Sensors and Materials, Vol. 35, No. 3 (2023) 813–833
MYU Tokyo

S & M 3212

Algorithm of Mask-region-based Convolution Neural Networks for Detection of Tire Sidewall Cracks

Jui-Chuan Cheng* and Chih-Ying Xiao

Department of Electronic Engineering, National Kaohsiung University of Science and Technology, No. 415, Jiangong Rd., Sanmin Dist., Kaohsiung City 807618, Taiwan

(Received September 12, 2022; accepted January 16, 2023)

Keywords: deep learning, Mask R-CNN, crack detection, image processing, instance segmentation

The tire sidewall is the weakest part of the entire tire. Although the tire sidewall is not directly in contact with the ground, it often undergoes great deformation. Weather, road conditions, and driving habits can also affect the tire life. Cracking is one of the earliest signs of tire aging and deterioration. If a driver does not regularly inspect their vehicle, damage to a tire may remain undetected and an uncontrolled tire explosion may occur. In this study, we use deep-learning-based artificial intelligence computer vision to train a deep neural network model using a large number of digital images to detect tire sidewall cracks instead of traditional sensors, inspection devices, or visual inspection methods. In this study, tire sidewall crack images were preprocessed and annotated using the annotation program VGG Image Annotator (VIA). Residual network 50 (ResNet50) is used as the backbone of mask-region-based convolutional neural networks (Mask R-CNNs). The preprocessing training and test results of our dataset show that the improved Mask R-CNN has better mean accuracy (mAP) and detection accuracy than the original Mask R-CNN and Faster-R-CNN and can not only reduce inspection costs and time, but also improve the efficiency of tire crack analysis.

1. Introduction

The performance of tires has an important influence on the safety of vehicles. Tires in practical use inevitably encounter a variety of complex and harsh road conditions that can cause tire wear, scratches, fatigue cracks, and other defects, making tire quality inspection very important for the tire industry. Tires, most of which are made of synthetic rubber, harden over time. Usually after four years of installation, small cracks appear in the tire sidewall. Cracks in the sidewall are more dangerous than cracks in the tread, because most sidewalls are only half the thickness of the tread, and the steel wire is the weakest at the sidewall. Thus, if a serious crack occurs in the sidewall, the tire should be replaced immediately.

The tire plunger tester system has been used by Taiwan Rubber Research & Test Center (TRC) as the tire quality inspection standard in Taiwan. The test items include the bead unseating test; plunger test; lateral, vertical, and envelope stiffness tests; footprint analysis;

*Corresponding author: e-mail: eagle@nkust.edu.tw
https://doi.org/10.18494/SAM4123

ISSN 0914-4935

dimension measurement; and inflation and vertical pressure tests.[1] There are other similar testing systems used in the tire industry.[2] However, these systems are expensive and not specifically used to detect tire cracks. Behroozinia *et al.* proposed a health monitoring algorithm to predict the location of cracks by comparing the acceleration signals of undamaged and damaged tires obtained from a triaxial accelerometer connected to the tire's inner liner.[3] However, this system requires the installation of additional sensors and is not suitable for practical crack inspection.

Computer vision is a technology that uses image sensors and computers to replace the human eyes to recognize, track, and measure targets, and then uses machine learning or deep learning techniques to achieve accurate recognition through further image processing. Computer vision combined with artificial intelligence has been widely used in various types of sensors, such as light detection and ranging (LiDAR) and radio detection and ranging (RADAR), for sensing the physical environment around self-driving cars,[4–6] and computer vision is often paired with cameras to identify and classify people, objects, and debris. It can significantly improve the shortcomings of traditional manual inspection and realize high-speed and accurate inspection in automatic production. In recent years, many algorithms based on convolutional neural networks (CNNs) have developed rapidly in the field of image recognition. Object detection is a very important field in computer vision, which has undergone major changes since the introduction of deep learning. The most well-known regions with CNN features (R-CNN) is the CNN series, as well as You Only Look Once (YOLO) and single-shot object detection (SSD). The development history of the R-CNN series includes R-CNN in 2014,[7] and Fast-R-CNN[8] and Faster-R-CNN[9] in 2015. As an alternative to the traditional visual methods, the mask-region-based CNN (Mask R-CNN) is a case segmentation model combining fully convolutional networks (FCNs) and Faster-R-CNN.[10]

