Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm

Yuexin Zhang * **Jie Huang**† ** **Fenghuang Cai** ***

* *College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China(e-mail: N180120079@fzu.edu.cn).*
** *College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China(e-mail: jie.huang@fzu.edu.cn).*
*** *College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China(e-mail: caifenghuang@fzu.edu.cn).*

Abstract: An improved bridge surface crack detection algorithm based on a further developed You Only Look Once version 3 algorithm (YOLO v3) is proposed to realize the fast and accurate detection of bridge surface cracks for timely repair application scenarios. The proposed algorithm is combined with MobileNets and convolutional block attention module (CBAM), which can detect bridge surface cracks in real time. The standard convolution is replaced by the depthwise separable convolution of MobileNets so as to reduce the number of network parameters. Moreover, in order to solve the problem of precision decline caused by depthwise separable convolution, the inverted residual block of MobileNetV2 is introduced. Furthermore, the proposed algorithm selectively learn the feature by multiplying the attention map with the input feature map through CBAM, and focus on channel and spatial attention mechanisms simultaneously. Finally, the feasibility of the algorithm is verified by experiment.

Keywords: bridge surface, crack detection, MobileNets, YOLO v3, inverted residual.

1. INTRODUCTION

Concrete bridge is an important node of traffic. It will inevitably face with high winds, rain and snow, earthquakes and freezing. Moreover, bridges also be affected by overloading and impacting, which may lead to various damage of bridge piers (Su et al., 2013). Among the damage of bridges, crack is difficult to be detected, which endangers the safety of the bridge (Zhang et al., 2017). Large cracks may directly destroy the integrity of bridge structure, which caused the carbonization of concrete, the peeling of protective layers and the corrosion of steel bars, and even lead to the bridge collapse accidents (Zhang et al., 2011). Therefore, effective measures are needed to monitor and prevent cracks in bridges, which play an important role in ensuring the safety and normal operation of bridge traffic. However, manual detection relies heavily on the experience of inspectors, which may lead to incorrect evaluation. Among al the crack detection techniques, visual detection is the most convenient and quick method (Adhikari et al., 2014). Computer image processing technique can recognize the crack in bridge, by automatically processing and analyzing a large number of images (Yeum et al., 2015).

In crack detection, the paper (Abdelqader et al., 2003) compared four crack detection algorithms: the fast Haar transformation, the fast Fourier transformation, the Sobel algorithm and the Canny algorithm. Among the four detection algorithms, the performance of the fast haar transformation is obviously better than others. However, all the above algorithm are difficulties to process the image data containing noise, and the detection accuracy is lower than manual detection. In the more advanced methods, the paper (Zalama et al., 2014) proposed a feature extraction method by the Gabor algorithm. The method used Adaboost algorithm to select and combine classifiers, which improve the classification accuracy of individual classifiers. Genetic Planning algorithm is applied to design image filters that eliminate the image noise towards improving crack detection. A machine Learning algorithm is used to classify image features, but the complex noise of the image greatly affects the accuracy of the algorithm detection. Therefore, a detection method which can accurately identify and locate bridge cracks needs to be improved.

Hinton, et al. presented the AlexNet (Krizhevsky et al., 2012) and won the 2012 ImageNet large scale visual recognition challenge (ILSVRC). Since then, convolutional neural network (CNN) has become the major algorithm in image classification. It has breakthroughs in the fields of image classification, target detection, image semantic segmentation and image processing. Unlike traditional neural networks, CNN is able to automatically learn appropriate feature from datasets. Object detection algorithm based on CNN can effectively overcome the difficulties of object detection in complex environment. Therefore, it gradually dominates the target detection field.

According to the different prediction process, the object detection algorithm based on deep learning mainly includes one-stage and two-stage. The two-stage object detection algorithm has two steps on the detection process.

⋆ This work was supported in part by the National Natural Science Foundation of China under Grant 61603094 (Corresponding Author: Jie Huang).

