Mathematical Biosciences and Engineering

MBE, 16(6): 7982–7994.
DOI: 10.3934/mbe.2019402
Received: 02 June 2019
Accepted: 15 August 2019
Published: 03 September 2019

http://www.aimspress.com/journal/MBE

Research article

Embedded system for road damage detection by deep convolutional neural network

Siyu Chen[1], Yin Zhang[1], Yuhang Zhang[2], Jiajia Yu[1,*] and Yanxiang Zhu[3]

[1] Department of Electronic and Computer Engineering, Southeast University Chengxian College, Nanjing 210088, China
[2] Department of Mechanical and Electrical Engineering, Southeast University Chengxian College, Nanjing 210088, China
[3] Verimake Research, Nanjing Qujike Info-tech Co., Ltd., Nanjing 210088, China

* **Correspondence:** Email: jjyu@cxxy.seu.edu.cn; Tel: +8613913020893.

Abstract: Road pavement could be damaged due to various reasons, causing damages such as cracks and pits. These damages cause potential dangers in traffic safety. It is necessary for road maintenance departments to find damages in time before maintenance. At present, maintenance departments of some high-level roads are equipped with specialized detection vehicles such as laser scanning vehicles to detect road damages. These kinds of devices can get good detection performance, but the economic cost is very high.

In this paper, we use a road damage image dataset to train an object detection model based on deep convolutional neural network and deploy it on a low-cost embedded platform to form an embedded system. The system uses a common camera mounted on windshield of a common vehicle as sensor to detect road damages. The embedded system consumes about 352 ms to process one frame of image and can achieve a recall rate of about 76% which is higher than some previous related works. The recall rate of this scheme using common cameras is less than that of high-level specialized detectors, but the economic cost is much lower than them. After subsequent development, the road maintenance department with limited funds can consider about schemes like this.

Keywords: road damage detection; road survey; embedded system; object detection; convolutional neural network; deep learning

1. Introduction

With continuous growth of economy, transportation industry is also developing at high speed and road mileage in the world is increasing. While total length of road is increasing, old roads are slowly becoming damaged over time and road maintenance departments in various areas are facing increasing maintenance demands.

Finding damages of road surface is a necessary step before road maintenance department perform repair work on road surface. Traditionally, road damage detection is mainly through manual search, which is extremely time-consuming and labor-intensive. A scheme that can easily find road damage is urgently needed to be developed. In fact, there have some scholars [1] researched this problem previously. At present, automatic detection of road damage mainly through three methods of laser, radar and vision.

In 2006, Hinton et al. proposed the concept of deep learning [2]. Since then, technology based on deep learning have begun to develop rapidly. Research and product development of computer vision based on deep neural network are rapidly emerging. Using image processing technology and deep neural network, a new round of exploration and practice has been carried out by scholars [3,4] on scheme of road damage detection.

In this paper, an embedded system is designed based on SSD object detection method proposed by W. Liu et al. [5], MobileNet object classification method proposed by A. G. Howard et al. [6], road damage image dataset proposed by H. Maeda et al. [4]. Main processor of the embedded system is Rockchip RK3399Pro SoC which integrates a separate neural processing unit (abbr. NPU). It runs an object detection model that can detect road damages through common camera. When road damage is detected, the system can automatically acquire geographic location information through GPS module and save related data in memory including image and location information. The system has been tested to achieve a useful detection recall rate with low economic cost.

In the following section 2, this paper will introduce some related work, including various schemes of road damage detection proposed by previous works and some current object detection methods based on deep convolutional neural network. In section 3, we will from top to bottom introduce overall architecture, detection process and core object detection method of our embedded system. In section 4, this paper will present performance test results of the system we designed and compare it with a previous work.

2. Related works

2.1. Road damage detection

Maintenance departments of highways and other high-level roads usually have sufficient funds to use expensive specialty devices to detect road damages. These expensive specialty devices usually can achieve very good accuracy [1,7]. Scholars began trying to use various schemes to detect road damages many years ago. After a long period of development, some products for road damage detection have come out, such as JG-1 laser 3-D road condition intelligent detection system [7], ZOYON-RTM road detection vehicle system [1], road condition rapid detection system based on line sweep technology CiCS [1] and so on. These devices typically can achieve an accuracy of over 95% and are widely used in practice.

