Pothole Detection Using Deep Learning

Authors: Amita Dhiman[1], Waqar Khan[2] and Reinhard Klette[1]

Affiliations: [1]School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology, Auckland
email: amita.dhiman@aut.ac.nz
[2]Weltec, Wellington

Context

Potholes cause significant costs to country's budgets, lead to vehicle damages, and may create life-threatening situations to vehicle passengers, bicyclists or other road-traffic participants.

Relevance

In this study, road pothole detection has been achieved by using state-of-the-art deep learning techniques. Identified results can be used for determining primary maintenance (fixing of potholes) or can be embedded into an application to alert drivers.

Focus

We use transfer learning, a concept of transferring the acquired knowledge of a deep convolution neural network trained on one task, to the pothole identification task. Results produced using two different networks serve two different purposes. The first method identifies potholes at pixel level and produces quite an accurate mask which can be further used for analysis or annotation. The second method facilitates identification of potholes in real time. Experimental results prove that pothole identification can be automated using a single camera. Potholes are identified in video data, and the developed model is applicable to real-time scenarios. The produced model is applicable to image or video data recorded by any camera mounted on any vehicle.

1 Introduction

Pothole is a perennial problem which starts with imperceptible microscopic cracks in the road surface. Potholes may arise because of traffic overloading or weather affecting the road surface [1]. Due to rain and other weather factors, these cracks clog water inside which freeze and expand during cold weather. This void hole shape structure becomes pothole either water or snow filled or dried one. See Fig. 1 for an example.

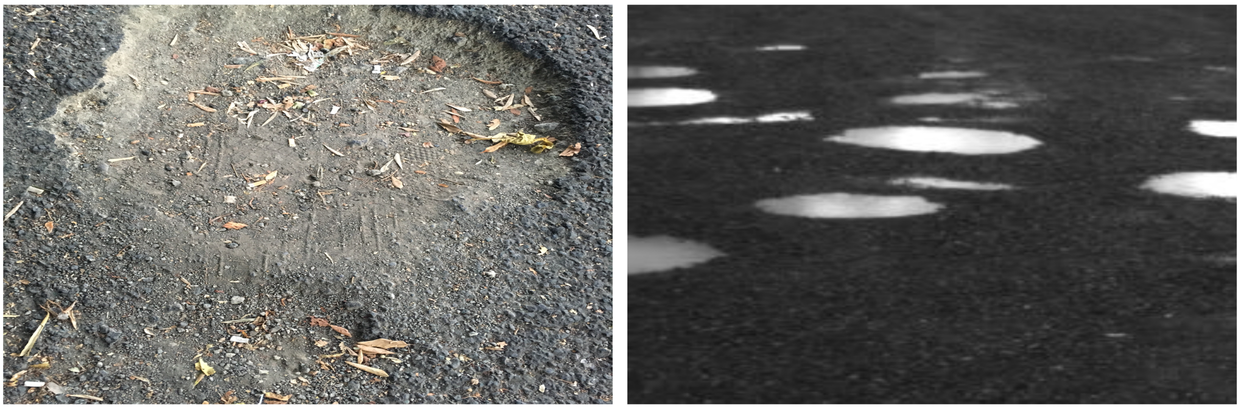


Fig. 1: Dried leafy pothole on left and water filled pothole on right, showing different intensities of light

Potholes cost life: Currently, of approximately $33,000$ traffic fatalities each year in the USA, one-third involve poor road conditions; in the UK, about 50 cyclists are seriously injured every year because of Britain's poor roads [2]. In October 2017, Auckland transport, New Zealand, received 276 requests only for compensation for damages or injuries that have been related to roads [3]. In India alone, potholes killed over $11,000$ people in the last four years [4]. In Rome, potholes caused untold number of accidents and shredded the 15 vehicles tires [5].

Potholes cost money: Potholes put a big dent in driver's budget and country's economy. New Zealand councils such as Christchurch spent $525,000$, Wellington $12,782$, Invercargill city $60,000$, and Dunedin city around $27,000$ in order to fix potholes [6]. In 2018, one of the largest pizza chain Domino's, dispensed grant of $\$5000$ to fix 53 potholes around 20 locations in the U.S. Domino's business commented on that: "*potholes cause irreversible damage to pizzas during delivery to home*" [7]. The UK government announced a budget of 420 million euros [8] and Rome 17 million euros to fix potholes in 2018 [5].
Hence, potholes cost both life and money. This is a significant road distress and a grave danger to all transport vehicles including cyclists.

Automotive industries inventions: The automotive industries use a variety of systems for road surface scan. For instance, the 2018 Jaguar land rover also invested in pothole detection technology. The system measures vibrations caused in the vehicles and adjust its suspension for more driving comfort. However, using this technology vehicle still drives over the pothole [9]. The 2017 Ford Fusion used 12 high-resolution cameras to adjust computerized controlled dampers in car for best ride comfort [10]. The 2013 Mercedes Benz S class uses a stereo-vision system to facilitate a road surface scan. Forward-looking cameras of the stereo-vision system are mounted near the central rear-view mirror. It facilitates more driving comfort when there is a bump or speed breaker on the road [11]. Road surface inspection is also done commercially using specialized vehicles [12, 13].