First, the region proposal algorithm is used to generates region proposals that may contain object. Then classify these region proposals and regress their positions through the convolutional neural network to obtain the final bounding boxes. The region convolutional neural network (R-CNN) is a representative of the algorithm (Girshick et al., 2014). Because R-CNN repeatedly extract features in all region proposals, resulting in low efficiency. To avoid repeatedly extract features from image and improve detection efficiency, Fast R-CNN directly generates region proposals on the feature map (Girshick, 2015). As the improved algorithm of Fast R-CNN, Faster R-CNN (Ren et al., 2015) uses the region proposal network (RPN) directly to generate region proposals, which has higher recall, higher average accuracy and faster processing speed. However, Faster R-CNN is still difficult to achieve real-time detection on devices with GPUs.

YOLO (Redmon et al., 2016) is a representative of the one-stage object detection algorithm. Unlike R-CNN, region proposal have been removed in the algorithm. The algorithm integrates the object positioning and classification into a CNN. Therefore, it only needs one forward operation to detect various objects. On the basis of YOLO, YOLO v2 (Redmon et al., 2017a) uses some optimized methods to improve average accuracy and detection rate. YOLO v3 (Redmon et al., 2017b) is the third version of YOLO, which extracts more representative features by combining shallow and deep features. The algorithm not only maintains the fast detection speed of YOLO v2, but also greatly improves the accuracy, especially in the detection of small objects.

YOLO v3 is one of the best object detection algorithms at present, and many scholars applied it to detect different objects and optimize accordingly. Although YOLO v3 has achieved good detection results in conventional dataset such as common objects in context (COCO) and visual object classes(VOC), it still has difficulties in detecting cracks on bridge surface. The structure of YOLO v3 is highly complex, which is more suitable for multi-category detection tasks. However, in this paper only needs to detect the bridge cracks, so leads to the mismatch between the magnitude of the training dataset and the complexity of the model. It's easy to appear overfitting phenomenon during the training process, resulting in accuracy decrease. Moreover, too many network parameters will make harder training and more time consumed, and the image detection time will also be longer.

Therefore, on the base of YOLO v3, this paper integrates the structure of MobileNets (Howard et al., 2017) and the CBAM (Woo et al., 2018). YOLO v3 feature extraction network was replaced by the fusion network, which became a streamlined and fast bridge crack detection network. Considering the complex structure of YOLO v3, this paper redesigns the feature extraction network and reduce the number of bounding box prediction. Standard convolution is replaced by the depthwise separable convolution of MobileNets to reduce the amount of network parameters. In order to eliminate a slight decrease in detection accuracy caused by depthwise separable convolution, this paper introduces the inverted residuals structure of MobileNetV2 (Sandler et al., 2018). Finally, CBAM is embedded into the convolutional layers, the algorithm can select information that is more critical to the object, while suppressing other useless information. Therefore, the algorithm has fewer parameter and faster detection speed. The improved algorithm can predict the location and size of the cracked area in real-time which is important to improve the efficiency of bridge detection.

The rest of paper is organized as follows: in section 2, the basic theory concept and network structure of YOLO v3 object detection algorithm is introduced; section 3 proposes a scheme to improve the YOLO v3 for bridge crack detection. The experiment and result analysis are shown in section 4; and the last section concludes this paper.

2. YOLO V3 OBJECT DETECTION ALGORITHM

The structure of residual neural network (ResNet) (He et al., 2016) is used in the feature extraction network of YOLO v3. YOLO v3 introduces multiple residual unit and uses multi-scale prediction to overcome the defects of YOLO v2 in the recognition of small object. Because of the high accuracy and detection speed, the algorithm is one of the best algorithm in the field of object detection. The algorithm uses a number of excellent 3×3 and 1×1 convolution kernel, and some residual structures are used in the later multi-scale predictions. Because the feature extraction network of YOLO v3 has 53 convolution layers, it is also known as DarkNet-53.

The structure of YOLO v3 is shown in Figure 1. The network after the 79th layer through a convolution layer to obtain the first scale of the detection box. Compared with the input image, the downsapling multiple of feature map for detection is 32. For example, if the input image is 416×416, the output feature map is 13×13. Because of the downsapling multiple is large, the receptive field of the feature map is relatively large, so it is suitable to detect the larger objects in the image. In order to detect small object, after upsampling the feature map of the 79th layer, which concatenate with the 61st layer feature map. In this way, the fine-grained feature maps can be obtained through the 91st layer. Compared with the input image, the downsapling multiple of feature map for detection is 16. The feature map has medium-scale receptive field which is suitable for detecting medium-scale objects. Finally, the output obtained by the 91th layer feature map upsampling, which concatenate with the 36th layer feature map. Compared with the input image, the downsapling multiple of feature map for detection is 8. The feature map has small receptive field which is suitable for detecting small objects.

