Road Damage Detection and Classification based on Multi-level Feature Pyramids

Junru Yin, Jiantao Qu, Wei Huang, and Qiqiang Chen*
College of Computer and Communication Engineering, Zhengzhou University Of Light Industry
Zhengzhou, Henan, 450001, P. R. China
[e-mail: yinjr@zzuli.edu.cn, qujiantao@gmail.com, hnhw235@163.com, chenqq@zzuli.edu.cn]
*Corresponding author: Qiqiang Chen

Received December 6, 2020; revised February 9, 2021; accepted February 16, 2021; published February 28, 2021

Abstract

Road damage detection is important for road maintenance. With the development of deep learning, more and more road damage detection methods have been proposed, such as Fast R-CNN, Faster R-CNN, Mask R-CNN and RetinaNet. However, because shallow and deep layers cannot be extracted at the same time, the existing methods do not perform well in detecting objects with fewer samples. In addition, these methods cannot obtain a highly accurate detecting bounding box. This paper presents a Multi-level Feature Pyramids method based on M2det. Because the feature layer has multi-scale and multi-level architecture, the feature layer containing more information and obvious features can be extracted. Moreover, an attention mechanism is used to improve the accuracy of local boundary boxes in the dataset. Experimental results show that the proposed method is better than the current state-of-the-art methods.

Keywords: Multi-level Feature Pyramids, Road Damage Detection, VGG16, Multi-scale, Multi-level

1. Introduction

Road damage has always been a critical problem; because road conditions are closely related to pedestrian/vehicle safety and economic development, road damage detection is essential. Many different automatic methods have been proposed to replace manual detection of road damage. Generally, these automatic methods use a variety of images with GPS information captured by vehicle-mounted smartphones. They can also be used to handle computer vision tasks (object detection [1], image enhancement [2], image classification [3], etc.) which have been addressed by a deep convolutional neural network (DCNN) [4-6]. These methods based on deep learning can perform well in road damage detection.

Laha Ale et al. [7] proposed the one-stage detection method based on RetinaNet in road damage detection. The head of a RetinaNet network is divided into two paths, one for classification prediction and the other one for bounding box prediction. In addition, RetinaNet can use different backbone networks to learn feature maps, which are the input to the Feature Pyramid Networks (FPN) [8]. They trained and compared RetinaNet models with different backbones, including DenseNet, ResNet, VGG, and InceptionResNetV2. The experimental results demonstrate that the RetinaNet approach can detect road damage with high accuracy.

Seungbo Shim et al. [9] proposed a road damage detection method based on Fast R-CNN, which is an extension of R-CNN. First, Selective Search (SS) is used to select multiple high-quality Region Proposals for each input image. Then the image is input into the convolutional neural network to extract features. The proposed regions are mapped to the last layer of the convolutional neural network, then the ROI (Region of Interest) Pooling layer is used to extract the same size output for each proposed region. Finally, Fast R-CNN uses Softmax for classification prediction, whereas R-CNN uses Support Vector Machines (SVM). Seungbo et al. observed excellent performance when using Fast R-CNN in road damage detection. Another famous object detection is YOLO which include YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, different versions have different architectures and techniques. Alfarrarjeh et al. [10] use YOLOv3 and fine-tuned the darknet53 module, they use two augmentation strategies, first is brightening or gray-scale to augment the lower number damage types, the second is to use cropping.

Building on the foundation of Fast R-CNN, Wang et al. [11] proposed using Faster R-CNN for road damage detection. Faster R-CNN extracts the feature map of the image using a background network, then Region Proposal Network (RPN) is used to generate proposal regions The ROI Pooling layer collects the input feature map and proposal regions, mapping the proposal regions to the feature map and pooling them into a uniform size region feature map, which is then sent to the full connection layer to determine the target class. Finally, the region feature map is used to calculate the class of the candidate regions, while the bounding box regression is used again to obtain the exact final location of the detection box. Wang et al. also expanded the data set before training and found that this method can achieve good performance in road damage detection.

To improve upon Faster R-CNN, Singh et al. [12] proposed using Mask R-CNN for road damage detection. Mask R-CNN takes a set of images as input, extracts them through a convolution network to create the feature map, and then uses RPN to predict the proposal region. Mask R-CNN differs from Faster R-CNN by using ROI Align instead of ROI Pooling. Finally, it generates a region proposal which is divided into three branches: box, class and mask, and the final result is obtained by a series of convolution operations. This method can achieve a Mean F1-score of 0.528 after training in a big data set.

Of these methods above, two-stage detectors, such as Fast R-CNN and Faster R-CNN, can

achieve high accuracy but are computationally inefficient. One-stage detectors, such as RetinaNet, YOLO, are more efficient but cannot achieve such high accuracy, and perform poorly in detecting objects from fewer samples. To achieve greater accuracy and efficiency, we proposed a Multi-level Feature Pyramids method which is based on M2det [13]. In addition, an attention mechanism is used to improve the accuracy of local detecting bounding boxes in the dataset. We also analyzed each layer of the backbone network and finally chose the last two layers of VGG16 for fusion. The proposed method results in an mAP of 63.96. In summary, the major contributions of this paper are at least threefold.

