Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 10, No. 3, 2023

Semantic segmentation of pavement cracks based on an improved U-Net

Siyu Jia

Southwest Minzu University, Chengdu 610200, China

Abstract: Cracks are one of the types of pavement defects that can affect the safety and quality of roads, so identifying such defects is an important part of road maintenance. In this paper, based on the U-Net coding-decoding structure, an ECA channel attention module is added to the coding stage, thus improving feature extraction without introducing too many extra parameters and computational effort. The insertion of the FCNhead decoding dock in the decoding stage can improve the performance of the model on semantic segmentation tasks while maintaining its efficiency and interpretability, thus better meeting the needs of practical applications. Experimental results on the CFD dataset and the crack500 dataset show that the algorithm improves the accuracy of crack detection and has better robustness.

Keywords: U-Net; Semantic segmentation; ECA; FCNhead.

1. Introduction

In recent years, China's road construction has made world-renowned achievements, the development of road paving in a great degree to promote China's economic development, but also to make some of China's remote areas, plateau areas, especially the ethnic minority gathering areas more convenient for people to travel. However, the later maintenance of roads requires great labour costs, and the health of roads will also become worse and worse due to the aggravation of vehicles, bad weather, natural aging and other factors, which is especially serious in the harsh environment of the plateau minority gathering areas. Pavement cracks as the initial manifestation of pavement disease, but also the most common pavement disease, in the pavement disease detection occupies a major position. Traditional manual inspection methods are not only time-consuming, labour-intensive, less accurate and less safe. Therefore, research into faster pavement crack detection is important to ensure the safety of traffic.

Researchers have conducted in-depth studies on road crack detection and have proposed a number of problem-solving methods, ranging from image processing to machine learning methods, including deep learning methods that are widely used today. Image processing algorithms include mainly methods such as threshold segmentation, edge detection and region growing, which are mainly used to process images and identify crack features. Traditional machine learning methods based on crack detection, such as neural networks and support vector machines, still rely on features that have been handcrafted using image processing techniques. Deep learning methods have fundamentally changed the way crack detection is done and have greatly improved detection performance [1].

With the excellent performance of deep learning in image processing tasks such as image classification, target detection and semantic segmentation, more and more research has been conducted in recent years on the use of deep learning methods for crack detection. The emergence of network models such as Inception [2] and Resnet [3] has increased the depth and width of the network structure and extracted more accurate and complete image features. Networks based on these models for classification, target detection and image segmentation have been effectively applied to the task of pavement crack detection [4]. Wu, Xiangdong et al. [5] combined conditional random fields with convolutional neural networks to refine crack segmentation boundaries and identify cracks; Anh [6] proposed an encoding-decoding full convolutional network (FCN) for concrete pavement crack detection; MeiQ et al. [7] used densely connected convolutional neural networks for crack detection; Lu, Yinju [8] proposed a multi-scale feature extraction module for learning rich deep convolutional features, so that the obtained crack features are more discriminative in complex backgrounds; Jia-Luo Park[9] proposed an end-to-end deep convolutional neural network for pavement crack detection, which can still obtain the complete features of cracks in complex scenes.

Although these crack detection methods have achieved a certain degree of success, the edges of the crack detection results are still blurred and the detection accuracy is not high in complex contexts. This study addresses two problems of crack feature extraction and target area location information learning, and proposes an improved UNet semantic segmentation network model based on the ECA attention mechanism to extract deeper features for defects in pavement cracks, enhance the model generalisation capability, and improve the accuracy and precision of defect segmentation.