With the change of the number and scale of the output feature map, the scale of the bounding box are adjusted accordingly. K-means clustering algorithm is used in YOLO v3 to get 9 scale of the bounding box, setting 3 bounding boxes for each downsapling scale. The 13×13 feature map has the largest receptive field, and the corresponding largest bounding box is suitable for detecting larger-sized objects. The 26×26 feature map has the medium receptive field, and the corresponding medium bounding box is suitable for detecting medium-sized objects. The 52×52 feature map has the smallest receptive field, and

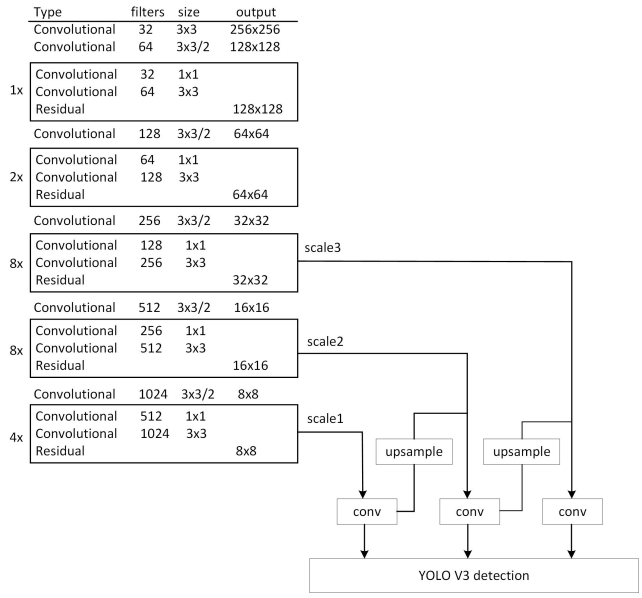

Fig. 1. YOLO v3 network structure.

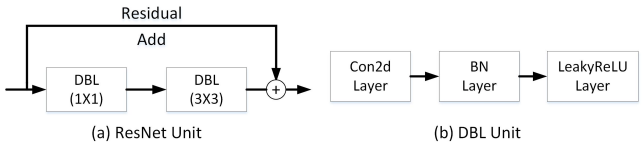

Fig. 2. Structural unit of Darknet53.

the corresponding smallest bounding box is suitable for detecting smallest-sized objects.

Darknet53 consists of five residual blocks, introducing the concept of ResNet. Each residual block consists of multiple residual units, which contain two DBL units, as shown in Figure 2(a). The DBL unit contains convolution layer (Con2d Layer), batch normalization layer (BN Layer) and leaky rectified linear layer (LeakyReLU Layer), as shown in Figure 2(b). By introducing residual units, the number of layers in the network can be significantly increased, while avoiding the gradient disappearing.

3. BRIDGE CRACK DETECTION OBJECT DETECTION ALGORITHM BASED ON IMPROVED YOLO V3

In order to solve the problem that YOLO v3 is difficult to apply to the bridge surface crack detection, this paper proposes a real-time object detection algorithm based on improved YOLO v3, and the structure of algorithm is shown in Figure 3. This paper makes three improvements to the structure of YOLO v3. The improved YOLO v3 network framework is shown in Figure 3, where the Conv is standard convolution operation, the Bottleneck is deepthwise separable convolution with inverted residual structure. The first 13×13 feature map is obtained after 5 downsapling, predicting three detection boxes. The 26×26 feature map is obtained by the feature map upsampling and concatenate with the 26×26 feature map, predicting three detection boxes.

