance detection accuracy in the case of previously known instances of defects.

**Keywords:** Road Digital Twins, Condition Information, Enriching and Maintaining, IFC, Mask RCNN, Depth Map

## 1. Introduction

The UK's strategic road network carries a third of all traffic and two-thirds of all freight (National Highways, 2022). It is vital to the UK's economy, and its effective maintenance is crucial to the nation both economically and socially. The UK spent £4.88 billion on road maintenance between 2018 and 2019, though road conditions still left a lot to be desired. 36% of B and C roads in the UK were classified in red and amber states, indicating bad road quality. 424 accidents in 2020 were caused by defective road surfaces. Over 1 million tons of excessive CO2 was emitted on Virginia interstate highways over a 7-year period due to defective road surfaces (Louhghalam et al., 2017). Additionally, £329,379 of compensation was paid to claimants who have made vehicle damage report claims due to potholes and road defects in 2018/19 by Highways England (Highways England, 2018). The monetary, safety and environmental costs of suboptimal road conditions are significant, and current road inspection practices are not capable of solving this problem. The current state of practice, i.e., the current ways to address the problem, and their potential drawbacks, are investigated.

Currently, most road inspections are done manually, but there exist some automatic road inspection methods. For example, TRACS (TRAc-speed Condition Surveys) and SCANNER (Surface Condition Assessment for the National Network of Roads) surveys are two of the main automated visual methods currently used for inspecting road surface conditions (Department for Transport, 2021a). However, there are significant problems associated with these methods. First, they can only detect the deformation of road surfaces. Second, these methods can only give a Road Condition Indicator (RCI) for each 10m or 100m subsection length of the road and identify sections of the road that are in need of further investigations, so further manual inspections are still required to locate, categorise and measure the specific defects. Third, these automated methods are only used on classified roads at a very low frequency, as these dedicated data collection vehicles are expensive, and only a limited amount of these vehicles are available across the country.

Local authorities managed unclassified roads forms the majority (60%) of the road network in England (Department for Transport, 2021). There are also some other commercial solutions available for inspections of unclassified roads, such as RoadBotics, Vaisala, and Gaist. And some of the local authorities are using these solutions (Department of Transport, 2021). However, they all have their own problems. For example, similar to the TRACS and SCANNER surveys, RoadBotics and Vaisala only produce a rating for each subsection of the road, and further manual inspections are still required. Gaist (Gaist, 2022) identifies and records some common types of road defects automatically, but it only focuses on the road surfaces and does not inspect other road assets, such as signs and road markings. In addition, all of the methods mentioned above only use 2D visual data. None of these methods makes use of 3D depth data of the road pavement.

As a result, most road inspections remain manual. Subjective interpretations and decisions are involved, and results from different inspections cannot be easily compared (Bianchini et al., 2010), leading to inconsistent conclusions. Inspection frequency is also low as some roads are only being inspected at a frequency that is up to every 12 months (Turner et al., 2020). The root cause of this problem is that the perceived value of increasing its frequency is too low to justify the costs.

These problems motivated the search for a digitalised approach to defect detection and registration, with the aim of increasing the value of inspection through more accurate detection and better integration with defect information storage standards, while at the same time enabling a more scalable approach that can make more frequent inspections possible.

## 2. Research Background

Currently, two main types of methods used for road defects detection are image processing based methods and machine learning based methods (Cao et al, 2020). Image processing methods mainly include 1) threshold segmentation methods, which simply threshold the intensities of the pixels  (Zhu et al, 2007); 2) edge detection methods, which perform gradient calculations to identify edges in the images (Zhao et al, 2010); 3) region growing methods, which select seeds in the image and find similar adjacent pixels around the seeds to identify regions of defects (Zhou, et al, 2016). These methods are relatively easy to implement, but generally do not give very good results and cannot categorise the defects detected. Instead, only regions of defects can be identified, which is not suited for our application; Machine learning based methods include unsupervised learning methods and supervised learning methods. Unsupervised learning methods (Li et al, 2019) can potentially remove human subjective factors from the results, but these methods are unable to classify the defects detected, rendering them unsuitable for this application as well. Supervised learning methods include 1) classification based methods, which divide the input into overlapping blocks, and then classify the block images into either binary or multiple classes (Li et al, 2020). These methods can automatically classify the block images, but lack the level of accuracy that would enable measurements of the defects; 2) pixel segmentation methods, which assign a label or a score to each pixel in the image (König et al, 2019). However, these methods are not capable of grouping the pixels into instances, so different objects of the same type are not distinguished; 3) object detection methods, which locate an object in the image and assign its object type. Different types of output can be produced by these methods, including bounding boxes and instance masks. The instance mask outputs are more useful since it enables the extraction of accurate measurements from the output. One of the most commonly used and best-performing algorithms which output instance masks is Mask R-CNN (He et al, 2017), and it will be used for 2D instance segmentation.

