Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020)
IEEE Xplore Part Number:CFP20F70-ART; ISBN:978-1-7281-4685-0

Deep Learning based Detection of potholes in Indian roads using YOLO

Dharneeshkar J
Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India-641112
dharneeshkar@gmail.com

Soban Dhakshana V
Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India-641112
sobandhakshana.v@gmail.com

Aniruthan S A
Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India-641112
saaniruth11@gmail.com

Karthika R
Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India-641112
r_karthika@cb.amrita.edu

Latha Parameswaran
Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India-641112
p_latha@cb.amrita.edu

***Abstract*— In countries like India road maintenance is a challenging task. Year after year, the accident rates are increasing due to the up-surging potholes count. As the road maintenance process is done manually in most places, it consumes enormous time, requires human labor and subjected to human errors. Thus, there is a growing need for a cost-effective automated identification of potholes. In recent trends, many approaches proved good results in applying deep learning [1] for different object detection. Convolutional Neural Networks (CNNs) have the ability to learn the art of extracting relevant features from an Image. But in countries like India, there is no potholes dataset available to train the CNN. In this paper, a new 1500 image dataset has been created on Indian roads. The dataset is annotated and trained using YOLO (You Only Look Once). The new dataset is trained on YOLOv3, YOLOv2, YOLOv3-tiny, and the results are compared. The results are evaluated based on the mAP, precision and recall. The model is tested on different pothole images and it detects with a reasonable accuracy.**

Keywords— ADAS, CNN, Deep Learning, Potholes Detection, R-CNN, Vision based approach, YOLOv2, YOLOv3

I. INTRODUCTION

One of the growing fields in Science and Technology is the Advanced Driver Assistance Systems (ADAS) [2], [3]. Over the years transportation has gained paramount importance in daily life. Currently, the world is experiencing an exponential advancement in intelligent transportation systems. The ADAS has automated a plethora of features in vehicles leading to a smart and sophisticated driving experience. When the past present future of ADAS is examined, the word "safety" is always given prime importance. In essence, ADAS's main job is to reduce the fatalities caused by road due to human errors.

While speaking about road safety, potholes play a crucial role. In India, over a period of three years (2015-2017), there have been over 9300 deaths and 25000 injuries exclusively only due to potholes not taking into consideration any other object that caused accidents [4]. When driving for prolonged hours or under stress or tension, the driver tends to miss out concentration. One of the primary reasons behind such accidents is the driver's inability to pay attention to every single detail on the road and that's when the ADAS comes into the picture. The data can either be directly reported to the driver by giving an alert symbol in the cabin of the vehicle or the data can be used by the autonomous driverless system where the system decides on what action it needs to mete out in order to prevent a collision and ensure a safe and comfortable riding experience for the passengers.

Potholes detection can be broadly classified into three categories namely vibration-based methods, 3D reconstruction-based methods and vision-based methods (2D image) [5]. Out of the three vision-based methods prove to be the most cost-effective ways of detection. At the same time, accurate detection of potholes using a 2D image is very tough. So, there is a need for a system which can detect Potholes from a 2D image with better accuracy and at greater speed. Especially countries like India need more focus as the potholes count is huge. Also, the system will have technical differences when implemented in different places depending on the degree of road maintenance.

Detection of potholes is much difficult when compared to other objects such as a pedestrian, vehicles, traffic signs, etc. because the former has a wide range of geometrics. When

comparing the recognition algorithms, Convolutional Neural Networks (CNN) has proven to be one of the best [6]. In this paper, potholes detection is implemented using one of the CNN family's unique representatives You Only Look Once (YOLO) to a newly created dataset for Indian roads.

II. RELATED WORKS

There are so many architectures and methodologies that can be used for potholes detection. Evaluating a pothole in most cases requires 3D equipment which is very expensive and not feasible on a large scale [7]. Instead, detecting a defect in the road using an image and then further processing it into defective and not-defective regions and analyzing the patterns in both the regions can help differentiate a pothole from the non-defective region which is essentially the road. Another method of analyzing these problems is by using three features of an image which are, the texture, shape, and dimensions of the defective area in the frame [8], [9]. But the method doesn't have machine learning at the core of it. Therefore, one of the most basic ways to carry out potholes detection is by using Convolutional Neural Networks (CNNs).