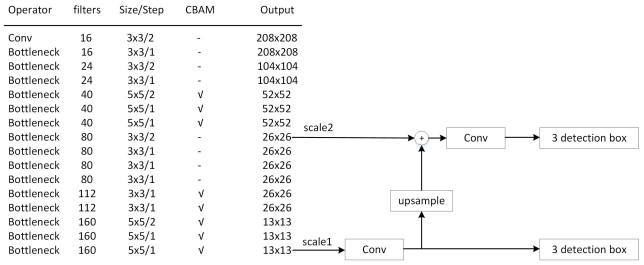

Fig. 3. Algorithm framework on the proposed improved YOLO v3

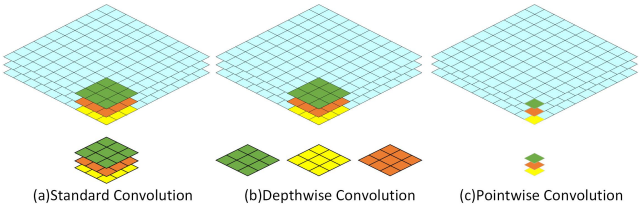

Fig. 4. Standard convolution filters and depthwise separable convolution

3.1 Depthwise separable convolution

In the design of the feature extraction network, this paper reduces the number of convolution layers to 16 layers, and reduces the output of network prediction box from 9 to 6. The depthwise separable convolution of MobileNets turns standard convolution operation into two steps. As shown in Figure 4, a standard convolution 4(a) is broken down into a depth convolution 4(b) and a 1×1 pointwise convolution 4(c). Depth convolution individually filters each input channel of input feature. Pointwise convolution uses 1×1 convolution kernel to combine the output of depth convolution. However, standard convolution is a step that combines the input feature map to produce the output. However, the standard convolution once filter the input feature map to get the output.

Suppose an input feature map F is $D_F * D_F * M$, get a feature map G of $D_G * D_G * N$ after convolution. D_F is the width and height of the input feature map, M is the number of input channels, D_G is the width and height of the output feature map, and N is the number of the output channels. The computation of the standard convolution is $D_K * D_K * M * N * D_F * D_F$. The computation of depthwise separable convolution is $D_K * D_K * M * D_F * D_F + M * N * D_F * D_F$. The ratio of the two computation is:

$$\frac{1}{N} + \frac{1}{D_K^2} \tag{1}$$

If the size of the convolution kernel is 3×3, the depthwise separable convolution is 8 to 9 times less computational than the standard convolution, but the accuracy is only slightly reduced. Therefore, Depthwise separable convolution can reduce the amount of computation and simplify model.

3.2 Linear bottleneck layer and inverted residual block

An input image is embedded into a n-dimensional space using random matrix T followed by ReLU, and then projected back to the image using T^{-1}. When n is very

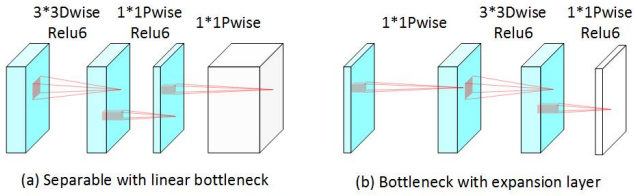

Fig. 5. Separable with linear bottleneck and bottleneck with expansion layer

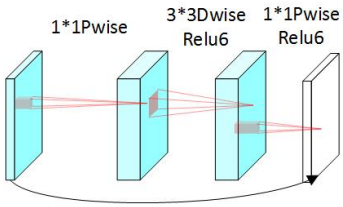

Fig. 6. Inverted residual block

small, the ReLU nonlinear transformation will loss a lot of information. If the dimension is higher, the output has more similarities with the image of input. A picture that is restored from a feature map with a high dimension, which is more similar to the input picture. Because dimension of depthwise separable convolution is smaller than the standard convolution kernel, using nonlinear layer destroys some of the information. In order to ensure the presentation ability of the model, the linear activation layer behind the small dimension output layer is removed and the linear bottleneck layer used.

Figure 5(a) is depthwise separable convolution with a linear bottleneck layer, and the input feature map obtains the feature map of the same dimension in the middle by means of a 3×3 depth convolution and a linear activation function. Then, the feature map passes through the 1×1 convolution layer and linear activation layer, getting the feature map after descending dimension. Finally, the dimensions of feature map increase by 1×1 convolution. Figure 5(b) is a relatively low-dimensional feature, by 1×1 pointwise convolution, and then 3×3 depth convolution, in which linear activation function to maintain the number of features. Then, the output is through the 1×1 depth convolution and linear activation function. Finally, the features of next layer after descending dimension is obtained.

