# Pavement Crack segmentation based on cDCGAN and attunet

## Abstract

Pavement crack segmentation is important to the condition assessment in civil engineering. In this work, a novel pixel level semantic segmentation structure, Attunet was proposed to do the pavement crack segmentation task accurately and efficiently. At the same time, an image augmentation method cGAN was proposed, which was modified from DCGAN. An open dataset DeepCrack was used to train and test the functionality of the Attunet and cGAN. The result from Attunet was compared with U-Net, DeepLab, FCN and lraspp\_mobilenet\_v3\_large. The result shown that

Also the cGAN was compared with DCGAN, and traditional augmentation methods by the segmentation accuracy. The result show that

## Keywords

Crack segmentation, DCGAN, U-Net, deep learning

## 1. Introduction

Crack has become one of the primary defects in pavement, which seriously affects the service life of pavement [1]. The traditional crack detection method is counting cracks manually, however, it is labor-intensive and time-consuming [2]. In addition, the specification required engineers to evaluate the damage level of each crack. Although road detection vehicle was used these years to detect pavement defects, time in counting the length and area of cracks is still huge while the accuracy is relatively low because engineer need to use mouse to measure the length which is easily introduce error. An automated method for crack classification and segmentation has attracted attentions in intelligent pavement surface inspection system.

The automation of crack detection and segmentation is important for civil engineering and highway agencies. A lot of traditional computer vision methods have been proposed in crack detection and segmentation. Image processing and machine learning are common methods in crack detection and extraction [3, 4]. The main idea is to learn and summarize the characteristics of crack and then build a model to make predictions, such as Support Vector Machine (SVM) [5] and K-means algorithm [6], which have already been used in the classification and segmentation of crack images. A crack detection approach based on Local Binary Patterns (LBP) with SVM was proposed by Cheng [5], which can extract the LBP feature from each frame of the video taken from the road. Ai, D. et al. [7] used multi-neighborhood information to segment cracks with the F1-score of 0.8. The accuracy of the algorithm proposed by Kaddah, W. et al. [8] is about 75%, which is based on the improved minimum path method, an image processing method, to segment pavement cracks.

However, image processing and traditional machine learning which heavily depends on the engineer’s knowledge, may limit the overall performance, Deep learning is becoming one of the most advanced pixel-level target detection methods in road condition inspection. Convolution Neural Network (CNN), a deep learning method, has been gradually utilized in road crack detection and segmentation [9, 10]. For example, Bang, S et al. [11] used ResNet-152 to classify cracks and got the results as Precision of 77.68% and Recall of 71.98%. Cao Vu dung et al. [12] used Full Convolution Neural Network (FCN) to detect and segment cracks and got an average accuracy of 90%. Yahui Liu proposed a DeepCrack for crack segmentation, which was consisted of the extened Fully Convolutional Networks (FCN) and the Deeply-Supervised Nets (DSN), and it showed a comparable result when compared with typical segmenting methods like AutoCrack and SegNet. Liu J W et al. [13] proposed a two-step pavement crack detection and segmentation method by combining YOLO v3 and a modified U-Net model together. Chengjia Han[14] proposed a U-Net based CNN model, CrackW-Net, by adding a skip-level round-trip sampling block to segment the pavement images from the Crack500 dataset and a self-built dataset and shows a good result. Wenjun Wang proposed a pyramid attention network which uses pre-trained DenseNet121 and a feature pyramid attention module. It was tested on the Crack500 and MCD dataset and achieves a IoU of 0.6235.

Although deep learning is the most advanced pixel-level segmentation method, it requires a large amount of data to train the network. Due to the large consumption of data, some data augmentation methods were popular in deep learning. Traditional augmentation methods include image crop and image flip are often used in the preprocess part to augment the dataset. In 2014, Goodfellow proposed the concept of generative adversarial networks (GANs), which can produce real-like images through a battle between the generator and discriminator. Alec Radford proposed deep convolutional generative adversarial networks (DCGANs) based on the conception of GAN, and it showed good representations of images. Because of using the convolution structure, DCGAN is popular in computer vision and has been applied in many areas including pavement crack data augmentation. For example, Lili Pei[15] used variational autoencoder (VAE) to encode crack images and the results from VAE was input to DCGAN model to generate the fake images. Boqiang Xu recaches a quite high identification accuracy in pavement crack classification tasks based on DCGAN and VGG16. However, there are still some problems in DCGAN. For example, the original DCGAN structure is more suitable for small size images like an image with resolution of 32 \* 32 pixels. The discriminator in DCGAN studies too fast which would lead the loss of discriminator to 0 very rapidly.

