A Comparative Evaluation of the Deep Learning Algorithms for Pothole Detection

Roopak Rastogi, Uttam Kumar, *Senior Member, IEEE*, Archit Kashyap, Shubham Jindal and Saurabh Pahwa
Spatial Computing Laboratory, Center for Data Sciences,
International Institute of Information Technology Bangalore (IIIT-B)
26/C, Electronics City Phase-1, Bangalore – 560100. India.
Email: roopak.rastogi@iiitb.org, uttam@iiitb.ac.in, archit.kashyap@iiitb.org,
shubhamj1996@gmail.com, saurabhphw@gmail.com

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342558

Abstract—**Potholes are a menace on roads and their presence compromises the safety of both drivers and pedestrians. In most developing countries, it is one of the major reasons for road accidents and loss of human life and property. Therefore, there is a need to consistently collect and update the data on the latest road conditions, so that the drivers can be advised for alternate routes and the concerned Government department can take immediate measures to fill up the potholes for the benefit of the commuters. A simple and efficient way to detect potholes on roads is through the application of object detection algorithms on images acquired from a smartphone camera. Therefore in this paper, we focus on evaluating the performance of state-of-the-art neural network algorithms such as YOLO and Faster R-CNN with VGG16 and ResNet-18 architectures for pothole detection that is both fast and accurate. Further, an improved YOLOv2 architecture is proposed to solve the class imbalance problem of "pothole" and "normal road" classes, and its performance is compared with other object detection techniques using precision, recall, intersection over union, and the number of frames processed per second (FPS). The results showed that the modified YOLOv2 architecture outperformed all the considered models with the lowest number of parameters (35 million) and the highest FPS (28), precision (0.87), and recall (0.89). This model can be deployed in autonomous vehicles for real-time geotagged pothole detection from photographs or video streams. The pothole detection application can also suggest potential alternate eco-friendly routes and guide the commuters in low light navigation.**

***Index Terms*—potholes, images, object detection, neural network, deep learning.**

I. Introduction

India has the second largest road network in the world with a total length of around 6 million kilometers. This network transports 65% of the merchandise and helps in commuting 90% of India's passenger traffic. Road transportation in India has attracted huge investments, thereby witnessing improvement in connectivity between cities, towns, and villages [1] and have a pivotal role in the success of the country's economy. Recently (in April 2020), the Indian Government budgeted US$213 billion (INR 15 lakh crores) for new road construction over the next two years. However, regular movement of both light and heavy motor vehicles, and subsequent delays in maintenance and improvement of the existing roadways gradually lead to wear and tear of the roads with the development of small, and medium to large size potholes. The presence of potholes on the roads not only increase the travel time and fuel consumption, thereby reducing the average speed of the traffic, but also lead to fatal accidents claiming lives and causing damages to vehicles. Monsoon season and excessive rainfall cause more distress to the commuters when potholes get filled with rainwater and sometimes sewage water in the absence of a proper drainage system. Therefore, along with the construction of new roads, timely maintenance of the existing roads are equally important.

A significant challenge towards the regular maintenance and repair of potholes is their timely and automatic detection. Pothole detection was classified into three categories namely, vibration, 3D reconstruction, and vision-based methods [2]. Among the vision-based methods, three visual specific characteristics of the pothole, namely, oval in shape, dark in color, and rougher surface than its surroundings were defined. These characteristics were used for shape extraction and image segmentation, and the extracted features were compared with the surroundings to determine a pothole [3]. Image processing techniques have also been used in detecting defects on the roads using texture, shape, and dimensions of the defective area [4], [5]. Further, pothole, crack, and patch area measurements were classified to identify the road distress [4].