However, such specialty devices are very expensive, only maintenance departments of high-level roads can afford to choose them. On common roads, most road maintenance departments still use method of manual searching which greatly slows down the efficiency of road maintenance work.

In recent years, object detection technology based on deep convolutional neural network has developed rapidly and related technologies have gradually landed in various application scenarios. In 2016, L. Zhang et al. used a low-cost smartphone to take 500 photos of road cracks on campus of Temple University and trained a simple 10-layer convolutional neural network model to determine whether there were road cracks in a photo. This model can achieve about 92.51% recall rate for recognition of the images in test set. However, because of the small size of their train set, generalization ability of their model is poor. Recognition performance of images from test set and other images is quite different and it cannot be applied in real world. However, it is undeniable that this is the first proposed method to detect road cracks by deep neural network [3].

In 2018, H. Maeda et al. took 9,053 photos containing 15,435 road damages in seven cities of Japan [4] and trained two models based on SSD-MobileNet or SSD-Inception. They packaged one model into an Android app runs on a common Android smartphone. Images are captured by smartphone camera for road damage detection, their device can achieve a recall rate of about 50% and a precision rate of about 75% on weighted average. Our system retrained a new model using dataset shared by them on GitHub and further optimized it.

2.2. Object detection

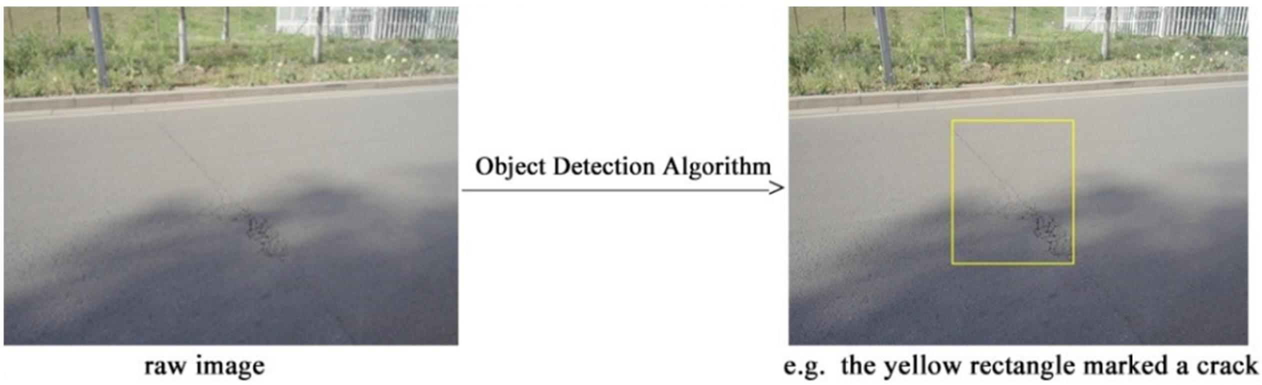

Figure 1. Example of object detection.

Road damage detection through vision is a problem of object detection. Object detection is a basic problem in the field of computer vision. As an example shown in Figure 1, input of object detection is a frame of image; output is all existing targets in the image and each target is given class information and position information. In many real-world scenarios including road damage detection, object detection is often the first step in visual perception, which has been researched by many institutions and scholars [5,8–11]. We have several following algorithms to select to apply on our system.

2.2.1. Faster R-CNN: Faster regions with CNN feature

R-CNN is an object detection algorithm proposed by R. Girshick et al. [8] in 2013. This algorithm can be said to be the first pioneering work of object detection using deep learning. After development of R-CNN and Fast R-CNN, S. Ren et al. proposed Faster R-CNN [9] in NIPS2015. Faster R-CNN unifies four basic steps of object detection (proposal generation, feature extraction, region classification, region refinement) into one same deep neural network framework, to reduce repetitive computation and greatly improve running speed compared with previous generations. However, compared with other common networks, Faster R-CNN is still running more slowly and not suitable for running on mobile poor-performance embedded platforms.

2.2.2. YOLO v3: You only look once v3

YOLO is an object detection algorithm proposed by J. Redmon et al. [10] in CVPR2016. Core idea of YOLO is to transform object detection into a regression problem to solve and based on a single end-to-end network to complete the object location and category output from the original image input. YOLO v3 [11] is the third version of YOLO, which has a fast running speed but is less sensitive to small object detection such as road cracks.