The use of data augmentation to increase the diversity and amount of training data has become an essential part of deep learning model training for image data. Mikołajczyk and Grochowski proposed a data augmentation approach and used the generated output to train deep learning models, demonstrating the importance of data enhancement in deep learning image classification models.[11] To demonstrate the use of Mask R-CNN in entertainment and how the model performs on different instances during training, Paste and Chickerur used different cartoon episodes featuring the cartoon characters Tom and Jerry and converted them into frames. They selected around 1500 images, which were arranged so that no two images had a similar object size, color, background, camera angle, and so forth. Mask R-CNN was used to segment the cat and mouse images with the purpose of studying the behavior of the model while performing semantic segmentation. This enabled the authors to infer some insights about how the model is trained and to analyze the types of results achieved in a variety of instances during the training phase of the model.[12]

Zhu *et al.* applied Mask R-CNN to automatic tooth detection and segmentation. Among the 100 images obtained from a hospital, 80 images were used as training data, 10 images were used as validation data, and the remaining 10 images were used as testing data. They found that Mask R-CNN also performs well in the segmentation of complex and crowded tooth structures.[13] In the field of image identification, Saha *et al*. proposed a combination of image segmentation and

image restoration to enhance existing road images and create an obstacle-free (vehicle) road dataset. The method used image inpainting to remove vehicles from input images. Mask R-CNN was employed to detect vehicles, and an in-image model was used to remove the detected objects. To improve the efficiency of identification, a morphological transformation method was used. Morphological transformations were also adjusted to the output, and multiple iterations were sometimes required to improve the results.[14]

The use of a deep learning algorithm to automatically detect road potholes was proposed by Rohitaa *et al.* Three deep learning models, namely, CNN, Mask R-CNN, and YOLOv3, were trained and tested using a dataset. The results of the three models were compared using evaluation metrics. Their system also incorporated hardware components for reporting potholes so that action could be taken to repair and maintain roads and to warn drivers of potholes.[15] Zhang *et al.* proposed a segmentation algorithm for vehicle damage detection based on migration learning and an improved Mask R-CNN.[16] They first collected car damage images for preprocessing, then used Labelme to make dataset labels and divided the dataset into a training set and test set. The results showed that compared with the original Mask R-CNN, the improved Mask R-CNN has better average precision value, detection accuracy, and mask accuracy.

To reduce the cost and increase the accuracy of identification, in this study, we applied Mask R-CNN to deep-learning-based artificial intelligence computer vision for object detection and instance segmentation of tire cracks as an alternative to expensive testers and additional sensors.

2. Materials and Methods

In this study, the whole process from a standard crack diagram to the generation of an accurate segmentation diagram is shown in Fig. 1. The set of preloaded object images must be preprocessed to a fixed size, then the fixed-size dataset is labeled to enhance the training data by data augmentation. Unlike the original Mask R-CNN, in this stage, we add a step of image

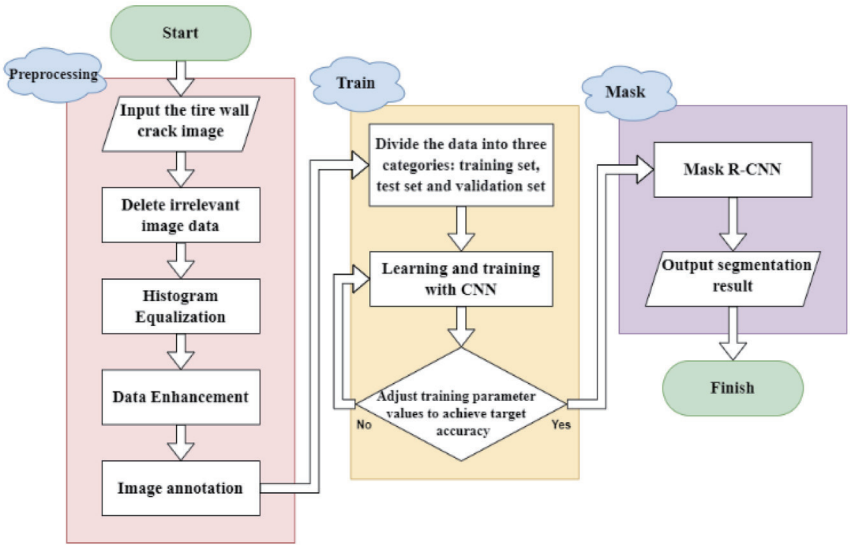


Fig. 1. (Color online) System flowchart.

histogram equalization (HE) to obtain uniformly distributed color intensities to improve object recognition. The processed dataset is fed into a neural network model for training, and the masked region is substituted into the CNN after the training to obtain the final result.