(1) We propose a Multi-level Feature Pyramids method based on M2det which can extract more information and obvious features since it has multi-scale and multi-level architecture.
(2) We use a Multi-level Feature Pyramids method in road damage detection and classification, it can effectively detect and classify damage of road images.
(3) An attention mechanism is used in the Multi-level Feature Pyramid to make the detection result bounding box more accurate, at the same time, the method get a better performance than other methods in experiment.

The remainder of this paper is organized as follows: Section 2 describes the proposed method. In section 3, the experiments and results are detailed. Conclusions are presented in section 4.

2. Proposed Method

The proposed Multi-level Feature Pyramids is composed of a backbone network and a Multi-level Feature Pyramid Network (MLFPN). It extracts two layers of features from backbone network and fuses them into the base feature. In MLFPN, there are some Thinned U-shape Modules (TUM), the first TUM processes the base feature, then other TUMs process the output of the previous TUM and the base feature. These features generated by TUM are added to the attention mechanism in Scale Feature Aggregation Module (SFAM). Finally, SFAM generates six valid feature layers used for prediction. The proposed method uses Multi-level Feature Pyramids in road damage detection; with multi-level and multi-scale architecture, it can extract the feature layer which contains more information and obvious features. The attention mechanism is used in SFAM to make the detection more accurate. The Multi-level Feature Pyramids architecture for road damage detection is shown in **Fig. 1**.

We chose VGG16 as the backbone network, and all the fully connected layers are removed while maintaining the convolutional layer and the maximum pooling layer. There are two different FFMs (Feature Fusion Modules) in the Multi-level Feature Pyramids. FFM1 takes out the last two layers of the VGG16 for preliminary fusion, while FFM2 is used for feature-enhanced fusion. TUM has an architecture similar to the feature pyramid, it obtains six valid feature layers by compressing the feature layer and up-sampling for feature fusion. SFAM adjusts the attention of different channels and determines their weight. Finally, the valid feature layers with the attention mechanism are used to predict the detecting result.

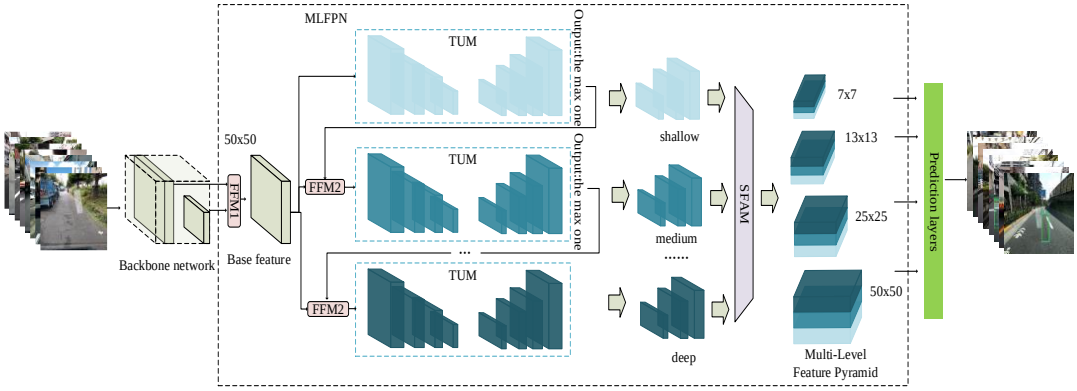


Fig. 1. Multi-level Feature Pyramid architecture for road detection and classification

In this section, we describe the Multi-level Feature Pyramid in detail. Specifically, subsection 2.1 introduces the backbone network, subsection 2.2 introduces the MLFPN, subsection 2.3 introduces the detail of FFM, subsection 2.4 introduces the architecture of TUM, and subsection 2.5 introduces the architecture of SFAM. Additionally, subsection 2.6 introduces the loss function.

2.1 Backbone Network

VGG was proposed by the Visual Geometry Group in Oxford [14]. There are two types of VGG structures, VGG16 and VGG19. VGG19 has three more convolutional layers than VGG16, in detail, VGG19 has one more convolutional layer before pooling layer in each of the 3, 4 and 5 layer than VGG16. VGG16 was selected as the backbone network because too many convolutional layers may lead to gradient disappearance, gradient explosion, or even overfitting and falling into local optimum. For the road damage target we detect, the number of convolution layers of VGG16 is relatively more reasonable. In addition, the experiment show that the mAP of VGG16 is higher than VGG19 . In this paper, all the fully connected layers of VGG16 were removed while the convolutional layer and the maximum pooling were retained. The architecture of VGG16 is shown in **Fig. 2**.