2.2.3. SSD: Single shot multiBox detector

SSD is an object detection algorithm proposed by W. Liu et al. [5] in ECCV2016. It has obvious speed advantage compared with Faster R-CNN and obvious accuracy advantage compared with YOLO.

Comprehensively evaluated strengths and weaknesses of each model, combined with the effectiveness of training and detection of road damage dataset, we select to use SSD as our object detection network.

3. Proposed method

3.1. System architecture

L. Zhang et al. and H. Maeda et al. have tried to use a common Android phone for road damage detection [3,4]. After our tests, computing and imaging performance of smartphones are not so satisfying with road damage detection. Considering that more suitable computing chips and cameras can be selected, we decided to design a dedicated embedded system. The embedded system architecture we proposed is as follows.

Main processor of this embedded system is Rockchip RK3399Pro SoC, which is connected to a High-Dynamic-Range (abbr. HDR) camera for capturing road images and a GPS module for acquiring geographical location. It is also connected to peripherals such as touch screen to facilitate user operation.

We select RK3399Pro SoC as main processor cause the chip integrates a separate NPU which can accelerate computation of deep neural network. The NPU has computing performance of 2.4 TOPs.

We select an HDR camera to capture road image. Compared with ordinary smartphone camera, it performs better in backlight condition and can clearly capture road surface image when it is directly exposed to sunlight. Using HDR camera allows the system can be used in more illumination condition. Overall architecture of our embedded system is shown in Figure 2 and a usage example is shown in Figure 3.

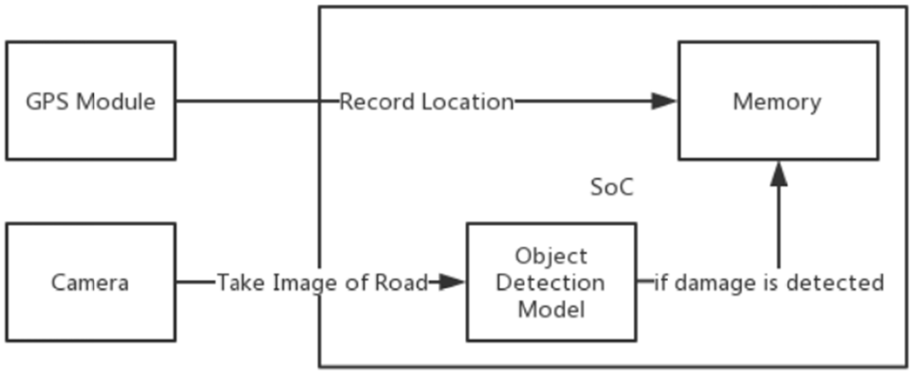

Figure 2. Overall architecture of our embedded system. Other road damage detectors can also use similar architecture.

Figure 3. Usage example of our system. A camera and a GPS module hang on windshield of a car.

We convert the trained object detection model into a ".rknn" file that can be accelerated using Rockchip NPU, then use OpenCV, NumPy, pySerial, PyQt5 and other Python packages to write programs for reading images, post-processing, reading GPS data, Graphic User Interface (abbr. GUI), etc.. That constitutes the entire software system.

3.2. Overall detection process

Mount the camera on front windshield of car, align the road ahead with a suitable angle and run the software to start the detection. System will continuously capture images of road ahead and input images into object detection model. After computing of the model and related post-processing, it can be determined whether a frame of image contains road damages. If an image is considered to contain road damages, system will automatically acquire geographic location information through GPS module and save related data in memory including image and location information for manual query and processing in later stage.

A description in flowchart form is shown in Figure 4.

