Data Craft

CE 784A Project Report: Semester 2020-21 (III)

Application of Machine Learning Algorithms in Pavement Damage Detection

Udita Jana, Shashank Rajput, Shrey Agrawal, Sahil Jaiswal, Chirag Jaiswal

1. **Introduction**- Pavement cracking is an early sign of pavement distress which directly affects the quality of pavement. Such distresses may severely deteriorate the pavement structures as they are subjected to repetitive traffic loads and natural factors. Owing to which, cracks are often regarded as the most important pavement quality index and crack detection has always been the major focus in pavement maintenance and monitoring. If these cracks are detected beforehand and monitored properly, the development and further propagation of cracks can be analysed and controlled which not only prolongs the service life of pavement structures, but also decreases fuel consumption and road maintenance costs. Most importantly, it significantly improves the traffic safety aspects. However, the manual or semi-automatic methods for crack detection purpose, has several shortcomings- 1) Creates interference to traffic flow during data acquisition and measurement 2) Requires a lot of human and financial resources, time consuming, 3) Subjective evaluation results in poor accuracy. Compared to the above approaches, automatic crack detection technologies have shown improved efficiency and are also economical and convenient for data storage and analysis purposes.

**2. Study Objective**- Generally, automatic crack detection consists of three main aspects, acquisition of crack Images, crack extraction, and classification. Starting from photography based image acquisition methods to present three dimensional laser scanning techniques, the pavement image acquisition technology has evolved conveniently. However, recognizing and classifying the cracks from the images are still regarded as a challenging task. Under this backdrop, the current study aims

* To develop an efficient algorithm for crack recognition and classification.
* To extract more useful information and inferences from 2D images that can be further aid in decision making process for planning pavement maintenance work.

Researchers have performed in-depth studies on road crack detection and suggested many ways for handling such problems, ranging from image processing to machine learning methods, including deep learning methods which has gained much popularity in past few years. However, with the advanced machine learning applications the prospect of efficient road crack detection has risen to next level. This study intends to exploit emerging machine learning techniques in order to achieve the objectives stated above. The next section briefly reviews crack detection techniques based on various machine learning algorithms.

**3. Brief Overview of Previous Studies**

Machine learning has become a popular domain and widely used in various areas. It can provide predictions by understanding the underlying patterns present in dataset. Various Supervised learning [1][2] and unsupervised learning [3][4][5] techniques have been adopted by researchers for cracks detection and analysis purpose.

In the recent years, deep learning techniques became popular as it achieved remarkable success in various computer vision tasks like image classification and object detection. Researchers have proposed deep learning based methods, especially deep convolution neural networks (DCNNs) for handling crack detection problems [6]– [10]. These methods can be broadly divided into three categories based on the approach for solving such problems. a.) crack detection based on image classification b.) crack detection based on pixel- level segmentation c.) crack detection based on object detection. This study intends to explore the first two approaches for achieving the study objectives. Some of the previous works in this context are summarised below.

*Crack detection based on image classification*- In these approaches, the input image is divided into several overlapping blocks, the block image is then classified and labelled as defective provided the number of defective pixels present in the block exceeds a certain value.

Crack detection based on binary classification- in this case the input images are first divided into number of overlapping blocks and a deep convolution neural network (DCNN) is used to decide whether the block contains a crack or not. Lei *et al*. divided pavement images of 3264 × 2248 into small patches of size 99 × 99 × 3, and classified these patches using DCNN and obtained the probabilities for the patches to contain crack [11]. Later, cha *et al*. [12] used ‘MatConvNet’ [13] to classify the input pavement images of resolution 256 × 256. Leo *et al*. [14] used a self-designed CNN to understand the relation between network depth and network accuracy [15]. Chen et al. implemented CNN for classifying image patches from processed pavement videos in [16].

