Road Damage Detection Using YOLO with Smartphone Images

Conference Paper · December 2020

DOI: 10.1109/BigData50022.2020.9377847

Dongjun Jeong
OP.GG

Road Damage Detection Using YOLO with Smartphone Images

Dongjun Jeong
SDU Robotics
University of Southern Denmark
Campusvej 55, 5230 Odense, Denmark
doj@mmmi.sdu.dk

***Abstract*—Deep learning-based technology is a good key to unlock the object detection tasks in our real world. By using deep neural networks, we could break a problem that is dangerous and very time-consuming but has to be done every day like detecting the road state. This paper describes the solution using YOLO to detect the various types of road damage in the IEEE BigData Cup Challenge 2020. Our YOLOv5x based-solution is light-weight and fast, even it has good accuracy. We achieved an F1 score of 0.58 using our ensemble model with TTA, and it could be an adequate candidate for detecting real road damage in real-time.**

***Index Terms*—Deep Learning, Road Damage Dataset, YOLO**

I. Introduction

In our life, road structure is an essential component. We use public transport to commute almost every day and get a delivery service such as food, clothes or furniture through the road. From the perspective of autonomous driving technology, the road infrastructure has to be managed and kept to be maintaining as perfect as possible to get rid of uncertain obstacles on the road. But, how could we find out the road damage before it acts up? As monitoring the road surface, we could notice where is a problem on the road and prevent the accident.

A human-based road damage monitoring system could be the first answer but not a perfect solution because it is affected by a different condition such as weather, speed of the vehicle, the complexity of the road, and the difference of criteria from the individual inspection. Thus, researchers have developed much more robust and accurate automatic road surface detectors through various methods. For example, using a probabilistic relaxation technique based on 3D information [1], combining 2D gray-scale image and 3D laser scanning data [2] or implementing a deep learning-based model such as CrackNet [3].

In this challenge, the dataset is gathered by a smartphone-based method [4] and we evaluated with various scenarios using YOLO [5] based on a deep learning-based algorithm. It is light-weight and fast in the object detection task, so it is available to improve the smartphone-based model for detecting road damage. The rest of this paper is organized as follows: Section II introduces the road damage dataset used in the IEEE BigData Cup Challenge 2020, explains YOLO of the deep learning-based object detection algorithm. In Section III, we describe the strategy of our solution such as dataset pre-processing, sampling the dataset, training phase, and trials to increase the performance of the model. Section IV contains the final results have developed from the experiments, then we finally conclude from our results in Section V. Also you could find out the detail codes here: (https://github.com/dongjuns/RoadDamageDetector)

II. Related Work

A. Road Damage Dataset

The image dataset is given by the IEEE BigData Cup Challenge 2020 for road damage detection, it consists of three countries such as Czech, India, and Japan. Each image is gathered by a smartphone application in JPEG format. The total number of images is 26,620 for training the road damage detector, 3,595 images from the Czech Republic, and 9,892 images from India were collected with a resolution of 720 x 720 pixels, and 13,133 images were captured with a resolution of 600 x 600 pixels in Japan. There are four damage types of pavement deterioration, such as D00 for the wheel-marked part, D10 for the equal interval, D20 for partial/overall pavement, and D40 for a pothole, as shown in Table I.

TABLE I
Specific Road Damage Types and Definitions considered in Maeda et al. (2018)

Damage Type		**Detail Information**	**Class Name**
Crack	**Longitudinal**	**Wheel-marked part**	**D00**
	Transverse	**Equal interval**	**D10**
	Alligator	**Partial / Overall pavement**	**D20**
Other Damage		**Pothole**	**D40**

B. Object Detection Using Deep Learning

Nowadays, deep learning has an important role in image classification. It extracts the feature maps from an input image using a neural network with hidden layers, and several deep learning networks based on Convolutional Neural Networks (CNNs), such as AlexNet [6], VGGNet [7], ResNet [8], etc, achieved a successful performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9]. A main point is that object detection could be a combination of classification and localization, thus many approaches have developed to solve object detection tasks using deep learning-based technology. The detection model is trained with the image dataset which contains the bounding-boxes and the labels to detect an object. From the perspective of region proposal-based methods, they propose a region that may include the object, classify the object, refine and get rid of overlapped bounding boxes, and score them based on other objects in the input image. And there are representative region-based models such as R-CNN [10], Fast R-CNN [11], and Faster R-CNN [12], and they also called by two-stage object detectors.

