 doi:10.20944/preprints202102.0569.v1

Vision-Based Deep Learning Algorithm for Detecting Potholes

Kanushka Gajjar [1], Theo van Niekerk [2], Thomas Wilm [3] and Paolo Mercorelli [4,*]

[1] Engineering, Built Environment and IT, Nelson Mandela University, Port Elizabeth, South Africa ; kanushkagajjar0@gmail.com
[2] Institute of Product and Process Innovation –PPI, Leuphana University of Lueneburg, Universitaetsallee 1, 21335 Lueneburg; mercorelli@uni.leuphana.de
+* Correspondence: kanushkagajjar0@gmail.com; Tel.: +27 788732986

Abstract: Potholes on roads pose a major threat to motorists and autonomous vehicles. Driving over a pothole has the potential to cause serious damage to a vehicle, which in turn may result in fatal accidents. Currently, many pothole detection methods exist. However, these methods do not utilize deep learning techniques to detect a pothole in real-time, determine the location thereof and display its location on a map. The success of determining an effective pothole detection method, which includes the aforementioned deep learning techniques, is dependent on acquiring a large amount of data, including images of potholes. Once adequate data had been gathered, the images were processed and annotated. The next step was to determine which deep learning algorithms could be utilized. Three different models, including Faster R-CNN, SSD and YOLOv3 were trained on the custom dataset containing images of potholes to determine which network produces the best results for real-time detection. It was revealed that YOLOv3 produced the most accurate results and performed the best in real-time, with an average detection time of only 0.836s per image. The final results revealed that a real-time pothole detection system, integrated with a cloud and maps service, can be created to allow drivers to avoid potholes.

Keywords: CNN; Faster R-CNN; SSD and YOLOv3

1. Introduction

Since South Africa is a developing country, the development and maintenance of roads is of great significance. Well-maintained roads contribute largely towards the country's economy and tarred roads have become an absolute necessity in the 21st century in light of economic demands on government by the citizenry. Potholes can be defined as areas of a road surface that have formed a hole after being cracked and worn away [1]. Potholes start as small cracks, which if not repaired timeously, can increase considerably in size. Flat and smooth surfaces are required to ensure a comfortable drive, however, potholes can result in an unpleasant and potentially dangerous ride. The tyres are likely to get damaged and would require re-alignment. Currently, there is no mechanism to caution drivers of approaching potholes. Consequently, potholes remain a serious hazard. However, technology is available that reduces the impact of a pothole on a vehicle. Variations of this technology have been installed in several vehicles including the S-class Mercedes, Ford Focus, Jaguar Land Rover and Audi A8. The technology in the S-class Mercedes is referred to as Magic Body Control, which is a suspension that can predict surface unevenness and prepare accordingly [2]. This technology does not completely eliminate the effect of a pothole but simply reduces it. However, it is possibly more beneficial for a driver to detect any damage in the road beforehand. Driving over a pothole may have serious implications for a vehicle, for example, a flat, or damage to a tyre, rims, suspension, steering and possibly the body of the car [3]. Complete avoidance of a pothole would eliminate these possibilities. The overall aim of this project is to identify the most suitable deep learning algorithm to detect a pothole whilst driving, determine the GPS location thereof, display the determined position on a map as pin and send the data to a cloud

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

server for information storage and sharing. An investigation of various deep learning detection algorithms will be conducted. The algorithms selected for this project include Faster R-CNN, SSD and YOLOv3. Each of the selected algorithms will be compared and the most suitable will be utilised for the purpose of this project. The IoT system will collect the GPS data from its module and store it in the cloud. Hence, no manual data must be collected. The data must be in real-time and monitored through the cloud.