Recently, deep convolutional neural networks (CNNs) have become popular for classification [6], object detection and recognition [7], [8], images captioning [9], image colorization [10], image denoising [11], super-resolution imaging [12], [13], etc. as they can automatically extract and fine-tune the relevant features without any interventions. CNNs are also gaining importance in semantic and instance segmentation [14]–[17]; semantic segmentation helps in understanding the content of the image at a pixel level whereas instance segmentation refers to the identification of each instance of every object in an image. State-of-the-art CNN architectures include two-stage detectors that have a proposal-driven mechanism and were popularized in the R-CNN framework [18]. While the first stage is used to generate a sparse set of candidate object locations, the second stage classifies that location as

978-1-7281-6916-3/20/$31.00 ©2020 IEEE

either foreground or background class. On the challenging COCO benchmark [19], this two-stage framework consistently achieved the highest accuracy. Generally, real-time applications require faster detectors like YOLO (You Only Look Once) [20], [21] and SSD (Single Shot Detector) [22], [23] that render higher accuracy than a threshold (which is user-defined) relative to the two-stage detectors. Use of CNN based models [24] have shown significant improvements over the previous models. A few studies have attempted to use deep learning architectures for object detection, for example, YOLO framework [25]–[28], and for the image, segmentation using a multi-step deep learning model [29] with promising results. However, some of these studies exist in isolation and their performances are limited by the underlying extracted features (depending on the model selected), which are more often than not carried out manually. So far, there are no studies reporting their comparison with other state-of-the-art methods that use the selective search algorithm for reason proposals such as R-CNN (Region-based Convolutional Neural Networks) with different backbone architectures. Moreover, real-time object detection such as detecting potholes on varying roads (streets, lanes, avenues, narrow and wide roads, highways, roads made up of gravel, cement, and bricks, etc.) and changing weather conditions (dry and monsoon season, cloudy and foggy winters, etc.) seek robust and time-efficient algorithms. Therefore, the following two objectives were formulated for this study:

- To compare the existing state-of-the-art deep learning architectures for pothole detection.
- To develop a modified YOLOv2 architecture to deal with class imbalance problem.

Firstly, YOLOv2 base model is implemented on two image resolutions of 416 x 416 and 608 x 608 for detecting the potholes. Secondly, Faster R-CNN with a backbone of VGG16 and ResNet-18 (Residual Neural Network) were trained and tested on the pothole dataset. Finally, a modified YOLOv2 architecture is presented. Performance evaluation is carried out using precision, recall, number of frames processed per second (FPS), and IoU (Intersection over Union). The paper is organized as follows: Section 2 reviews state-of-the-art deep learning algorithms for object detection, followed by description of the dataset and methodology in section 3. Section 4 discusses the results with concluding remarks in section 5.

II. Review of Deep Learning Algorithms for Object Detection

Object detection is a combination of object localization and object classification. In object localization, the goal is to locate an object in the scene through bounding boxes, while object classification classifies a given image into classes. Thus, object detection includes both locating and classifying an object. In this section, a review of two object detection algorithms viz. YOLO and Faster R-CNN are presented. Finally, we propose a modified YOLO algorithm to resolve the class imbalance problem between “pothole” and “normal” classes.

A. YOLO

YOLO (You Only Look Once) object detection algorithm proposed by Redmon et al. [20] requires only one single forward propagation through the neural network to make predictions faster. Non-maximum suppression ensures the detection and recognition of each object once with the bounding boxes. Unlike other traditional classifiers that learn on local regions of the images to develop models, YOLO works on the whole image, thereby making predictions on the global context. It divides the image in a grid of S x S cells where each grid cell predicts B bounding boxes using the information on the center of the object (x, y), dimensions of the object (w, h), conditional class probabilities C, and the corresponding values of confidence score. The confidence score is the likelihood of the bounding box containing the object (also called objectness) and the accuracy of the boundary box. The class confidence score is a function of box confidence score and conditional class probability. The final target prediction is denoted as S x S x (B x 5 + C) tensor which can be used to measure the classification as well as the localization. The class confidence score is calculated as:

Class confidence score = box confidence score x conditional class probability

YOLO has 24 convolutional layers followed by 2 fully connected layers. Reduction layers are often used to reduce the depth of feature maps, resulting in 2 boundary box predictions per location. It uses sum-squared error (between ground truth and predictions) to compute the localization, confidence, and classification losses. For true positive class, the bounding boxes with the highest IoU with the ground truth are selected. YOLOv2 [21] overcomes the problems of localization errors and low recall values that were observed in YOLO. To increase recall, convolutional anchor (also called prior) boxes are used in YOLOv2. Further, batch normalization, high-resolution classifier, k-scale customized dimension clusters, and direct location prediction increase its performance significantly.

B. Faster RCNN (Faster Region-based Convolutional Neural Networks)

In the R-CNN family of algorithms (R-CNN [18], Fast R-CNN [29] and Faster R-CNN [30]), the network is usually made up of these components/steps:

- *Location generation of objects:* A region proposal algorithm (such as selective search [31]) generates possible location of the objects in the image through bounding boxes.
- *Feature generation:* A stage for generating features of the objects using CNNs.
- *Classification:* A layer which classifies the objects.
- *Regression:* A layer for fine-tuning the coordinates of the bounding boxes.

In R-CNN, the input to CNN is a region at pixel level while in Fast R-CNN, the input is a feature map. On the other hand, the similarity between R-CNN and Fast R-CNN is that the region proposal algorithm and CNN are decoupled in both, which affects the accuracy and detection time. This is because when the region proposal algorithm has false negatives, the performance of CNN is vulnerable, therefore, coupling actually improves the speed and accuracy. R-CNN takes a longer time for training, making it unsuitable for real-time applications. Moreover, the selective search algorithm is fixed, therefore no learning happens at this stage, leading to the generation of bad candidate region proposals. On the contrary, Fast R-CNN is faster than R-CNN because the convolution operation is done only once per image to generate a feature map. However, including region proposals in Fast R-CNN slows down the algorithm affecting its performance.

Since selective search is slow and time-consuming, Faster R-CNN was proposed where instead of using a selective search algorithm on the feature map, a separate network (such as CNN) is used to predict the region proposals. The predicted region proposals are then reshaped using an ROI (region of interest) pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes. Therefore, Faster R-CNN is much faster for real-time object detection. The CNN used here could be either InceptionNet [32], ResNet [33], VGGNet [34] or any other network capable of image classification.

C. Modified YOLOv2

YOLOv2 has shown great success for multiclass object detection. However, for one class detection problem (like pothole detection where class imbalance exists), deep neural network architectures are slightly overrated. Therefore, the original YOLOv2 architecture was modified by removing few layers and introducing some residual connections to mitigate the vanishing gradients and to unlearn the insignificant features learned during training, with the added weighted classification loss (as shown in Fig. 1). Classification loss forces the previous layers to tune the learned features correctly with "pothole" and "normal road" images and increase the accuracy along with the detection speed.

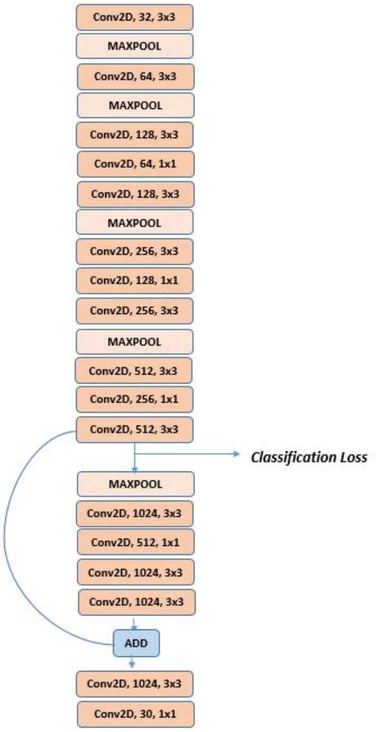

Fig. 1. Architecture of the proposed modified YOLOv2 network which has around 35MN parameters after removing few layers, introducing the residual connections and additional weighted classification loss.