The Object Detection field started off with Region-Based Convolutional Neural Networks (R-CNN) and moved to faster and more advanced algorithms like YOLO (You Only Look Once) and SSD (Single-Shot Detection). The R-CNNs use selective search algorithms to extract 2000 bounding boxes from the input image in the first step itself to stick with processing only the most important features in an input based on color, pattern, shape, and size [10]. The R-CNNs use a three-stage mechanism: feature extraction via Selective Search Algorithm, SVM classification and Regression modelling for tight bounding boxes. The Fast R-CNN model uses a single stage mechanism where it directly passes the input to a CNN and the output from this CNN is the Regions of Interest (RoI) [11]. Then an RoI pooling layer is applied to the output from the CNN to warp the images to the size the Network is accustomed to work with. These RoIs are given to a fully connected Neural Network (NN) that segregates them and returns bounding boxes on the RoIs using linear regression and softmax networks working in a parallel manner. This provides drastic speed gains as well as savings in terms of the size of the model. Although Fast R-CNN uses a one-stage mechanism, it relies on the selective search method initially and this consumes time. The Faster R-CNN passes the input image to a CNN which returns the feature map of the input and these feature maps are passed to Region Proposal Networks which gives object proposals along with a score of how much of an object that particular prediction resembles [12] [13]. This is passed to an RoI pooling layer which warps all the proposed Regions to the same size to make them compatible with further hidden networks that will be used in the process. The final layer is similar to that of the R-CNN which is a fusion of both the softmax and Linear Regression Layer which classifies and draws bounding boxes for the various objects available in the image. Even though the Faster R-CNN has improved speed, it has a fundamental drawback as the other two R-CNNs, it doesn't look at the complete image at once, instead they focus on parts of the image sequentially and this requires many complex stages. To work around this issue, the SSD and YOLO methods were created.

The framework of YOLO yields drastic speed gains. It processes up to 45 fps, which is a massive improvement compared to Region-based CNNs. To view the real-world events in real-time smoothly, 30 fps is a sufficient speed, but with 45 fps YOLO favors real-time Object Detection.

SSD [14] is similar to YOLO but is more lenient in terms of aspect ratio compatibility. It works with 6 more additional aspect ratios than YOLO. This enables tighter object wrapping than YOLO. It also has higher accuracy than YOLO as it has a greater number of hidden layers. This added advantage of better accuracy comes with a trade-off, that is speed. Since cars move at very high speeds, potholes detection at such high speeds calls for high-speed detection and one of the highest speeds can be obtained by using YOLO as per the explanations available and illustrated above.

III. METHODOLOGY

A. YOLO

The idea behind YOLO [15] was to detect and classify the objects in an image at a single glance. YOLO is suitable for real-time applications and provides good accuracy. In this method both, the multiple bounding boxes and the class probabilities of each bounding box are calculated simultaneously.

The first version of YOLO ends up with a lot of localization errors and has very low recall when put against its predecessors and YOLOv2 [16] was designed with that in mind and has been improved drastically on those fronts. Applying Batch Normalization improved the mAP by 2%, using the High-Resolution Classifier improved it further by ~4%, using convolutional anchor boxes improved the recall values, using k-scale customized Dimension Clusters boosted the performance by 1% and by using Direct location prediction YOLOv2 has improved by 5% over the Original YOLO version. The performance gains of YOLOv2 are very significant when compared to the original version.

YOLOv3 [17] predicts how much a bounding box resembles an object based on logistic regression. Logistic classifiers are used since the softmax Layer didn't prove to be of much use for boosting performance. The performance of YOLOv3 on small objects has improved by several folds but the performance isn't as strong and promising as the results from YOLOv2 when it comes to large and medium-sized objects. In yolov3, there are fifty-three (53) convolution layers used. Three different scales are used for predicting bounding boxes. For every bounding box, an objectness score is calculated. The class of objects in the bounding boxes are calculated using multilabel classification. The final layer produces a 3d tensor with the bounding boxes, objectness and the class prediction encoded in it. Fig. 1 shows the flow of the pothole detection.

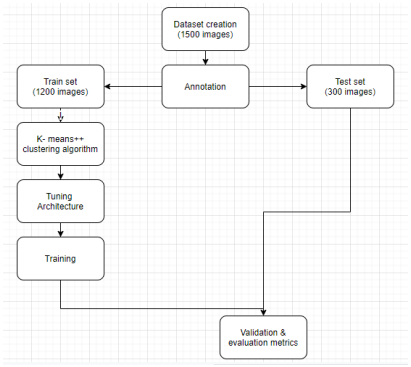

Fig. 1. Flow diagram for pothole detection.

