

Intelligent Preprocessing Selection for Pavement Crack Detection based on Deep Reinforcement Learning

Yan Lin
Beijing University
of Posts & Telecommunications
Beijing, China
linyuan@bupt.edu.cn

Guosheng Xu
Beijing University
of Posts & Telecommunications
Beijing, China
guoshengxu@bupt.edu.cn

Siyi Li
Beijing University
of Posts & Telecommunications
Beijing, China
lsy_curry@qq.com

Guoai Xu
Beijing University
of Posts & Telecommunications
Beijing, China
xga@bupt.edu.cn

Jie Deng
National Engineering Center
of Road Maintenance Technologies
Beijing, China
dengjie@roadmaint.com

Jiankun Cao
National Engineering Center
of Road Maintenance Technologies
Beijing, China
caojiankun@roadmaint.com

Abstract—With the rapid increase of traffic, the pressure on road maintenance is gradually increasing. Pavement crack is a common problem in all kinds of pavement diseases. In the actual production process, pavement images have different kinds of noise influence. The proposed algorithm is to select optimal preprocessing methods for pavement images in various conditions to improve the accuracy of crack detection. The algorithm includes two parts, a crack detection network and an intelligent preprocessing decision system. The crack detection network identifies the cracks in road images. The intelligent preprocessing decision system selects the best preprocessing method for pavement images based on the deep reinforcement method. The experiment results indicate that the validity and effectiveness of our proposed method.

Index Terms—crack detection; deep reinforcement learning; intelligent preprocessing decision system.

I. INTRODUCTION

The highway transportation system is one of the most important large-scale public infrastructures. With the rapid growth of traffic, the pressure on road maintenance is gradually increasing. Pavement crack is a common problem in various pavement diseases. How to effectively improve the detection effect of pavement cracks is an urgent problem to be solved. With the continuous advancement of technology, automatic image acquisition and detection methods have gradually replaced manual detection. In the last few years, with the swift development of the identification techniques, Kim et al. [1] discuss the methods of detecting and recognizing road cracks in 2017. Iqbal et al. present that the image preprocessing is the basic step of image processing and computer vision. It includes basic operations, such as noise removal, cropping, brightness increase, and non-uniformity correction [2]. In the

actual production process, there may be water stains, uneven illumination, shadows, and other interference information on the road surface. Different feature preprocessing methods are usually adopted for different road noise information. It is hard to shape a general system to detect pavement cracks in different conditions. In order to better identify the pavement cracks, we propose an algorithm that can autonomously select the optimal preprocessing method for road pavement images in various conditions. Cracks can be accurately detected when selecting the most suitable preprocessing method.

II. RELATED WORK

Deep reinforcement learning is based on reinforcement learning, supplemented by the strong generalization and feature extraction ability of depth models. Deep reinforcement learning has made great success in continuous decision-making problems, such as game playing [3] and machine control [4]. Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment [5].

Deep reinforcement learning has four key elements: environment, state, action, and reward [6]. A reinforcement learning model can be built with these elements. The problem of reinforcement learning is to obtain an optimal policy for a specific problem, maximizing the reward obtained under this strategy. The policy is the sequential data that represents a series of actions [7]. Motivated by the advances and the features of deep reinforcement learning, we propose an intelligent preprocessing system for pavement crack detection based on deep reinforcement learning method. 1) The system can choose different preprocessing methods for pavement images in various conditions. 2) The system selects the optimal

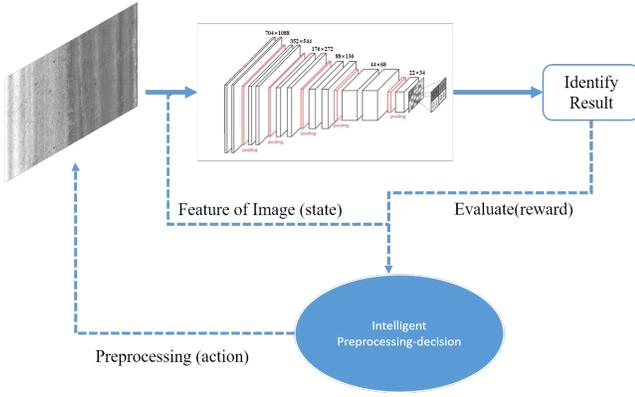


Fig. 1. Structure overview. The upper part of the figure shows the crack detection network, whereas the lower part presents the intelligent preprocessing-decision system.

preprocessing method for a certain image to improve the accuracy of the crack detection model.