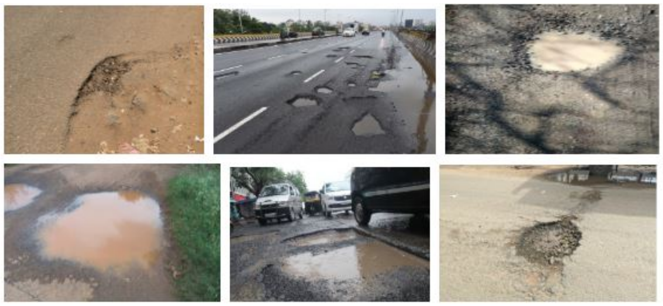

Fig. 2. Images showing potholes in different lightning, weather and traffic conditions.

III. DATA AND METHODOLOGY

A. DATA

Images and videos of "pothole" and "normal road" in RGB wavelength were acquired with smartphones in multiple field visits in the month of October and November, 2019. Separate instances of data were collected during sunny, dry, cloudy, and rainy days in Electronics City Phase 1, Bangalore City, India (shown in Fig. 2). Additional attribute information about the potholes such as pothole size (small, medium, and large as decided through visual inspection), dry/filled with rainwater, depth of potholes, location information, etc. was also recorded for each observation. To avoid bias towards "only pothole images" (as all roads do not have potholes), negative samples i.e. images with normal roads (roads without potholes) were also included. Further, roads with no traffic, low, medium, and heavy traffic with pedestrians were also considered during data collection. Other sources of data included crowdsourcing and publicly available data on the web representing varying conditions from different cities across the country. The final dataset had 1300 images with 1334 potholes in total, with nearly identical number of images representing dry potholes and potholes clogged with water. Training datasets were generated signifying varying road conditions and scenarios from multiple spatial locations.

B. METHODOLOGY

This section discusses the overall methodology adopted in this study.

1) Data preprocessing, training and test data generation: The final set of images in the dataset were manually selected after careful inspection. Images with glare, low contrast, blurred vision, and pothole shadowed by large vehicles were

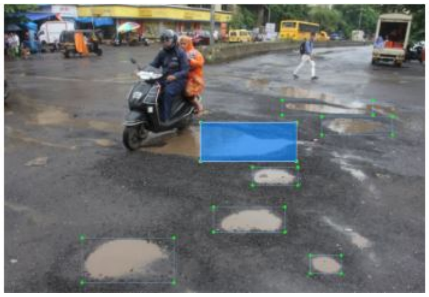

Fig. 3. Image showing the pothole annotated with the bounding boxes. The box coordinates are further used as ground truth to train the object detection models.

omitted from the database. For the training and test data generation, potholes in the images were annotated with bounding boxes to treat them as objects using LabelImg tool [35] as shown in Fig. 3, resulting in a total of 1334 annotated boxes. The training and test data contained 1150 and 150 images respectively, with a uniform mixture of single, multiple, and negative samples (that is roads with no potholes). All the models were trained on RTX 2080Ti with 11GB memory on Google Colab. YOLOv2 and Faster R-CNN were implemented in Keras [36], an open-source neural network library in Python.

2) Pothole detection using YOLO: YOLOv2 architecture with two image resolutions of 416 x 416 and 608 x 608 were considered for detecting the potholes on the test dataset. 416 is the minimum grid cell size that can be tested in this architecture, and 608 was the maximum high-resolution grid that the architecture could implement on the available computing platform.