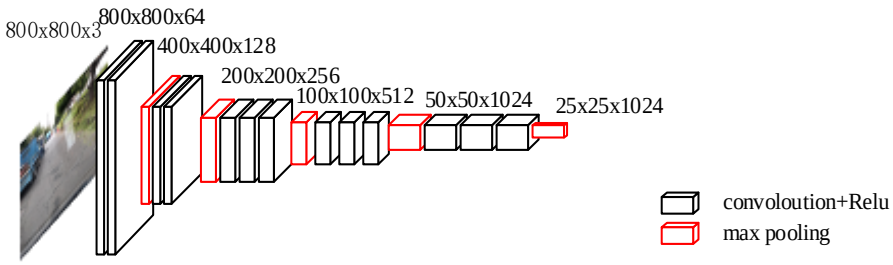


Fig. 2. The architecture of VGG16

We can see the architecture of each layer in the figure directly. The input images were resized to (800, 800, 3), the resized images are processed by a series of convolution and pooling which is divided into five layers. The output of the fourth and fifth layer are fused in FFM1.

2.2 MLFPN

As shown in **Fig. 1**, the MLFPN is composed of four parts: FFM1, FFM2, TUM and SFAM. There are a total of 8 TUMs in the MLFPN. The first TUM learns from X_{base}, the other TUMs take two inputs, X_{base} and the output of the previous TUM. The output of multi-level and multi-scale features are described as shown in (1).

$$X_l = \begin{cases} T_l\left(X_{base}\right), l = 1 \\ T_l\left(F\left(X_{base}, X_{l-1}\right)\right), l = 2...L \end{cases} \tag{1}$$

where X_{base} denotes base features, X_l represents the valid feature layer with the max scale of the $l-th$ TUM, L indicates the number of TUMs, T_l denotes the $l-th$ TUM processing, and F represents the FFM2 processing. Finally, SFAM combines feature maps of the same scale of each TUM to form the feature pyramid.

2.3 FFM

There are two different FFMs in Multi-level Feature Pyramids; FFM1 is used for preliminary fusion and FFM2 is used for feature-enhanced fusion. The details of FFMs are shown in **Fig. 3**.

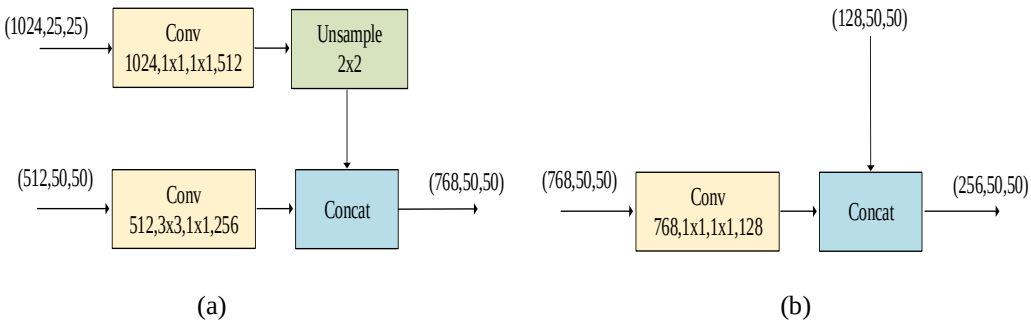


Fig. 3. FFM detail, (a) represents FFM1, (b) represents FFM2

In FFM1, the outputs of the last two layers from VGG16 are fused. Specifically, the (25, 25, 1024) feature layer is convoluted and up-sampled to change the shape to (50, 50, 512), and the (50, 50, 512) feature layer is convoluted to change the shape to (50, 50, 256). Then these two results are stacked to a (50, 50, 768) preliminary fused feature layer. This fused feature layer is called the base feature. In FFM2, the (50, 50, 128) feature layer of the six valid feature layers from the TUM is fused with the initial fused base feature from FFM1. Finally, it outputs a (50, 50, 256) enhanced fused feature layer. This feature layer can be passed into the TUM as input again.

2.4 TUM

TUM uses a thinner U-shaped network which is same as the feature pyramid, it contains multi-level and multi-scale features. The architecture of TUM is shown in **Fig. 4**.