A newer area of research is using 3D information to detect defects. Multiple RGB cameras can be used to recover 3D views through, for example, stereo vision (Zhang et al, 2014) or multi-frame fusion (Dhiman and Klette, 2020). Alternatively, LIDAR scanners can capture depth information directly, and point cloud data generated can be used as inputs instead (Li et al., 2020). Even though the resolutions of RGB cameras are higher than LIDAR data, using the former to generate a 3D model has multiple practical constraints. The main problem is that it takes a lot of computational power, as point correspondences need to be established. Moreover, their performance is subject to lighting conditions. These factors limit their usefulness and practicality in the real world. Hence, RGB cameras are unsuitable for 3D geometry generation for the purposes of this project. This leaves only LIDAR scanners to be considered. Libraries such as the Point Cloud Library (Ruse and Cousins, 2011) and Open3D (Zhou et al, 2018) provide numerous ways of manipulating and processing point clouds, in preparation for model training. PointNet is a type of neural network that is designed to work directly on point clouds, and does not require rasterisation into regular 3D voxels (Qi et al, 2017). However, limitations do exist for LIDAR data as well. Their relatively low resolution means that smaller defects, like cracks, can be missed. Furthermore, they do not recover data on colors, which usually contains valuable information about the defects. To add to this, detection in 3D is not well-studied for defects. There are no public datasets of defects available in 3D, which makes training and benchmarking models difficult.

There are still gaps to be solved in order to automate road defect identification and tracking. Firstly, we have yet to understand the strengths and weaknesses of each data modality. In particular, it would be useful to understand the condition in which one modality would be more suitable than the other. Secondly, no research has been done in keeping track of the evolution of road defects in the context of a road digital twin. This task is critical in constructing a digital twin of the road as past defect information needs to be updated periodically to ensure the information remains relevant and accurate. To achieve this, a systematic system of registration and storage of defect data in a universal format is required.

This research aims to enable more informed decisions about how multi-modal data can improve the detection of road defects by comparing how each modality performs in different scenarios, and to enable past defect information to be used to help with updating current defect information and tracking defect development over time.

## 3. Proposed Solution

Based on the research objectives, the methodology was proposed and the process is shown in Figure 1. Considering the need for data collection, it was decided that the defects that this project will focus on are cracks, alligator cracks, and potholes, as they are commonly found in the area where data collection would take place. Cracks, alligator cracks, and potholes can all be detected in 2D RGB images, but only potholes can be detected in 3D depth images.

The two main inputs are collected road data and an existing road digital twin. The road data consists of both RGB and LIDAR data, with the LIDAR data undergoing further processing to output a depth map so that they can be transformed into 2D depth images. This was motivated by the reduced file storage required by 2D images and the ease of applying well-studied 2D imagery-based machine learning methods as opposed to 3D-based ones. The process then starts with analyzing whether a specific location of the road has a previously registered defect in the input digital twin. If it does, then it will be forwarded to the bounding box detector to gather precise information about the evolution of the defect. If it does not, then segmentation will be performed to determine if a new defect is present. Both streams are mostly similar, with the processes both involving segmentation of the image into clusters of defects and measurement of the clusters to determine their severity and geometry. The two main differences are: (1) in the case of a previously identified defect, the registered bounding box would be the area where segmentation takes place, as opposed to a larger, more general area. This is to increase the quality of the detection due to the prior knowledge that there is a high chance that a defect is present in that location. (2) The registration process of the case of a previously identified defect involves a comparison of the new state with the registered state, whereas the case of no previous registration requires a new registration with the digital twin.