Unfortunately, all these technologies are equipped with expensive sensors and without technology disclosure, therefore limiting its accessibility to the general public. This paper aims to fill these gaps.

The rest of the paper is organised as follows. Section 2 presents a review of related literature. Section 3 gives a brief introduction to the transfer learning and Mask R-CNN and YOLO architectures. Section 4 demonstrates experimental results of the proposed work. Section 5 concludes.

2 Related Work

Research around the world has comprehensively explored strategies for identification of road distress. Current methods use a variety of sensors such as inertial measurements, 3D scanners, and optical sensors. Regardless of the technological improvements, the identification and reporting of pothole still depends mainly on public reporting. In this study, we have reviewed *neural network* (NN) based techniques used for pothole identification. NN consists of a trained network for the purpose of object detection. NN mainly relies upon pre-defined datasets along with significant computing power to train its network. Fortunately, in this age of data science the pre-defined datasets are improving. Processing cost has also significantly reduced over time.

The choice of training a network has shifted from machine learning to deep learning. Deep learning requires lots of data to be processed and the choice of network is dependent highly on the output.

t

Table 1: Examples of CNNs for image segmentation, and used data sets

CNN	Year	Used datasets
FCN	2014	PASCAL VOC
$SEGNET$	2015	CamVid
$DilatedConvolutions$	2015	VOC2012, COCO
$DeepLab$	2014-2017	PASCAL 2012, CityScapes
$RefineNet$	2016	PASCAL 2012
$PSPNET$	2016	PASCAL 2012, CityScapes
$LargeKernelMatters$	2017	PASCAL 2012, CityScapes
$MaskR-CNN$	2017	COCO

Convolutional neural network CNN is a class of deep neural networks mainly applied in image and video recognition. Extensive research has been carried out already for image segmentation using CNNs (see table 1) such as PSPNet [14], RefineNet [15], or Large-Kernel-Matters [16]. *Fully convolutional neural network* (FCN) by Long et al. [17], is another relevant example of image segmentation where a final fully-connected layer is replaced by another convolutional layer for a large receptive field to capture the global context of a scene. However, this results in a coarse segmentation maps due to the upsampling layers of the FCN.

Badrinaraynan [18] proposed *Segnet*, a multiclass deep-encoded-decoder-based CNN, that is more memory-efficient than the FCN and performs semantic pixelwise segmentation. Segnet eliminates the need of upsampling, as this decoder uses pooling indices, computed in the max-pooling step of the corresponding encoder, for non-linear upsampling.

One more class of CNNs, which uses dilated or atrous convolutions, is proposed in DeepLab by Chen et al. [19]. However, this type of convolutions is computationally expensive as its application uses high-resolution feature maps.

Although deep learning based object detection and recognition is becoming very common, however the main limitation lies in the lack of training data. A pothole too is like an object in the scene which can be detected/recognised through deep learning. To the best of our knowledge, we have not been able to find any publication and dataset about detecting potholes mainly through deep learning. Therefore, we have proposed their detection through transfer learning based approach.

3 Transfer Learning

Transfer learning is one of the current research area in CNN. In a CNN, the process of feature extraction is coarse in earlier layers and becomes fine (more specific) in later layers. In case of object detection, the later layers extracts the information about the position of the object. Whereas, for object recognition, all layers of CNN serve the common purpose of extracting features of the object being identified. Transfer learning is particularly useful when CNN cannot be trained from scratch, usually due to the lack of larger training datasets. It instead extracts the knowledge gained from a source domain D_s and transfers it to a target domain D_t.

3.1 Mask R-CNN

For the purpose of feature extraction, object detection and finally object location identification; multiple layers of CNN can be combined to work together. For example, *region-based convolutional neural network* (RCNN) by [21]. However, RCNN involves computationally expensive as well as time consuming training time. The RCNN consists of three independent models including a CNN based feature extraction, an SVM classifier and finally a regression model for object location identification based on bounding boxes (also known as *Region of Interest* (ROI)).

This problem of unifying three different models was solved, and evolved into a new network known as Fast R-CNN [22]. Fast R-CNN improved the training time of RCNN; however, the region proposals were still generated by an additional model. Hence, the training was still expensive.

Faster R-CNN removed this limitation of a generation of region proposals by a separate model and integrated the region proposal algorithm into the CNN model. Thus, Faster-RCNN is a single, unified model composed of a *region-proposal network* (RPN) and fast R-CNN with shared convolution feature layers.

Mask R-CNN [20] is an extension of Faster R-CNN to pixel-level instance segmentation. Mask R-CNN separates the classification and pixel-level mask prediction step, and added a third branch to predict an instance level segmentation along with other two branches of classification and localization. To predict a pixel level segmentation mask, a small fully-connected network is applied to each ROI.

The Mask R-CNN implementation uses a ResNet101 [23] and Feature Pyramid Networks [24]. A ResNet is a standard feature extractor which detects low-level features at early layers, and high-level features at later layers. The network accepts an image of 1024×1024 pixels. A smaller image is padded with zeroes to match with the expected image resolution. FPN is another improved feature extractor, a second pyramid that allows features at every level to have access to both lower- and higher-level features.