The rest of the paper is structured in the following manner. In Section III, we describe our proposed system. In Section IV, we explain the details of the components of the system. In Section V, we demonstrate the experiments and the analysis of results. In Section VI, conclusions are provided.

III. SYSTEM OVERVIEW

A. Overall Structure

The general architecture of the proposed algorithm in this paper is illustrated in Fig. 1. The network is divided into two parts: the crack detection network and the intelligent preprocessing-decision system.

The goal of our algorithm is to raise the accuracy of the crack identification model by autonomously choosing a suitable preprocessing method for images. We utilize the deep reinforcement learning method to check the quality of the features extracted by the crack identification network and to judge the recognition of the crack identification model. If the accuracy of crack identification reaches a certain value, then the crack detection network can output the result. Otherwise, images will be transformed by the preprocessing method selected by the intelligent preprocessing-decision system.

The decision ability of the deep reinforcement learning method is crucial. It can fully understand the abstract features and make decisions on the basis of the operations of these features by integrating the functions of automatic feature extraction. We transform pictures with poor recognition results and re-value them by using the crack recognition model. Using the same parameters to raise the accuracy of the crack identification model is possible through the autonomous learning ability of the intelligent preprocessing-decision system.

B. Crack Detection Network

Crack detection network is responsible for the feature extraction, road image input, and crack recognition. The network is composed of several convolutional layers and five pooling

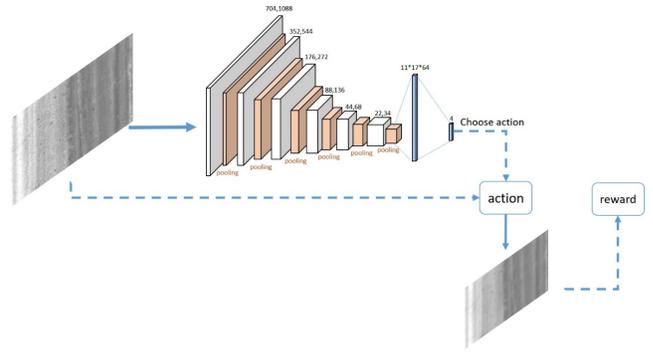


Fig. 2. Structure of the intelligent preprocessing-decision system

layers alternately. These pooling layers can reduce the input images to their 1/32. Original 2200×3400 images are initially resized to 704×1088 and then reduced to 22×34 after going through five pooling layers.

The output of the crack detection network is a 22×34 matrix, each element of which represents if the corresponding block contains cracks. The value of 1.0 indicates that cracks may appear in this block. Such labeling is a one-to-one correspondence with the output of the neural network, which can efficiently conduct the end-to-end training of image labeling. Given the existence of multiple convolutional layers, the top-level neurons use the context information around its corresponding block to determine whether the small area is a crack, effectively utilizing the overall characteristics of images.

The output of the crack identification model and the target are matrices. Measuring the distance between each element in the two matrices is necessary. The number of blocks containing cracks is relatively small, and the number of blocks containing cracks in each picture can be rather different. We use positive sample dice coefficient (PSD) to measure the similarity of vectors as the objective function for training. The common formula for measuring vector similarity is defined as follows:

$$PSD = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (1)$$

where X represents the output matrix of the crack identification network, and Y indicates the manually labeled matrix. $|X|$ and $|Y|$ represent the $L1$ norm of the two matrices. $X \cap Y$ indicates the Hadamard product of X and Y . When X equals Y , $PSD = 1$; otherwise $PSD < 1$. The value of PSD becomes large when the coincidence of X and Y becomes large. The value is only related to the proportion that the crack area is correctly identified and has no relation to the number of crack areas. Therefore, a common evaluation standard for different pictures exists. In practice, X and Y may be zero at the same time. To prevent the divide-by-zero error, we add a smoothing term to this formula:

$$PSD = \frac{2|X \cap Y| + \epsilon}{|X| + |Y| + \epsilon} \quad (2)$$

C. Intelligent Preprocessing-decision System

In our paper, the intelligent preprocessing-decision system is a crucial part to improve the ability of the crack detection model. The main structure of this part is shown in Fig. 2. The system based on the deep reinforcement learning method judges the current result of the crack identification and transforms images with bad recognition results by using the selected preprocessing method. The transformed images are then placed in the crack identification network again, and a new round of calculation is performed.

The reinforcement learning method is used to select a suitable preprocessing method for the pavement images. To achieve this goal, four main components: environment, state, action, and reward should be defined.

Intuitively, the environment of the intelligent selection system is the feature space composed of the whole pavement image dataset. An intelligent system must learn which preprocessing method is beneficial for crack identification. It should transform images with bad recognition results by using the suitable preprocessing method.

We consider the feature space of images as the state. The initial state is the feature space of original images. After preprocessing images, the state is transformed into the next step, which is the feature space of the transformed images.

The intelligent system involves the preprocessing methods for pavement images, including Contrast-limited adaptive histogram equalization, bilateral filter, and morphological opening operator. We also add no operation on the images to the action set. Therefore, four actions are performed in total. In Section III, the reasons for selecting such preprocessing methods are explained in detail.

The system is trained to obtain the best strategy by the reward information received. The reward can show the performance of the identification algorithm and reflect whether the result of the crack recognition is better after a certain image preprocessing method. For each image, the accuracy of crack identification varies with different preprocessing methods. We regard the difference between the accuracy of the crack identification as the measurement of reward. If the accuracy of the transformed image becomes higher, then the intelligent preprocessing-decision system can receive a positive reward. By contrast, the negative reward can be received. That is,

$$r_t = \begin{cases} r_a, & ACC_t > ACC_{t-1} \\ -r_a, & ACC_t \leq ACC_{t-1} \end{cases} \quad (3)$$

where ACC_t represents the accuracy of the crack identification of the current action, and ACC_{t-1} indicates the accuracy of the crack identification of the last action. r_a is a positive value to represent the positive reward, and $-r_a$ is a negative value to represent the negative reward of the action. The structure of the intelligent preprocessing-decision system can be described in Table I.

IV. ACTIONS

The actions of the intelligent preprocessing-decision system are to transform the original image. Different preprocessing

TABLE I
ALGORITHM OF INTELLIGENT PREPROCESSING-DECISION SYSTEM

<p>Input: Maximum training cycle I_{max}, Maximum number of exploratory rounds E_{max}. Maximum number of steps per round S_{max}, update frequency of target network E_{update}; Data amount of each batch M, memory pool P, and decay parameter γ.</p> <p>Initialization: Random initialization of eval network $Eval_{net}$ parameters θ^Q, and target network $Target_{net}$ parameters $\theta^{Q^t} \leftarrow \theta^Q$.</p>
<ol style="list-style-type: none"> 1. while the current training step $< I_{max}$: 2. while the current round step $< E_{max}$: 3. choose a picture from the training set; 4. obtain the feature matrix of the image as the initial state; 5. if the picture cannot be correctly read: 6. pass 7. while the number of current steps $< S_{max}$ and the episode has not finished: 8. select action a_t, satisfying $a_t = \operatorname{argmax} Q(s_t, a, \theta)$ with probability $1 - \epsilon$ or randomly select action a_t with probability ϵ; 9. transform the image by using the selected preprocessing method; 10. take the feature of the transformed image as the next state; 11. calculate the accuracy of the transformed image and obtain the reward; 12. store the memory (s_t, a_t, r_t, s_{t+1}) into memory pool P; 13. randomly select data from P ; 14. calculate the target Q value $q_{target} = r_t + \gamma \times \max(Target_{net}(s_{t+1}, a_{t+1}))$; 15. perform a gradient descent step on $(q_{target} - q_{eval})^2$ with respect to network parameters θ^Q; 16. every E_{update} steps reset $\theta^{Q^t} = \theta^Q$; 17. end 18. end