2. Materials and Methods

2.1 Data collection and processing

It was necessary to identify an area in South Africa where potholes are prevalent. St. Francis Bay and Jeffrey's Bay in the Eastern Cape Province were identified for the purpose of this study. Additional data containing pothole images was found online. The latter was produced by [4] who completed a project relating to pothole detection using vision and machine learning methods. The images taken by [4] were taken in the Vaal Triangle area in Gauteng, Stellenbosch and Somerset West in the Western Cape. Initially, it was assumed that these two sets of data would be adequate, however, with the advent of time, it was revealed that more images were required to enhance training for the network. If the dataset is too small, overfitting, which refers to a model that models the training data too well can occur. The detail and the noise in the training data has been mastered to such an extent that the performance of the model on new data is negatively impacted [5]. Thus, it was necessary to produce additional images. These images which were produced by VWSA were taken in Uitenhage and surrounding areas in the Eastern Cape Province. These areas were identified as those containing a large number of potholes.

The entire dataset of instances of potholes comprised of 1910 images. The images of the datasets was gathered through a GoPro Hero 3+. This camera was selected because it produces good quality images whilst moving. Thus, it was not necessary to de-blur the images in the pre-processing steps, thereby reducing the processing time. The camera was set at a resolution of 1280 x 720pixels. The goPro used by [4] was set to the time lapse mode at an interval of 0.5 seconds/image and at a resolution of 3680 x 2760pixels.

The collected data was arranged into positive and negative sets. The positive set of data contained images with potholes, while the negative dataset comprised of images without potholes. The data collected by [4] was divided into two scenarios, positive and negative sets as well as divided into a complex and simple scenario. In the simple scenario, it was assumed that the roads were well lit and open, while in the complex scenario, a more real-world scenario is depicted. This scenario includes instances of shadows on the road. This dataset comprised of images which were taken at various times throughout the day. All of the data was separated into the training and test sets.

The images were cropped such that only the road surface was illustrated. The sky and other unnecessary information was cropped out of the image. This had to be done because the computer utilised for training had inadequate GPU memory for training. Reducing the size of the image and eliminating unnecessary data such as the sky and grass resulted in less memory to train the network. The computational time of the algorithms was also reduced as a result of cropping the images. The images were not resized because significant aspects of the image would have been lost thus making it difficult to train the network.

The cropped images containing potholes were annotated by utilising the Image Processing Toolbox in MATLAB. The annotation process involved colouring in the regions in which there were potholes manually. This colouring process was completed by utilising the WACOM Cintiq Pro 13. This compact device is an advanced creative pen display which enables a user to have a direct pen-on-screen connection. Using this device saved time because it would have been extremely difficult and time consuming to colour each pothole using a mouse. Once the masks were created from the semantic segmentation, these were utilised to create the bounding boxes. Since the VWSA dataset was acquired at a later stage, the images within this dataset were annotated using a different graphical annotation tool: labelImg. Only bounding boxes were created for this dataset. Initially, the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

results acquired using semantic segmentation and bounding boxes would have been compared, however, at a later stage it was decided that comparisons between various other networks using only bounding boxes would be made. The overall process to be followed to achieve the desired results is illustrated in Figure 1 below.