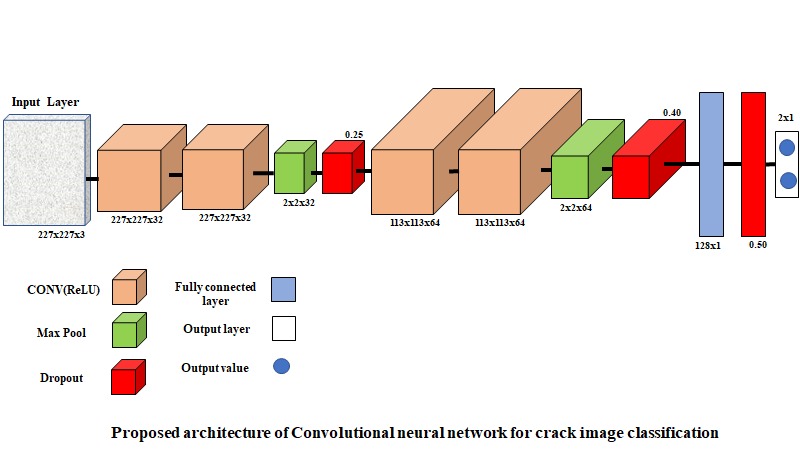
Crack detection based on multi-class classification- Crack detection based on binary classification often fails to provide information related to the pavement defect. For this purpose, often multi-class classification techniques are used while detecting pavement cracks. CNN model has been efficiently used to address such multi-class classification problems [17] [18]. Implementation of these techniques require 3D pavement images in order to classify the extent of pavement damage. Since, the present study aims to identify pavement cracks from 2D pavement images, binary classification techniques have been adopted. The detailed framework of the adopted method is discussed in the next section.

**4. CNN Framework for Crack Classification**

The overall architecture of the CNN adopted for this study is explained in this section. Also, the layers used in this study, and the backgrounds of each layer is briefly stated. Generally, CNN architecture consists of multiple layers such as input and output layer, convolution layer, pooling layer and activation layer. In a deep CNN framework, apart from the above mentioned layers some auxiliary layers i.e. dropout, batch normalisation, standard normalisation are also used.

*4.1 Overall architecture*

The proposed architecture of CNN for crack image classification is represented in Figure 1. The first layer is the input layer of 227x227x3 pixel resolutions RGB image i.e. height and width are of 227 pixels each and 3 channels. The input data was passed through the convolutional layers (L1, L2, L3. L4, L5, L6) see in table 1. The constant filter size of 3x3 is used for feature extraction process and dropouts are introduced after each set of 2 convolutional and 1 pooling operation to reduce the overfitting of the model. After convolutional operations the matrix is flatten into 1x1x128 column vector and is fed into the rectified linear unit (ReLU) layer. Finally, the sigmoid layer is used as output layer for binary classification of image into crack and non-crack. Detailed dimensions of all the layers and operators shown in the architecture are listed in Table 1.



**Fig. 1.** Proposed Architecture of CNN framework

**Table 1**: Dimensions of layers and operations

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Layer** | **Height** | **Width** | **Depth** | **Operator** | **Height** | **Width** | **Depth** | **No.** | **Stride** | **Padding** | **Dropout** |
| **Input** | 227 | 227 | 3 | **C1** | 3 | 3 | 3 | 32 | 1 | Same | - |
| **L2** | 227 | 227 | 32 | **C2** | 3 | 3 | 32 | 32 | 1 | Same | - |
| **L3** | 227 | 227 | 32 | **P1** | 2 | 2 | - | - | 2 | - | 0.25 |
| **L4** | 113 | 113 | 32 | **C3** | 3 | 3 | 32 | 64 | 1 | Same | - |
| **L5** | 113 | 113 | 64 | **C4** | 3 | 3 | 64 | 64 | 1 | Same | - |
| **L6** | 113 | 113 | 64 | **P2** | 2 | 2 | - | - | 2 | - | 0.4 |
| **L7** | 1 | 1 | 128 | **ReLU** | - | - | - | - | - | - | - |
| **L8** | 1 | 1 | 128 | **C5** | 1 | 1 | 128 | 2 | 1 | - | 0.5 |
| **L9** | 1 | 1 | 2 | **Sigmoid** | - | - | - | - | - | - | - |
| **Output** | 1 | 1 | 2 | **-** | - | - | - | - | - | - | - |

*4.2 Convolution layer*

A convolution layer first performs element-by-element multiplications between a sub-array of an input array and a kernel or receptive field. The initial weights assigned to the kernels are generated randomly. There are different ways of selecting bias according to the network. The dimensions of subarray is always same as that of the receptive field and the kernel dimensions chosen are always smaller than the input array. the multiplied values are then summed, and bias part is added to the summation. The convolution layers have advantages as they decrease the input data sizes thereby reducing computational costs.