3.2 YOLO

YOLO is an objet detection network which uses a single regression problem [25]. We have used YOLO version 2-YOLOv2 for our experiments. It is an improved version of YOLO [26] as it generalizes better over image size. YOLO consists of 24 convolutional layers which is followed by two fully connected layers. An input image of size $[n \times n]$ pixels is passed through a CNN and output is a vector of bounding boxes (b). The input image gets divided into a $g \times g$ grid cells. Where, $g = \frac{n}{s^p}$ is computed from stride s and maxpool layers p.

For example, in YOLOv2 the input image size is $[n \times n] = [416 \times 416]$ pixels with $s = 2$ and $p = 5$. So, grid cell size is 13.

These grid cells produces N bounding boxes along with their confidence scores. Next step in YOLO is to perform non-max suppression, which is a process of removing bounding boxes with low object probability and highest shared area. However, YOLO has a limitation of one object rule which limits how close detected objects can be.

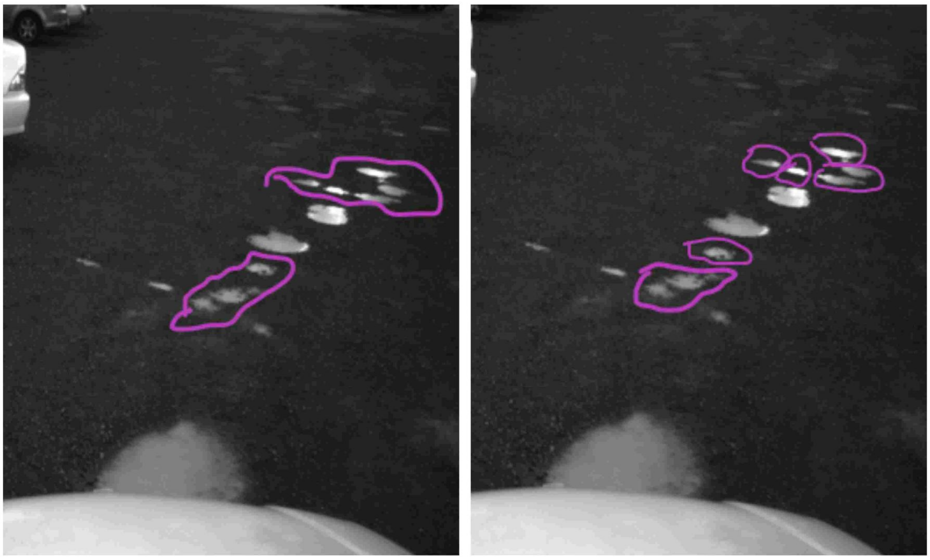


Fig. 2: Annotation challenge while labelling it for ground truth(Images from DLR dataset).

The bounding box consists of 5 numeric predictions: confidence, x, y, width, height. Where, confidence score represents *Intersection over Union* `IoU` between predicted and ground truth box. x, y are coordinates centre of box relative to grid cell, width and height are relative to input. During testing the confidence score represents how likely and accurately the bounding box has the object. YOLO produces number of bounding boxes per grid cell, only one of them is responsible for the object being detected. So, the bounding box with higher `IoU` is selected. The loss function in YOLO mainly comprises of classification, localization and confidence loss.

4 Experiments

Previous studies have demonstrated that “there is no uniform road damage dataset available openly, leading to the absence of a benchmark for road damage detection” [27]. Manual labelling of irregular potholes for producing ground truth is a laborious and challenging task. For pixel-wise annotation in an image, it takes around $30 - 50$ seconds depending upon the choice of tool and network. Annotating a pothole using a bounding box is difficult as pothole can be of any shape such as circular, longitudinal or multiple potholes adjacent to each other (see Fig. 2).

The potholes marked in purple colours can be perceived as one big pothole or can be counted separately. Two potholes in left image, six potholes in right image, other potholes are not marked here, but considered in experiments.

4.1 Datasets

As there is no public benchmark available for potholes to carry out experiments. So, for training dataset, we used:

1. `CCSAD Urban sequence` 1 [28]
2. `DLR`. Recorded using the *integrated positioning system* (IPS) [29, 30]
3. `Japan`. Consists of $163,664$ road images each of resolution $[600 \times 600]$ pixels collected in Japan [27]
4. `Sunny`. Consists of 48913 frames recorded on a very sunny day [31]

For testing we used the following datasets:

1. `CCSAD` `Urban sequence` 2
2. `CCSAD` `Urban sequence` 1 with exclusive images different from training images
3. `PNW` Pacific Northwest dataset consisting of $19,784$ images [32]

4.2 Transfer Learning with MASK R-CNN

For *Transfer learning with MASK R-CNN* (LM1) we used per pixel manually labelled ground truth images. As training dataset, we used 247 training images from `CCSAD's Urban Streets sequence`1, `DLR`, `Japan` and manually labelled ground truth annotations. As test dataset, we used `CCSAD's` sequences no. 2 and very challenging `PNW` snow dataset.

For hyper-parameters, we used grid search and experimented with learning rate from 1 to 0.000001 and realized that majority of the learning rates failed to train our model. A very slow learning rate such as 0.00001 and very high such as 1 never converged causing instability. So,we used a learning rate of 0.001 as it helps to avoid the problem of an exploding gradient.

We train the network using stochastic gradient descent with a learning momentum of 0.9 to identify an object class as pothole.