B. DATASET

Indian potholes are distinct and stand out from other country's potholes. So, there is a need for the creation of a new dataset that can clearly represent the current Indian road conditions. We traveled places namely, Coimbatore, Idukki, Kumily, etc. and generated an entirely new 1500 image dataset. The images were taken from the dashboard of a car with an iPhone 7 camera. The images were captured with an assortment of different angles, distances and lightening conditions. Furthermore, the images were taken in a mixture of climatic conditions (Rainy, clear, overcast, etc.). The quality of the images is taken care of as it plays a major role in the deep learning process. The images were resized to 1024*768. Some images from the dataset are shown in Fig. 2.

Fig. 2. Some images from the newly created potholes dataset

After the dataset collection, the subsequent step is annotation. Annotation can be performed using a lot of free open source tools such as LabelImg, BBox, Yolo_mark, etc. As the chosen method is YOLO, the annotation should be done in YOLO format i.e. class-id, center_X, center_Y, width, height. Hence, LabelImg annotation tool is chosen as it can directly label images in YOLO format. Fig. 3 shows a sample annotation of an image and its corresponding output file. As the focus is only on potholes detection, the number of class is one and the class-id is fixed as zero. The dataset is randomly split into train and test sets. The Train set contains 80% of the images and the test set contains the remaining 20%.

Fig. 3. Sample annotation of an image using LabelImg tool

C. EXPERIMENT

The experiment was carried out on a GeForce GTX 1060 with MAX-Q design and the operating system was Ubuntu 18.04. To increase the computation speed and efficient use of graphics the system is installed with CUDA. CUDA is used when there is a need for GPU computation. It ensures a super-fast neural network but it supports only Nvidia GPU. By using CUDA the training time is drastically reduced. AlexeyAB's Darknet neural network has been used for the pothole detection analysis. The AlexeyAB's darknet package comes handy as it contains many direct command-line commands related to the object detection. Also evaluating the results is much easier. Fig. 4 shows K-means++ clustering algorithm implemented on the pothole training dataset. The algorithm is used to fix the optimal number of anchor boxes. It is an unsupervised algorithm that clusters the data based on the specified K-clusters. Initially, average IOU's are generated for each value of K (from 1 to 9). The resultant data is plotted in a graph and the elbow point is found. Now, based on the elbow point's K value the anchors are generated. Using the new anchors has improved the mAP percentage.

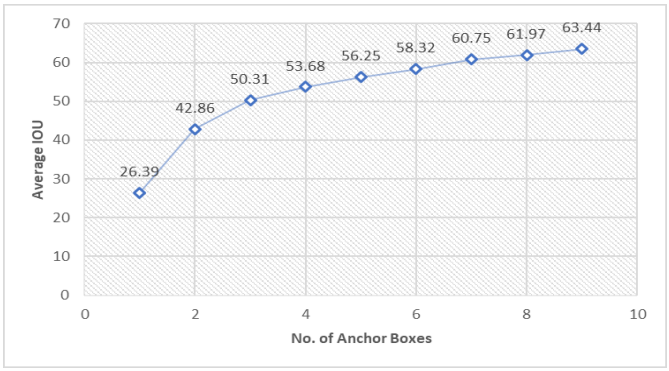

Fig. 4. Average IOU vs number of anchor boxes

Our experiment is carried out on neural network architectures such as yolov3, yolov2, and tiny-yolov3. Each architecture has its own trade-off. Tuning the architecture model is cardinal in getting the required mean average precision. For training, Batch size is fixed as 64 and subdivisions as 16. The batch and subdivision sizes can be changed based on the system's graphic capabilities. For yolov3, the filter size is given as 18 and whereas 30 in case of yolov2. In addition, the newly generated anchors are updated to the architecture. Also, the max_batches are reduced to 5200 as we have only one class and its corresponding steps been given. For yolov3, darknet53 is the pre-trained weights for the convolutional layers and for yolov2, darknet19 is used. After

implementing the above changes, training is initialized and for every 1000 iterations, the weight file is saved.

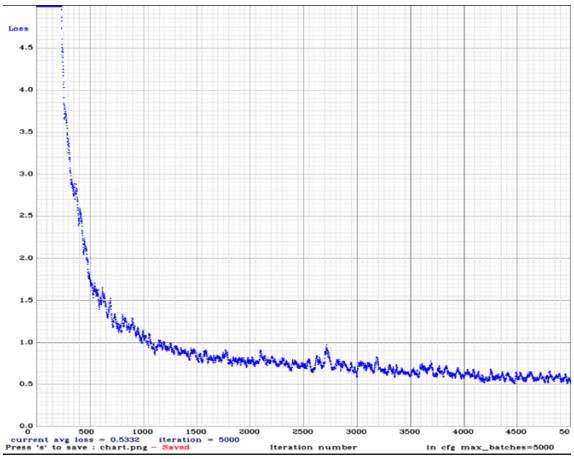

Fig. 5. Average loss vs number of iterations

Fig. 5 demonstrates the decaying of the average loss when trained with yolov2 architecture. From the figure it's evident that the average loss is optimized to a value of 0.5 and saturates. Training is stopped when the average loss no longer decreases. The weight file with the lowest average loss is chosen for the evaluation.