Based on these studies, this paper proposed a new pixel-level pavement crack segmentation network, AttuNet. This new net was tested and compared with other classic semantic segmentation models including Unet, FCN, DeepLab….

Meanwhile, a modified cGAN was proposed as an image augmentation method to generate fake images for the deep learning model. the output result was compared with traditional augmentation methods and DCGAN. The accuracy of the AttuNet was used as a metric to evaluate the performance of the augmentation methods. The results show that the proposed cGAN method could siginificantly improve the accuracy of the AttuNet in the same crack dataset.

## 2. Methods

### 2.1 Data Preparation

A DeepCrack dataset was utilized in this work to test the performance of the networks. The DeepCrack dataset is an opensource dataset published in Github (<https://github.com/yhlleo/DeepCrack>). This dataset consists of 537 RGB crack images with manually annotated segmentations. The image has a resolution of 544 \* 384 pixels. The images were divided into two subsets: 300 images for training and 237 images for testing. As we can see, the amount of images is relatively for a deep learning training. Thus, image augmentation methods were applied to this dataset including traditional augmentation methods, DCGAN and cGAN. The number of the augmented images were shown in Table 1. Before importing to the Attunet structure, all the images including the crack image and its label, would be reshaped to 256 \* 256 pixels.

Table 1. The number of raw images in dataset and the generated images from augmentation methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Training | Testing |  |  |
| DeepCrack | 300 | 237 |  |  |
| cDCGAN | 100 |  |  |  |
| Traditional | 100 |  |  |  |
| DCGAN | 100 |  |  |  |
|  |  |  |  |  |

### 2.2 Data Augmentation

Data augmentation has proven to be an important thing before deep neural networks training. Instead of traditional augmentation methods like crop and flip, a cGAN was proposed in this work based on the structure of DCGAN. The structure of cGAN was shown in Figure 1.

As for the cGAN model, the initial learning rate was set to 0.0001, and adjusted during training, where the learning rate would decrease by 15% for each 10000 steps. Binary Cross Entropy (BCELoss) was used as the loss function and Adam was utilized as the optimizer to update the network. The batch size of the dataset was set to 16.

Normally, GAN only works well at much lower resolutions like 32 \*32 or 64 \*64. In order to make the model architecture for better results, some modifications were made. The contributions of this work can be summarized as follows:

1. Larger kernel size was used. The kernel size was increased by 1, from 3\*3 to 4\*4 both in generator and discriminator. For generator, a larger kernel at the top convolutional layers could cover more area and thus, could capture more information maintain the smoothness of the image. For discriminator, a small kernel may cause the discriminator loss to rapidly approach 0 while a larger kernel size can ease this situation.

2. The number of filters was increased compared in cGAN to the original DCGAN. A small amount of filters, especially in generator, would make the produced images very blurry while more filters can help capture additional information which can eventually add sharpness to the final produced images.

3. A batch normalization layer was followed by the convolutional layer. Batch normalization acts as a regularize which can reduce the accelerating training and improve the generated image quality.

4. A Gaussian noise layer was added as the first layer of the discriminator. It can prevent the discriminator from studying too quick.



Figure1: A structure diagram of the proposed cDCGAN model. The input is a random vector with length 4096. In Generator part, a GBlock (256@4\*4) is comprised of a convolution layer with filters 256 and kernel size , and then a batch normalization with momentum 0.7, an Rectified Linear Unit (ReLU) and an upsampling layer are performed on the result from convolution. In Discriminator, A DBlock(16@3\*3) is comprised of a convolutional layer whose kernel size is , and then followed a batch normalization with momentum 0.7, an activation function leaky Rectified Linear Unit (leakyReLU), a Dropout layer with parameter 0.25 and an average pooling layer. The convolution layer can process the image to produce a set of feature maps. The activation functions used in this cGAN model include ReLU, leakyReLU and sigmoid, which can make the network learn a non-linear task. The average pooling in Discriminator was utilized to translate invariance and reduce the parameter size of the networks.