C. YOLO

YOLO has a single neural network architecture, predicts a set of bounding boxes and class probabilities at a sitting for every test image. First of all, it divides the full image by several a grid with a specific size, and anchor boxes are generated in every grid of input image by predefined scale and size. Each anchor box predicts the objectness score, box center offset x, box center offset y, box width, box height, and class scores at one time in contrast to a two-stage detector. Thus, YOLO is an extremely fast end-to-end algorithm to detect the objects, and it is called a one-stage object detector. Also, the performance of YOLO has improved over the development of deep learning technology, so there are updated versions [13] - [16] for improving the light-weight, inference speed, and accuracy.

III. Experiments

We used YOLOv5 for this road damage detection challenge because it is state-of-the-art in the YOLO family for now. Also, YOLOv5 has useful components such as data augmentation, state-of-the-art activation functions, utilization of multi-GPU training, and a convenient manual. YOLOv5 uses CSPNet [17] as the backbone to extract the feature map from the image, and it has a Spatial Pyramid Pooling layer (SPP) for using various input image size and improving the robust. In our experiments performed on two V100 GPUs, we fine-tuned the YOLOv5x model which is a pre-trained checkpoint on COCO dataset [18].

We eliminated the bounding boxes of useless classes in the dataset of Japan, every image is checked whether it has an object or not. After pre-processing the images, we could use 1,072 images for the Czech Republic, 3,223 images for India, and 7,900 images for Japan to train our model. Following the previous research with the conjugation of multiple source road images [19], we also wrote many scenarios to compare the performance between only feeding single country images and using two different countries together in the Czech Republic, India, and Japan. We trained our models by splitting them into training dataset 80% and validation dataset 20% such as 5-Fold cross-validation, used various data augmentation options, for example, hue, saturation, value for HSV, image translation, image scale, mosaic, etc, therefore the input images are augmented, as shown in Fig. 1, to train the model. The default hyper-parameters are applied such as SGD optimizer, learning rate 0.01, momentum 0.937, and weight decay 0.0005. And,

TABLE II
F1 score for the experiments: Czech and India

Dataset	Model Name	Validation
Czech	**Czech_1**	**0.345**
	Czech_2	**0.429**
	Czech_3	**0.388**
	Czech_4	**0.36**
	Czech_5	**0.336**
Japan 1k + Czech	**Japan1k_Czech_1**	**0.503**
	Japan1k_Czech_2	**0.437**
	Japan1k_Czech_3	**0.421**
	Japan1k_Czech_4	**0.415**
	Japan1k_Czech_5	**0.382**
Japan 2k + Czech	**Japan2k_Czech_1**	**0.471**
	Japan2k_Czech_2	**0.427**
	Japan2k_Czech_3	**0.386**
	Japan2k_Czech_4	**0.426**
	Japan2k_Czech_5	**0.459**
India	**India_1**	**0.461**
	India_2	**0.424**
	India_3	**0.468**
	India_4	**0.439**
	India_5	**0.455**
Japan 1k + India	**Japan1k_India_1**	**0.37**
	Japan1k_India_2	**0.454**
	Japan1k_India_3	**0.461**
	Japan1k_India_4	**0.504**
	Japan1k_India_5	**0.457**
Japan 2k + India	**Japan2k_India_1**	**0.4**
	Japan2k_India_2	**0.376**
	Japan2k_India_3	**0.458**
	Japan2k_India_4	**0.496**
	Japan2k_India_5	**0.459**

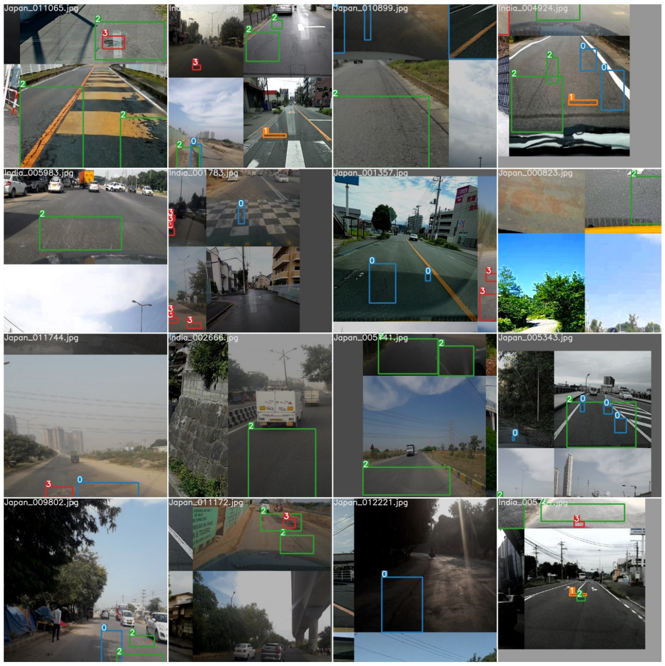

Fig. 1. An example of input images with augmentation for training the model

when we used 50 epochs and 32 batch sizes, the model was trained stably and showed steady performance.