3) Pothole detection using Faster R-CNN: Faster R-CNN was implemented with a backbone of VGG16 (also called OxfordNet) and ResNet-18 (residual neural network). VGG16 consists of 16 layer CNN for which the input image size was 224 x 224 that is suitable for object localization and classification. In localization, the class score was replaced by various candidates for the bounding box represented by a 4-D vector. The loss function was altered from classification to a localization loss (such as a mean squared error). In ResNet-18, residual connections were introduced to handle the vanishing gradient problem so that the model converges faster (more description is presented in the discussion section). Fundamentally, the training procedure for Faster R-CNN differs from Fast R-CNN since the convolutional layers can be shared. The major steps in training Faster R-CNN are: (a) The RPN (region proposal network) and detection network were trained using the pre-trained model on ImageNet. (b) The fixed layers were shared, and unique layers of RPN were fine-tuned using the detection network. (c) Finally, the unique layers of the detection network were fine-tuned, while the shared layers were fixed.

4) Pothole detection using Modified YOLOv2: The proposed modified YOLOv2 architecture was trained and tested on the pothole dataset where the layers could tune the learned features about the pothole and normal classes.

5) Accuracy assessment: Precision, recall, number of frames processed per second (FPS), and IoU (Intersection over Union) were used as validation metrics for evaluating the detection results. IoU represents the overlap area of a ground truth bounding box and the detected bounding box represented as A and B respectively, over the union of both the bounding box areas.

$$IOU = |A \cap B| / |A \cup B| \tag{1}$$

It shows the similarity between detection and ground truth. Precision and recall are given by equations (2) and (3).

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

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

Here, TP (true positive) occurs when a pothole is correctly detected as a pothole by the model. FP (false positive) is the false detection of a pothole by the model when it is actually not a pothole. FN (false negative) refers to a situation when the pothole is not detected by a model while it actually exists. Therefore, the detection of potholes on negative samples is FP, and non-detection of potholes on positive samples (pothole images) is FN.

IV. RESULTS AND DISCUSSION

Fig. 4 shows sample images of dry and wet potholes with bounding boxes detected from various algorithms. Table I highlights the number of parameters, FPS processed by the models, precision, and recall for the different architectures.

Table I indicates that the proposed modified YOLOv2 architecture has outperformed all the models with the lowest number of parameters (35 million) and the highest FPS (28), precision (0.87), and recall (0.89) (highlighted in bold font) followed by YOLOv2. In the modified YOLOv2 model, the induction of classification loss increased the accuracy and decreased the execution time for pothole detection. On the other hand, YOLOv2 base model was trained on different image dimensions as mentioned earlier. Here, the image size of 416 x 416 resulted in 13 x 13 grids (as there are five max-pooling layers in YOLOv2 and each layer downsamples the image by a factor of 2), where each grid contained five anchors and each anchor contained the bounding box coordinates along with the probability scores.

Faster R-CNN produced a high number of parameters with the lowest FPS, precision, and recall. In Faster R-CNN implementation, RPNs were used for locating the potholes through the bounding boxes that were passed through the detection network. Non-maximum suppression was applied to account for the highly correlated proposed areas which significantly reduced the proposed regions, although we did not conduct any controlled experiments to assess the magnitude of this reduction. Very deep neural networks like VGGNet [34] that contains 19 layers are hard to train due