### 2.3 Crack segmentation

An AttuNet was proposed as a crack segmentation approach in this work. Figure 3 shows the main architecture of the proposed structure, in which the details of each operation are presented.



Figure 3. The structure of the proposed Attunet. The output is a Nc channel map of the probabilities where Nc is the number of classes (Nc=1 in this work). The attention module filters the features rather than connect the features directly. Each convolutional operation would follow a batch normalization to standardize the inputs and stabilize the learning procedure.

There are some differences made in the AttuNet compared to U-Net model:

(1) An attention module was introduced to increase the accuracy of the model. The details of attention module were shown in Figure 2. This is because when applying image feature extraction, it will cause the lack of spatial local information and loss of pixel positioning, which will lead to the precision loss in the final crack segmentation.



Figure 2. The structure of the used attention module. The attention module got two inputs g and x from former layers. Each input was processed through a convolution layer and batch normalization and then they are added together which can fuse the features under the two different scales. And finally the result from the attention module would concatenate with the input x after a ReLU activation, a 1\*1 convolution, a batch normalization and a sigmoid function. By doing this, the attention module can fuse the different features from different scale layers to improve the consistency of the feature map and thus improve the model performance.

(2) Each convolution layer was followed by a batch normalization layer, which can standardize the inputs from convolution operation and also stabilize the learning procedure.

(3) BCEWithLogitsLoss was used as the loss function which combines a sigmoid layer and the Binary Cross Entropy in one single class. Root Mean Squared Propagation (RMSProp) with a momentum 0.9 was utilized as the optimizer to update the network.

The initial learning rate was set to 0.00001. The batch size of the dataset was set to 16.

For the crack segmentation task, another version of AttuNet, called AttuNet-min, were designed in this work. In this version, the max pooling layer was replaced by the min pooling layer. This is because that the crack pixels are always the darkest part in an image, using a min pooling layer can extract the crack information accurately. At the same time, the ReLU was replaced by logsigmoid function as the logsigmoid function would put more attention on the small pixel value.

The data augmentation methods and CNN models were all implemented in Python and computed under the following machine speculations: Windows 10, Intel(R) Core (TM) i9-10900X CPU, NVIDIA RTX A4000 with 16 GB memory, 64GB RAM.

### 2.4 Evaluation metrics

Precision (P), Recall (R), F1 score, Intersection over Union (IoU), Average Precision (AP) were utilized to evaluate the semantic segmentation.

The Precision, Recall and F1-score are calculated by Equation (6) to Equation (8).

(1) Precision can measure how accurate your predictions is. The precision can be calculated by Equation (6) where TP is true positive and FP is false positive.

, (6)

(2) Recall suggests the level of sensitivity for prediction results. Recall can be calculated by equation (7) where FN is false negative.

, (7)

(3) F1 score is defined based on the harmonic average of Precision and Recall.

. (8)

(4) IoU measures the overlap between 2 boundaries. It was used to measure how much the predicted boundaries overlaps with the ground truth. IoU was calculated according to Equations (9).

, (9)

Where represents the number of pixels belonging to class i but predicted as class j;.

(5) Average precision calculates the average precision value for recall value over 0 to 1. It can be calculated by finding the area under the precision-recall curve.

## 3. Results

Data augmentation

The results from cDCGAN were shown in Figure 5.

To make the experiments convincing, other three segmentation methods were compared including DeepLab, FCN and lraspp\_mobilenet\_v3\_large. We fine-tuned these three methods by changing the number of classes to 1, the loss function and optimizer to fit the dataset.

