Manhole Cover Detection Using Depth Information

Haotian Ren¹ and Fang Deng²

¹ School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100876, China
² School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100876, China

*Corresponding author’s e-mail: rht@bupt.edu.cn

Abstract. Nowadays, the security of manhole cover has become a growing concern. In the past, manual inspection and Internet of Things were often used to ensure the safety of manhole cover. With the advent of smart cities, people look forward to using cameras as a more convenient method to detect whether manhole covers are damaged. Based on the structure of object detection network, we integrate the depth information in the picture and create a new model which improves the accuracy of detecting damaged manhole covers.

1. Introduction
Municipal infrastructure includes many aspects and the manhole cover is one of the key aspects. The management of manhole covers involves the benefits of urban residents. The damaged manhole covers not only affects the tidiness of the city, but also poses a great threat to public safety. In response to the above problems, many scholars look forward to the Internet of Things, hoping to monitor the safety information of the manhole cover in real time by installing sensors inside the manhole cover to realize the smart manhole cover. This method solves the management problem of manhole cover safety to a certain extent, but there are still some shortcomings. First of all, each sensor under the smart manhole cover only plays a single role, and the current smart manhole cover often needs to have multiple sensors to achieve the corresponding function, which is too expensive to popularize in society. Secondly, the scope of sharing information of the smart manhole cover is limited which caused the fact that the signal sent by the sensor can only be received within a sufficiently close range from the manhole cover. Thirdly, sensors under the smart manhole cover runs on batteries, which need to be replaced regularly. Therefore, in view of the problems of smart manhole covers, this paper proposes a manhole cover detection method fused with depth information, which can detect manhole covers more specifically than traditional target detection methods.

2. Research actuality
Detection of manhole cover belongs to object detection mission in computer vision. The purpose of object detection is detecting the positions and classes of specific items. The structures of object detection networks are similar, which consist of backbone, neck and head[1]. The structure is shown in figure 1.
In the structure, backbone is used to extract image feature information. Some backbones are used with GPU, such as: VGG[6] and ResNet[5]. While some are focus on CPU, such as: SqueezeNet[9] and MobileNet[10], which sacrifice a little precision for less computation.

In the procedure of extracting features, the low-level features which are import to predict tiny objects are usually dropped. Neck is a special part connecting backbone and head, whose function is enriching the multiscale information and merging the high-level features with the low-level features.

Head is responsible to output the final prediction results. Different head networks also divide the existing object detection networks into 3 parts: single-stage object detection network, multi-stage object detection network and anchor-free object detection network.

Single-stage object detection network has only one output, which contains the position information and category information of the target to be detected. YOLO and SSD[7] is the classic single-stage object detection network. The multi-stage target detection network has multiple outputs. Take Fast R-CNN series[11] as an example, the first head network needs to distinguish the foreground and the background according to the feature map of the backbone network and output all possible targets. After that, the network extracts the feature maps corresponding to these targets, changes the feature maps to the same size after ROI Pooling, and then inputs them to the second head network to classify and position regression for each head network. Due to the requirement of operating speed, this paper decided to use a single-stage target detection network to detect damaged manhole covers.

3. Manhole cover detection using depth information

3.1. Depth information

Compared with an intact manhole cover, a damaged manhole cover will have a certain degree of difference in depth information. This difference plays a vital role in detecting whether the manhole cover is damaged or not. We plan to use binocular cameras to obtain the depth information in the picture, but due to equipment limitations, we use Fully Convolutional Residual Networks (FCRN)[12] to simulate the depth information in the picture. Part of the result is shown in Figure 2.
3.2. Our model

Our model is proposed based on YOLOv4[1]. In addition to fusing depth images, our model has also made some improvements based on YOLOv4.

First, we change the input of the network. As we all know, YOLOv4 uses the CSPDarknet53[2] module to extract features while converting the picture into higher-dimensional information. At the same time, it loses some low-dimensional information, which is detailed information. Some of the detailed information is particularly effective for detecting whether the manhole cover is damaged. Therefore, we incorporate depth information when inputting the picture, and the input is changed from the original 3 dimension of RGB to the 4 dimension of RGBD.