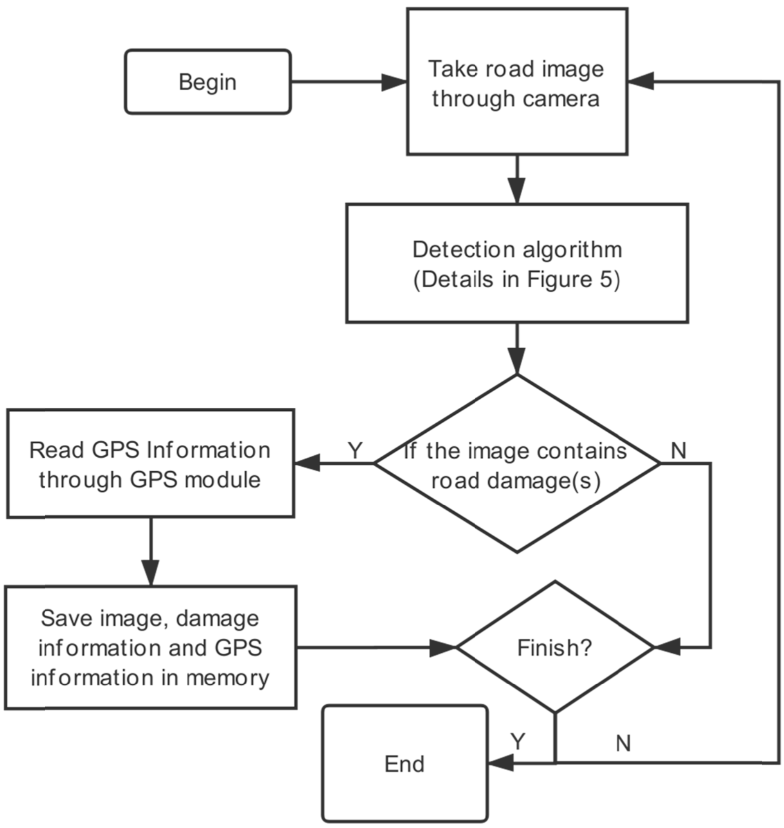

Figure 4. The main process of the detection.

3.3. Core object detection model

Single Shot MultiBox Detector (SSD) [5] is a one-stage detector for multiple categories, which is slightly less accurate than two-stage network (such as Faster R-CNN [9], etc.). However, two-stage network needs to process image to generate a series of proposals and then classify proposals by convolutional neural network, while one-stage network directly convert problem of object border locating into regression problem. Therefore, one-stage network such as SSD usually runs faster than two-stage network.

In the demo code given in original paper of SSD, VGG [12] is used as the backbone network for feature extraction. In real-world scenario applications, when SSD is used for object detection on mobile platforms such as smartphones or embedded systems, MobileNet [6] is usually used to replace VGG network to achieve faster computing speed. MobileNet is a convolutional neural network proposed by A. G. Howard et al. particularly designed for mobile platforms. It uses depth-wise separable convolution to replace traditional convolution, in order to reduce network weight parameters, which makes model more lightweight and can achieve a reasonable balance between efficiency and accuracy.

Considered about computing performance of the embedded platform and actual application scenario, we select SSD whose backbone is replaced to MobileNet, namely SSD-MobileNet, for road damage detection. We used RoadDamageDataset [4] to train an SSD-MobileNet-based object detection model using TensorFlow deep learning framework on platform of NVIDIA GeForce GTX 1060 GPU. Input of the model is a matrix of 300 × 300 × 3 (i.e., an RGB image of 300 × 300 pixels) and output is a matrix contains information of position and confidence of road damages existing in the image.

The algorithm flow from camera capturing image to the computing of detection result is shown in Figures 5 and 6.

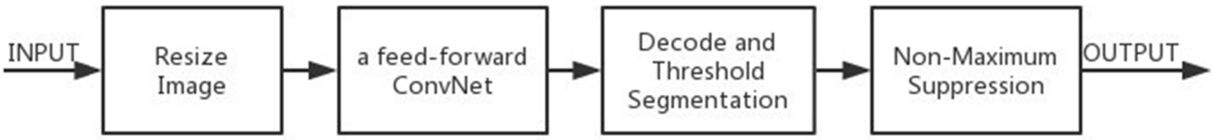

Figure 5. Core process of the detection algorithm [5,6].