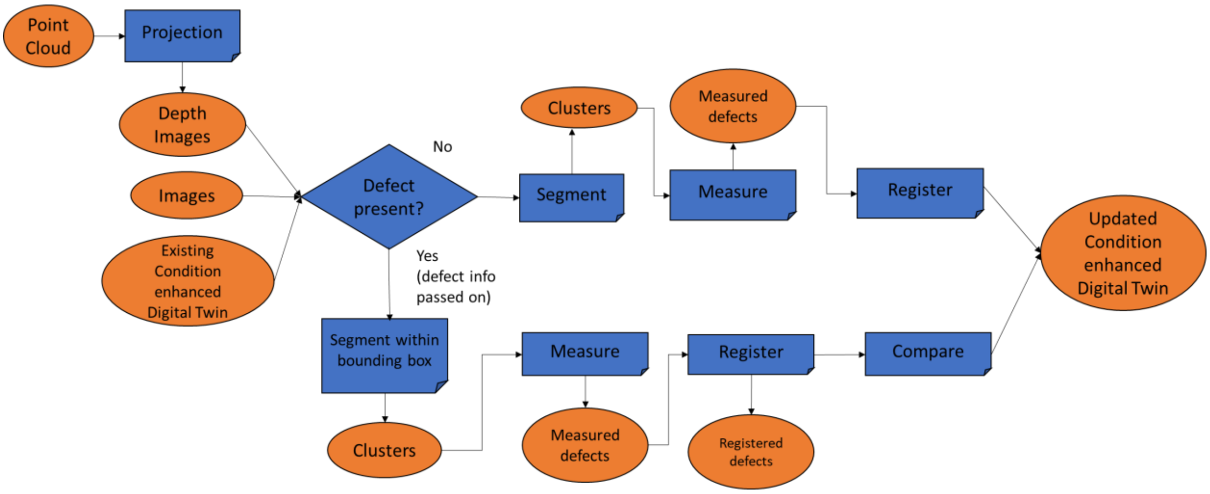
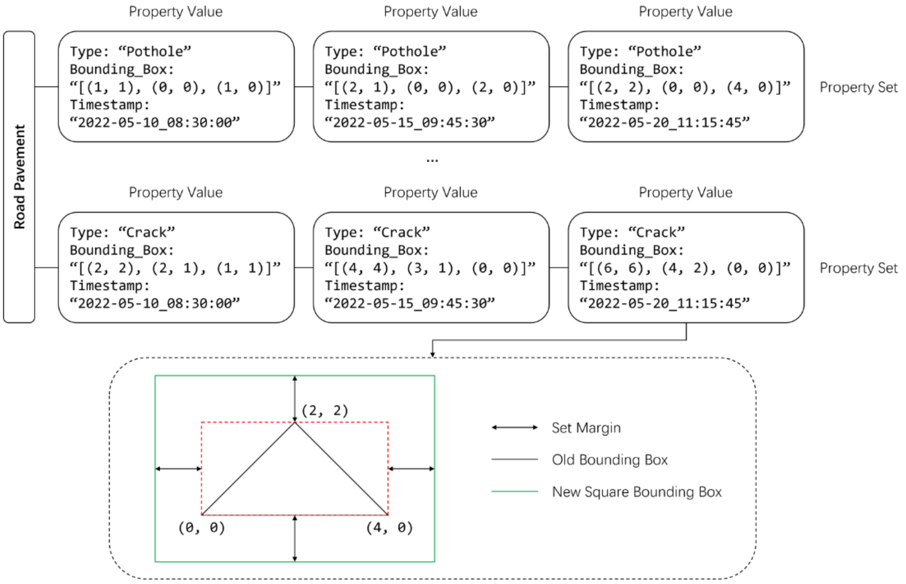


Figure 1: The proposed solution



```
Type: "Pothole"
Bounding_Box:
"[(1, 1), (0, 0), (1, 0)]"
Timestamp:
"2022-05-10_08:30:00"

Type: "Pothole"
Bounding_Box:
"[(2, 1), (0, 0), (2, 0)]"
Timestamp:
"2022-05-15_09:45:30"

Type: "Pothole"
Bounding_Box:
"[(2, 2), (0, 0), (4, 0)]"
Timestamp:
"2022-05-20_11:15:45"

Type: "Crack"
Bounding_Box:
"[(2, 2), (2, 1), (1, 1)]"
Timestamp:
"2022-05-10_08:30:00"

Type: "Crack"
Bounding_Box:
"[(4, 4), (3, 1), (0, 0)]"
Timestamp:
"2022-05-15_09:45:30"

Type: "Crack"
Bounding_Box:
"[(6, 6), (4, 2), (0, 0)]"
Timestamp:
"2022-05-20_11:15:45"
```