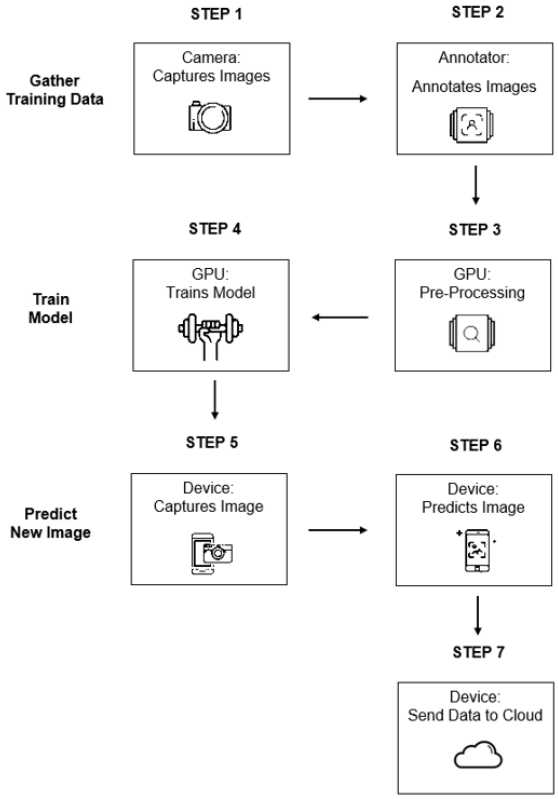

Figure 1. Overall process to be followed to achieve the end goal

2.2 Development and methodology

Various network architectures were utilized for this project. More than one network architecture was utilized to determine which performed the best with the given dataset. The network architectures included SSD (Single Shot Detection) with the Inception v2 backbone, Faster RCNN (Region-based CNN) with Inception v2 backbone and YOLOv3 (You Only Look Once). This section discusses why YOLOv3 was selected as the network used for the application of this project. The overall network architecture is also expounded upon in this section.

The results revealed that although Faster R-CNN was more accurate than SSD, the inference time was extremely slow and unsuitable for real-time detection. Although SSD was fast, the accuracy of the detection was unsatisfactory. Further research was completed and it was revealed that videos generally shot at a speed of at least 24fps, the Faster R-CNN would likely not be able to keep pace. Since Faster R-CNN is a regional based method which comprises of two phases, proposing regions and processing these, it proves to be somewhat inefficient for real-time detection.

After this comparison was made, it was decided that a third network should be explored to optimise the results. Since single stage-detectors produce a higher inference, further research was conducted on such detectors. Through thorough research, it was revealed that the YOLOv3 network performs better for real-time applications because the detection time per object is less when compared to Faster R-CNN and SSD [6]. Generally, when dealing with objects of large sizes, SSD performs well. SSD utilises upper layers for

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

detection. Consequently, the performance for small objects is not sound. For the application of this project, the size of the potholes could vary significantly, thus resulting in poor accuracy.

2.3 Training YOLOv3

The Darknet deep learning framework developed by Joseph Redmon [7] was utilised for this project. It was necessary to first download and build this framework onto the computer being used. The first task is to prepare the dataset. The YOLOv3 annotated files are required to be in a text file format. Thus, the ".xml" files were converted to ".txt" files. The ".txt" file comprises of rows which represent a single bounding box in the image and contains the following information about the bounding box:

<object-class-id> <centre-x> <centre-y> <width> <height>

The first field, object-class-id, is an integer that represents the class of the object. This number ranges from 0 to (number of classes-1). Since this project only comprises of one class, this number is always set at 0. The second and third field, centre-x and centre-y are the x and y co-ordinates of the centre of the bounding box divided by the image width and height. The fourth and fifth field, width and height are the width and height of the bounding box divided by the image width and height.

The second to the fifth entries are all floating-point values that range between the values 0 and 1. The dataset was then split into the training and test sets. The process of transfer learning was used in the next step. A pre-trained model containing convolutional weights trained on ImageNet was thus utilised. By utilising these weights, the training time was reduced significantly [8]. The darknet.data, classes.names and darknet-yolov3.cfg files, which is included in the code distribution, requires information relating to the specifications of the object detector and relevant paths. Thus, these files were edited such that the relevant information was provided for each. Various hyper-parameters were configured in the configuration file. Once all the various components for training were set, the training could take place. The training continued till the loss value dropped below a specific threshold [9].

Once the YOLOv3 algorithm was able to accurately detect potholes on the road, it was necessary to upload the GPS co-ordinate of the detected pothole to a cloud server.

Comparisons were made between various USB GPS receivers to determine which would be the most suitable for the application of this project. After thorough research was completed, it was decided that the best suited USB GPS receiver would be the GlobalSat BU-353-S4 receiver for this project. This receiver comprises of a highly sensitive, low power consumption chipset in an ultra-compact form factor. It is powered by a SiRF Star IV GPS chipset and provides superior performance in urban canyons and dense foliage. The device is built with a magnet which can stick to the top of a vehicle. Elemental exposure is not a concern because it is able to withstand both freezing and extreme hot temperatures [10].