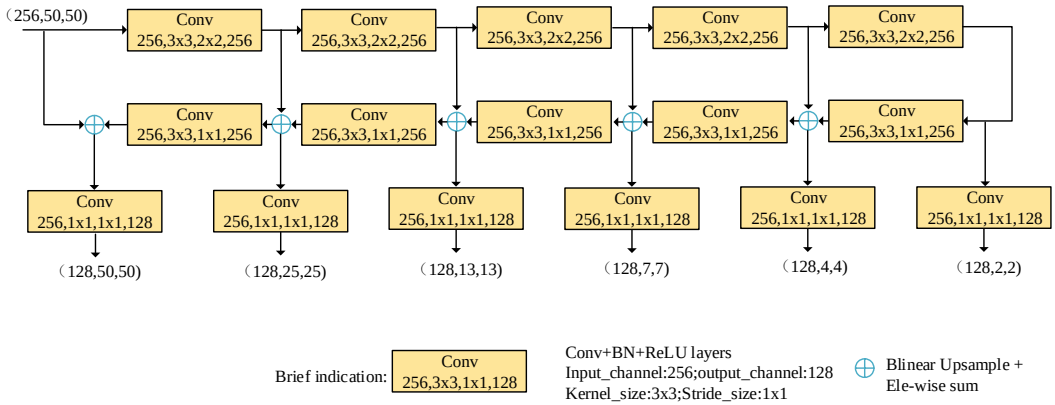


Fig. 4. The architecture of TUM

In TUM, the encoder is made up of a series of 3×3 convolutional layers with a stride of 2. The decoder uses the output of these layers as reference features. In addition, after up-sampling and element-wise sum operation, we added 1×1 convolution layers at the decoder branch to enhance learning ability and keep smoothness for the features. All of the outputs form the multi-scale features of the current level in the decoder of each TUM. The outputs of stacked TUMs form the multi-level and multi-scale features; shallow-level features are provided by the front TUM, medium-level features are provided by the middle TUM, and deep-level features are provided by the back TUM.

2.5 SFAM

The purpose of SFAM is to aggregate the multi-level and multi-scale features generated by TUMs into a multi-level feature pyramid. The architecture of SFAM is shown in **Fig. 5**.

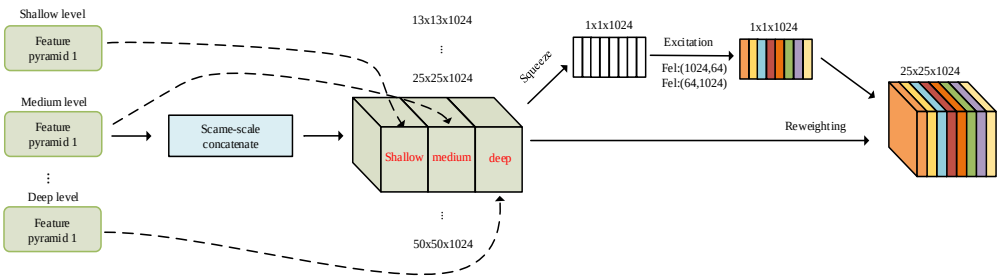


Fig. 5. SFAM architecture

SFAM uses the attention mechanism, which makes the detection result bounding box more accurate. The attention mechanism is inspired by the physiological perception of the environment. The human visual system has a tendency to select specific information from an image, focusing on those parts and ignoring extraneous information. Hu et al. [15] presented a very significant structure of the SENet model. The core of SENet is an attention mechanism module which is divided into three parts: squeeze, excitation and attention. The channel attention mechanism is based on the importance of each feature. For different tasks, features can be simply and effectively assigned based on input. SFAM aims at aggregate the multi-level multi-scale features generated by TUMs into a multi-level feature pyramid. The

first step is to overlay the channel of same scale features generated by different TUMs, the overlapping channels can be expressed as (2).

$$Y = \left[Y_1, Y_2, \ldots, Y_i, \ldots\right] \tag{2}$$

Of these, Y represent the feature map of different scales, it can be represented as (3).

$$Y_i = Concat\left(y_i^1, y_i^2, \ldots, y_i^l\right) \in R^{W_i \times H_i \times C} \tag{3}$$

In detail, i represent the $i-th$ scale and l represent the $l-th$ layer, each scale of the aggregated feature pyramid contains the same scale of features from different layers.

$$s = \sigma\left(W_2 \delta\left(W_2 z\right)\right) \tag{4}$$

$$\tilde{Y}_i^c = s_c \cdot Y_i^c \tag{5}$$

In (4) - (5), σ represent the Rectified Linear Unit, δ represent the Sigmoid function. The second step of SFAM is to introduce the attention mechanism based channel domain to focus on the channel that is most helpful for detection. The channel information z is generated by global pooling during the squeezing phase, in order to capture channel dependencies fully, the following step learns the attention mechanism through two fully connected layers. Finally, the obtained weights s with the attention mechanism are multiplied with the channels in the input Y to generate the final output.