Our experimental results are performed with a confidence threshold of 0.4, non-maximum suppression (NMS) with the intersection of union (IoU) 0.5. As shown in Table II, using Japan road damage dataset and Czech image together usually showed better performance on the validation set than only using a single Czech or India dataset, but not always. In Table II, 'Dataset' means dataset used to train the model, and 'Model Name' shows a specific version number of splitting the dataset by five, such as training dataset 80% and validation dataset 20%. In the case of the model name 'Japan1k_Czech_1', it used 1,000 Japan road images and Czech road images version 1 to train the model, and it predicts the Czech road damage types. When we used 1,000 Japan road damage images with the Czech dataset, the validation result on the Czech road was higher than other Czech model scenarios. And we could check that its loss functions are optimized faster than the single-source dataset when we trained the model using a multi-source dataset. In some cases of India, even Japan dataset affected badly to detect the road damages. As shown in Table III, Japan dataset showed much higher F1 scores than the Czech and India, we experimented with various mix scenarios but it was very time-consuming, so could not try to report the detailed results. The predicted images of our solution are shown in Fig. 2, average inference speed is almost 30 ms for a single image with a resolution 608 x 608 pixels and 40 ms with a resolution 720 x 720 pixels.

TABLE III
F1 SCORE FOR THE EXPERIMENTS: JAPAN

Dataset	Model Name	Validation
Japan	**Japan_1**	**0.575**
	Japan_2	**0.58**
	Japan_3	**0.578**
	Japan_4	**0.572**
	Japan_5	**0.586**

IV. RESULTS

We used ensemble models that are trained with a multi-country source dataset to predict the road damage for each country dataset with confidence threshold 0.4, NMS with IoU 0.5. Also, we applied Test-Time Augmentation (TTA) but it did not improve the performance of our model. We achieved F1 scores of 0.568 for the test1 dataset and 0.571 for the test2 dataset using our solution. After the challenge is finished, we performed several experiments to improve our model. For the Czech dataset, the 5 ensemble model using the Czech and 1,000 Japan images together showed the highest F1 score. If we need to consider inference speed too, the lower ensemble model could be a proper one for a real-time detector on the Czech road. And from India test1 dataset, we could improve the F1 score up to 0.584 for the test1 dataset and 0.577 for test2 dataset using 5 ensemble models which are trained using only the India dataset with TTA. An average inference speed and F1 score of various ensemble models are shown in Table IV.