For training, the batch size is 2. We trained for 30 epochs, which took 14 hours on our Ge Force GTX GPU achieving overall precision and recall on testing dataset as 88.7 and 84.6 respectively. Some of the frames from testing dataset is shown in Fig. 6. Loss in *LM1* is calculated using the following equations.

Loss is defined following [20] as

$$LM1_{loss} = LM1_{class} + LM1_{bbox} + LM1_{mask} \tag{1}$$

where

$$LM1_{class} = \frac{1}{N_{class}} \sum_{i} -a_i^* log a_i - (1 - a_i^*) log(1 - a_i) \tag{2}$$

$$LM1_{bbox} = \frac{\lambda}{N_{bbox}} \sum_{i} a_i^* \cdot L_1^{smooth}(b_i - b_i^*) \tag{3}$$

$$LM1_{mask} = -\frac{1}{m^2} \sum_{1 \leqslant i,j \leqslant m} [l_{ij} log \hat{l}_{ij}^t + (1 - l_{ij}) log(1 - \hat{l}_{ij}^t)] \tag{4}$$

where $LM1_{class}$ is a loss function over two classes, $LM1_bbox$ is bounding box loss, and $LM1_mask$ is a mask loss. And: N_{class} is a normalization value, a_i, a_i^* are predicted and ground truth probability of an object being detected respectively. λ is a balancing constant and b_i, b_i^* are predicted and ground truth four coordinate values. l_{ij} is the label of cell(i, j) in the true mask and $\hat{l}_{ij}^t$ is predicted value of same cell for ground-truth class t.

The $LM1_{class} + LM1_{bbox} + LM1_{mask}$ during training and validation is shown in Fig. 3. The average training and validation loss in Fig. 4 shows after certain number of iterations loss does not increase further.

To preserve the aspect ratio of uniform size $1,024 \times 1,024$, zero padding is added to the top and bottom of an image as shown in Fig. 5, top left.

LM1 is composed of a three-stage framework. The first stage scans the whole image for generating proposals. The second stage classifies the proposals, generates bounding boxes, and third stage produces masks of an object.

Stage 1: Region proposal network (RPN). RPN is a lightweight neural network that scans over the backbone's feature map, using a sliding window to generate anchors that are typically boxes distributed over the image area. The sliding window operation is handled by its convolutional nature, and this is very fast on a GPU. The output of the RPN is an anchor class and a bounding-box refinement. The output of the RPN is a grid of anchors (see Fig. 5, top right) at different scales.

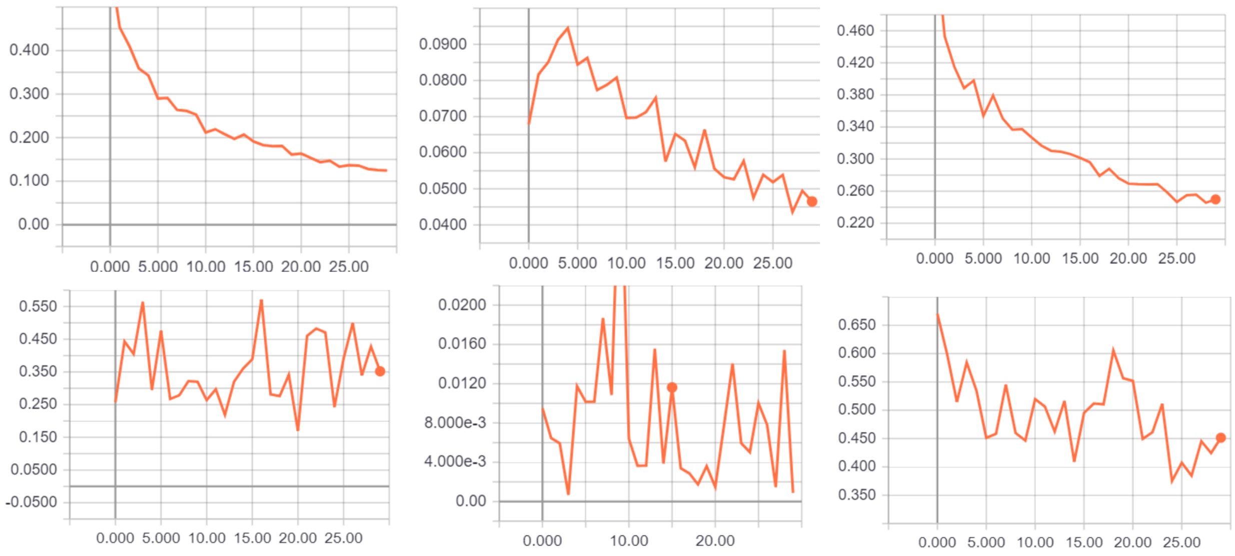


Fig. 3: $LM1_{box} + LM1_{class} + LM1_{mask}$ during training (upper row) and validation (bottom row). Here horizontal axis represents number of epochs and vertical axis represents loss.