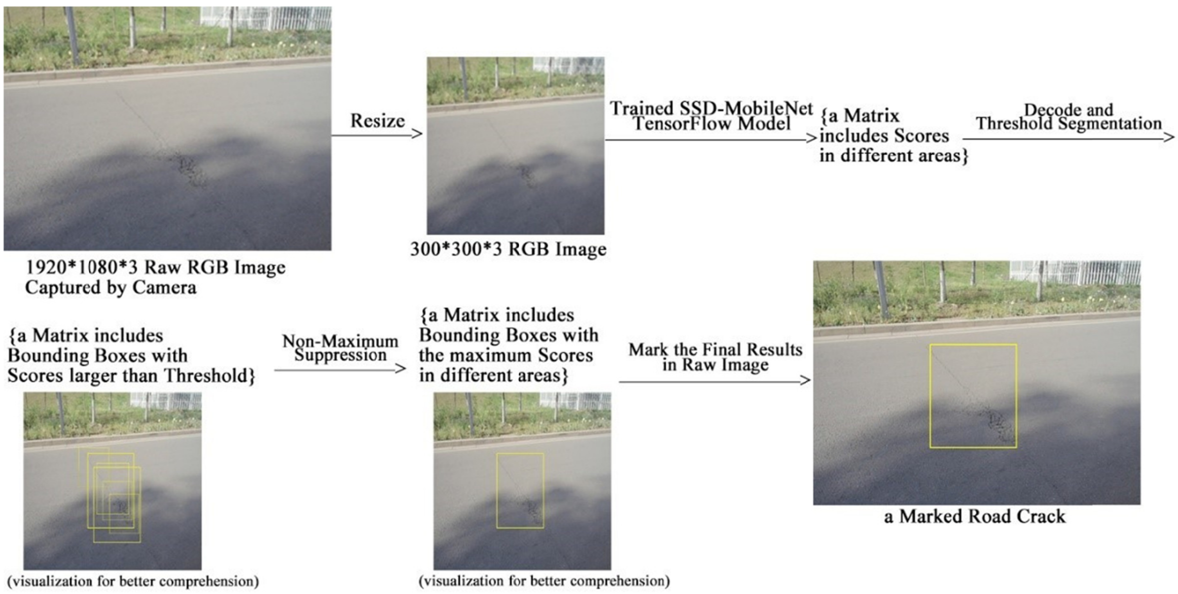

Figure 6. Process of detection on an example image.

The input is an image captured by camera (e.g., the image size captured by camera in our test system is 1920 × 1080), then the image is resized to 300 × 300 using OpenCV and it is fed into the SSD-MobileNet model running on NPU for inferencing. Decode the computing result from NPU and do threshold segmentation, a matrix is obtained, which contains the boundary boxes of road damages and their respective confidence levels, indicating which locations may have targets. However, the boundary boxes usually have a large number of repetitive results, so it is necessary to perform non-maximum suppression and the results with almost the same location but low confidence scores are removed, then the final inferencing results are obtained.

4. Experimental evaluation

4.1. Experiment on testset

Generally, we can use benchmarks such as precision and recall [13] to evaluate performance of an object detection model.

For such models, Confusion Matrix [13] between prediction results and ground truth can be represented as shown in Table 1.

Table 1. Prediction results confusion matrix [13].

Truth \ Prediction	Positive	Negative
True	True Positive (TP)	False Negative (FN)
False	False Positive (FP)	True Negative (TN)

Precision is defined as:

$$Precision = \frac{TP}{TP + FP}$$

That is, in N_P images which the system thought contain road damages, about $N_P \cdot Precision$ images of them contain true road damages. A high precision rate also means a low false positive rate.

Recall is defined as:

$$Recall = \frac{TP}{TP + FN}$$

That is, when there are N_T damages on road, the system can detect about $N_T \cdot Recall$ damages of them. A high recall rate also means a low false negative rate.

SSD-MobileNet object detection model densely samples the whole inputted image, then compute confidence score of objects to be detected in each boundary box.

We can set a threshold manually, when score outputted by the model is higher than threshold, it is considered that there are road damages existing in image.

That is:

If score > threshold:

Read GPS information

Save detect result

Else:

Do nothing

For the same model, precision and recall are a pair of contradictory measures, one of which increases while the other inevitably decreases. Generally, precision and threshold are positively correlated, recall and threshold are negatively correlated.

The setting of threshold depends on application requirements of different detection systems. For road damage detection, we should pursue a lower false negative rate, that is, pursue a higher recall, so as to mark as many road damages as possible.