2.6 Loss Function

Given a training data set to train the proposed model, we use a multitasking loss function which contains classification and regression. In order to classify different road damage types, we define the classification loss function as L_{cls}, as shown in (6).

$$L_{cls}\left(p_j, p_j^*\right) = -\log p_j p_j^* \tag{6}$$

In addition, we define the regression loss function as L_{reg} in order to make the positioning of the detection bounding box closer to the real bounding box, as shown in (7).

$$L_{reg}\left(t_j, t_j^*\right) = smoothL1\left(t_j - t_j^*\right) \tag{7}$$

Finally, the total loss is given by L_{cls} and L_{reg}, as shown in (8).

$$L\left(\left\{p_j\right\}, \left\{t_j\right\}\right) = \frac{1}{N_{cls}} \sum_j L_{cls}\left(p_j, p_j^*\right) + \lambda \frac{1}{N_{reg}} \sum_j p_j^* L_{reg}\left(t_j, t_j^*\right) \tag{8}$$

where j represents the index of an anchor in a mini-batch, p_j indicates the predicted probability of anchor being an object region. The ground-truth label is denoted by p_j^*, 1 for positive and 0 for negative. Girshick [16] defined the predicted box t_j and the ground-truth box t_j^*. In (8), $L_{cls}\left(p_j, p_j^*\right)$ and $p_j^* L_{reg}\left(t_j, t_j^*\right)$ are normalized with N_{cls}, N_{reg}, and a balancing weight λ, respectively. The function of $smoothL1$ is defined as shown in (9).

$$smoothL1(x) = \begin{cases} 0.5x^2, if\ |x| < 1 \\ |x| - 0.5, others \end{cases} \tag{9}$$

The proposed method algorithm is shown in **Algorithm 1**.

Algorithm 1: Road damage detection and classification based on Multi-level Feature Pyramids

Input: X, X represents road image

Step 1: VGG16 creates multi-layer feature of input image,

$F = \{F_1, F_2, F_3, F_4, F_5\}$.

F_j represents the layer of VGG16, where j represents the $j-th$ layer.

Step 2: Fuse the part of multi-layer features by FFM1 to generate the base feature,

$Base = FFM1(F_4, F_5)$.

Step 3: Use TUMs to obtain the multi-scale feature. Input the base feature into the first TUM, in TUMs, the feature layer of input is compressed constantly and then upsampled continuously to obtain six valid feature layers with different scale feature. Of these, fuse the largest valid feature layer of this six and the base feature, then input the fused feature into the next TUM. repeat the same operation from the second TUM to eighth TUM.

Step 4: SFAM adds the attention mechanism to the six valid feature layers generated by TUMs.

Step 5: For these valid feature layers with attention mechanism, they are convoluted twice, The first convolution predict the variation of each prior bounding box at each grid point of the feature layer. The second convolution predict the type of each prediction bounding box at each grid point of the feature layer.

Output: Output the road damage bounding box and the type of damage according to these prior bounding box.

3. Experiments and Result

In this section, we show the details of the implementations and design of the ablation experiments. We also use the same dataset in different methods and compared the results. Section 3.1 introduces the implementation details, 3.2 introduces the datasets, 3.3 introduces the evaluation criteria, 3.4 introduces the ablation experiments, and 3.5 introduces the AP and

mAP of the detecting result.

3.1 Implementations Details

Our experiment was run with Keras v2.1.5, the hardware environment was CUDA 10.0.13 and cuDNN 7.4.1. The GPU used was NVIDIA GeForce RTX 2070 on Windows 10 with 16GB memory.

In our method, the original images are resized to (800, 800, 3) and the resized images are input to the backbone network. In VGG16, we removed all fully connected layers. We extract the fourth and fifth layers of VGG16 and fuse them to form a base feature. This base feature is used in the MLFPN. In the MLFPN, TUM extracts the features in a U-shape feature compression and then up-sample for feature fusion in the TUM. We can obtain six valid feature layers using TUM. We then take out the (50, 50, 128) feature layer from the six valid feature layers and fuse it with the base feature extracted by FFM1, and output a (50, 50, 256) enhanced fused feature layer. At the same time, the enhanced fused feature layer from FFM2 can be passed into the TUM again for U-shaped feature extraction. The six feature layers from the TUM are attached to an attention mechanism to adjust the layer weight. Finally, these valid feature layers and attention mechanism are used to predict the result.

3.2 Datasets

We use 7240 images of damaged road as the datasets; as introduced in Microsoft coco [17], all images were photographed by a vehicle-mounted smartphone. There are nine types of damage labeled in the datasets. In **Fig. 6**, we show some examples of the eight common damage types. **Table 1** shows the road damage types in our data set and their definitions.