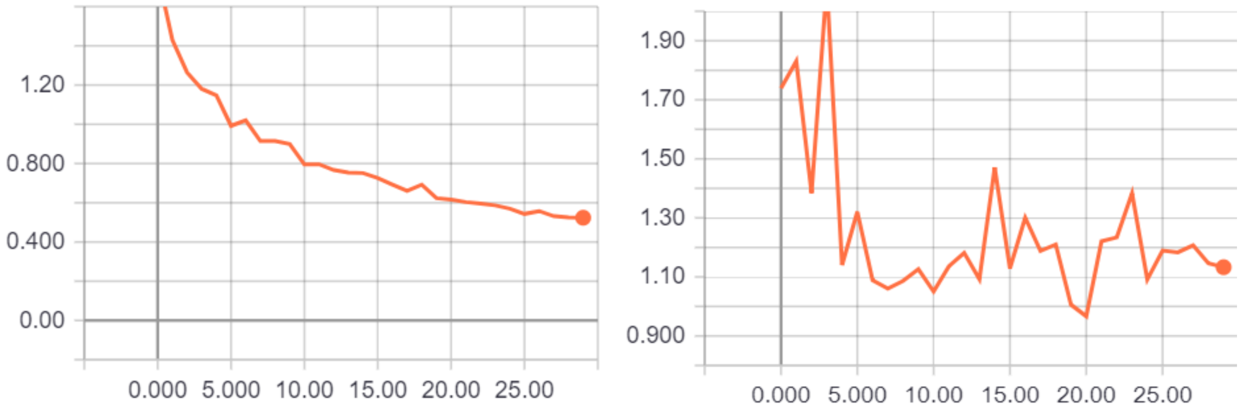


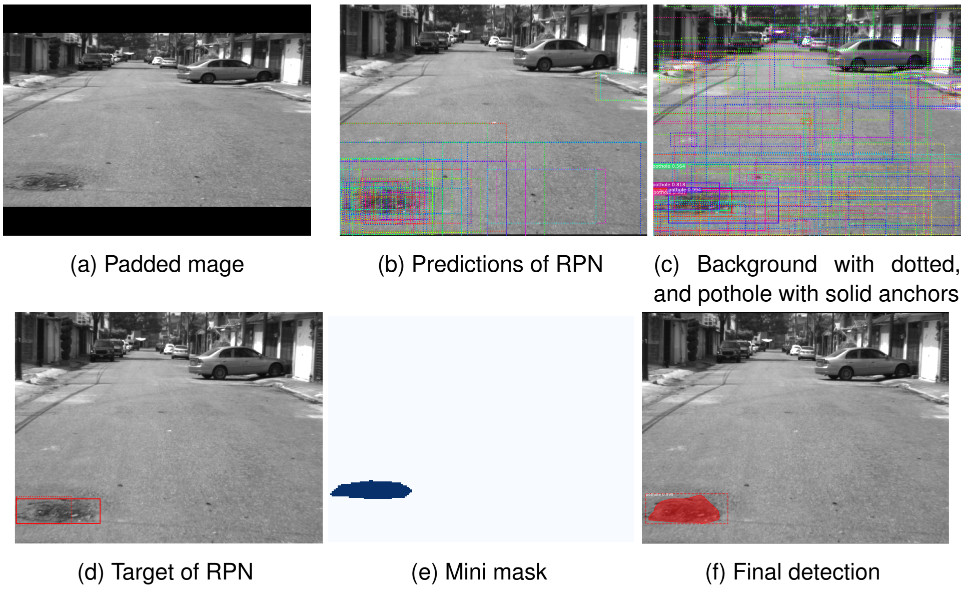
Fig. 4: $LM1_{loss}$ during training and validation. Here horizontal axis represents number of epochs and vertical axis represents loss.

We fine-tune the RPN end-to-end for a region proposals, initialized by a pre-trained CNN image classifier. IoU > 0.7 and < 0.3 define positive or negative samples, respectively. A small $n \times n$ window slides over the convolved feature map of the entire image. Anchor is produced to predict the multiple regions, at each sliding position. We used 256 anchors per image for RPN training.

Stage 2: Refined bounding boxes. To precisely map bounding boxes to the regions of an image, Mask R-CNN improves the RoIAllign layer of the network for pixel-level segmentation. It removes the harsh quantization of the RoIPool layer to properly encapsulate the extracted features with the input. This stage accepts the refined anchors from the RPN and classifies the anchors as shown in Fig. 5, middle left. A refined bounding box with a final detection is shown in Fig. 5, middle right. It trains an object detection model by using the proposals obtained by RPN.

Stage 3: Instance masks. The mask branch is a CNN that accepts positive regions as input generated by the classifier during Stage 2, and predicts a low resolution 28×28 soft mask for it(see Fig. 5, bottom row). A soft mask differs from a binary mask as these are represented by float numbers and hold more details. We fine-tune the layer of Mask R-CNN, according to our object class name.

The reasons for false-positives resulted in some of the frames during training is due to our network tried to identify a pothole which might be a patch or an emerging pothole. As a pothole has no defined shape or features, it is hard to label them manually.



(a) Padded mage

(b) Predictions of RPN

(c) Background with dotted, and pothole with solid anchors

(d) Target of RPN

(e) Mini mask

(f) Final detection

Fig. 5: Illustration of detection steps.