*4.3 Pooling layer*

The pooling layer reduces spatial size of an input array also known as ‘downsampling’. Two different pooling layers i.e. max pooling and mean pooling layers are popularly used in CNN applications. Previous literatures have indicated that max pooling layer outperforms the mean pool operations, for this study also max pooling layer has been used.

*4.4 Activation layer*

The generate nonlinearity in the standard ANN framework sigmoid functions are often used as activation layers. But, owing to the fact that saturating nonlinearities slows down the computations, now a days ReLU was introduced as a non-linear activation function and as it does not provide bounded output values it facilitates faster computation with better accuracies. Considering these facts ReLU has been also used as an activation layer in this study.

*4.5 Auxiliary layer*

Machine learning algorithms often suffer from over-fitting issues. To address this issue dropout layers are implemented. This occurs when the network classifies a training data set effectively but fails to provide satisfactory validation results. To address this issue, dropout layers are used. Training a network with a large amount of neurons often results in overfitting due to complex co-adaptations. A dropout layer randomly disconnects some connections between the neurons with a certain dropout rate and accordingly the CNN can function more efficiently. To bring the dataset to a common scale without distorting the range of values. It is important when the features of the image have different ranges, though it is not always required in machine learning algorithms. Standardization is helpful when the data follows Gaussian (Normal) distribution, however this need not to be true. The basic understanding of the Gaussian distribution is given by the following expression.

Z=

**5. Dataset Used for Analysis**

The original dataset used for this study, [19] contains a total number of 400 raw images with 448 × 448 pixel resolutions which were annotated as cracked or non-cracked. There is not much variation in the image’s lighting intensity. In the pre-processing, the images were resized to 227 × 227 and the database for training the model was updated accordingly. From the database, images were randomly selected to generate training and validation datasets. Among the 400 images, 280 images were used to train the model and rest 120 images were later used to validate the adopted CNN framework.

*Hyperparameters*

The deep neural networks is usually trained by the Adam (Adaptive Moment Estimation) which is a common adaptive learning rate optimizer. Adam can be understood as the combination of the RMSprop and Stochastic Gradient Descent with momentum. It involves the squared gradients from the RMSprop and moving average of the gradient. The moving average varies exponentially and computes gradient on current mini-batch.

mt=β1mt-1+(1-β1)gt ….(1)

vt=β2vt-1+(1-β2)gt2 ....(2)

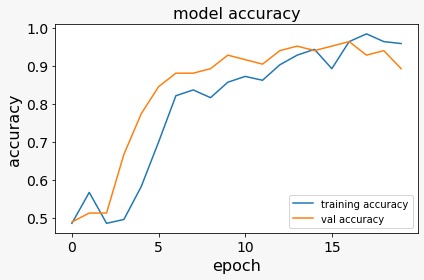
where m and v are the moving averages and g is the gradient of current mini-batch with β1 and β2 as the hyper parameters of the proposed algorithm. The default values of β1=0.9 and β2=0.999 has been taken into account. Subsequently, the updated weights are calculated using:

wt+1=wt- ….(3)

where is the learning rate taken as 0.001 and ε=10-8

**6. Training and validation results**

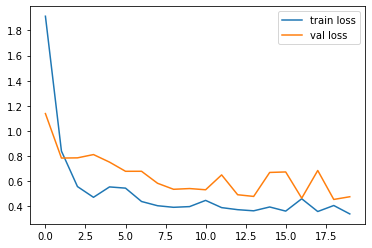
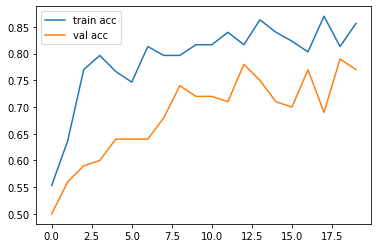
The training and validation results are shown in Figure 2. The ratio of the number of crack and intact images in the updated dataset was 1:1, and a split of 7:3 was used for validation. Therefore, the obtained training accuracy is calculated using 280 image sample, and validation accuracy is calculated using remaining 120 image sample. The highest accuracies for training and validation are observed to be 98.47% at the 18th epoch and 96.43% at 17th epoch, respectively.