Figure 2: The IFC File Structure and the Generation of Bounding Boxes

Figure 2 shows how the system stores tracked records of development of example defects. Each time for a new scan, either a new node will be added onto the recording chain, or an end node will be added to the recording chain if the defect is not detected in the new scan. To store the record chains in an IFC schema, each of the tracked records of defect development is stored as a "Property Set" of the "Road Pavement" object, with each of the "nodes" stored as a "Property Value" of a "Property Set", as shown in Figure 2. For each new scan, the square bounding boxes for defects from the previous scan are generated first (shown as the red dashed box in Figure 2 below), and new square bounding boxes (shown as the green box in Figure 2 below) are then generated around the old bounding boxes with a pre-set margin, as shown in Figure 2. Then, image segmentation is performed within each of the new square bounding boxes generated.

## 4. Methodology

### 4.1 Data Collection

Data was manually collected due to the unavailability of suitable public datasets. An example of a popular public dataset is KITTI 360 (Liao et al., 2022). It has a wide coverage of around 74km of roads, though the resolution and annotations are limited as it is primarily designed for use in self-driving cars, which are only concerned with bigger road assets such as poles and other cars, and not road defects such as potholes and cracks. Two sets of data were collected, which contained RGB images and LIDAR data respectively. Each set of these data would have its associated training and testing data. The RGB images were collected using an iPad Pro, with a 12MP, $f/1.8$ aperture camera. The LIDAR dataset was collected using the FARO scanner.

Training data for the RGB dataset consisted of 269 top-down images of defects, with each image consisting of one or more defects. It is worth noting that the two sets of training data covered different segments of roads. Testing data for the RGB and LIDAR datasets cover the same road segment in order to ensure the comparability of the results. Two stretches of roads, which had a total length of 150m, were used as testing data. One of the two segments was a road with a dry surface, whereas the other was wet. This aids in allowing for an investigation into how weather can affect detection performances.

To find a compatible representation of both sets of data, the RGB images would need to be stitched together for testing in order to form a single image of the entire road, which allows one point in one modality to directly correspond to one point in another. The stitching operation was done using an app on the iPad called Polycam, which takes advantage of the iPad’s own built-in LIDAR sensor to convert a video feed of a slow-moving motion along the road to a textured 3D object, with the RGB image of the road overlayed as the texture. A top-down image of the 3D object is then extracted using Blender, which allows a singular top-down image of the road to be extracted using a virtual camera, with factors such as the inclination of the road being taken into account. The LIDAR dataset consisted of multiple scans of the road at different locations, and the registration of these scans was done automatically using Autodesk Recap.

### 4.2 Data Processing

To enhance the quality and usability of the collected data, multiple operations were performed for the two sets of data. The RGB testing data collected from Polycam contained blurred spots, therefore requiring certain areas to be rescanned. Different scans were stitched together manually using Photoshop. The LIDAR data was significantly processed so that they can have a 2D representation that retains 3D information. This involved mapping a quadratic surface to the road surface using CloudCompare, then projecting points on the road to the surface, and outputting an image of the height of each point above it. This resulted in a depth map, with the colour informing whether a point is above or below the road, where points below the surface indicating a possible pothole.

### 4.3 Data Annotation

Once the data was processed, they were annotated using CVAT. The RGB dataset contained 3 labels: Pothole, Crack, and Alligator Crack. The LIDAR dataset was only labelled for potholes, as that was the only type of defect visible. The annotations were exported to the COCO format. A point worth mentioning is that instances of "training" above referred to a dataset that contains images used to train the model, which includes the actual training dataset used for gradient descent, and a validation dataset used to ensure generality by preventing overfitting.