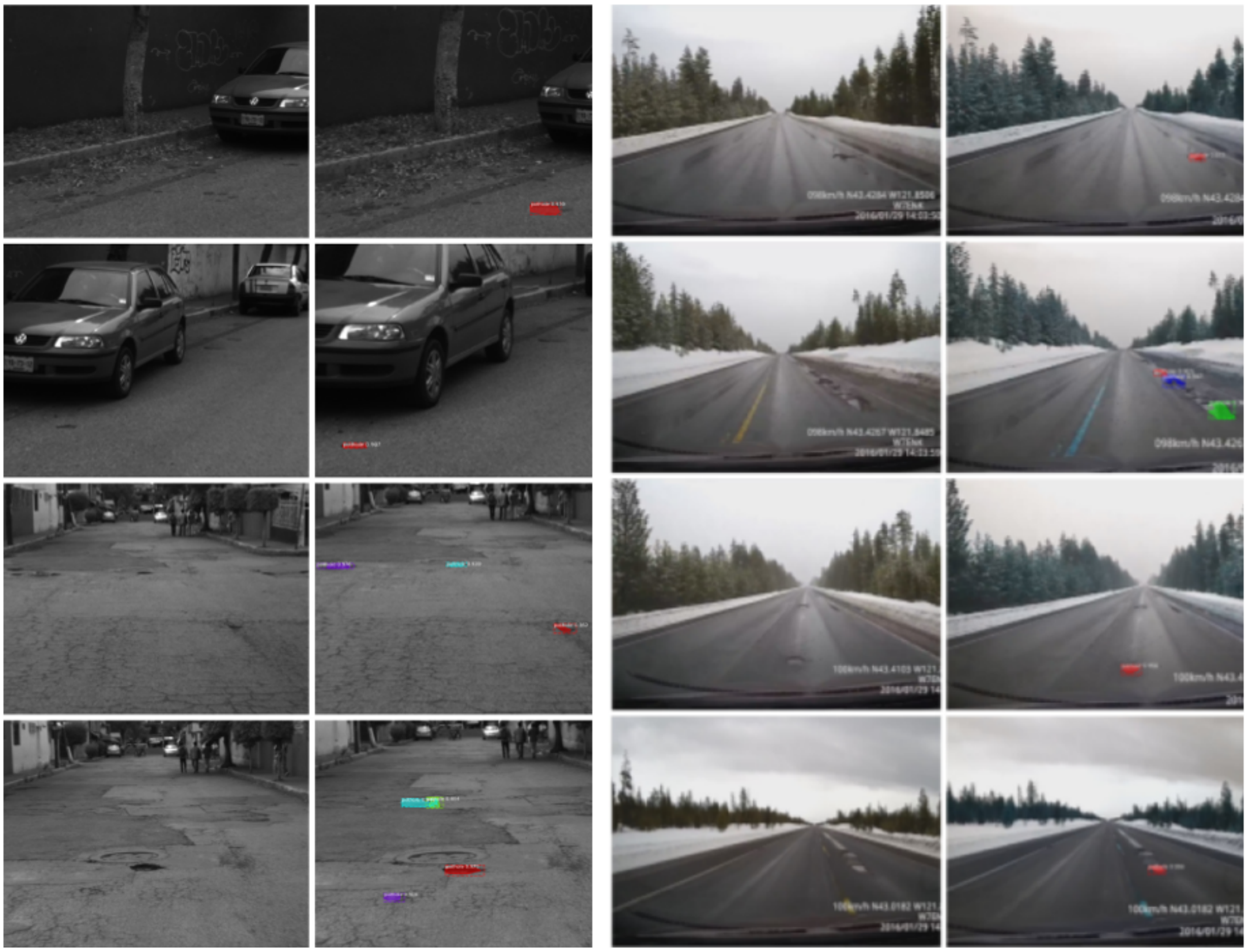


Fig. 6: Detected road potholes using *LM1*.

4.3 Transfer Learning with YOLO

We describe now our *Transfer learning with YOLO* (LM2). For real time detection of potholes, we used transfer learning using another object detector- *You only look once* (YOLO). The training and testing dataset for YOLO was same as MASK R-CNN. However, the annotation format is different in case of YOLO, which is a bounding box and not a mask. We trained our network using TESLA K80 GPU. We started with setting the input image subdivision value as 8, but due to a resulting high memory requirement we changed it to 32. We used 64 images per batch. To optimize the produced weights, this *LM2* model uses three loss values as functions as

$$LM2_{loss} = LM2_{class} + LM2_{bbox} + LM2_{conf} \tag{5}$$

$$LM2_{class} = \sum_{i=0}^{g^2} I_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}(c))^2 \tag{6}$$

$$\begin{aligned} LM2_{bbox} = \lambda_{bbox} \sum_{i=0}^{g^2} \sum_{j=0}^{b} I_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ \lambda_{bbox} \sum_{i=0}^{g^2} \sum_{j=0}^{b} I_{ij}^{obj} [(\sqrt{\omega_i} - \sqrt{\hat{\omega}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \end{aligned} \tag{7}$$

$$\begin{aligned} LM2_{conf} = \sum_{i=0}^{g^2} \sum_{j=0}^{b} I_{ij}^{obj} [(c_i - \hat{c}_i)^2 + \\ \lambda_{nobj} \sum_{i=0}^{g^2} \sum_{j=0}^{b} I_{ij}^{nobj} [(c_i - \hat{c}_i)^2 \end{aligned} \tag{8}$$

where $LM2_{class}, LM2_{bbox}, LM2_{conf}$ are classification, bounding box and confidence loss respectively. I_i^{obj} is one when pothole is present otherwise it is 0. $p_i(c)$ is conditional probability for class c in cell i. $\hat{p}$ is conditional class probability. (x, y) is the predicted and $(\hat{x}, \hat{y})$ is the actual bounding box position, similarly (ω, h) is width and height of the bounding box. λ is a constant to penalize bounding box predictions. c is the confidence score associated with the bounding box predictor and $\hat{c}$ is the IoU of the predicted to the ground truth bounding box.