2. Image pre-processing

2.1. Introduction to the datasets

(1) CRACK500: The CRACK500 dataset [10] was derived from 500 images of road cracks ranging in size from 2560 pixels x 1440 pixels to 2448 pixels x 3264 pixels taken by mobile phones at Temple University by Yang et al. The cracked images were annotated pixel by pixel. In order to fit the pixel requirements of the input images to the network model and to facilitate the training of the model, 408 of the images with 2560x1440 pixels were selected and each original image was cropped to 512x512 pixels. The CRACK500 dataset consists of a training set and a test set, some of the original training images and labels are shown in Figure 1, the training images and the labeled images have the

same names, but the training images are in .jpg format and the labeled images were in .png format. The training set contains 285 images and the validation set contains 123 images. labeled images in CRACK500 corresponding to the original figure are divided into 2 colours, black for the background and white for the crack area. Figure 1 Part of the crack images and image labels According to the requirements of the model input, the original images and the labeled images in the training set need to be separated into 2 folders, and the data name text for image segmentation needs to be established according to the image segmentation, specifying the image names of the training set and the validation set respectively, to facilitate the model to read and train the images.

(2) CFD: CFD is a dataset of concrete pavement cracks with labelled images proposed by Shi et al. [11] and contains 118 images of size 480x320 pixels. These images contain background noise, such as shadows and stains. All images have hand-labelled labeled images.

2.2. Data enhancement

In deep learning, in order to improve the robustness of recognition models and prevent overfitting during training, data augmentation operations are often performed on the training dataset in order to improve the generalisation ability of the model. Commonly used data enhancement methods include rotation transformation, flip transformation, translation transformation, contrast transformation, etc. As the actual crack pictures taken on the road have characteristics such as variable crack directions, low brightness and high noise, the data enhancement methods selected in this paper are rotation transformation, inversion transformation, contrast transformation and noise scrambling to make the pictures in the training set more closely match the characteristics of the road pictures taken during vehicle driving [12].

3. Experimental methods

3.1. UNet algorithm

The UNet [13] network architecture is shown in Figure 1, which consists of two parts: the encoder on the left and the decoder on the right. Each downsampling module of the encoder consists of two repeated 3 × 3 convolutions and a 2 × 2 maximum pooling, and for each downsampling of the feature map, the module increases the number of channels by a factor of 1 and reduces the feature map size by 1 2; each upsampling module of the decoder consists of upsampling the feature map first using a 2×2 deconvolution to up-sample the feature map, expanding the feature map size by a factor of 1 and reducing the number of feature channels by 12, and then stitching the feature map between the encoder and decoder in the channel dimension to compensate for the image semantic information lost due to the decoder downsampling, and then further extracting the image feature information through two 3×3 convolutions.

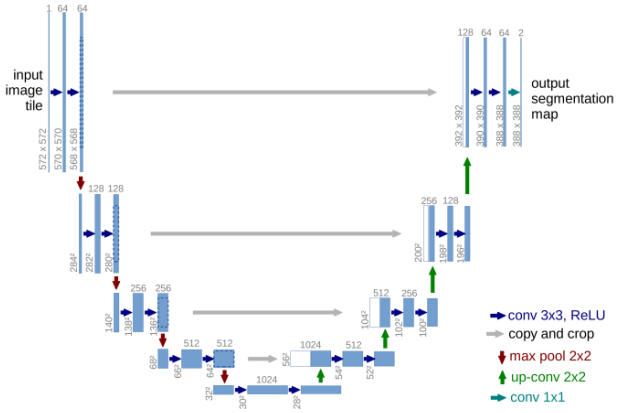

Figure 1. UNet model structure diagram

3.2. Decode head

The encoder-decoder network used in this paper is the encoder-decoder network. An encoder is usually a pre-trained classification network, like VGG, ResNet, followed by a decoder network. Where these architectures differ is mainly in the decoder network. A decoder is usually a series of neural network layers used to decode and restore high-level feature information back to the resolution of the input image to enable pixel-level classification and labelling. The main task of the decoder is to restore the detailed information of the input image from the low-resolution feature maps extracted from the encoder. The decoder dock is usually the final layer of the decoder output, which maps the decoder-extracted feature maps to pixel-level classification results and outputs the final semantic segmentation results. The main task of the decoder dock is to translate the feature information from the decoder output into the corresponding pixel labels.