**Fig. 2.** Training and validation accuracies for each epoch.

**7. Transfer Learning Based Approach**

In this machine learning technique, a pre trained model which is proposed for a particular task is utilised to solve similar but different problems involved in other tasks. In deep learning applications, model learns the necessary weights and features in order to classify different images. These pre trained models can be employed for solving a similar problem statement. The initial layers present in the network architecture which is useful for feature extraction are kept intact, only the layers at the end are modified according to the specific task requirement. Transfer learning has wide applications in tasks related to computer vision. Several algorithms had been proposed by researchers for image classification purpose. For this study, pre-trained model ResNet50 is used for crack detection purpose. The results obtained from this approach is represented in Figure 3.



1. (b)

**Fig. 3**. a. Training and Validation accuracies at each epoch b. loss for training and validation dataset at each epoch

**8. Pixel-wise Crack Segmentation-** Apart from traditional image-level or block-level based evaluations, pixel- level analysis techniques are also used for automatic crack detection. However, pixel-wise crack segmentation techniques are generally regarded as more favourable for extracting useful information as using this approach several crack characteristics can also be obtained. A significant amount of pixel-wise crack segmentation techniques has been developed in the recent years, among those this study aims to implement the U-net model.

*8.1 Fully convolutional network*

Pixel-wise object segmentation or semantic segmentation methods are based on fully convolution networks (FCN). Unlike traditional CNNs, in these techniques, fully convolutionallayers are used instead of full connected layers.Fully connected layers used in traditional CNNs requires input of a fixed length, this criterion in some cases i.e. while processing randomly sized images, this criterion appears as a big bottleneck. Hence, the fully convolutional layers are used in place of the fully connected layers. FCN has shown satisfactory results in several applications. However, FCN adds hundred paddings to original inputs to avoid feature map size reduction, which often makes such arrangements more complex and redundant.