Initially, the learning rate was set to 0.01. However, after $1,000$ iterations the average loss kept on increasing. Therefore, we used 0.0001 for learning rate. The average loss drops from 16.582247 to 0.635149 in first 100 iterations. Therefore, we slowly increase the learning rate from 0.0001 to 0.001.

We trained the network for around $8,000$ iterations and checked the *mean average precision* mAP [33] value for different iterations at $5,400$, $6,400$, $7,400$. The mAP value of $5,400$ iterations was higher than other iteration weights and also $Loss_LM2$ does not decrease further from 0.090189 after five thousand iterations. Hence, using Early Stopping Point, the model is selected after $5,400$ iterations. A sample of frames ares shown in Fig. 7 from PNW testing dataset.

The originality of our work lies in focusing on pothole identification under challenging illuminating and weather conditions. Recent advancements in the area of deep learning supported the field of object detection and one can now autonomously identify potholes. The LM2 model saves identified pothole images separately, which can be used for analysing to plan in advance to patch up potholes. This will help to automate the laborious and expensive task of manually pothole identification. Though the accuracy of LM2 model is not high as *LM1* model, however it still shows the interesting trade off between speed and accuracy.

5 Conclusions

We utilized modern approaches for object detection and demonstrated that pothole identification can be made automated in real time. This research also fills the gap of different datasets recording under different scenarios. We found out that LM1 method outperforms LM2 method with high accuracy. However, with more training, labelled datasets and use of more than one GPU the reliability of the models can be increased.

Considering the success of our models on testing dataset, our methods allow for future possibilities in numerous ways such as using the output of LM1 method as an annotated image to train the LM2 method.

While this research provides promising steps toward pothole identification, one could extend these models to extract a variety of other metrics such as depth and size of identified potholes.

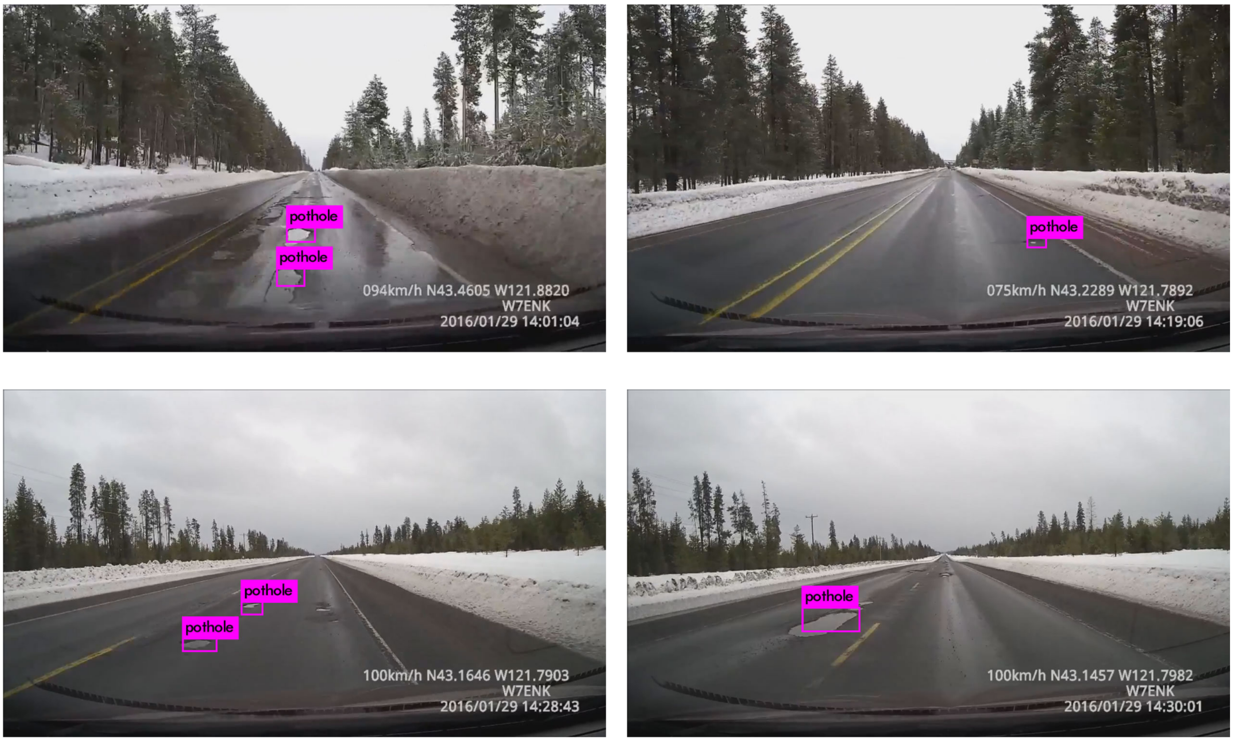


Fig. 7: Detected road potholes using *LM2*.