ASPPHead (Atrous Spatial Pyramid Pooling Head) is a dock solving structure commonly used in image segmentation tasks to efficiently extract multi-scale features from images in order to improve the segmentation accuracy of the model. ASPPHead was first proposed in the DeepLab [14-15] family of models, and has since been widely used in other semantic segmentation models.At the heart of ASPPHead is the Atrous Spatial Pyramid Pooling (ASPP) module, which consists mainly of multiple parallel Atrous Convolutional Layers to process feature maps with different cavity rates. In each Atrous Convolutional Layer, the sampling rate of the convolutional kernels can be adjusted to different void rates to control the size of the perceptual field, thus obtaining multi-scale perception without loss of image detail. the Atrous Convolutional Layers in the ASPPHead can also employ multiple The Atrous Convolutional Layers in ASPPHead can also be pooled at multiple scales to further increase the range of the perceptual field. In ASPPHead, the ASPP module is usually inserted after the last layer of the encoder to capture larger contextual information.

PSPHead is a component of PSPNet[16] whose main role is to input the feature graph into the pyramid pooling module to obtain contextual information at different scales in order to improve the accuracy of semantic segmentation. PSPHead is designed to include four main components: a convolutional layer, a pooling layer, an upsampling layer and a feature fusion layer. First, the convolutional layer performs dimensionality reduction and feature extraction on the input feature map to produce a more discriminative feature map. The pooling layer then pools the feature maps at different scales, capturing contextual information at different scales in

the image. Next, the upsampling layer upsamples the pooled feature maps at different scales so that they have the same dimensions as the original feature maps. Finally, the feature fusion layer stitches the feature maps at different scales and fuses them with features using a convolutional layer to capture global and local contextual information.

The FCNHead is a commonly used deconvolution part for performing pixel-level semantic segmentation tasks in a Fully Convolutional Network (FCN)[17] . The FCNHead typically consists of a series of convolutional layers and upsampling operations to convert the network's low-level feature map into an output feature map of the same size as the input image, and and the feature vector at each pixel location corresponds to the predicted probability of the category at that location. In FCNHead, the common upsampling methods include bilinear interpolation upsampling and deconvolution upsampling. Bilinear interpolation upsampling is a simple operation that up-samples a low-resolution feature map to a target feature map of the same size as the input image by scaling and smoothing the feature map. Deconvolution upsampling, on the other hand, is a method of performing upsampling through a convolution operation that better preserves the detailed information in the feature map, thus producing more accurate segmentation results.

3.3. ECA Attention Module [18]

In order to balance model performance and computational complexity, this paper embeds a lightweight ECA (Efficient Channel Attention) channel attention module in the coding part of the model, so that the channel weights can be dynamically adjusted by calculating the global features in each channel feature map without introducing too many extra parameters and computational effort, which ensures feature richness while This improves the expressiveness and generalisation capability of the model while ensuring feature richness.

The ECA attention module first uses the operation of global averaging pooling on the input feature map u(i) (where i denotes a pixel point in the feature map) to take an average of the pixels in each feature channel, thus obtaining the global features in the image channel dimension. The formula is as follows:

$$g(u) = AvgPool(u) \tag{1}$$

Next, the obtained feature map is fed into a one-dimensional convolutional layer of size 1×k. The feature map can obtain the correlation between different channels through this convolutional layer, so as to obtain the appropriate weight distribution between different channels, and then use the Sigmoid activation function to normalize the weight values of different channels, with the following formula:

$$\omega = Sigmoid(C1D)k(g(u)) \tag{2}$$

where C1D denotes a one-dimensional convolution. Here the size k of the one-dimensional convolution is set to 3 and the Sigmoid activation function is formulated as follows:

$$f(x) = 1/(1+e^{-x}) \tag{3}$$

Finally, the learnt weights are extrapolated with each channel of the original feature map separately, with the following formula:

$$Z = \omega \otimes u \tag{4}$$

where: ω denotes the learned channel weights; Z denotes the feature map output by the ECA attention module.