## 4.4 Evaluation

Metrics were needed to be able to evaluate and benchmark the performance of the models. The evaluators used by the COCO dataset were used in this project (COCO dataset, 2021). They were built on top of other key machine-learning concepts and metrics. The most important metrics for object detection are precision (1) and recall (2).

$$precision = \frac{tp}{tp+fp} \tag{1}$$

$$recall = \frac{tp}{tp+fn} \tag{2}$$

where $t_p$, $f_p$, $f_n$ are true positives, false positives, and false negatives respectively. Precision is the proportion of correct predictions in terms of all positive classifications, while recall is the proportion of correct predictions over all labels of that class. Using these definitions, a precision-recall (p-r) curve can be plotted. A key characteristic of this curve is that it should be curving outwards, with higher recalls being associated with lower precisions and vice versa. The intuition behind this is that a model with high recall tends to be more generous at classifying objects as positives, which will increase the proportionality of false positives, reducing the precision. A new metric can be computed from this using the area under the p-r curve, called the Average Precision (AP).

Another metric specific to object detection is called Intersection over union (IoU). It gives a measure of how aligned the predictions and annotations are. A $t_p$ is classified when the IoU reaches a certain threshold, and each threshold would produce a unique precision-recall curve with a unique AP value. In most cases, an IoU > 0.5 is considered to be a true positive. However, this introduces a bias as this system would not distinguish between very accurate predictions with higher IoUs, and less accurate ones with lower values. This would not take into account how closely the prediction and annotation match, even though this metric is already computed. The COCO standards solve this problem by computing the mean of the AP calculated over a range of IoUs, as shown in equation (3):

$$AP_{COCO} = \frac{AP_{0.50}+AP_{0.55}+\cdots+AP_{0.95}}{10} \tag{3}$$

Each of these APs was computed for a single class. To evaluate the performance of the entire model, the mean average precision (mAP) can be computed by finding the means of the APs of each of the classes. AR is defined in a similar way as equation (4):

$$AR_{COCO} = \frac{AR_{0.50}+AR_{0.55}+\cdots+AR_{0.95}}{10} \tag{4}$$

COCO also includes AP and AR evaluations for two different tasks: bounding box detection and segmentation mask. The former refers to detecting a rectangle bounding the edges of an object, while the latter refers to detecting the actual outline of an object. Bounding box detections are usually easier since it is a more "loose" method.

During training, a special loss function, total_loss, was used to measure performance. It is the average of various loss metrics defined in (He et al, 2017), which covers areas specific to this architecture, such as the Region Proposal Network and ROI head. The precise definition of this metric will not be discussed in this paper for simplicity.

## 4.5 Model

A model was set up to implement Mask RCNN, detailed below.

*Model Configuration:* Firstly, the model needed to be configured so that its performance can be tailored to this project. An open-source object detection package developed by Facebook

Research called Detectron2 was used to implement Mask RCNN. The default configuration included in the Detectron2 tutorial, was used in this project, which also initialised the weights. The details of this configuration can be found at GitHub COCO. The learning rate was set to 0.00025, with a batch size of 2.

*Training:* Training data was divided into an actual training set and a validation set. The training set is the set of inputs used to perform gradient descent in order to adjust the model weights and constitutes the target for the model. The validation set is held out in weight training, though it would periodically be used as inputs to test the generality of the model weights. This would be done by plotting the validation loss as a function of iterations. The model is considered to have overfitted if the validation loss is increasing while the training loss is decreasing. The validation loss curve usually follows a U-shape, and the number of training iterations that correspond to the trough is considered to be the optimum. Two separate models were trained for RGB images and depth maps respectively. The optimal training iteration for RGB images was 1500, while for depth maps it was 750.

### 4.6 IFC generation

A condition-enhanced digital twin is created using the 3D scan captured by iPad. The 3D scan was first exported as a 3D object and imported into Blender. This 3D object is then used to generate an IFC model of the road using the blender. Then the defect information is manually added onto the IFC model using ifcOpenShell.

## 5. Results and discussion

Figure 3 illustrates a comparison between the performances of the model on different defect types on the two road segments. Road 1 generally has better results than Road 2, which is an indication that the wet road conditions in Road 2 made it more difficult for the model to identify defects. For instance, cracks would appear thicker due to water seeping deep into the cracks, causing areas around the crack to remain wet for longer. The performances of the detection of two types of cracks did indeed suffer from a larger penalty in the wet road compared to the dry one, relative to the reduction in performance of the detection of potholes.