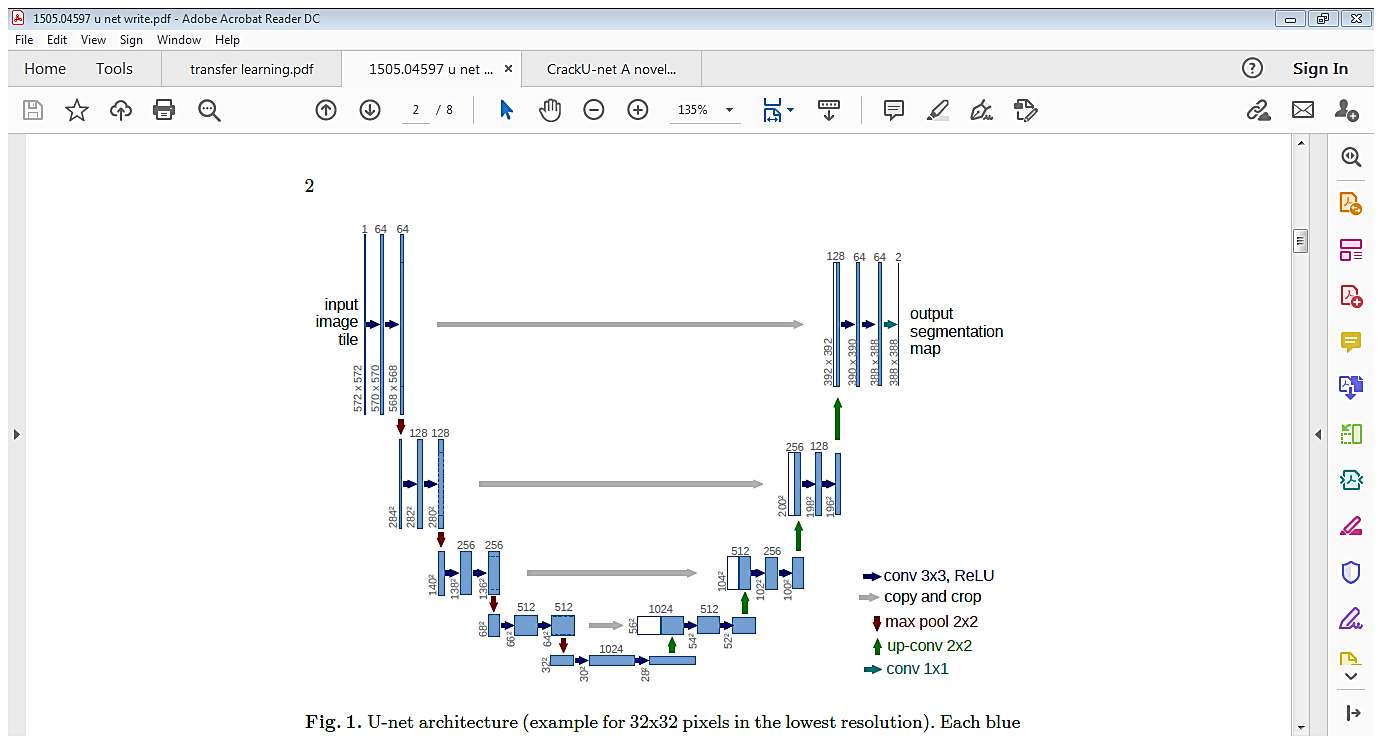
*8.2 Introduction of U-Net*

In 2015, U-net [20] was proposed as a modified extension of FCN in order to work with less number of training samples and yet not comprising the precision of segmentation process. The name “U-net” arises due to the “U” shape of the proposed architecture.

As mentioned earlier in case of FCNs, up sampling operations take place instead of pooling operations which produces high resolution output images. These high resolution features in combination with the up sampled output is used in localisation. Additionally, many feature channels are included in the U-net architecture up sampling part. This modification allows transmission of relevant information to the subsequent layers with high resolution. Another important aspect is the U-net framework, only utilises the valid portion of the convolutions, i.e., only the pixels, having full context present in the input images are considered in the segmented map. This approach can be efficient for segmentation of large images. An overlap-tile strategy (based on mirroring) is deployed for filling in the missing context of the input image which allows prediction of pixels near edges or border areas. Moreover, U-net discards the pre-trained portions which is essential for FCN architectures and yields a less complex and flexible network.

*8.3 Network Architecture*

The proposed architecture comprises of two parts – a) contracting path in the left part b) expansive path in the right part as illustrated in Figure 4. The contracting path has a similar architecture like convolutional networks. Here two 3x3 convolutions (followed by ReLU) are applied one after another (without any padding). A 2×2 max pooling operator (stride=2) is used for down sampling purpose. Here at each consecutive step the number of feature channels are twice of the previous step. On the right part of the architecture, the converse operations are carried out i.e. in each consecutive step the feature map is upsampled. Here 2×2 up convolutions are applied at each stage followed by 3×3 convolutions (with ReLU) that reduces the number of feature channels in consecutive steps. The resulting feature maps at each stage of the contracting path are concatenated with the corresponding stage at the expansive part. In the final step, a 1×1 convolution is used for mapping the last feature vector in order to obtain desired output classes. The overall network architecture includes 23 convolution layers in total, the details are shown in Figure. 4. Here, the rectangular blue box represents multi-channel feature maps. The number at the top of these boxes denotes number of channels. The feature map dimensions are indicated at the bottom left edges. White rectangular boxes in the right side represent feature maps copied from the contracting path. Different operators are denoted by the arrows.

.