When it comes to the backbone network, we improved CSPDarknet53[2] which is the backbone network in YOLOv4 but still use its basic unit. The CSPDarknet53 in YOLOv4 is composed of 5 CSP blocks, and the structure of each CSP block is shown in Figure 4.
Figure 4. The structure of CSP block.

Each CSP block contains N residual units, which is the structure in the dashed box in Figure 4. First, the network will use a 3X3 convolution kernel to convolve the input, and then divide the output into two parts, part1 and part2. After the operation of N residual units, part1 will concatenate with part2 and the result will pass through a 1X1 convolution kernel to reduce complexity. This operation allows the network to increase accuracy while reducing the amount of calculation.

Compared with the traditional convolutional layer, the CSP module refers to the design of Resnet, which effectively solves the problem of gradient disappearance and over-fitting caused by the excessive depth of the network. At the same time, half of the channels are not involved in the residual calculation, which greatly reduces the calculation. Therefore, our model replaces all the convolutional layers whose kernel size is 3X3 convolutional layers and stride size is 1 with CSP block in YOLOv4. The specific structure of our model is shown in Figure 5.

Figure 5. The structure of our model.

Here is the specific illustration of the structure of our model. In the input stage, we fuse the depth information with the picture, and then change the number of channels of the feature map through 30
3*3 convolution kernels, and input it into the first CSP block, and record the output as CSP1. We take CSP1 as the input and send to a convolutional layer with a stride size of 2 for down-sampling, and then inputs the result to the second CSP block, and the output is recorded as CSP2. Then we down-sample the CSP2 and input the result to the third CSP block, the output is recorded as CSP3. And then like YOLOv4, we use SPP (Spatial Pyramid Pooling)[3]. Firstly, down-sample CSP3 and put the result through 3 max-pooling layers with the kernel size of 5×5, 9×9 and 13×13. After that, we concatenate the results of three max-pooling layer with the origin input (which is not through the max-pooling layer). This step not only increase receptive field, but also fuse multi-scale information. Then we input the connection result into the fourth CSP block, and the output is recorded as CSP4. Next, we use the PANet[4] structure which is similar to YOLOv4 to up-sample CSP4, and concatenate the result with CSP3. The output will pass through the fifth CSP block, and the output is marked as CSP5. In YOLOv4, the PANet structure use a 3×3 convolution kernel to fuse the features after concatenating the channel while we replace it with a CSP block, which improves the accuracy while reducing the amount of calculation. After CSP5 is up-sampled, concatenate the output with CSP2. We input the result to the sixth CSP block, and the output is called CSP6. CSP6 will be input into the first head, a convolutional layer with 3 × (NC + 5) kernels whose size are 3 × 3. Here, NC is the number of categories to be detected. The result of the convolutional layer is the first output of our model, whose scale is 80*80. Then we concatenate the down-sampled CSP6 with CSP5 and input the result to the seventh CSP block, whose output is recorded as CSP7. Like CSP6, CSP7 will be input into the second head and generate the second output of the model whose scale is 40*40. Then we down-sample CSP7, and concatenate the result with CSP4. After that, we input the concatenating result into the eighth CSP block and record the output as CSP8. Similar to CSP6 and CSP7, CSP8 will be input into the third head and generate the third output of the model whose scale is 20*20. The purpose of multiple outputs is to adapt to detection objects of different sizes, which is the same as YOLOv4 and will not be repeated here.

In addition to replacing the multiple convolutional layers of YOLOv4 with the CSP block, we also replaced the activation function from Mish[8] to Leaky Relu in order to reduce the amount of calculation of the model. The calculation formula is shown in (1):

\[ y = \max(x, ax) \] (1)

In addition to replacing the multiple convolutional layers of YOLOv4 with the CSP block, we also replaced the activation function from Mish to Leaky Relu. In formula (1), a is a small number, usually below 0.2. The formula of the Mish function is shown in (2):

\[ y = x \cdot \tanh(\ln(1 + e^x)) \] (2)

Although the whole process can be guided, it requires more calculations. So we decide to use Leaky Relu as the activation function.