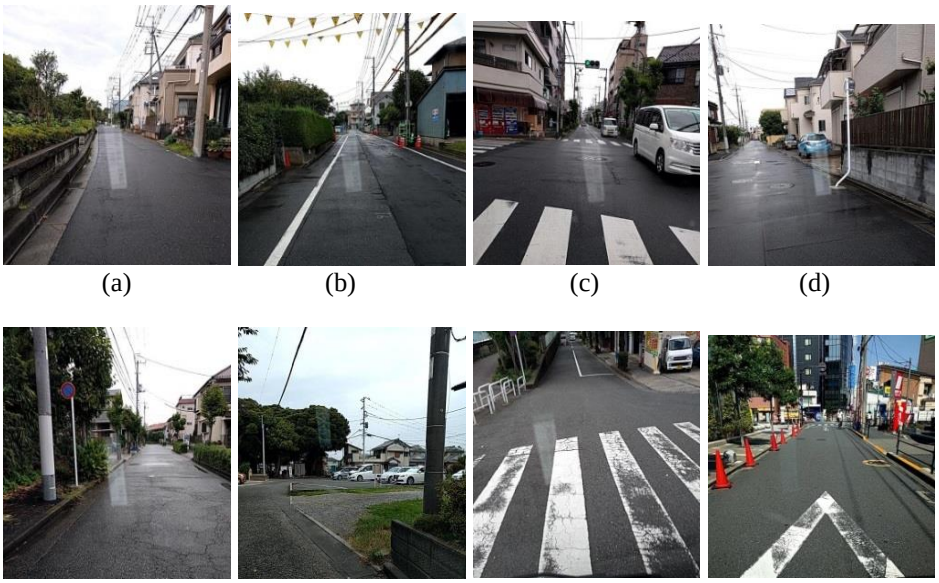


Fig. 6. Examples of each damage type in the dataset. (a) D00, (b) D01, (c) D10, (d) D11, (e) D20, (f) D40, (g) D43, and (h) D44

Table 1. Road damage types in our data set and their definitions

Damage type			Class name	Detail
Crack	Linear crack	Longitudinal	D00	Wheel mark part
			D01	Construction joint part
		Lateral	D10	Equal interval
			D11	Construction joint part
	Alligator crack		D20	Partial pavement, overall pavement
Other corruption			D40	Rutting, bump, pothole, separation
			D43	Crosswalk blur
			D44	White line blur

The datasets are divided into training set and testing set in a ratio of 7:3, that means 5068 of these images are used for training and the other 2172 images are used for testing. In addition, we take 10% of the training set as the validation set.

3.3 Evaluation Criteria

We chose mAP (mean Average Precision) as a measure of detection accuracy in object detection. As shown in (10)-(12).

$$mAP = \frac{\sum_{i=1}^{n} (AP)_i}{n} \tag{10}$$

$$(AP)_i = \frac{\sum_{j=1}^{m} \left(\sum_{i=1}^{n} P_i^j \right)}{m} \tag{11}$$

$$P_i = \frac{T_i}{S_i} \tag{12}$$

where $(AP)_i$ refers to the average precision of one road damage type, P_i^j refers to the precision of the $i-th$ category in the $j-th$ image, n refers to the number of categories and m refers to the number of images. In (12), T_i refers to the number of correctly detected objects in one category in a picture, and S_i refer to the total number of categories in a picture.

3.4 Ablation Experiments

We ran a number of ablations to analyze Multi-level Feature Pyramids. The results are shown in **Table 2** and discussed in detail below.

Table 2. The value of AP and mAP on Multi-level Feature Pyramids with different fusion layers of VGG16

Layer of VGG16	AP (%)								mAP (%)
	D00	D01	D10	D11	D20	D40	D43	D44	
VGG:4,5	75.44	88.88	26.8	9.24	82.15	63.66	91.78	90.08	**63.96**
VGG:3,5	41.24	64.83	15.02	5.2	59.47	8.5	69.93	67.14	41.41
VGG:3,4,5	39.19	63.19	16.33	4.24	52.23	1.01	62.38	62.84	37.68
VGG:1-5	11.45	43.78	6.38	1.14	44.19	0.98	59.35	49.57	27.11

Since shallow feature layers contain rich information but the features are not obvious enough, and deep feature layers are more obvious but contain too little information, we selected multiple layers to fuse. In the ablation experiments, we selected combinations of layers 4 and 5; layers 3 and 5; layers 3, 4 and 5, and layers 1-5 of VGG16. Using the same dataset and the same environment, our experiments demonstrate that the first method, with layers 4 and 5 of VGG16, achieved the best performance.

The proposed method yields excellent detection results after training with a large dataset. The result of the detection shows the damage labeled with a bounding box in the image. Detecting results of eight damage types using the proposed method are shown in **Fig. 7**.