In engineering, a common method to evaluate how to select the appropriate threshold is to observe the P-R Curve. Horizontal axis of P-R Curve is recall and vertical axis is precision. Relationship between precision and recall under different threshold of the same model can be observed more intuitively, so that it is convenient to select the appropriate threshold according to application requirements. Hence, through drawing and analyzing P-R Curve, threshold can be selected suitably in order to make detector have better performance.

Using different values of threshold, test the model with same group of images, sorting out the relationship between the inference results and ground truth, recording the number of TP, FP, FN, and then calculating precision and recall, P-R Curve can be drawn.

We tested and calculated precision and recall when threshold is taken from 9 values from 0.1 to 0.9, then draw P-R Curve (shown in Figure 7). Observing the curve, we can find that when threshold is taken as 0.3, i.e., when the confidence score is higher than 0.3 the system thinks the image contains road damage, recall can be as high as possible without too low precision.

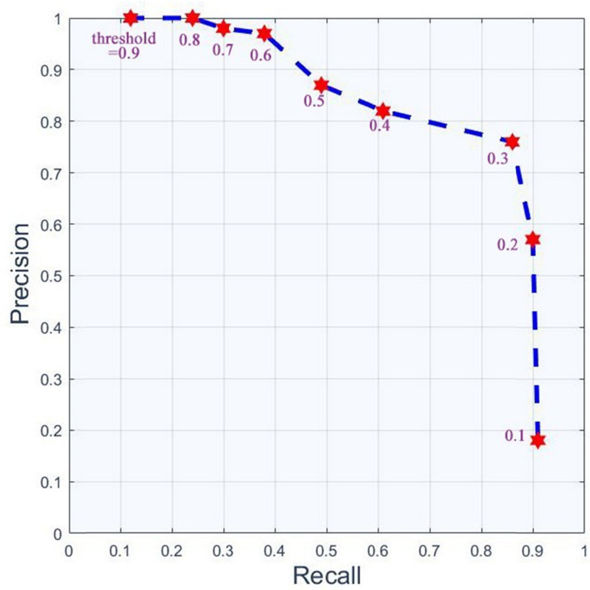

Figure 7. P-R Curve obtained by testing for 1000 random images. Suitable threshold can be selected through analyzing P-R Curve.

Then, we randomly picked 1000 images containing road damages from RoadDamageDataset [4] and run model on NVIDIA GeForce GTX 1060 GPU to inference for these images. This GPU consumes about 47 ms to finish inference of one frame of image on average. Finally, the model can detect about 86.7% road damages of all, that is, recall is about 86.7%.

4.2. Experiment on real roads

Detection on real roads can be affected by various lighting conditions, different road designs and other factors, it will not be so ideal as experiment on testset. However, it is not enough to have a good performance only on testset, the road damage detect system we designed must take into account performance while using on real roads. We deployed the model on a Rockchip RK3399Pro platform described in 3.1 and tested it. With the help of acceleration of neural network computing by dedicated NPU, RK3399Pro SoC consumes about 352 ms to finish inference of one frame of image, with much lower power consumption than GTX1060 GPU (RK3399Pro~7W vs GTX1060~206W). Appearance of our test system is shown in Figure 8.

We use PyQt5 to make a GUI for using with touch screen connected to RK3399Pro SoC, to facilitate user to operate the system when detecting on real roads. A screenshot of GUI is shown in Figure 9.

Figure 8. Appearance of the whole embedded system.

We randomly selected about 100 km of non-repetitive ordinary roads for testing in Jiangbei New Area, Nanjing, Jiangsu Province, China. During the test, the system was always running and a person is arranged for directly observing road surface to record the number of road damages.

After completion of the test, 636 images were saved in system. After manual inspection, the system marked 209 true road damages (TP) in 636 photos. Compared with that, the person directly observing road surface counts 274 road damages. After sorting out and calculating, recall of this test is about 76.28% and precision is about 32.86%. In other words, approximately 76% of true road damages can be detected by the system.