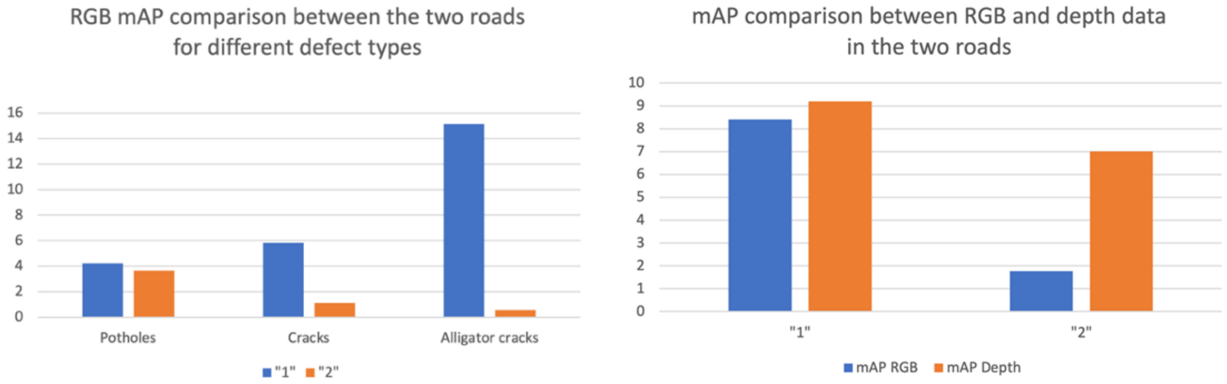


Figure 3 (left): Comparison of the performances of the model on different defect types on the two road segments; Figure 4 (right): Comparison of the performances between the two sets of data (RGB vs Depth) on the two road segments

Figure 4 shows a comparison between the detection precisions of the two models on the two road segments. The two models had similar results in road 1, which was dry. However, the performance diverges significantly on road 2, which was wet. The depth model was a lot more robust against a change in weather conditions, only suffering from a minor reduction in performance. This supports the hypothesis that wet roads make detections less accurate as they have different colour intensities, which would affect the RGB model but not the depth model.

Figure 5 is an example of the detection results of the same section of the road. RGB models are shown at the top, while the depth model is at the bottom. The results demonstrate that there is potential for depth maps to be used alongside other existing data in order to improve the detection performance on road defects.

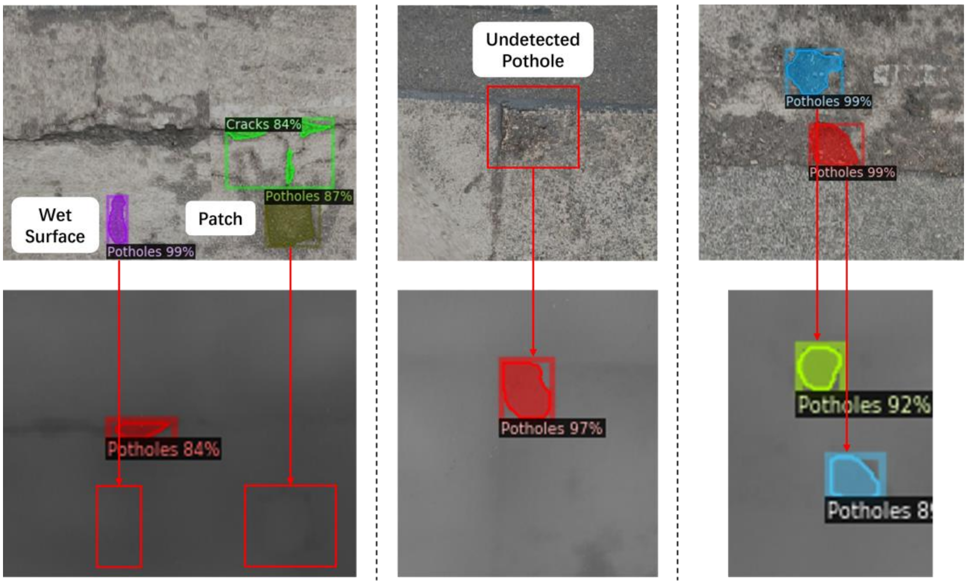


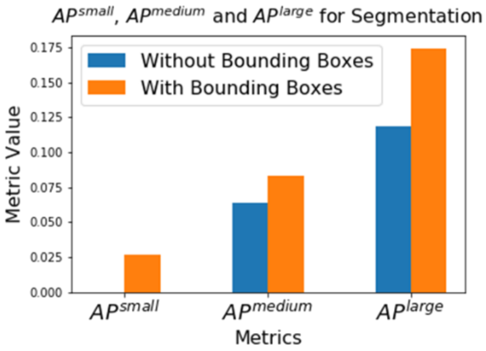
Figure 5 example of the detection results of the same section of the road