4. Experimental results and analysis

4.1. Experimental environment

The experiments are based on the Pytorch (1.8) framework; the GPU computing device model is NVIDIA GeForce GTX 2080Ti; the CUDA version is 10.2; the programming language is Python3.7; the stochastic gradient descent (SGD) training iteration is used, the initial learning rate is set to 0.01, the momentum factor momentum is set to 0.9, the weight decay factor is set to 0.0005, the number of iterations is 40,000, and the semantic segmentation toolkit MMSegmentation is used. was set to 0.9, the weight decay factor was set to 0.0005, the number of iterations was 40,000, and the semantic segmentation toolbox MMSegmentation was used.

4.2. Evaluation indicators

MIOU and mAcc are one of the two commonly used deep learning target detection evaluation metrics, which are very effective in analysing pavement crack segmentation results [19]. The metric is defined as follows: Pixel Accuracy, pixel accuracy is the percentage of correctly marked pixels to the total pixels. The formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

mAcc (mean accuracy), the average accuracy is averaged over all categories of Accuracy. The formula is as follows:

$$mAcc = \frac{1}{K+1}\sum_{i=0}^{k}\frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

MeanIOU (mean intersection over union), the mean intersection ratio has been used as a standard metric in semantic segmentation, and the IOU formula is as follows:

$$IOU = \frac{TP}{TP + FP + FN} \tag{7}$$

MIOU is averaged over all categories of IOUs. The formula is as follows:

$$MIOU = \frac{1}{K+1}\sum_{i=0}^{k}\frac{TP}{TP+FP+FN} \tag{8}$$

Where: TP indicates that the target is cracked and the detection result is also cracked (positive detection); TN indicates that the target is non-cracked and the detection result is also non-cracked (negative detection); FP indicates that the target is non-cracked and the detection result is cracked (false detection); FN indicates that the target is cracked and the detection result is background (missed detection). The formulae of the above two indices show that when Accuracy and MIOU are larger, it proves that the extraction of the model is more effective.

4.3. Analysis of results

In order to verify that different dockers have different segmentation performance on the model, under the same running environment and parameter settings, this paper designs a semantic segmentation model with UNet and adds the mainstream dockers FCNhead, ASPPhead and PSPhead to do comparison experiments on two public datasets respectively, and the experimental results are shown in Table 1 and Table 2.

Table 1. Test results of different algorithms on CFD dataset

Algorithms	mIou	mAcc
UNet+ASPPhead	75.3	83.09
UNet+PSPhead	75.49	84.15
UNet+FCNhead	**75.71**	**86.44**

Table 2. Test results of different algorithms on the crack500 dataset

Algorithms	mIou	mAcc
UNet+ASPPhead	62.17	67.46
UNet+PSPhead	63.3	67.67
UNet+FCNhead	**63.97**	**70.3**

From Tables 1 and 2 and it can be seen that using UNet as the base network model and adding FCNhead to the dock solver achieves better results on both the CFD dataset and the crack500 dataset. In the UNet + FCNHead structure, the FCNHead is inserted into the last layer of the UNet to extract higher level features using convolutional layers and upsampling operations and convert them into pixel level segmentation results. The advantages of FCNHead over other dock solving components are that it has a simple, efficient structure that produces high quality segmentation results quickly and is very interpretable, giving a clear indication of the category prediction probabilities for each pixel point.

In order to verify the effect of the ECA module on the model performance, this paper first UNet+FCNhead model as the basis, add the ECA attention module to the coding part of UNet for experiments. The results are shown in Table 3 and Table 4 by comparing the experiments on crack500 and CFD datasets.