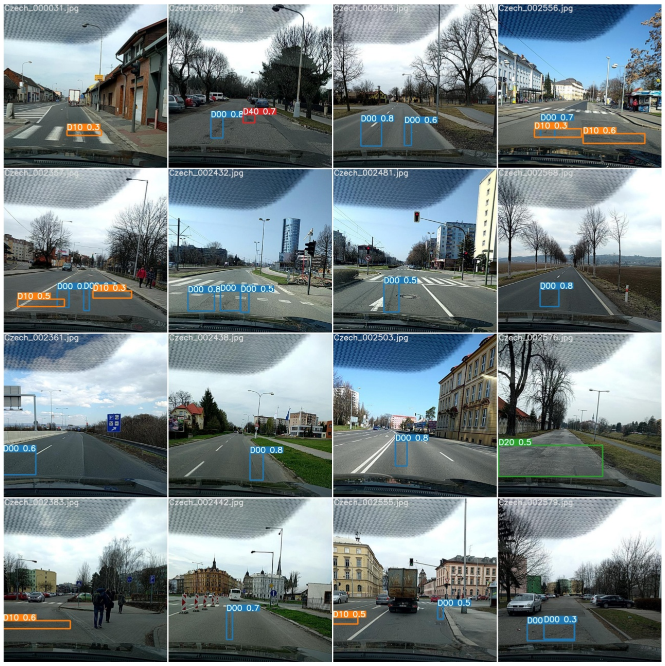

Fig. 2. An example of predicted bounding boxes with the YOLOv5x model

TABLE IV
INFERENCE SPEED AND F1 SCORES OF VARIOUS ENSEMBLE SCENARIOS

Dataset Type		Inference Speed (s)	Test1	Test2
1 Ensemble	**Czech**	**0.033**	**0.517**	**0.509**
	India	**0.032**		
	Japan	**0.03**		
2 Ensemble	**Czech**	**0.066**	**0.523**	**0.543**
	India	**0.063**		
	Japan	**0.06**		
3 Ensemble	**Czech**	**0.096**	**0.528**	**0.537**
	India	**0.095**		
	Japan	**0.089**		
4 Ensemble	**Czech**	**0.131**	**0.537**	**0.542**
	India	**0.129**		
	Japan	**0.12**		
5 Ensemble	**Czech**	**0.16**	**0.537**	**0.541**
	India	**0.163**		
	Japan	**0.15**		
5 Ensemble	**Czech**	**0.18**	**0.584**	**0.577**
5 Ensemble + TTA	**India**	**0.484**		
6 Ensemble + TTA	**Japan**	**0.566**		

V. CONCLUSION

Road damage detection is a crucial problem, and many kinds of researches [20] have developed to break it in this challenge. As one of the deep-learning way, we used a YOLO-based solution [21] to detect road damage in the Czech Republic, India, and Japan. The dataset is collected by Smartphone applications from each country by 1 FPS. We evaluated various dataset scenarios using multi-country images within the Czech, India, and Japan, and it showed some interesting points. Using Japan road damage dataset with the Czech or India could affect the schedule of convergence of the model and generalization positively, but it does not always improve the performance of the model. For our YOLOv5x-based solution, one pre-trained weight needs just 170 MB memory and its inference speed is very fast. We achieved 0.584 for test1 dataset and 0.577 for test2 dataset with 5 or 6 ensemble models and TTA but could make a fast object detector also using lower ensemble and without TTA. In the perspective of real road damage detection problem, Not only accuracy but also inference speed is important even FPS can be a much more crucial point. Therefore, this solution could be an appropriate candidate for road damage detection on smartphone applications in real-time.

REFERENCES

[1] E. Salari and G. Bao, "Automated pavement distress inspection based on 2d and 3d information," in Electro/Information Technology (EIT). IEEE, 2011, pp. 1–4.

[2] J. Huang, W. Liu, and X. Sun, "A Pavement Crack Detection Method Combining 2D with 3D Information Based on Dempster-Shafer Theory," Comput. Aided Civ. Infrastructure Eng. (CACAIE), vol. 29, no. 4, pp. 299–313, 2014.

[3] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network," Comput. Aided Civ. Infrastructure Eng. (CACAIE), vol. 32, no. 10, pp. 805–819, 2017.

[4] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," Comput. Aided Civ. Infrastructure Eng. (CACAIE), vol. 33, no. 12, pp. 1127–1141, 2018.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

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

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014. [Online]. Available: arXiv:1409.1556.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, vol. 0, pp. 580–587.

[11] R. B. Girshick, "Fast R-CNN," in ICCV, 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[13] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525.

[14] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: arXiv:1804.02767.

[15] Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020. [Online]. Available: arXiv:2004.10934.

[16] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, Adam Hogan, lorenzomammana, tkianai, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Hatovix, Jake Poznanski, Lijun Yu, changyu98, Prashant Rai, Russ Ferriday, Trevor Sullivan, Wang Xinyu, YuriRibeiro, Eduard Reñé Claramunt, hopesala, pritul dave, and yzchen, "ultralytics/yolov5: v3.0," Zenodo, 13-Aug-2020, doi: 10.5281/zenodo.3983579.

[17] Chien-Yao Wang, H. Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen and Jun-Wei Hsieh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571–1580.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., "Microsoft COCO: Common Objects in Context," 2014. [Online]. Available: arXiv:1405.0312.

[19] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama and Y. Sekimoto, "Transfer Learning-based Road Damage Detection for Multiple Countries," 2020. [Online]. Available: arXiv:2008.13101.

[20] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama and Y. Sekimoto, "Global Road Damage Detection: State-of-the-art Solutions," 2020. [Online]. Available: arXiv:2011.08740.

[21] A. Alfarrarjeh, D. Trivedi, S. H. Kim and C. Shahabi, "A Deep Learning Approach for Road Damage Detection from Smartphone Images," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5201-5204, doi: 10.1109/BigData.2018.8621899.