The location of the detected pothole was required to be sent to a cloud server so that other drivers can be informed of the location of the potholes. The cloud server utilised for this project was Ubidots STEM. Ubidots is a two platform company, which comprises of Ubidots and Ubidots STEM. The standard STEM package is a non-commercial license that allows easier access to students, researchers and hobbyists globally [11]. Ubidots is an Internet of Things (IoT), data analytics and visualisation company. Data that is gathered from various sensors can be transformed into useful information, allowing for business decisions and machine-to-machine interactions be made. Educational research is possible through the utilisation of Ubidots. This platform allows for integration of IoT into business and research [11].

Figure 2 below illustrates an overview of the overall system architecture. The image on the road will first be captured via the camera. Once this image is captured, the neural network will run, if a pothole is detected. The GPS location of the detected pothole will be determined via the GPS module. The collected GPS data will be sent to the cloud server,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

Ubidots. This information will be available for all those who have access to the Ubidots account used to store the data. The relevant parties can then be informed of the potholes on the road and the location thereof.

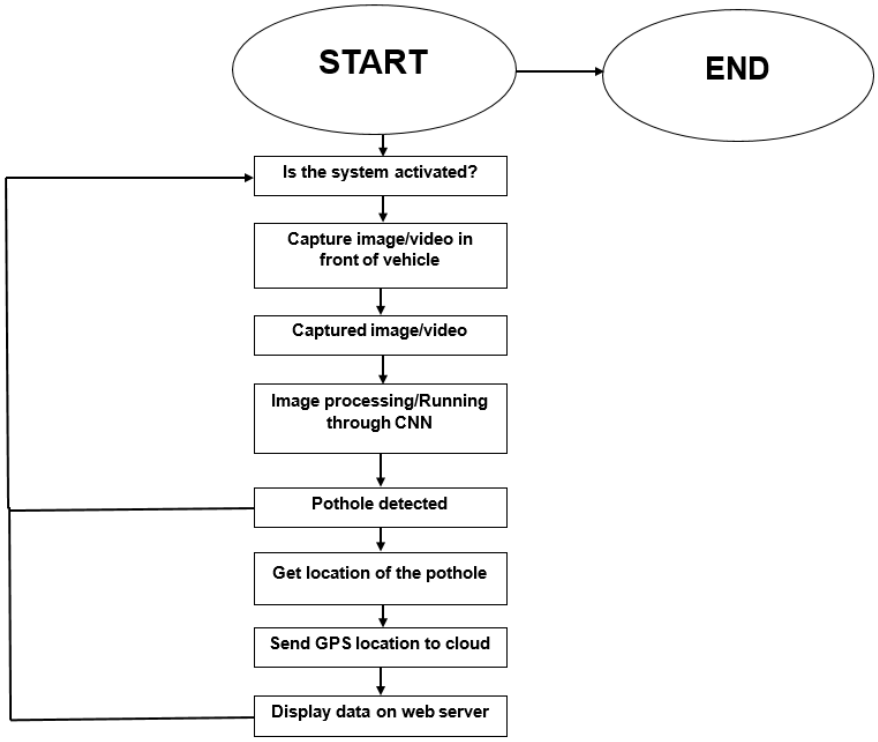

Figure 2: Overall system architecture

3. Results

In this section, the various evaluation metrics to determine which network performed the best is discussed. The three networks are compared and the results gathered from the discussed evaluation metrics is presented. The integrated system performance is discussed in this section too.

3.1. Evaluation Metrics

Mean average precision is a metric utilised to evaluate object detectors. It is the average of the average precision. To comprehend MAP, it is necessary to define the terms 'precision', 'recall' and 'IoU' (Intersection over union). Precision can be defined as how repeatable a measurement is [12]. It is the percentage of the results that is relevant. An example of precision is how close a second arrow is to the first arrow on a dart board [13].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall can be defined as the percentage of the total relevant results that is classified correctly by the algorithm [14].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

If precision is increased then recall will decrease and vice versa.

IoU can can be defined as the ratio of the area of intersection and area of union of the ground truth and predicted bounding boxes. The "ground truth bounding box" is the bounding box and its co-ordinates are provided in the training set [15]. Figure 3 below illustrates that the green box represents the ground truth box while the red box is what the model predicts. It is clear that these two boxes have different co-ordinates. The area of intersection is where the one box overlaps the other and the area of union is the total area covered by both bounding boxes [16].