## 5.1 Bounding boxes

The performance of the model is evaluated using the metrics with and without the bounding boxes, and the results are then compared. It can be seen from Figure 6 that the detection results are significantly improved by using bounding boxes, especially for the detection of small defects. None of the small defects was detected when bounding boxes were not used. This improvement in the detection of small defects is predictable since in a normal search, the whole image is segmented into smaller sections of the same size, which does not take into account the actual size of the defect being detected, whereas in a more targeted detection with the aid of bounding boxes, essentially the area of the defect is being focused on and "zoomed into", enabling an easier detection by the model. An example of this is given in Figure 7.

The results also show a big improvement on large defects, which is because that the bounding boxes help to "zoom" the defect into a more appropriate scale, resulting in large performance improvements in the detection of both small and large defects, and consequently increasing the number of True Positives (TP). The detection results are also improved by bounding boxes because the search area is much more limited, hence many of the False Positives (FP) outside the bounding boxes are directly eliminated. Using the definition of precision in equation 1, as the number of True Positives (TP) increases, and the number of False Positives (FP) decreases, there is an overall increase in the value of precision.

It can be seen from Figure 8 that the second set of metrics also indicates a large improvement, especially in the detection results of small defects. This is due to similar reasons as explained above. In addition, the bounding boxes are expected to give an improvement here since normally only one defect is expected per bounding box, hence the density of defects is reduced, resulting in easier detections by the model. As the number of TP increases, the number of FN decreases since |TP| + |FN| = the total number of samples within the testing dataset. As a result, the overall value of recall, as defined in equation 2, increases.



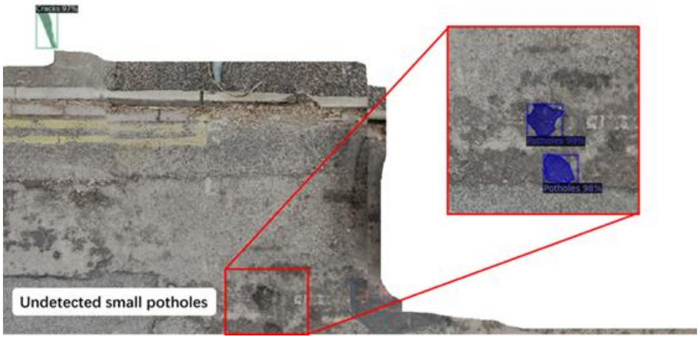


Figure 6 (left) Comparison of AP Values across Different Scales for Segmentation; Figure 7 (right) An Example of Detection of Small Potholes using Bounding Boxes

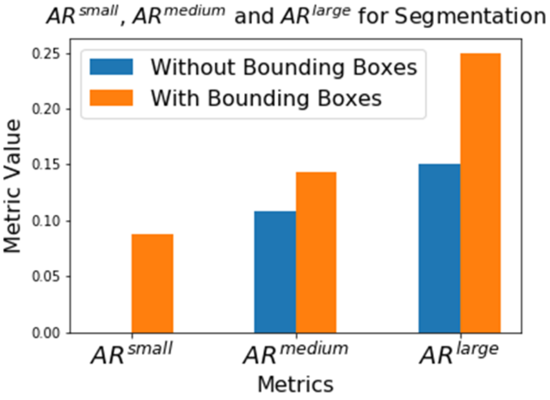


Figure 8: Comparison of AR Values across Different Scales for Segmentation

## 6. Conclusions

This research studied road digital twins with condition information and proposed an innovative methodology for enriching and maintaining road digital twins. There are three main contributions in this study: 1) Depth-based detections can improve the performance of defect detection, in that it allows for more consistent performance under different weather conditions that can affect the appearance of the road surface. By incorporating geometrical data about defects, it allows for more information about the defect to be factored in when detecting defects. 2) The bounding box method can improve the performance of defect detection during the updating scans. 3) A way to store past defect information within the IFC schema has been developed, which allows the history of defect developments to be accessed by the system to help with updating defect information in subsequent detections. The proposed solution can bring performance improvements in road defects detections by making use of data in multiple modalities and historical defects.