operations on original images are performed for such transformation. We choose four actions for crack detection and explain the reasons for selecting these operations. These operations include Contrast-limited adaptive histogram equalization, bilateral filtering, morphological opening operator, and direct output of original images (i.e., without preprocessing method), a total of four actions.

A. Contrast-limited Adaptive Histogram Equalization

Histogram counts the probability of each gray level appearing in images [8]. Histogram equalization (HE) utilizes the histogram to adjust the gray value of images for enhancing their global contrast and making their gray value evenly distributed in the histogram. The algorithm finds a gray value mapping:

$$D_b = \frac{D_{max}}{A_0} \sum_{i=0}^{D_a} H_i, \quad (4)$$

where A_0 represents the area of the image (the total number of pixels), D_{max} indicates the maximum gray value of the original image, D_a is the gray value of the original image, D_b is the gray level of the converted image and H_i is the number of i -level grayscale pixels.

Contrast-limited adaptive histogram equalization (CLAHE) [9] is a good complement to the shortcomings of HE. CLAHE [10] can preserve further details for images and minimize noise increase.

Road images can have low or high overall brightness due to camera lighting. In this case, the contrast between cracks and surrounding road surfaces is reduced, and the characteristics of cracks are not obvious, which can cause missing recognition. The use of HE can effectively improve the contrast between cracks and surroundings, enhance the characteristic expressions of cracks, and make crack identification easy. CLAHE algorithm can reduce the noise interference enhanced by HE. CLAHE can also make processed images reduce noise and global contrast as much as possible while improving the local contrast and enhancing crack recognition.

We utilize CLAHE to avoid the problem of excessive brightness in certain areas of images and maximize the crack characteristics on the basis of enhancing the local contrast of images. The basic process is as follows:

1. Split images into $m \times m$ pieces.
2. HE of each block is carried out after the slope of the cumulative return function to avoid the increase of noise limiting the contrast.
3. Bilinear interpolation is used to eliminate the boundary between blocks.

B. Bilateral Filter

The purpose of an image filter is to eliminate noise interference while preserving the image edge information and contour as much as possible. Bilateral filter [11] is an edge-preserving filter based on the Gaussian filter, which only considers the distance between pixels, whereas bilateral filter considers the distance and gray value of pixels:

$$H(y) = \frac{1}{k(y)} \sum_{x \in S} p(x) G_d(x, y) G_r(p(x), p(y)) \quad (5)$$

$$k(y) = \sum_{x \in S} G_d(x, y) G_r(p(x), p(y)) \quad (6)$$

In Formulas (5) and (6), $G_d(x, y)$ represents the Gaussian weight of the distance between pixels x, y and $G_r(p(x), p(y))$ denotes the gray value Gaussian weight between pixels x, y . Bilateral filtering can blur image noise by using different distances of pixels.

Asphalt-stirred stones generally repeatedly flatten Road surfaces; the largest difference between road surface images and other kinds of images is that many salt-and-pepper noises exist in road surface images. Such noises are determined by the inherent characteristics of roads. These salt-and-pepper noises may confuse the crack characteristics and cause the missing recognition or misidentification of the cracks. The bilateral

filter can eliminate these noises to an extent and retain crack information.