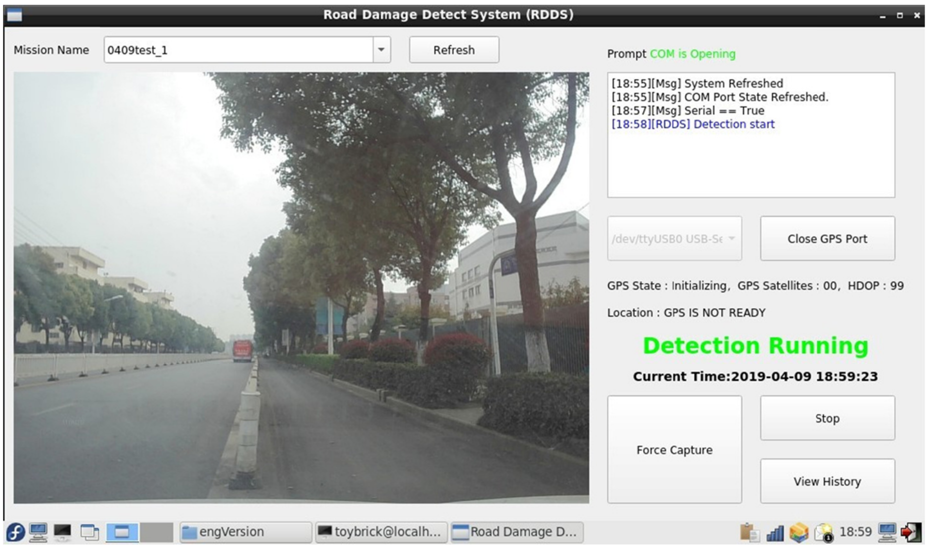

Figure 9. Example of our graphical user interface.

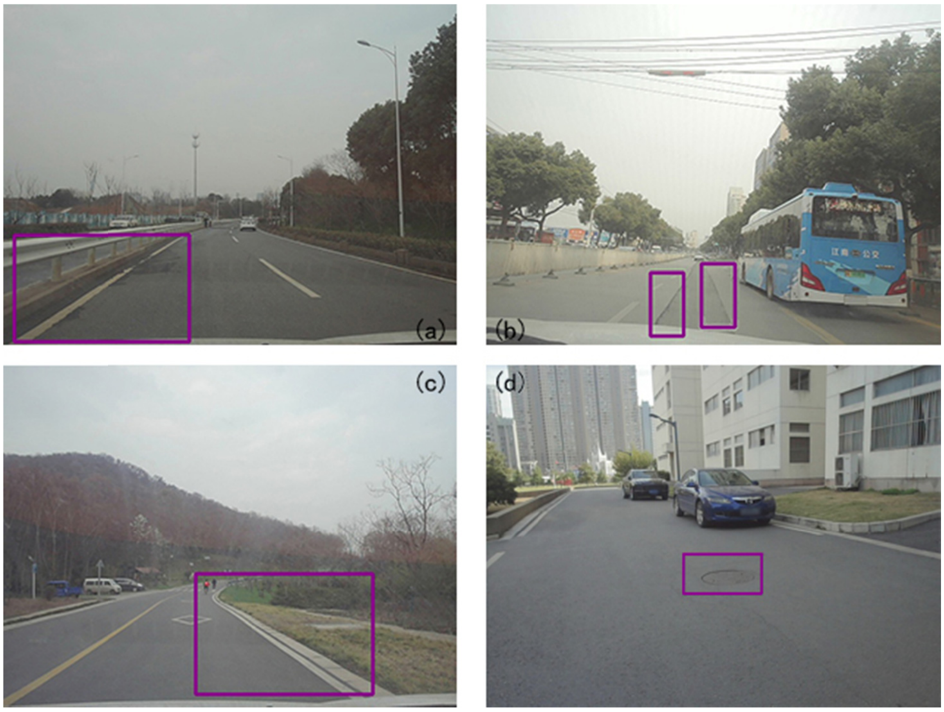

Figure 10. Some examples of saved images in this test. (a): White line is blurred (True Positive); (b): Two cracks (True Positive); (c): White line is normal, but the system thinks it is blurred (False Positive); (d): There is a manhole cover, but the system thinks it is a crack (False Positive).

When creating RoadDamageDataset, H. Maeda et al. built a test system running on Android smartphone, it consumes about 1.5 s to process one frame of image and it can achieve recall of about 50% and precision about 79% in real roads in Japan [4]. It has a higher precision but only about half of real road damages can be detected. With help of NPU acceleration, our system is four times faster as their Android smartphone. We also optimized some parameters of model to improve recall on real road detection to 1.5 times of their system. However, when we improve recall, we sacrifice a certain precision and there are more false positive images. After each detection, the saved images need to be manually reviewed to remove about two-thirds images with false positive.