Because the input of the bottleneck layer contains all the necessary information. To prevent the information loss, remove activation layer from the back of some layers. Enough channels can keep the information of image, so the feature map needs to be ascend dimension within the bottleneck layer. After ascending dimension, dimension information is more abundant. At the same time, descending the dimension after the activation layer, theoretically can keep all the necessary information not lost. Inverted residual blocks are improved on the basis of traditional residual blocks, as shown in Figure 6. First, use pointwise convolution to ascend dimension, and then use the 3×3 convolution and activation functions to filter the features. To obtain the output of the layer feature, the feature was descended the dimension using the 1×1 convolution and activation function. Finally, the feature map of output adds up to the low-dimensional features of the input.

3.3 Convolution block attention module

In order to select the information that is more critical to the current task objective from the many feature information and improve the efficiency and accuracy of image information processing, this paper introduces the attention mechanism in deep learning. The attention mechanism in deep learning is essentially similar to the selective visual attention mechanism of humans. It can quickly scan images to get the object areas. Then use more attention resources to this area, and quickly get high-value information from a large amount of information, suppressing other useless information. CBAM is a simple and effective attention module for convolutional neural networks. For the feature map generated by the convolutional neural network, The attention map of feature map is calculated by CBAM from channel and spatial. Then multiplies the attention map with the input feature map for adaptive learning of the feature. CBAM is more effective than the attention mechanism of SeNet (Hu et al., 2017) which only focus on channel. As a lightweight module, CBAM can be integrated into the convolution layer of object detection network for training.

For an input feature map: $F \in R^{C*H*W}$, where C, H and W are the channels, height, and width of the feature maps, respectively., CBAM gets a 1-dimensional channel attention feature map $M_C \in R^{C*H*W}$ and 2-dimensional spatial attention feature map $M_S \in R^{1*H*W}$. The overall attention process can be summarized as:

$$F^{'} = M_C(F) \otimes F \tag{2}$$

$$F^{''} = M_S(F^{'}) \otimes F^{'} \tag{3}$$

where $\otimes$ is element-by-element multiplication. $F^{'}$ is calculated by multiply the channel attention with the input feature map. $F^{''}$ is obtained by multiply spatial attention map of $F^{'}$ with the channel attention map. The calculation process of the channel attention module and the spatial attention module is as follows:

Channel attention module focuses on meaningful features in input images. As shown in Figure 7, in order to efficiently calculate channel attention, CBAM compresses feature maps on spatial dimensions using max-pooling (MaxPool) and average-pooling (AvgPool), generating two different spatial context descriptors: F^c_{max} and F^c_{avg}. Using a shared network of multi-layer perceptron (MLP) to calculate two descriptors to get channel attention features: $M_C \in R^{C*1*1}$. The calculation is as follows:

$$M_C(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{4}$$

$$M_C(F) = \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \tag{5}$$

where $W_0 \in R^{C/r*C}$, $W_1 \in R^{C*C/r}$, and Relu was used as an activation function after W_0.

The spatial attention module focuses on location information. As shown in Figure 8, two different spatial context descriptors are firstly used on the dimensions of the channel

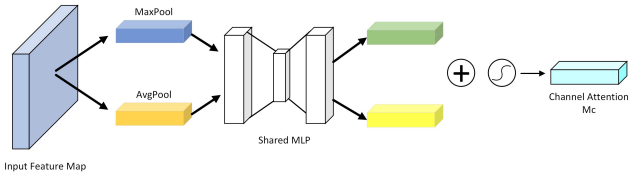

Fig. 7. Channel Attention Module

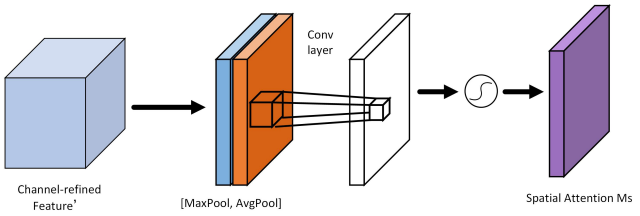

Fig. 8. Spatial Attention Module

Fig. 9. Partial sample image of bridge crack data set

using max-pooling and average-pooling $F^s_{max} \in R_{1*H*W}$, $F^s_{avg} \in R_{1*H*W}$. Then the two feature descriptors are used to merge and convolution is used to generate spatial attention feature map $M_S(F) \in R_{H*W}$. The channel attention is computed as:

$$M_S(F) = \sigma(f^{7*7}([AvgPool(F); MaxPool(F)])) \quad (6)$$

$$M_S(F) = \sigma(f^{7*7}([F^S_{avg}; F^S_{max}])) \quad (7)$$

where σ denotes the sigmoid function, r is the reduction ratio and $f^{7\times7}$ represents a convolution operation with the filter size of 7 × 7.

4. EXPERIMENT AND RESULT ANALYSIS

This paper collected 1500 pictures of bridge surface cracks and the resolution is 1024 × 1024. Used the LabelImg to label the location of all crack in the image and generate xml file corresponding to the file name of image. The file records the location of all crack in the image. In order to enhance the reliability of the data, resize all images at a resolution of 416 × 416. The dataset is divided into 3 groups, 960 for training data, 240 for valuing data and 300 for testing data. To increase data diversity, the random flips, translation, blurs, and changes brightness, contrast, and exposure are used to the training data. Some images of the dataset are shown in Figure 9.

The experimental environment for this paper is: Windows 10 operating system, CPU for Intel Core i5-8500, Python 3.6, deep learning framework for Keras, and cuda10.0 for the accelerated computing, Intel Core i5-8500 processor, NVIDIA GeForce GTX 2070 with 8GB memory, 16GB DDR4 RAM.

4.1 Network training

The k-means algorithm is used to cluster the dataset to get the bounding box priors and uses the intersection over union (IOU) instead of Euclidean distance as the standard for clustering:

$$d(box, center) = 1 - IOU(box, center) \quad (8)$$

The bounding box priors are (145, 437), (155, 434), (168, 444), (213, 212), (218, 223), (228, 231), (230, 219), (240, 238), (458, 150). In this paper, two prediction box scales of the network are designed, each of which is assigned three bounding box priors for training.

Training in the improved YOLO v3 and YOLO v3, the initial learning rate of the weight is set to 0.001. The loss of the model is monitored by TheuseLROnPlateau, a callback function of TensorFlow. If the valuing data does not decline for 10 epoch, the model automatically reduces the learning rate to one-tenth. Because the structure of deep learning network is complex, it is easy over-fitting when the accuracy of the training data is improved and the accuracy of the valuing data is reduced during training. In order to obtain the best generalization ability, the early stop method is introduced during training. When the accuracy of the model on the valuing data begins to drop, stop training to avoid over-fitting.

4.2 Testing and analysis

300 bridge crack images were used to test the performance of the algorithm in this paper. Do the same test on YOLO v3 and improved YOLO v3. The results are shown in Table 1. The IOU threshold is the limit of IOU in prediction box, that 0.1 is no more than 10%. The network weight is the size of the parameters after the network is trained. Precision is the number of correctly predicted divided by all object detected. Recall is the number of correctly predicted divided by the objects number of testing data. Detection speed is the number of detected images in one second.

Table 1 shows that under the same test conditions, the *Precision* of the improved YOLO v3 was 89.16%, 3.55% lower than YOLO v3. The *Recall* is 91.16%, 2.25% higher than YOLO v3. The detection speed is 20.56 fps, 6.04 fps higher than YOLO v3. However, compared with the 235M weight of YOLO v3, the improved network weight is only 11.1M. It can reduce the convergence time of the network. Four experiment results of the proposed algorithm are shown in Figure 10. The part of the red box in the figure represents the area of crack. According to the figure, all the cracks in the picture can be detected accurately by the proposed algorithm.