C. Morphological Opening Operator

The two most basic operations in morphological transformation are erosion and dilation, and the convolution kernel sliding on original images is used to change pixel values. In the erosion operation, the center element maintains its original pixel value when all the pixels in the kernel are 255, otherwise, it becomes 0. This operation can erode the boundary of the foreground (white pixels). In the dilate operation, the pixel value of the center element is 255 as long as one pixel in the convolution kernel is 255. This operation can swell the foreground boundary, that is, etch away the background (black pixels). In mathematical morphology, using \ominus for erosion and \oplus for dilation is common. The opening operator can be described as follows:

$$A \circ B = (A \ominus B) \oplus B \quad (7)$$

Different sizes of the structural elements in the opening can result in various filtering effects, and the selection of different structural elements can lead to different segmentation, which means that different features are extracted. White noise is the main type of noise in road images. Therefore, the opening operation that can effectively remove white noise is a suitable preprocessing method for road images. White noise is removed after erosion, but black noise is increased, which can affect the accuracy of crack recognition. Images are then dilated to remove black noise, which can retain the crack characteristics. Therefore, the overall noise of images can be reduced, and the crack feature can be highlighted.

V. EXPERIMENTS AND ANALYSIS

The models used in the experiments are all based on Python and Pytorch. Pytorch is the python-version of the torch. A neural network is used in our algorithm, and a few parameters and computations are needed. Thus, we utilize GPU (Graphics Processing Unit) to accelerate the training process [12]. A server with Titan X Pascal GPU, which has a single-precision floating-point computing power of about 11 TFlops, accomplishes the experiments.

A. Dataset Collection

A dedicated digital camera with a vertical downward shooting angle takes the road pavement images. The vehicles with the camera must move along the road at a uniform speed in a straight line, taking a road pavement picture every two meters. The road pavement pictures are then numbered in sequence.

To fully reflect the algorithm performance, we collect approximately 10,000 HD road photos for training and testing. The road pavement dataset is composed of 8-bit single-channel gray-scale images. The image size is 2200×3400 pixels. The images are sliced into non-overlapping blocks, and each block includes 100×100 pixels. Therefore, 748 labeled blocks are found in each image. The blocks containing crack pixels are labeled as "1", whereas those without obvious crack pixels

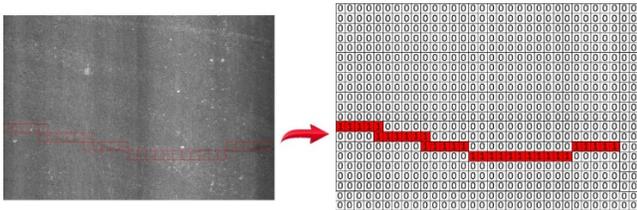


Fig. 3. Road image and its labels

are labeled as “0”. A matrix M containing only 0 and 1 can represent a road pavement image. A sample of the labeling image is illustrated in Fig. 3.

B. Evaluation

1) *Comparison with Single Preprocessing Methods:* The crack identification network is trained first. To evaluate our system, we identify the road cracks by using the crack identification algorithm with a specific preprocessing method and the one with the intelligent preprocessing-decision system. As mentioned in Section III, the preprocessing methods are CLAHE, bilateral filter, and opening. The intelligent system selects the most suitable preprocessing method for images. If the accuracy of the crack detection network improves, then the system can receive a positive reward. Otherwise, the system can receive a negative reward. The system finishes a training step when the accuracy reaches the point of 0.8, or when the last action is chosen as no operation.

The results rely on common definitions, that is, precision, recall, and F-1 score. We evaluate the prediction by calculating the PSD of the models. PSD emphasizes the positive examples in the prediction, which plays a critical role in our task.

The agent collects the total reward in an episode, and we periodically calculate the reward during the training process. We use 7,000 road images as the training dataset. Additional 3,000 pavement images are selected as the test dataset to evaluate network performance. Adam [13] is used as the optimization function. We use LeakyRelu [14] as the activation function to avoid the vanishing gradient problem. The learning rate of the system is set as 0.001. The greedy parameter of the greedy policy regularly changes during the training process. The reward function of the system is set as follows:

$$reward_0 = p_2 - p_1 \quad (8)$$

As described in Section II, p_2 represents the accuracy of the crack identification of the current action and p_1 indicates the accuracy of the crack identification of the last action. In real production, accurately identifying road cracks is crucial. The results in Table II show that the crack identification network with our intelligent preprocessing-decision system achieves higher precision and PSD. Therefore, our method performs better than the one with a single specific preprocessing method. Although the recall is slightly lower than CLAHE and opening, our proposed method achieves an overall F1 score of 78.36%, which is higher than the others. Such a score