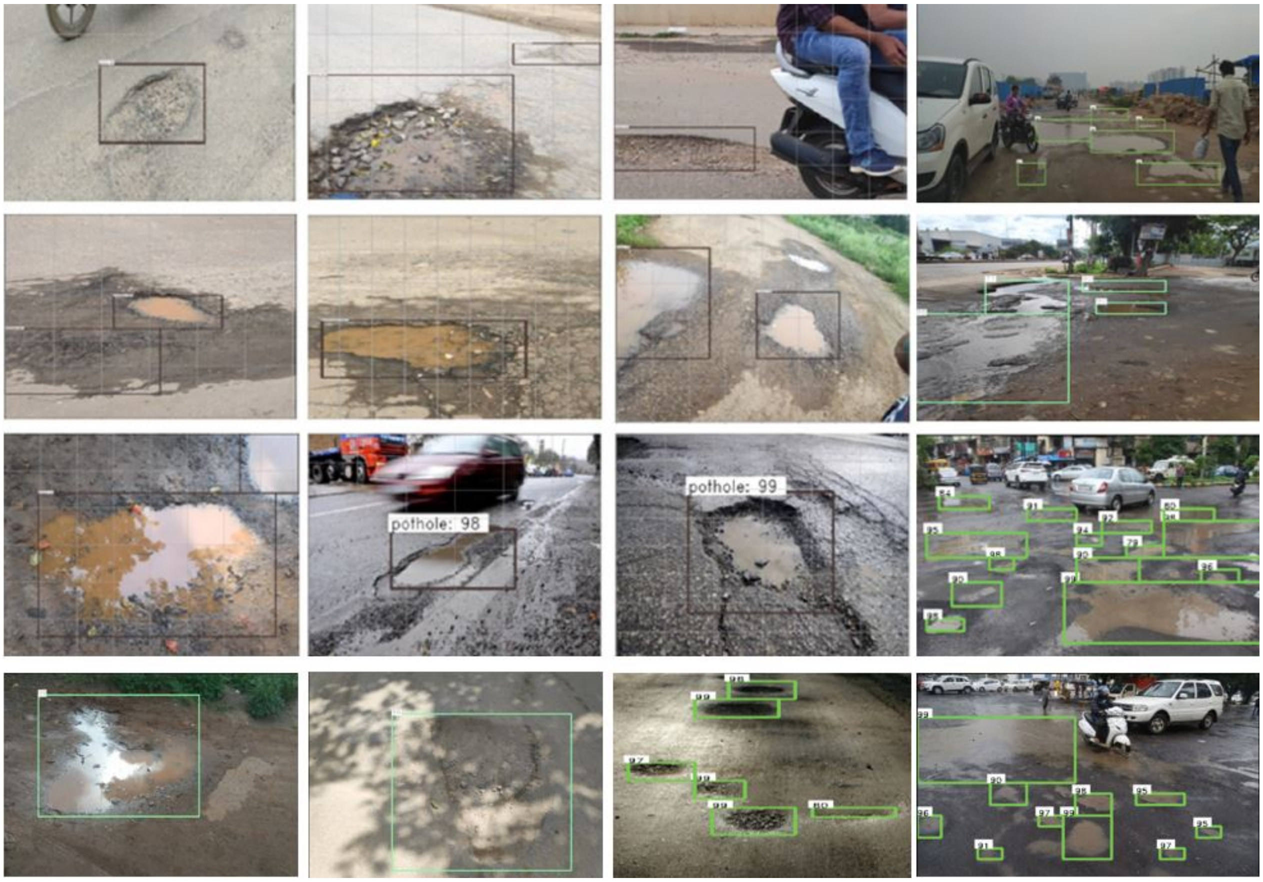

Fig. 4. Sample dry and wet potholes under varying light and traffic conditions with bounding boxes detected from various architectures. Images in the last column show the model predictions on multiple potholes.

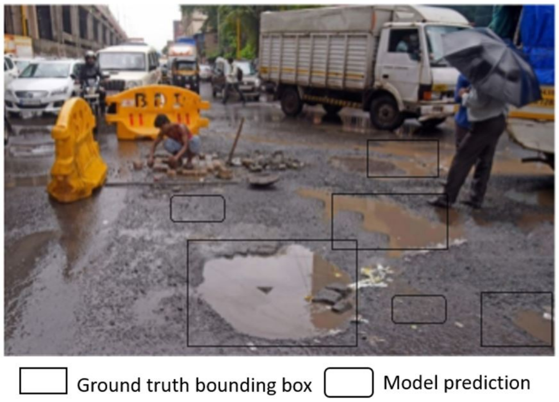

Fig. 5. Example of image with multiple potholes and model prediction that does not overlap with the ground truth.