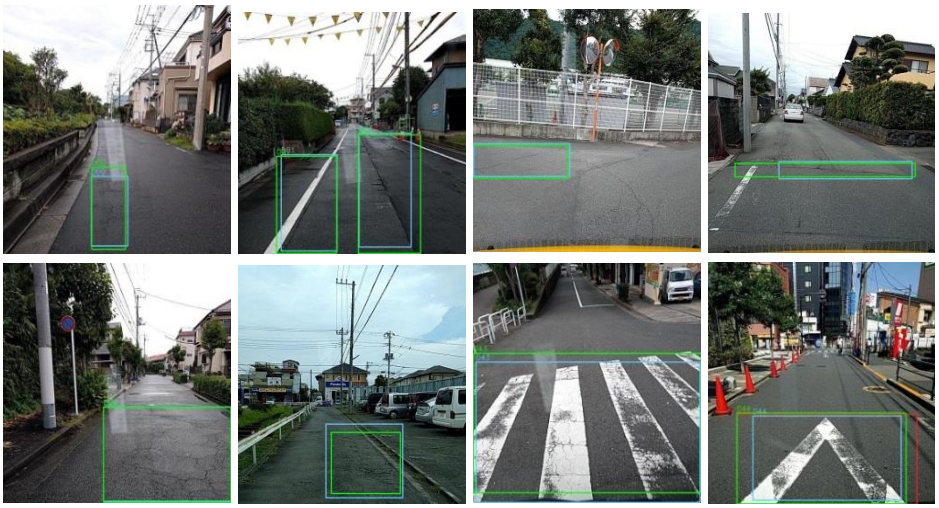


Fig. 7. The result of detection; detecting result (blue) and ground truth (green)

3.5 AP and mAP of Detecting Result

We used Multi-level Feature Pyramids, RetinaNet and YOLOv3 [18] trained on the same dataset. We then compared AP values and mAP values obtained from different methods, as presented in **Fig. 8** and **Fig. 9**.

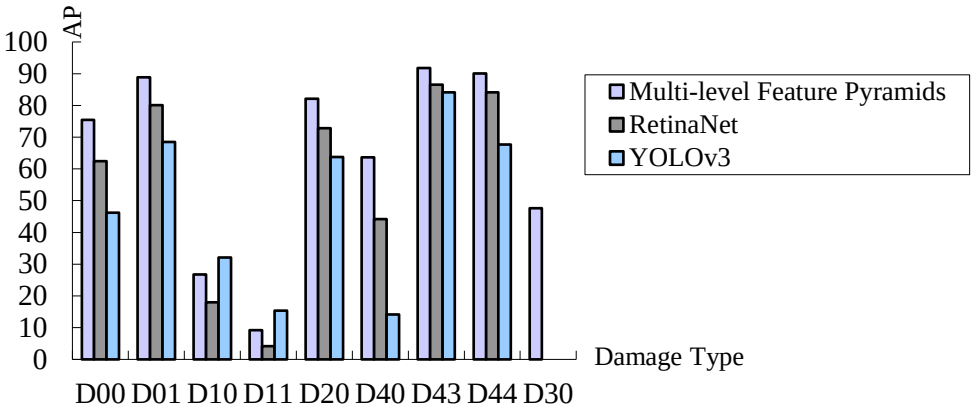


Fig. 8. AP value of different Damage Types

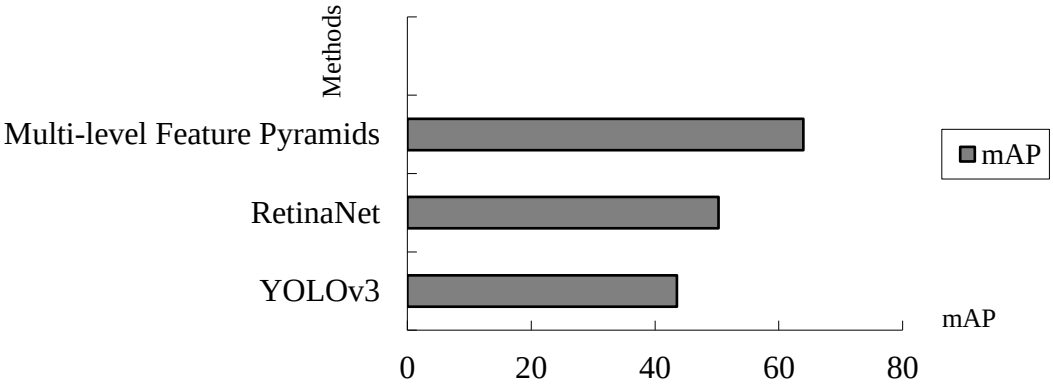


Fig. 9. mAP value of different methods

In **Fig. 8** and **Fig. 9**, the "D43" and "D44" cracks have the highest AP value. The "D01" and "D20" cracks also demonstrate good performance. We believe this is because these four types of crack have more samples in the training dataset. Although "D30" had few samples, Multi-level Feature Pyramids detected "D30" cracks with a 47.62% AP value, while RetinaNet and YOLOv3 could not detect "D30" cracks. Our method can detect types with few samples due to its multi-level and multi-scale structure. In addition, the attention mechanism makes the detection more accurate, allowing our method to achieve the highest mAP value.

4. Conclusion

In this paper, we proposed a method using Multi-level Feature Pyramids in road damage detection. We selected 70% of the dataset for training and the remaining 30% for testing. In addition, we took 10% of the training set as the validation set. To achieve the better results, we performed ablation experiments to evaluate which layer of VGG16 to choose. The ablation experiments demonstrated that layers 4 and 5 of VGG16 achieve the best performance. We

evaluated the detecting results by comparing the AP and mAP values obtained from three different methods. Our method achieved the mAP of 63.96, which is the highest value among the evaluated methods. We also compare the images of the detecting result, which demonstrate that our method can also achieve excellent performance in small target detecting. However, the proposed method is still not fast enough in detection, it is not suitable for real-time tasks. In future work we will simplify the network framework to improve the detection speed.