TABLE II
RESULTS OF DIFFERENT PREPROCESSING METHODS

-	Precision	Recall	F1	PSD
CLAHE	0.5092	0.9722	0.6683	0.5224
Bilateral	0.533	0.9602	0.6854	0.5307
Opening	0.4256	0.9721	0.5920	0.4298
Our system	0.6576	0.9701	0.7836	0.6585

TABLE III
RESULTS BASED ON DIFFERENT REWARDS

-	Precision	Recall	F1	PSD
$Reward_0$	0.6576	0.9701	0.7838	0.6585
$Reward_1$	0.6738	0.9788	0.7981	0.6774
$Reward_2$	0.8874	0.9792	0.9310	0.8906

suggests that our system is helpful for crack identification to select the most suitable preprocessing method and improve the identification performance.

2) *Comparison with different rewards:* To demonstrate the importance of the reward function, we define the two other reward functions:

$$reward_1 = \begin{cases} 1, & p_2 > p_1 \\ 0, & p_2 = p_1 \\ -1, & p_2 < p_1 \end{cases} \quad (9)$$

$$reward_2 = \begin{cases} 1, & 0.8 > p_2 > p_1 \\ 2, & p_2 > 0.8 \text{ AND } p_2 > p_1 \\ 0, & \text{else} \end{cases} \quad (10)$$

Table III shows that the reward function is crucial to system performance. $reward_2$ function performs better than the two other reward functions because the values of $reward_2$ function are always positive, which improves system learning.

3) *Comparison with other methods:* The last experiments reveal that our system achieves an improved performance with $reward_2$. We evaluate the performance of our system- $reward_2$ with several other existing methods. We select a few pavement images to show the crack detection results of different methods. The brighter color in Fig. 4 is detected as cracks by the corresponding algorithm. Table 4 shows the results of our system and the common edge and semantic segmentation methods: richer convolutional networks for semantic segmentation (RCF) [15], fully convolutional networks for semantic segmentation (FCN) [16], and DeepCrack [17]. Our system improves its performance in all aspects. As shown in Table IV, our system improves the performance by 17.2% relative to DeepCrack, second best, in terms of PSD.

VI. CONCLUSION

In the actual road detection work, the detected pavement images have kinds of background interference. Different pavement images adopt different preprocessing methods. This