5. CONCLUSION

In this paper, the improved YOLO v3 algorithm is proposed for the bridge surface crack detection. The number of convolution layers of feature extraction network is 16 layers, and the number of output prediction box is 6 on two scales. Such design can reduces network complexity for crack detection. Using depthwise separable convolution

Table 1. Comparison of detection efficiency of the standard YOLO v3 and improved YOLO v3

Network	IOU threshold	Weights of network	Precision	Recall	Detection speed
Standard YOLO v3	0.1	235M	92.17%	88.91%	14.52 FPS
Improved YOLO v3	0.1	11.1M	89.16%	91.16%	20.56 FPS

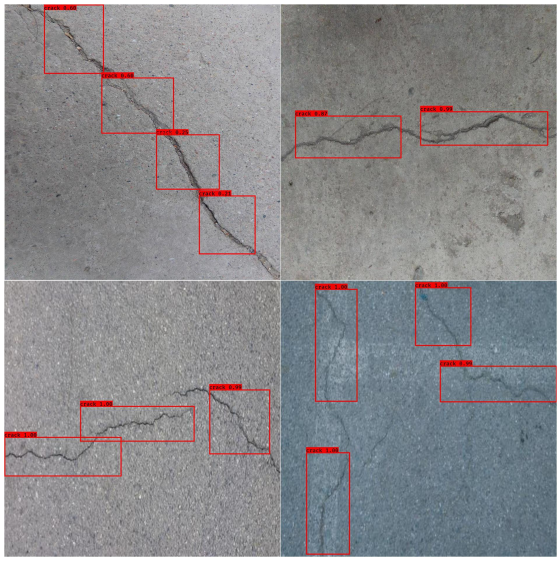

Fig. 10. Four experiment results of the proposed algorithm

instead of standard convolution can reduce the network parameters and obtain the lightweight network. Moreover, the inverted residual block was used to improve the prediction accuracy of the network. Finally, the CBAM attention mechanism is introduced to improve the efficiency and accuracy of crack detection. Experiment results show that the improved YOLO v3 has achieved excellent detection results in the bridge surface crack, and accuracy and detection speed are better than YOLO v3.

In the future work, we'll execute the network on unmanned aerial vehicle (UAV). Cracks in bridges can be detected through bridge images acquired by UAV. The main difficulty is how to run a deep neural network by the embedded device without GPU. Thus, the network structure needs to be further improved.

REFERENCES

Abdelqader, I., Abudayyeh, O., and Kelly, M. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, 17(4), 255–263.

Adhikari, R.S., Moselhi, O., and Bagchi, A. (2014). Image-based retrieval of concrete crack properties for bridge inspection. *Automation in construction*, 39(39), 180–194.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE international conference on computer vision*, 1440-1448.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets:efficient convolutional neural networks for mobile vision applications. *Proceedings of the IEEE conference on computer vision and pattern recognition*, arXiv:1704.04861.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2017). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-1.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, 1097-1105.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.

Redmon, J., and Farhadi, A. (2017a). YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6517-6525.

Redmon, J., and Farhadi, A. (2017b). YOLOv3:An incremental improvement. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6517-6525.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *In Advances in neural information processing systems*, 1137-1149.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). MobileNetV2: inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510-4520.

Su, T. (2013). Application of computer vision to crack detection of concrete structure. *International Journal of Engineering and Technology*, 5(4), 457–461.

Woo, S., Park, J., Lee, J.Y., and Kweon, I.S. (2018). CBAM: convolutional block attention module. *Proceedings of the European Conference on Computer Vision (ECCV)*, 3-19.

Yeum, C.M., and Dyke, S.J. (2015). Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, 30(10), 759–770.

Zalama, E., Gomezgarciabermejo, J., Medina, R., and Llamas, J. (2014). Road crack detection using visual features extracted by Gabor filters. *Computer-Aided Civil and Infrastructure Engineering*, 29(5), 342–358.

Zhang, H., Tan, J., Liu, L., Wu, Q.M.J., Wang, Y., and Jie, L. (2017). Automatic crack inspection for concrete bridge bottom surfaces based on machine vision. *2017 Chinese Automation Congress (CAC)*, 4938-4943.

Zhang, J., and Cheng, L. (2011). Investigation of combination characteristic of damage location and intensity in arch using wavelet coefficient. *2011 Second International Conference on Mechanic Automation and Control Engineering*, 370-373.