TABLE I
PERFORMANCE EVALUATION OF THE VARIOUS POTHOLE DETECTION ARCHITECTURES

Models	*Parameters*	*FPS*	*Precision*	*Recall*
YOLOv2 (416 x 416)	50MN	20	0.83	0.85
YOLOv2 (608x608)	50MN	12	0.85	0.86
Modified YOLOv2 (608x608)	**35MN**	**28**	**0.87**	**0.89**
Faster RCNN (VGG16)	136MN	7	0.79	0.80
Faster RCNN (ResNet18)	133MN	5	0.81	0.83

to the vanishing gradient problem (as the gradient is back-propagated to the layers, repeated multiplications make the gradient infinitesimally small). This saturates the performance of a network or even degrades it when the number of layers are increased. Therefore, in ResNet-18 [33] we introduced the residual connections to handle the vanishing gradient problem, although it did not perform well with the pothole dataset. For images with multiple potholes, there were two possible cases: (1) If the model detected a pothole without any IoU with the ground truth bounding box, then it was considered false positive, and false negative for the actual potholes present in the image. (2) If the model detected a pothole in the image having IoU with the ground truth bounding box, then it was considered true positive as depicted in Fig. 5.

Future work will focus on optimizing the parameter settings of the various models to increase the detection accuracy of very small potholes while reducing the execution time.

V. Conclusion

Detecting potholes on the road is important to safeguard human life and minimize vehicle damage. The pothole detection problem present more challenges with varying pothole size, diverse road construction materials used, different traffic conditions, and changing weather scenarios. In this work, 1300 pothole images were collected to test various object detection models. A modified YOLOv2 was proposed to unravel the class imbalance problem of "pothole" and "normal road" classes, and it outperformed the YOLOv2 base model and Faster R-CNN architectures. Pertaining to its high accuracy, the proposed architecture can be integrated with a smartphone camera using raspberry pi and can be mounted on the dashboard of manual and autonomous vehicles for real-time pothole detection. This will help the drivers and commuters with advisories for safe driving.

Acknowledgement

We are grateful to the International Institute of Information Technology Bangalore (IIIT-B), India for the infrastructure support. We are thankful to Infosys Foundation for the financial assistance and project grant through the Infosys Foundation Career Development Chair Professor.

References

[1] https://www.ibef.org/industry/roads-india.aspx [Last accessed: 15 August, 2020. 09:00 am.]

[2] K. Taehyeong, and S. Ryu, "Review and analysis of Pothole detection methods," J. of Emerg. Trends in Computing and Info. Sci., vol. 5(8), pp. 603-608, 2014.

[3] K. Christian, and I. Brilakis, "Pothole detection in asphalt pavement images," Adv. Engg. Info., vol. 25(3), pp. 507-515, 2011.

[4] H. Lokeshwor, L.K. Das, and S.K. Sud, "Method for automated assessment of Potholes, Cracks and Patches from road surface video clips," Procedia-Social and Behavioral Sci., vol. 104(2013), pp. 312-321, 2013.

[5] R. Karthika, and L. Parameswaran, "An automated vision-based algorithm for out of context detection in images," Int. J. Signal and Imaging Sys. Engg., vol. 11(1), pp. 1-8, 2018.

[6] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proc. NIPS, pp. 1097–1105, 2012.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. CVPR, pp. 580–587, 2014.

[8] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C.L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in Proc. ICCV, 2015.

[9] O.M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in Proc. British Machine Vision Conference, 2015.

[10] J.Y. Zhu, P. Krahenbuhl, E. Shechtman, and A.A. Efros, "Learning a discriminative model for the perception of realism in composite images," in Proc. ICCV, pp. 3943–3951, 2015.

[11] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in Proc. NIPS, pp.341–349, 2012.

[12] C. Dong, C.C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in Proc. ECCV, Springer, pp. 184–199, 2014.

[13] J. Justin, A. Alexandre, and F.F. Li, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," vol. 9906, pp. 694-711, 2016. 10.1007/978-3-319-46475-6_43.

[14] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester 2014 (2014).