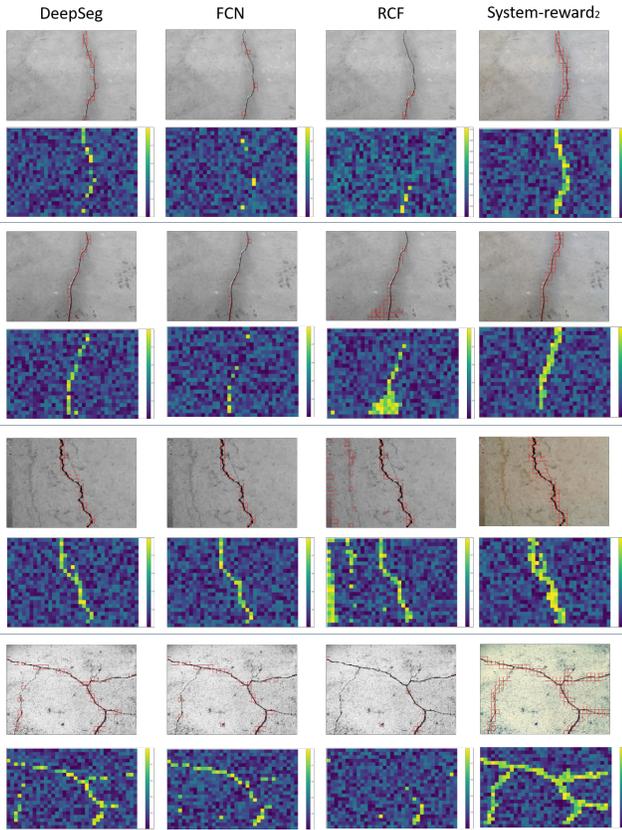


Fig. 4. Crack detection with different algorithms

TABLE IV
RESULTS OF COMPARED METHODS

-	Precision	Recall	F1	PSD
DeepCrack [17]	0.7495	0.6193	0.6782	0.7595
FCN [16]	0.7208	0.4883	0.5821	0.7319
RCF [15]	0.7118	0.6536	0.6814	0.7299
System- $reward_2$	0.8874	0.9792	0.9310	0.8906

paper proposed a method to select the optimal preprocessing method for pavement images in various conditions. The algorithm employs the deep learning method to form the strategy of independent selection of preprocessing method, so as to improve the detection effect of pavement cracks. The experiments and analysis show that our method performs better in the detection process which proves the validity and effectiveness of our method. And we are now planning for further expansion. In this study, we only select certain preprocessing methods. We believe that other preprocessing methods can be added to our system to improve its performance.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No. 2018YFB0803605).

REFERENCES

- [1] Hyunjun Kim, Eunjong Ahn, Soojin Cho, Myoungsu Shin, and Sung Han Sim. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement & Concrete Research*, 99:53–61, 2017.
- [2] Iqbal Zahid, Khan Muhammad Attique, Sharif Muhammad, Shah Jamal Hussain, ur Rehman Muhammad Habib, and Javed Kashif. An automated detection and classification of citrus plant diseases using image processing techniques: A review. *Computers & Electronics in Agriculture*, 153:12–32, 2018.
- [3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [4] Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.
- [5] L. P Kaelbling, M. L Littman, and A. W Moore. Reinforcement learning: A survey. *J Artificial Intelligence Research*, 4(1):237–285, 1996.
- [6] Zhenxin Wang, Sayan Sarcar, Jingxin Liu, Yilin Zheng, and Xiangshi Ren. Outline objects using deep reinforcement learning. *arXiv: Computer Vision and Pattern Recognition*, 2018.
- [7] Abhishek Das, Satwik Kottur, Jose M. F. Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [8] S Mathavan, Akash Kumar, Khurram Kamal, Michael Nieminen, Hitesh Shah, and M M Rahman. Fast segmentation of industrial quality pavement images using laws texture energy measures and k-means clustering. *Journal of Electronic Imaging*, 25(5):053010–053010, 2016.
- [9] M. S. Hitam, W. N. J. H. W. Yussof, Ezmahamrul Afreen Awalludin, and Z. Bachok. Mixture contrast limited adaptive histogram equalization for underwater image enhancement. In *International Conference on Computer Applications Technology*, pages 1–5, 2013.
- [10] Yudong Zhang, Xueyan Wu, Siyuan Lu, Hainan Wang, Preetha Phillips, and Shuihua Wang. Smart detection on abnormal breasts in digital mammography based on contrast-limited adaptive histogram equalization and chaotic adaptive real-coded biogeography-based optimization. *Simulation*, 92(9):873–885, 2016.
- [11] Cristiano Premebida, Luis Garrote, Alireza Asvadi, A Pedro Ribeiro, and Urbano Nunes. High-resolution lidar-based depth mapping using bilateral filter. In *International conference on intelligent transportation systems*, pages 2469–2474, 2016.
- [12] Kristina Doycheva, Christian Koch, and Markus Konig. Gpu-enabled pavement distress image classification in real time. *Journal of Computing in Civil Engineering*, 31(3):04016061, 2017.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer Ence*, 2014.
- [14] Yun Zhang, Qinglong Hua, Dan Xu, Hongbo Li, and Pengfei Zhao. A complex-valued cnn for different activation functions in polarsar image classification. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019.
- [15] Y. Liu, M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5872–5881, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.
- [16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [17] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.