References

[1] Y. Hou, W. Zhu, and E. Wang, "Hyperspectral mineral target detection based on density peak," *Intelligent Automation and Soft Computing*, vol. 25, no. 4, pp. 805-814, 2019. Article (CrossRef Link)

[2] L. Sun, C. He, Y. Zheng, and S. Tang, "SLRL4D: joint restoration of subspace low-rank learning and non-local 4-D transform filtering for hyperspectral image," *Remote Sensing*, vol. 12, no. 8, pp. 1-26, Sep. 2020. Article (CrossRef Link)

[3] Q. Ye, H. Zhao, Z. Li, X. Yang, S. Gao, T. Yin, and N. Ye, "L1-norm distance minimization based fast robust twin support vector k-plane clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4494-4503, 2018. Article (CrossRef Link)

[4] Y. Yang, J. Q. M. Wu, X. Feng, and T. Akilan, "Recomputation of dense layers for the performance improvement of DCNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2912-2925, 2020. Article (CrossRef Link)

[5] Y. Guo, C. Li, and Q. Liu, "R2N: A novel deep learning architecture for rain removal from single image," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 829-843, 2019. Article (CrossRef Link)

[6] H. Wu, Q. Liu, and X. Liu, "A review on deep learning approaches to image classification and object segmentation," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 575-597, 2019. Article (CrossRef Link)

[7] L. Ale, N. Zhang, and L. Li, "Road Damage Detection Using RetinaNet," in *Proc. of IEEE International Conference on Big Data (Big Data)*, 2018. Article (CrossRef Link)

[8] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 2117-2125, 2016. Article (CrossRef Link)

[9] S. Shim, C. Chun, and S. K. Ryu, "Road Surface Damage Detection based on Object Recognition using Fast R-CNN," *Journal of The Korea Institute of Intelligent Transport Systems*, vol. 18, no. 2, pp. 104-113, 2019. Article (CrossRef Link)

[10] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, "A deep learning approach for road damage detection from smartphone images," in *Proc. of IEEE International Conference on Big Data*, pp. 5201-5204, 2018. Article (CrossRef Link)

[11] W. Wang, B. Wu, S. Yang, and Z. Wang, "Road damage detection and classification with Faster R-CNN," in *Proc. of IEEE International Conference on Big Data*, pp. 5220-5223, 2018. Article (CrossRef Link)

[12] J. Singh and S. Shekhar, "Road damage detection and classification in smartphone captured images using mask r-cnn," *arXiv preprint arXiv:1811.04535*, 2018. Article (CrossRef Link)

[13] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 9259-9266, 2019. Article (CrossRef Link)

[14] Y. Pan, G. Zhang, and L. Zhang, "A spatial-channel hierarchical deep learning network for pixel-level automated crack detection," *Automation in Construction*, vol. 119, 2020. Article (CrossRef Link)

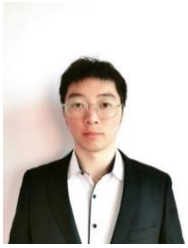
[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132-7141, 2018. Article (CrossRef Link)

[16] R. Girshick, "Fast R-CNN," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015. Article (CrossRef Link)

[17] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. of European Conference on Computer Vision*, pp. 740-755, 2014. Article (CrossRef Link)

[18] J. Choi, D. Chun, H. Kim, and H. J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. of IEEE International Conference on Computer Vision*, pp. 502-511, 2019. Article (CrossRef Link)

Junru Yin was born in Henan. China, in 1986. She received the Ph.D degree in forest management from Chinese Academy of Forestry, Beijing, China, in 2015.Since 2015, she has been with Zhengzhou University of Light Industry, Zhengzhou, China, where she is currently a lecture. Her current research interests include hyperspectral classification, machine learning and forest management.



Jiantao Qu was born in Henan, China, in 1994. He received the B.S. degree in physics from Zhengzhou University of Light Industry. Zhengzhou, China, in 2020. He is currently working toward the master's degree with School of Computer and Communication Engineering, Zhengzhou University of Light Industry. His research interests covers image classification, pan-sharpening, and deep learning.

Wei Huang was born in Henan. China, in 1982. He received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2015. He is currently a Lectuer with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His current research interests include pan-sharpening, hyperspecttal classification, image processing, and machine learning.

Qiqiang Chen was born in Henan. China, in 1984. He received the Ph.D. degree in radio physics from Lanzhou University, Lanzhou, China, in 2015. He is currently a Lectuer at Zhengzhou University of Light Industry, Zhengzhou, China. His current research interests include hyperspectral classification, image processing and machine learning.