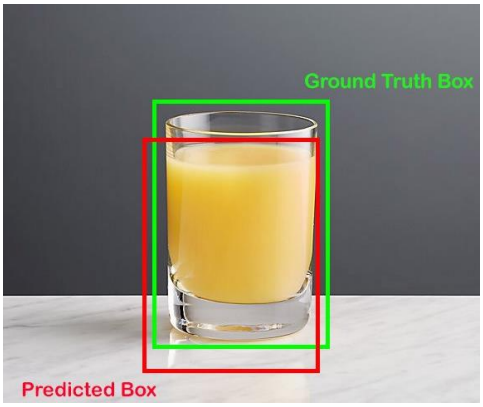

Figure 3: Representation of ground truth box and predicted box and IoU [16]

The confidence score can be defined as the probability that an anchor box contains an object. The confidence is predicted by a classifier. Both the IoU and confidence allows one to determine whether a predicted box is a true positive, false positive or false negative. A threshold value of 0.5 is predefined for the IoU [15].

A detection is considered a true positive (TP) only if the following three conditions are met [17]:

- Confidence score > threshold
- The predicted class matches the class of a ground truth
- The predicted bounding box IoU (e.g. 0.5) is greater than the threshold of the ground-truth

If the above conditions are not met, the predication is considered a false positive (FP). When the confidence score of a detection that is supposed to detect a ground truth is lower than the threshold, it is considered a false negative (FN) [17]. When the confidence score of a detection that is not supposed to detect anything is lower than the threshold, it is considered a true negative (TN). However, this is not of great significance in object detection [16].

A numerical metric, known as average precision (AP) can be utilised to evaluate the performance of a detector. AP is essentially the precision averaged over all unique recall levels. To reduce the fluctuations in the curve, it is necessary to interpolate the precision at multiple recall levels before actually calculating the metric. The interpolated precision p_{interp}, at a specific recall level (r) is defined as the highest precision found for any recall level $r' \geq r$ [18]:

$$P_{interp}(r) = \max_{r' \geq r} p(r') \quad (3)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

There are two ways to select the levels of recall at which the precision should be interpolated. Traditionally, 11 equally spaced recall levels are selected (i.e., 0.0, 0.1, 0.2,...1.0). A new standard selects all unique recall levels presented by the data. The new method is more advanced to enhance the precision and measure differences between methods with low AP. It is possible to define AP as the area under the interpolated precision-recall curve [18].

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (4)$$

Where r_1, r_2,...,r_n is the recall levels at which the precision is first interpolated. The calculation of AP only considers one class. Only one class, potholes, was used for this project.

Average recall (AR) is a numerical metric that can also be utilised to compare the object detector performance. AR is essentially the recall averaged over all $IoU \in [0.5, 1.0]$ and can be determined by the following equation [19]:

$$AR = 2 \int_{0.5}^{1} recall(o) do \quad (5)$$

Where o is IoU and $recall(o)$ is the corresponding recall.

3.2 Results

Table 1 below illustrates the results of the various performance metrics for each of the networks which were trained.

Table 1. Performance metrics results of each network

	Average Precision	Average Recall	F1-Score	mAP @ 0.5 IoU	Detection time per image (average)
Faster R-CNN	0.077	0.663	0.137	0.415	7.02s
SSD	0.043	0.326	0.076	0.185	4.815s
YOLOv3	0.347	0.32	0.42	0.347	0.836s

The YOLOv3 network achieved the precision value of 0.347 whereas Faster R-CNN and SSD achieved average precision values of 0.077 and 0.043 respectively. However, the recall value of YOLOv3 was lower than that of Faster R-CNN, indicating that while the YOLOv3 network has a higher proportion of positive results in the correctly predicted values, the Faster R-CNN network outperforms it in its ability to correctly predict the positive results. The mAP value of Faster R-CNN was also higher than that of YOLOv3, further proving that Faster R-CNN has a greater level of precision in detecting potholes. It should be noted that, although the size of each detected pothole is not the same, Faster R-CNN performs the best out of all of the networks regardless of the size of the object to be detected. The average detection time per image was the lowest for YOLOv3. Since the project had to be utilized in real-time, the YOLOv3 network was selected as the network on which the other operations would run i.e. acquire the GPS location of the detected

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 February 2021 doi:10.20944/preprints202102.0569.v1

pothole and storing it in the cloud. This network was selected because the inference time per image was the lowest.