The reported research was motivated by collaboration with *Northland Innovation Centre*'s N3T project; see [34–36] for joint publications so far, also including the *German Aerospace Centre* (DLR) in the context of their IPS [29, 30] (see above).

References

1. Road melts around NZ. www.newshub.co.nz/home/new-zealand/2019/01/roads-melt-around-new-zealand-under-brutal-summer-heat.html January 27, 2019.
2. The pothole facts. www.pothole.info/the-facts.
3. Auckland transport. www.neighbourly.co.nz/e-edition/east-bays-courier/27329 January 17, 2018.
4. Times of India, www.timesofindia.indiatimes.com/india/deadly-pits-potholes-claimed-11386-lives-during-2013-16/articleshow/60774243.cms.
5. J. Horowitz. www.nytimes.com/2018/03/25/world/europe/italy-rome-potholes.html. March 25, 2018.
6. Christchurch report, www.stuff.co.nz/the-press/news/100847641/christchurch-the-pothole-capital-of-new-zealand/, February 04, 2018.
7. D. O'Carroll, "For the love of pizza, Domino's is now fixing potholes in roads", article on stuff.co.nz, June 13, 2018.
8. www.openaccessgovernment.org/420-million-budget-for-potholes-welcome-but-it-is-not-enough/53805/ October 30, 2018.
9. JLR, www.landrover.com/experiences/news/pothole-detection.html.
10. Solution, www.corporate.ford.com/innovation/solving-the-bumpy-commute.html.
11. Magic Body Scan, www.mercedes-benz.com/en/mercedes-benz/innovation/magic-body-control/, 2013.
12. Fugro roadware, Canada, www.roadware.com/applications/.
13. Pavement management, Denmark, www.greenwood.dk/road.php.
14. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network", Proc. *Int. Conf. CVPR*, pp. 2881-2890, 2017.
15. G. Lin, A. Milan, C. Shen, and I. D. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation", Proc. *Int. Conf. CVPR*, 1(2), 2017.
16. C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters - Improve semantic segmentation by global convolutional network", Proc. *Int. Conf. CVPR*, pp. 1743-1751, 2017.
17. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", Proc. *Int. Conf. CVPR*, pp. 3431–3440, 2015.
18. V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE Trans. Pattern Analysis Machine Intelligence* 39(12): 2481 - 2495, 2017.
19. L. -C Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", *J. IEEE Trans. Pattern Analysis Machine Intelligence*, 40(4): 834–848, 2018.
20. H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask R-CNN", *CoRR*, 1703.06870, 2017.
21. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", Proc. *CVPR*, 580–587, 2014.
22. R. Girshick, "Fast R-CNN", Proc. *ICCV*, 1440–1448, 2015.
23. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", Proc. *CVPR*, 770–778, 2018.
24. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection", Proc. *CVPR*, 1(2), p. 4., 2017.
25. J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger", Proc.*CVPR*, pp. 7263-7271, 2017.
26. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, A, "You only look once: Unified, real-time object detection", Proc.*CVPR*, pp. 779-788, 2016.
27. H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone", *J. Computer-Aided Civil Infrastructure Engineering*, 33(12): 1127-1141, 2018.
28. R. Guzmán, J. -B. Hayet, and R. Klette, "Towards ubiquitous autonomous driving: The CCSAD dataset", Proc. *CAIP*, 582–593, 2015.
29. A. Börner, D. Baumbach, M. Buder, A. Choinowski, I. Ernst, E. Funk, D. Grießbach, A. Schischmanow, J. Wohlfeil, and S. Zuev, "IPS - A vision-aided navigation system" in Proc. *Advanced Optical Technologies*, 6(2):121–129, 2017.
30. D. Grießbach, D. Baumbach, and S. Zuev, "Stereo-vision-aided inertial navigation for unknown indoor and outdoor environments" in Proc. *Indoor Positioning Indoor Navigation* IEEE Xplore, 2014.
31. S. Nienaber, M.J. Booysen, and R.S. Kroon, "Detecting potholes using simple image processing techniques and real-world footage," in Proc. *Southern African Transport Conference*, July 2015.
32. PNW dataset, www.youtube.com/watch?v=BQo87tGRM74.
33. J. Hui, "mAP (mean average precision) for object detection", article on medium.com, March 07, 2018.
34. D. Baumbach, H. Zhang, S. Zuev, J. Wohlfeil, M. Knoche and R. Klette, "GPS and IMU Require Visual Odometry for Elevation Accuracy", in Proc. *Advanced Video and Signal-based Surveillance*, 2018.
35. I. Ernst, H. Zhang, S. Zuev, M. Knoche, A. Dhiman, H.-J. Chien, and R. Klette, "Large-scale 3D Roadside Modelling with Road Geometry Analysis: Digital Roads New Zealand", in Proc. *Pervasive Systems, Algorithms and Networks*, 2018.
36. H. Zhang, I. Ernst, S. Zuev, A. Börner, M. Knoche, and R. Klette, "Visual odometry and 3D point clouds under low-light conditions", in IEEE Proc. *Image Vision Computing New Zealand*, doi: 10.1109/IVCNZ.2018.8634769, 2018.