**Fig. 4.** U-net model architecture.

*8.4 Skeletonization* -The width of the predicted crack image can be measured by extracting the morphological aspects, which can be segmented into a thinned crack skeleton. In the skeletonization the main aim is to produce skeleton consisting of pixels with not more than two neighbours located at a junction. The skeleton is a global property of a binary object, and that the boundary should be used to locate the skeleton pixels.

*8.5 Crack measurement*

Skeletonization of segmented images of crack are done to visualize the topology of cracks in the pavement. These skeletons can be further used in calculating the total pixel length of cracks present in the image. The measurement can be done as:

....(4)

where, is the total length of cracks in the image

are the crack pixel in the 2D image at x (row) and y (column),

is the finite length of the crack element.

Therefore, average crack pixel width can be calculated from the pixel area covered by the cracks and dividing the total pixel area by the total pixel length of the cracks obtained after skeletonization.

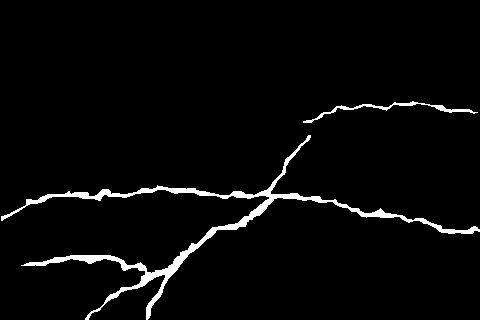
….(5)

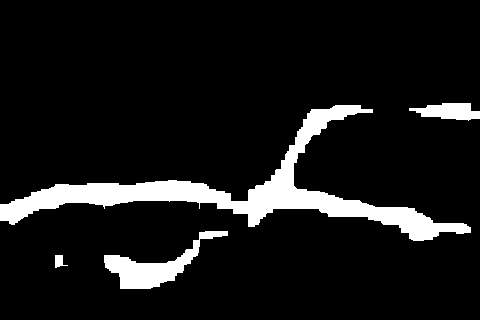
Where, is the average pixel width of the cracks in the image,

is the finite area of the crack element.

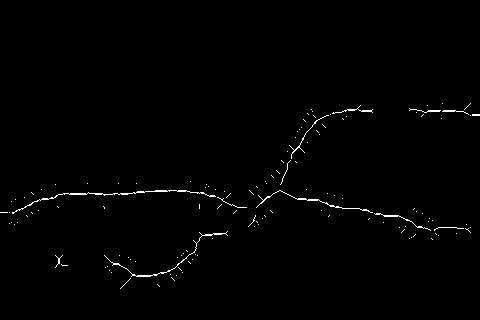
Dataset used- For pixel level crack segmentation and crack measurement two different data sources have been utilised, for training and validating the U-net model a dataset [21] containing around 11000 images was used. This dataset is actually a merged version of 12 crack segmentation dataset. On the other hand, a public database CFD [22] has been chosen as test dataset. This dataset contains 118 pavement crack images with pixel size of 320 x 480, and all these images were captured in Beijing, China using an iPhone 5. Various types of noise, oil spots, and shadows are present in the dataset.

Result and Discussion- Results obtained after implementing U-net and skeletonization are illustrated in Figure. 5. It was earlier found that the regular U-net model architecture could not segment the fine proportions of crack present in the images. To handle this problem, as a pre-processing step, the segmentation masks of images was dilated which yielded satisfactory result (Figure.5.c). Model achieved around 90 percent accuracy on both training and validation dataset.



**a b**

**c d**

****

**e**

**Fig. 5.** a. original image b. ground truth c. result obtained from U-net segmentation

d. ground truth for skeletonization e. result obtained from skeletonization

Results obtained from skeletonization was further used to compute the crack pixel dimensions, these values were compared with the actual measurements to understand how well these skeletonization technique can approximate the actual distress details. Results obtained from the comparison are shown in Figure 6. It was observed that variation in crack pixel length is much higher compared to crack pixel width. This can be also explained by the inaccuracies at the segmentation stage, as we can see from figure 5.b and 5.c. the prediction differs from the actual segmentation mask.