Table 1. comparison of different models on the test data.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | data | P | R | F1 | IoU | AP |
| Attention-U-Net | DeepCrack |  |  |  |  |  |
| Attention-U-Net | DeepCrack+DCGAN |  |  |  |  |  |
| Attention-U-Net | DeepCrack+crackDCGAN |  |  |  |  |  |
| Attention-U-Net | DeepCrack+traditional augment |  |  |  |  |  |
| Attention-U-Net | DeepCrack+crackDCGAN |  |  |  |  |  |
| DeepLabV3 | DeepCrack+crackDCGAN |  |  |  |  |  |
| FCN | DeepCrack+crackDCGAN |  |  |  |  |  |
| LRASPP | DeepCrack+crackDCGAN |  |  |  |  |  |

## 4. Conclusion

In the future, the number of the augmented images added to the training data can be studied.

## 5. Reference

[1] W. Wang *et al.*, "Pavement crack image acquisition methods and crack extraction algorithms: A review," *Journal of Traffic and Transportation Engineering (English Edition),* vol. 6, no. 6, pp. 535-556, 2019.

[2] Y. Yu, M. Rashidi, B. Samali, A. M. Yousefi, and W. Wang, "Multi-image-feature-based hierarchical concrete crack identification framework using optimized SVM multi-classifiers and D–S fusion algorithm for bridge structures," *Remote Sensing,* vol. 13, no. 2, p. 240, 2021.

[3] W. Lin, Y. Sun, Q. Yang, and Y. Lin, "Real-time comprehensive image processing system for detecting concrete bridges crack," *Computers and Concrete, An International Journal,* vol. 23, no. 6, pp. 445-457, 2019.

[4] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *Ieee Access,* vol. 8, pp. 14531-14544, 2020.

[5] C. Chen, H. Seo, C. H. Jun, and Y. Zhao, "Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM," *International Journal of Pavement Engineering,* pp. 1-10, 2021.

[6] J. Huyan, W. Li, S. Tighe, R. Deng, and S. Yan, "Illumination compensation model with k-means algorithm for detection of pavement surface cracks with shadow," *Journal of Computing in Civil Engineering,* vol. 34, no. 1, p. 04019049, 2020.

[7] D. Ai, G. Jiang, L. S. Kei, and C. Li, "Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods," *IEEE Access,* vol. 6, pp. 24452-24463, 2018.

[8] W. Kaddah, M. Elbouz, Y. Ouerhani, V. Baltazart, M. Desthieux, and A. Alfalou, "Optimized minimal path selection (OMPS) method for automatic and unsupervised crack segmentation within two-dimensional pavement images," *The Visual Computer,* vol. 35, no. 9, pp. 1293-1309, 2019.

[9] C. Chen, H. Seo, and Y. Zhao, "A novel pavement transverse cracks detection model using WT-CNN and STFT-CNN for smartphone data analysis," *International Journal of Pavement Engineering,* pp. 1-13, 2021.

[10] S. L. Lau, E. K. Chong, X. Yang, and X. Wang, "Automated pavement crack segmentation using u-net-based convolutional neural network," *IEEE Access,* vol. 8, pp. 114892-114899, 2020.

[11] S. Bang, S. Park, H. Kim, and H. Kim, "Encoder–decoder network for pixel‐level road crack detection in black‐box images," *Computer‐Aided Civil and Infrastructure Engineering,* vol. 34, no. 8, pp. 713-727, 2019.

[12] C. V. Dung, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction,* vol. 99, pp. 52-58, 2019.

[13] J. Liu *et al.*, "Automated pavement crack detection and segmentation based on two‐step convolutional neural network," *Computer‐Aided Civil and Infrastructure Engineering,* vol. 35, no. 11, pp. 1291-1305, 2020.

[14] C. Han, T. Ma, J. Huyan, X. Huang, and Y. Zhang, "CrackW-Net: A novel pavement crack image segmentation convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems,* 2021.

[15] L. Pei, Z. Sun, L. Xiao, W. Li, J. Sun, and H. Zhang, "Virtual generation of pavement crack images based on improved deep convolutional generative adversarial network," *Engineering Applications of Artificial Intelligence,* vol. 104, p. 104376, 2021.