The vehicle was driven at 60km/h to adhere to the speed limit set in the city. The webcam used for the real-time testing was positioned such that the view of the road was akin to a driver's viewpoint and the maximum area of the road was captured. The webcam was placed inside the vehicle. When driving at a speed of 60km/h and taking into consideration that the detection time per image of the YOLOv3 network was 0.836 seconds, the calculated distance from which potholes can be detected is 13.877m. Determining the distance in real-time was beyond the scope of this project. Once the pothole was detected, the GPS location of the detected potholes had to be acquired. The GPS location of the detected pothole was then sent to a cloud server, Ubidots. The GPS data is available for anyone who has access to the Ubidots account created for this project. The longitude and latitude values of each detected pothole is stored on Ubidots. The last detected pothole is illustrated as a pin on a map on Ubidots. This enables one to acquire a visual representation of where the pothole is situated. This visual representation is shown in Figure 4 below.

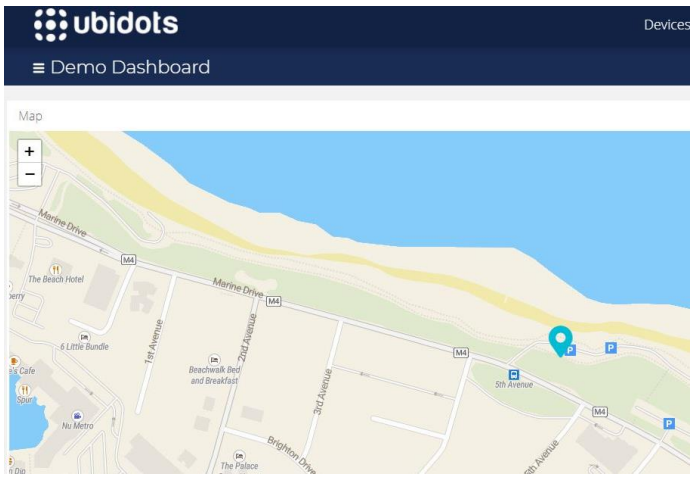

Figure 4: Screenshot of zoomed in map showing the pin of the last detected pothole on Ubidots

4. Discussion

The development and maintenance of roads in developing countries such as South Africa is of great significance. Potholes can result in an unpleasant and potentially dangerous ride. The tyres are likely to get damaged and would require re-alignment. It was thus necessary to create a system whereby driving over a pothole could be minimised. This research paper presented a comparative evaluation of state-of-the art CNN based object detection models to detect potholes on the road in real-time and from videos. Each of the networks were trained on a custom dataset and the performance was evaluated through the utilisation of various evaluation metrics. The best results for the application of this project was acquired through the YOLOv3 architecture, which worked the best for real-time applications because the inference speed was the fastest of the three evaluated architectures. The Faster R-CNN network proved the most accurate of the three models. SSD performed the worst in terms of accuracy, which could be attributed to the varying

sizes of the potholes. Once it was determined that YOLOv3 would be the most suitable architecture for this particular application, the location, i.e. the GPS co-ordinates of the detected pothole had to be determined. The next step was upload these GPS co-ordinates to a cloud server where these could be stored and later illustrated on a map.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER, grant number XXX" and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding. Any errors may affect your future funding.

Data Availability Statement: The dataset containing images of potholes can be obtained by contacting Kanushka Gajjar at kanushkagajjar0@gmail.com.