IV. RESULTS

TABLE I. COMPARISON OF DIFFERENT YOLO ARCHITECTURES

Model	**mAP@ 0.25**	**mAP@ 0.50**	**IoU**	**Precision**	**Recall**
Yolov3(416* 416)	58.79	38.41	46.29	0.69	0.35
Yolov2(416* 416)	64.05	39.46	31.6	0.46	0.45
Yolov2(608* 608)	68.57	45.33	34.77	0.52	0.5
Tiny Yolov3(608* 608)	72.12	49.71	50.96	0.76	0.4

The results obtained by training our new 1500 image dataset on different architectures is shown in Table 1. The confidence threshold is kept as 0.25. Also, the mean average precision is calculated at an Intersection over Union (IoU) threshold of 0.50. The Intersection over Union is the measure of the overlap between our prediction and the ground truth. In evaluation, the results mAP is one of the crucial metrics. The average precision is calculated by finding the area under the precision- recall curve, where Precision=TP/(TP+FP) and Recall=TP/(TP+FN). Precision is nothing but the ability of our model to identify only the relevant objects. Recall says about the percentage of finding out all the positives. Thus, when comparing the results, the highest mAP is obtained in tiny yolov3_ (608*608). Fig. 6 shows a sample detection results.

Fig. 6. Some sample detection results

V. CONCLUSION

Potholes detection is unique when compared to other object detections such as person, car, airplane and so on. Unlike other objects, potholes don't have a fixed shape. This makes it challenging for detection. Increasing the mean average precision for potholes detection is difficult due to the above-mentioned limitation.

In this paper, the newly created 1500 image dataset is trained using various versions of YOLO. Furthermore, the mean average precision is improved by making suitable architectural changes. In future, the system can be implemented real-time in a vehicle's dashboard using a raspberry pi with a camera. Also, the system can be embedded with a GPS [18] to track the location of the potholes detected which can be a great use to the road maintenance team.

VI. REFERENCES

[1] Geronimo, David, et al. "Survey of pedestrian detection for advanced driver assistance systems." IEEE transactions on pattern analysis and machine intelligence 32.7 (2009): 1239-1258

[2] Shaout, Adnan, Dominic Colella, and S. S. Awad. "Advanced driver assistance systems-past, present and future." 2011 Seventh International Computer Engineering Conference (ICENCO'2011). IEEE, 2011.

[3] Bengler, Klaus, et al. "Three decades of driver assistance systems: Review and future perspectives." IEEE Intelligent transportation systems magazine 6.4 (2014): 6-22.

[4] https://www.indiatoday.in/india/story/over-9300-deaths-25000-injured-in-3-years-due-to-potholes-1294147-2018-07-24

[5] Kim, Taehyeong, and Seung-Ki Ryu. "Review and analysis of pothole detection methods." Journal of Emerging Trends in Computing and Information Sciences 5.8 (2014): 603-608.

[6] Du, Juan. "Understanding of Object Detection Based on CNN Family and YOLO." Journal of Physics: Conference Series. Vol. 1004. No. 1. IOP Publishing, 2018.

[7] Koch, Christian, and Ioannis Brilakis. "Pothole detection in asphalt pavement images." Advanced Engineering Informatics 25.3 (2011): 507-515.

[8] Huidrom, Lokeshwor, Lalit Kumar Das, and S. K. Sud. "Method for automated assessment of potholes, cracks and patches from road surface video clips." Procedia-Social and Behavioral Sciences 104.2013 (2013): 312-321.

[9] Karthika, R., and Latha Parameswaran. "An automated vision-based algorithm for out of context detection in images." International Journal of Signal and Imaging Systems Engineering 11.1 (2018): 1-8.

[10] Girshick, Ross, et al. "Region-based convolutional networks for accurate object detection and segmentation." IEEE transactions on pattern analysis and machine intelligence 38.1 (2015): 142-158.

[11] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.

[12] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[13] Chebrolu, Koti Naga Renu, and P. N. Kumar. "Deep Learning based Pedestrian Detection at all Light Conditions." 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2019.

[14] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[15] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[16] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[17] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[18] Madli, Rajeshwari, et al. "Automatic detection and notification of potholes and humps on roads to aid drivers." IEEE sensors journal 15.8 (2015): 4313-4318.