Table 3. ECA-UNet results on CFD dataset

Index	ECA	mIou	mAcc
1		75.71	86.44
2	√	77.12	87.59

Table 4. ECA-UNet results on the crack500 dataset

Index	ECA	mIou	mAcc
1		63.97	70.3
2	√	66.31	72.11

The addition of the ECA attention module to the model coding location has improved the accuracy as can be seen in Tables 3 and 4. On the CFD dataset, the accuracy of the model improved from 86.44% to 87.59%, and the average cross-merge ratio improved from 75.71% to 77.12%. On the crack500 dataset, the accuracy of the model increased from 63.97% to 66.31%, and the average cross-merge ratio increased from 70.3% to 72.11%. This indicates that the ECA module embedded in the coding part of the model can better improve the extraction of features.

5. Conclusion

In this paper, for the detection of cracks based on pavement images, we propose to use the classical UNet network to implement the detection of crack regions in pavement images and achieve pixel-level crack region recognition. By comparing between the semantic segmentation network models with different solution docks incorporated, it is not only confirmed that the semantic segmentation network model can be used for pixel-level recognition of road pavement cracks, but it is also found that when FCNhead solution docks are incorporated, the network has a higher degree of crack recognition and better robustness, which is useful for the selection of models to achieve pixel-level recognition of road pavement cracks. A lightweight ECA attention mechanism module is then embedded in the encoder to improve the accuracy of the model. The training results of this paper can also be used as a pre-training model for pavement crack recognition, which can be combined with actual pavement image samples for more targeted training for different detected pavements in the next step.

References

[1] W. Cao, Q. Liu and Z. He, "Review of Pavement Defect Detection Methods," in IEEE Access, vol. 8, pp. 14531-14544, 2020, doi: 10.1109/ACCESS.2020.2966881.

[2] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[3] He K, Zhang X, Ren S, et al. Deep residual learning [J]. Image Recognition, 2015, 7.

[4] Zhai Junzhi, Sun Chaoyun, Pei Lili et al. A multi-scale feature-enhanced pavement crack detection method[J]. Journal of Transportation Engineering, 2023, 23(01): 291-308. DOI: 10.19818 / j.cnki.1671-1637.2023.01.022.

[5] Wu Xiangdong, Zhao Health, Liu Legend. Crack detection algorithm for bridges based on CNN and CRF [J]. Computer Engineering and Design, 2021, (42): 51-56.

[6] Dung C V. Autonomous concrete crack detection using deep fully convolutional neural network [J]. Automation in Construction, 2019, 99: 52-58.

[7] Mei Q, Gül M, Azim M R. Densely connected deep neural network considering connectivity of pixels for automatic crack detection [J]. Automation in Construction, 2020, 110: 103018.

[8] Lu Yinju, Li Zuzhao, Dai Shuguang. A pavement crack image segmentation algorithm incorporating high-order multiscale features [J]. Small Microcomputer Systems, 2022, 43(06): 1197-1203.

[9] Berjaro. Research on end-to-end road crack detection technology based on deep learning [D]. Wuhan: Huazhong University of Science and Technology, 2019.

[10] Yang F, Zhang L, Yu S, et al. Feature pyramid and hierarchical boosting network for pavement crack detection [J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(4): 1525-1535.

[11] Shi Y, Cui L, Qi Z, et al. Automatic road crack detection using random structured forests[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(12): 3434-3445.

[12] Wu Qiuyi. Pavement crack identification based on semantic segmentation network model[J]. Transportation Science Technology,2020, No.300(03):80-83+109.

[13] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation [C]//Medical Image Computing and Computer- Assisted Intervention-MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015: 234-241.

[14] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834-848.

[15] Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation [J]. arXiv preprint arXiv:1706.05587, 2017.

[16] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. 2881-2890.

[17] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[18] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C] // Proceedings of the IEEE / CVF conference on computer vision and pattern recognition. 2020: 11534-11542.

[19] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation [C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.