Acknowledgments: In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, R.; Orgunalp, U.; Hosking, B.; Liu, I. Pothole Detection Based on Disparity Transformation and Road Surface Modeling. *IEEE Transactions on Image Processing*. **2019**, 897-908.
2. First Drive: 2018 Mercedes-Benz-S-Class. Available online: https://www.automobilemag.com/news/first-drive-2018-mercedes-benz-s-class/ (accessed on 9 February 2019).
3. Namala, N.; Mallesh, S., Kumar, A. A Study on Potholes and its Effects on Vehicular Traffic. *IJCR*. **2018,** 1-2.
4. Nienaber, S.; Booysen, M.; Kroon, R. Detecting potholes using simple image processing techniques and real-world footage. 34th Southern African Transport Conference (SATC 2015), CSIR International Convention Centre, Pretoria, South Africa, 6-9 July 2015.
5. Nicholas, JA.; Herbert Chan, HW.; Baker MAB. Machine learning: Applications of artificial intelligence to imaging and diagnosis. *Biophys Rev*. **2019,** 11, 111-118.
6. Ukhwah, E.; Yuniarno, E.M.l Suprapto, Y.K. Asphalt Pavement Pothole Detection using Deep Learning method based on YOLO Neural Network. *10.1109/ISITIA.2019.8937176*, 35-40.
7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement.
8. Tsiakmaki, M.;Kostopoulos, G.; Kotsiantis, S.; Ragos, O. Transfer Learning from Deep Neural Networks for Predicting Student Performace. *Appl. Sci.* **2020,** 10, 2145.
9. Zhijian, H.; Fangmin, Li.; Xidao, L.; Zuowei, C. A Weekly Supervised Method for Mud Detection in Ores Based on Deep Active *Learning*. *Mathematical Problems in Engineering*. **2020**
10. Soundy, A.W.R.; Packhurst, B.J.; Brown, P.; Martin, A.; Molteno, T.C.A.; Schumayer, D. Comparison of Enhanced Noise Model Performance Based on Analysis of Civilian GPS Data. *Sensors* **2020,** 20, 6050.
11. Ferrandez-Pastor, F.J.; Garvia-Chamizo, J.M.; Nieto-Hidalgo, M.; Mora-Pascual, J.; Mora3-Martinez, J. Developing Ubiquitous Sensor Network Platform Using Internet of Things: Application in Precision Agriculture. *Sensors.* **2016,** 16, 1141
12. Xinliang, Z.; Wanru, W.; Yunji, Z.; Heng, X. An improved YOLOv3 model based on skipping connections and spatial pyramid pooling. *Systems Science and Control Engineering*. **2020**
13. Goutte, C.; Gaussier, E. A probabilistic interpretation of Precision, Recall and F1-Score, with implication for Evaluation. *Lecture Notes in Computer Science.* **2005,** 325-359.
14. Jingzhe Ma.; Shaobo, D.; Ye, Z.; Jing, W.; Zongmin, W.; Runzhi, Li.; Yongli, Li.; Liazhong, Z.; Huimin, M. Efficient Deep Learning Architecture for Detection and Recognition of Thyroid Nodules. *Computation Intelligence and Neuroscience.* **2020**, 1-15.
15. Drid, K.; Allaoui, M.; Kherfi, M.L. Object Detector Combination for Increasing Accuracy and Detecting More Overlapping Objects. *Lecture Notes in Computer Science*. **2020**
16. Li, C.; Luo, B.; Hong, H.; Su, X.; Wang, Y.; Li, J.; Wang, C.; Zhang, J.; Wei, L. Object Detection Based on Global-Local Saliency Constraint in Aerial Images. *Remote Sens*. **2020**, 12
17. Simhambhatla, R.; Okiah, K., Slater, R. Self-Driving Cars: Evaluation of Deep Learning Techniques for Object Detection in Different Driving Conditions. *SMU Data Science Review.* **2019,** 2.

 doi:10.20944/preprints202102.0569.v1

18. Zhang, Z.; Ai, X.; Chan, C.K.; Danhnoun, N. An efficient algorithm for pothole detection using stereo vision. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '14), Florence, Italy, 2014, 564-568.
19. Hosang, J.; Beneson, R.; Dollar, P.; Schiele, B. What makes for effective detection proposals?. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2016, 814-830.