**Fig. 6.** Comparison between actual and predicted crack dimension

**Conclusion**

This project mainly focused on crack detection and pixel level segmentation of cracked surfaces. For the automatic crack detection purpose a CNN framework was proposed. Transfer learning (fixed feature extraction with ResNet50) was also tried out. In the pixel level segmentation part U-net framework was utilised. The segmented crack pixels were skeletonized to visualise the crack topology. The crack width and length were measured based on the predicted crack maps. Since, dilation operation was used during segmentation the predicted crack length was much higher than the ground truth. This technique can be used to approximate pavement distresses. However, this approach will generate effective results only in the absence of geometric distortions and in static conditions.

**References:**

[1] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic road crack detection using random structured forests,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 12, pp. 3434–3445, 2016.

[2] A. Marques and P. L. Correia, “Automatic road pavement crack detection using svm,” Lisbon, Portugal: Dissertation for the Master of Science Degree in Electrical and Computer Engineering at Instituto Superior Técnico, 2012.

[3] A. Akagic, E. Buza, S. Omanovic, and A. Karabegovic, “Pavement crack detection using otsu thresholding for image segmentation.” IEEE, 2018, pp. 1092–1097.

[4] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, “Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2718–2729, 2016.

[5] H. Li, D. Song, Y. Liu, and B. Li, “Automatic pavement crack detection by multi-scale image fusion,” IEEE Transactions on Intelligent Transportation Systems, no. 99, pp. 1–12, 2018.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, p. 436, 2015.

[7] W. Cao, Q. Lin, and Z. He, “Hybrid representation learning for crossmodal retrieval,” Neurocomputing, vol. 345, pp. 45–47, 2019.

[8] W. Cao, J. Yuan, Z. He, Z. Zhang, and Z. He, “Fast deep neural networks with knowledge guided training and predicted regions of interests for real-time video object detection,” IEEE Access, vol. 6, pp. 8990–8999, 2018.

[9] M. Dan, L. Zhang, G. Cao, W. Cao, G. Zhang, and H. Bing, “Liver fibrosis classification based on transfer learning and fcnet for ultrasound images,” IEEE Access, vol. 5, no. 99, pp. 5804–5810, 2017.

[10] D. Meng, G. Cao, Y. Duan, M. Zhu, L. Tu, D. Xu, and J. Xu, “Tongue images classification based on constrained high dispersal network,” Evidence-Based Complementary and Alternative Medicine,2017, (2017-3-30), vol. 2017, no. 4, pp. 1–12, 2017.

[11] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network.” IEEE, 2016, pp. 3708–3712.

[12] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.

[13] A. Vedaldi and K. Lenc, “Matconvnet: convolutional neural networks for matlab,” 2015.

[14] L. Pauly, D. Hogg, R. Fuentes, and H. Peel, “Deeper networks for pavement crack detection.” IAARC, 2017, pp. 479–485.

[15] F.-C. Chen and M. R. Jahanshahi, “Nb-cnn: deep learning-based crack detection using convolutional neural network and naïve bayes data fusion,” IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4392–4400, 2017.

[16] Z. Fan, Y. Wu, J. Lu, and W. Li, “Automatic pavement crack detection based on structured prediction with the convolutional neural network,” arXiv preprint arXiv:1802.02208, 2018.

[17] B. Li, K. C. Wang, A. Zhang, E. Yang, and G. Wang, “Automatic classification of pavement crack using deep convolutional neural network,” International Journal of Pavement Engineering, pp. 1–7, 2018.

[18] X. Wang and Z. Hu, “Grid-based pavement crack analysis using deep learning.” IEEE, 2017, pp. 917–924.

[19]<https://data.mendeley.com/datasets/xnzhj3x8v4/2>

[20] O. Ronneberger, P. Fischer, T. Brox, “U-net: Convolutional Networks for Biomedical Image Segmentation,” International Conference on Medical image computing and computer-assisted intervention. Cham: Springer; 2015, pp. 234-241.

[21] https://github.com/khanhha/crack\_segmentation

[22] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, “Automatic road crack detection using random structured

Forests”, IEEE Trans. Intelligent Transportation Systems. 2016, 17, 3434–3445.

Results from Plagiarism Check- 18% Similarity.