4. Experiment

4.1. Dataset

In order to conduct a confirmatory experiment, we produced a damaged manhole cover detection data set. We photographed the manhole cover during the day and night to test the robustness of the algorithm under different light conditions. As damaged manhole covers are not easy to found relatively, we choose to crawl pictures of damaged manhole covers on Baidu and Google. The dataset includes three types of pictures: intact manhole covers pictures taken during the day, intact manhole covers pictures taken at night and pictures of damaged manhole covers. Examples are shown in Figure 6, Figure 7, and Figure 8:
The training set has a total of 458 pictures, including 203 pictures of intact manhole covers taken during the day, 147 intact manhole covers pictures taken at night and 108 pictures of damaged manhole covers. The test set has a total of 53 pictures, including 23 pictures of intact manhole covers taken during the day, 17 intact manhole covers pictures taken at night and 13 pictures of damaged manhole covers.

4.2. Experimental setup
In the experiment, we use Tesla-T4 to execute all the training in the batch size of 8. The training epoch is set to 200. And all the depth map used in our model is generated by well-trained FCRN.

4.3. Results
We used the YOLOv4 model and our improved YOLOv4 model for comparative experiments. The experimental results are in Table 1.

| Model     | Precision | Recall | mAP:0.5 | mAP:0.5~0.95 |
|-----------|-----------|--------|---------|--------------|
| YOLOv4    | 0.748     | 0.851  | 0.867   | 0.525        |
| Our model | 0.707     | 0.883  | 0.883   | 0.543        |

In the experiment, we use precision, recall, mAP(mean Average Precision):0.5 and mAP:0.5~0.95 as our evaluation index. From Table 1, we can see that although the YOLOv4 model has a slight advantage in precision, our model is ahead of YOLOv4 in terms of recall, mAP:0.5, and mAP:0.5~0.95. Our model leads by 3.2% on recall, 1.6% on mAP: 0.5, and 1.8% on mAP:0.5~0.95. It can be seen that the fusion of depth information and the modification of network structure are effective.
5. Conclusion
In recent years, the safety of manhole covers has become a growing concern. In order to better use the computer vision model to detect damaged manhole covers, this paper proposes a damaged manhole cover detection model based on YOLOv4 that integrates depth information. The main contributions are as follows:

- Fuse depth information with the picture, so that the damaged manhole cover can be detected pertinently.
- Modify the network structure of YOLOv4, replace the convolutional layer with CSP block as much as possible, and increase the accuracy of the network.
- Modify the activation function of the YOLOv4 convolutional layer from Mish to Leaky Relu, which reduces the amount of network calculations and achieves a higher speed.

Acknowledgments
This research work was supported by the Beijing University of Posts and Telecommunications. We would like to thank the anonymous reviewers for their valuable comments.

References
[1] Bochkovskiy, A., Wang C., Mark Liao, H. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. https://arxiv.org/pdf/2004.10934.
[2] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. (2020) CSPNet: A new backbone that can enhance learning capability of cnn. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop). pp. 2, 7.
[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(9):1904–1916. 2, 4, 7.
[4] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. (2018) Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768. 1, 2, 7.
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (2016) Deep residual learning for image recognition. In Proceed14 ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778. 2.
[6] Karen Simonyan and Andrew Zisserman. (2014) Very deep convolutional networks for large-scale image recognition. https://arxiv.org/pdf/1409.1556.pdf.
[7] Wei Liu, Dragomir Angelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. (2016) SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 21–37, 2, 11.
[8] Diganta Misra. (2019) Mish: A self regularized nonmonotonic neural activation function. https://arxiv.org/pdf/1908.08681.pdf.
[9] Forrest N landola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. arXiv preprint arXiv:1602.07360. 2
[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitri Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. (2017) MobileNets: Efficient convolutional neural networks for mobile vision applications. https://arxiv.org/pdf/1704.04861.pdf.
[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS). pp. 91–99.
[12] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. (2016) Deeper depth prediction with fully convolutional residual networks. In 2016 Fourth International Conference on 3D Vision (3DV), IEEE. pp. 239–248.