Several pictures of the system saved in the test are shown in Figure 10.

5. Conclusions

Referenced to the road damage detector [4] designed by H. Maeda et al. on Android smartphone, we retrained an object detection model, then made some improvement and designed a road damage detector running on the RK3399Pro embedded platform. Through analyzing P-R Curve, we adjusted the confidence score threshold and sacrificed a certain precision to achieve a higher recall. At the same time, we also transplanted the object detection model for road damage detection to an embedded platform which is nearly as cheap as Android smartphone but has a dedicated processing unit so it can compute neural network and process images faster. We also use HDR camera instead of common camera to make the system suitable for more lighting conditions.

Table 2. Comparison of some schemes.

Scheme	Specialized High-Level Detectors (e.g. JG-1 [7])	Detectors Run on Android Smartphone [3,4]	Our Method
Price	High (almost >$30000)	~$300	~$300
Precision	>95%	~79%	~32.86%
Recall	>95%	~50%	~76.28%
Time Consumption on One Frame	–	~1.5 s	~352 ms

However, due to limited size of dataset and limited algorithm performance, similar to the schemes using smartphones, our low-cost system has lower performance than specialized high-level detectors. Our system currently only detects about 76% of true road damages and manual reviewing to remove false positive results is needed after each detection task. Comparison of some schemes is shown in Table 2.

Road damage detection is different from detection of common objects such as cats and dogs. Road damage detection is less popular and there is no large-size dataset like MS COCO [14], while one of the ways to improve performance of deep learning model is to improve size of dataset.

In the future, low-cost road damage detectors using similar schemes will continue to be improved before they can be put into real applications.

Acknowledgements

The work of this paper is supported by 2018 Jiangsu Province University Student Innovation and Entrepreneurship Training Program (201812689012Y) and 2018 Southeast University Chengxian College Practical Innovation Training Program (1103100069).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. E. Chuo, Research history and prospect of domestic pavement automatic detection system, *China High-Tech. Enterp.*, **1** (2011), 1–3.
2. G. E. Hinton, S. Osindero and Y. W. Teh, A fast learning algorithm for deep belief nets, *Neural Comput.*, **18** (2006), 1527–1554.
3. L. Zhang, F. Yang, Y. D. Zhang, et al., Road crack detection using deep convolutional neural network, *2016 IEEE Int. Conf. Image Process. (ICIP)*, 3708–3712.
4. H. Maeda, Y. Sekimoto, T. Seto, et al., Road damage detection using deep neural networks with images captured through a smartphone, *Comput. Aided Civ. Infrastruct. Eng.*, **33** (2018), 1127–1141.
5. W. Liu, D. Anguelov, D. Erhan, et al., SSD: Single shot multiBox detector, *2016 European Conf. Comput. Vision (ECCV)*, 21–37.
6. A. G. Howard, M. Zhu, B. Chen, et al., MobileNets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861.
7. A. He, Advantages of JG-1 laser 3D pavement inspection system, *China Highw.*, **16** (2005), 94–95.
8. R. Girshick, J. Donahue, T. Darrell, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, *2014 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 580–587.
9. S. Ren, K. He, R. Girshick, et al., Faster R-CNN: Towards real-time object detection with region Proposal networks, *Adv. Neural Inf. Process. Syst.*, (2015), 91–99.
10. J. Redmon, S. Divvala, R. Girshick, et al., You only look once: Unified, Real-Time object detection, *2016 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 779–788.
11. R. Joseph and A. Farhadi, YOLO v3: An incremental improvement, preprint, arXiv:1804.02767.
12. S. Karen and A. Zisserman, Very deep convolutional networks for Large-Scale image recognition, preprint, arXiv:1409.1556.
13. Z. Zhou, Model evaluating and selecting, in *Machine Learning* (eds. Zhihua Zhou), Tsinghua University Press, (2016), 23–52.
14. T. Lin, M. Maire, S. Belongie, et al., Microsoft COCO: Common objects in context, *2014 European Conf. Comput. Vision (ECCV)*, 740–755.