In the future, more training data can be used to train more accurate models. Data had to be collected for this project, which limited the amount of training data accessible. By training and testing over a larger dataset, more accurate results can be obtained, and the conclusions can be better derived. Future work can be done by using more sophisticated data processing pipelines to generate road texture. This will allow for the testing image's resolution to match up better with that of the training data, which can improve accuracy. Additionally, we only used the positions and sizes of previous defects to help with the updating steps. We could also potentially use other properties of previous defects such as their types as additional inputs to the convolutional neural network. In order to train such a model, a significantly larger amount of training data would need to be collected over a long period of time. Moreover, more pavement defect types could also be added, like rutting, together with defects on other assets, such as sidewalks, traffic lights, and traffic signs.

## 7. Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/V056441/1] and the OMICRON Project EU H2020 [grant number 955269].

## References

National Highways. “Annual Report and Accounts 2022, National Highways”. In: (July 2022).

Louhghalam, A., Akbarian, M., & Ulm, F. J. (2017). Carbon management of infrastructure performance: Integrated big data analytics and pavement vehicle interactions. Journal of Cleaner Production, 142, 956-964.

Highways England. Freedom of information request-potholes and road defects. 2018

Department for Transport. Technical Guide to Road Conditions. In: (Nov. 2021a).

Gaist. Intelligence for highways. https://www.gaist.co.uk/intelligence-for-highways.

Andrew Turner. Middlesbrough council highway safety inspection manual, 2020.

Bianchini, A., Bandini, P., & Smith, D. W. (2010). Interrater reliability of manual pavement distress evaluations. Journal of Transportation Engineering, 136(2), 165-172.

Cao, W., Liu, Q., & He, Z. (2020). Review of pavement defect detection methods. IEEE Access, 8, 14531-14544.

Zhu, S., Xia, X., Zhang, Q., & Belloulata, K. (2007, December). An image segmentation algorithm in image processing based on threshold segmentation. In 2007 third international IEEE conference on signal-image technologies and internet-based system (pp. 673-678). IEEE.

Zhao, H., Qin, G., & Wang, X. (2010, October). Improvement of canny algorithm based on pavement edge detection. In 2010 3rd international congress on image and signal processing (Vol. 2, pp. 964-967). IEEE.

Zhou, Y., Wang, F., Meghanathan, N., & Huang, Y. (2016). Seed-based approach for automated crack detection from pavement images. Transportation research record, 2589(1), 162-171.

Li, H., Song, D., Liu, Y., & Li, B. (2018). Automatic pavement crack detection by multi-scale image fusion. IEEE Transactions on Intelligent Transportation Systems, 20(6), 2025-2036.

Li, B., Wang, K. C., Zhang, A., Yang, E., & Wang, G. (2020). Automatic classification of pavement crack using deep convolutional neural network. International Journal of Pavement Engineering, 21(4), 457-463.

König, J., Jenkins, M. D., Barrie, P., Mannion, M., & Morison, G. (2019, September). A convolutional neural network for pavement surface crack segmentation using residual connections and attention gating. In 2019 IEEE international conference on image processing (ICIP) (pp. 1460-1464). IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

Zhang, Z., Ai, X., Chan, C. K., & Dahnoun, N. (2014, May). An efficient algorithm for pothole detection using stereo vision. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 564-568). IEEE.

Dhiman, A., & Klette, R. (2019). Pothole detection using computer vision and learning. IEEE Transactions on Intelligent Transportation Systems, 21(8), 3536-3550.

Li, H. T., Todd, Z., Bielski, N., & Carroll, F. (2022). 3D lidar point-cloud projection operator and transfer machine learning for effective road surface features detection and segmentation. The Visual Computer, 38(5), 1759-1774.

Rusu, R. B., & Cousins, S. (2011, May). 3d is here: Point cloud library (pcl). In 2011 IEEE international conference on robotics and automation (pp. 1-4). IEEE.

Zhou, Q. Y., Park, J., & Koltun, V. (2018). Open3D: A modern library for 3D data processing. arXiv preprint arXiv:1801.09847.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).

Liao, Y., Xie, J., & Geiger, A. (2022). KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence.

COCO datasets (2021) https://cocodataset.org/#detection-eval