[15] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute 2 image: Conditional image generation from visual attributes," arXiv preprint arXiv:1512.00570 (2015).

[16] T.D. Kulkarni, W.F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in Proc. NIPS, pp.2530–2538, 2015.

[17] A. Dosovitskiy, J.T. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in Proc. CVPR, pp. 1538–1546, 2015.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proc. IEEE CVPR, Columbus, OH, pp. 580-587, 2014, doi: 10.1109/CVPR.2014.81.

[19] L. Tsung-Yi, M. Michael, B. Serge, H. James, P. Pietro, R. Deva, D. Piotr, and C. Zitnick, Microsoft COCO: Common Objects in Context, 2014.

[20] R. Joseph, D. Santosh, G. Ross, and F. Ali, "You Only Look Once: Unified, Real-Time Object Detection," pp. 779-788, 2016, 10.1109/CVPR.2016.91.

[21] J. Redmon, and A. Farhadi, "YOLO9000: Better, Faster, Stronger," pp. 6517-6525, 2017, 10.1109/CVPR.2017.690.

[22] L. Wei, A. Dragomir, E. Dumitru, S. Christian, R. Scott, Fu. Cheng-Yang, and B. Alexander, "SSD: Single Shot MultiBox Detector," vol. 9905, pp. 21-37, 2014, 10.1007/978-3-319-46448-0_2.

[23] F. Cheng-Yang, L. Wei, R. Ananth, T. Ambrish, and B. Alexander, "DSSD : Deconvolutional Single Shot Detector," 2014.

[24] V. Srivatsan, J. Sobhagya, S. Karan, W. Lars, and M. Christoph, "Vision for road inspection," in Proc. 2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014, pp. 115-122, 2014. 10.1109/WACV.2014.6836111.

[25] E.N. Ukhwah, E.M. Yuniarno, and Y.K. Suprapto, "Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network," in Proc. 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, pp. 35-40, 2019, doi: 10.1109/ISITIA.2019.8937176.

[26] D.J, S.D.V, A.S.A, K.R., and L. Parameswaran, "Deep Learning based Detection of potholes in Indian roads using YOLO," in Proc. 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, pp. 381-385, 2020, doi: 10.1109/ICICT48043.2020.9112424.

[27] L.K. Suong, and K. Jangwoo, "Detection of potholes using a deep convolutional neural network," J. Universal Computer Sci., vol. 24, pp. 1244-1257.

[28] Y. Sudhir, V. Girish, and C.V. Jawahar, "City-Scale Road Audit System using Deep Learning," pp. 635-640, 2019, 10.1109/IROS.2018.8594363.

[29] R. Girshick, "Fast R-CNN," in Proc. 2015 IEEE ICCV, Santiago, pp. 1440-1448, 2015, doi: 10.1109/ICCV.2015.169.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39(6), pp. 1137–1149, June 2017. DOI:https://doi.org/10.1109/TPAMI.2016.2577031.

[31] S. Koen, U. Jasper, T. Gevers, and S. Arnold, "Segmentation as selective search for object recognition," Pediatric Critical Care Medicine - PEDIATR CRIT CARE MED, pp. 1879-1886, 2011, 10.1109/ICCV.2011.6126456.

[32] S. Christian, L. Wei, J. Yangqing, S. Pierre, R. Scott, A. Dragomir, E. Dumitru, V. Vincent, and R. Andrew, "Going deeper with convolutions," in Proc. IEEE CVPR, pp. 1-9, 2015, 10.1109/CVPR.2015.7298594.

[33] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition,", pp. 770-778, 2016, 10.1109/CVPR.2016.90.

[34] S. Karen, and Z. Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, arXiv 1409.1556.

[35] Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg [Last accessed: 15 August, 2020. 09:05 am.]

[36] Huynh Ngoc Anh. https://github.com/experiencor/keras-yolo2. [Last accessed: 15 August, 2020. 09:05 am.]