2.1 Mask R-CNN

Mask R-CNN is one of the structures of instance segmentation, which is a prototype of Faster-R-CNN with the addition of branches of the mask. The region of interest (ROI) pooling in Faster-R-CNN is also changed to ROI Align in Mask R-CNN. Because the ROI pooling operation is not based on pixel-to-pixel alignment in the image, an integer that has little influence on the boundary box but a great influence on the accuracy of mask segmentation will be selected. A semantic segmentation branch is added to realize the prediction relationship between the mask and classification. The mask branch only performs semantic segmentation, and the other branch performs the classification prediction operation. Figure 2 shows the workflow of Mask R-CNN.

2.2 Backbone

CNNs have a very wide range of applications in the field of image classification. Theoretically, the deeper the network structure, the better its fitting ability should be. However, it has been experimentally found that when the depth of the network reaches a certain level, the performance of the network decreases rather than increases. ResNet is a residual network, which can be understood as a sub-network that is stacked to form a very deep network. ResNet utilizes skip connections to add the output of the previous layer to the output of the stacked layers, simplifying the training of these networks and improving the efficiency of deep neural networks with many neural layers while minimizing the percentage error. The numbers in the names of CNNs, such as 50 and 101, represent the number of convolutional plus fully connected layers.

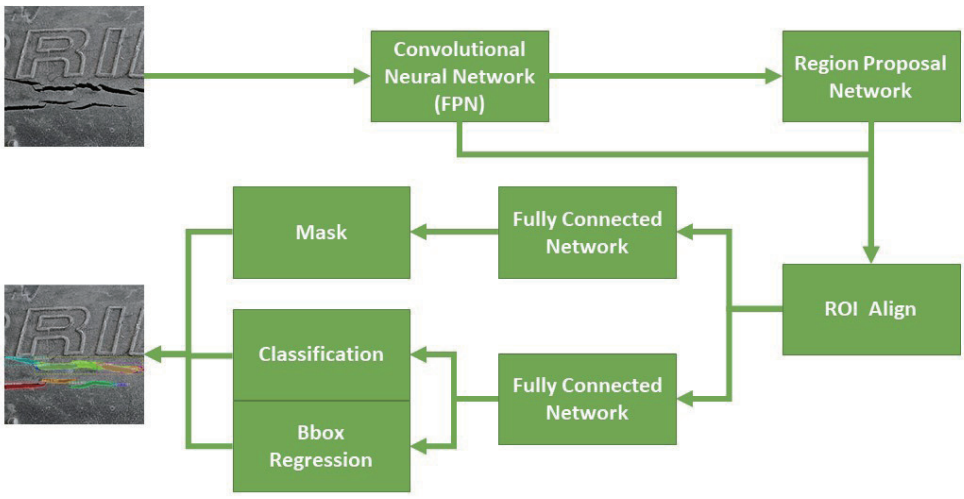


Fig. 2. (Color online) Mask R-CNN workflow.

Generally, residual network 101 (ResNet101) is used for the backbone network of Mask R-CNN, but too many layers will reduce the computing efficiency. The method of crack detection used in this study requires a low number of network layers. Therefore, to improve the computing speed, residual network 50 (ResNet50) is adopted in this study.

Different objects or their features are different in size (or scale) on different images. When we perform object detection, feature extraction at a single scale is often insufficient. Image pyramids are a common tool for solving problems at different scales. Most object detection algorithms only use the top-level features of an image pyramid for prediction; whereas the bottom-level features have less semantic information but accurate object locations, higher-level features have more semantic information but coarser object locations, and the detection accuracy may be low for small objects. The feature proposal network (FPN) algorithm combines the concepts of a top-down pathway and skip connections, making full use of the CNN to generate feature maps at each stage while allowing each feature map in the image pyramid to have higher-quality feature information. The FPN extracts side features at different scales from the ResNet backbone for each layer and uses the high resolution of the lower-layer features and the high semantic information of the higher-layer features for prediction by fusing the features of these different layers.

2.3 Region proposal network

Before the development of Faster-R-CNN, a selective search was used to extract the post-selection box in object detection architectures such as R-CNN and Fast-R-CNN. Compared with a region proposal network (RPN), a selective search is more time-consuming and cannot integrate the whole object detection into one network. The main goal of an RPN is to realize the function of region proposal. This is performed by scanning the image and identifying areas that may contain objects. An RPN runs at very high speed on a GPU. The use of weights in an FPN allows an RPN to efficiently reuse the extracted features and avoid double computation.

2.4 Feature pyramid network

An FPN is used to improve the feature extraction ability of a CNN by fusing feature maps of different scales. To extract features from an FPN, Mask R-CNN obtains input images through a CNN. In Fig. 3, the five gray boxes with different widths and heights depicted as C1 to C5 represent the five stages of ResNet50.[17] A CNN is a collection of filters with training weights and biases. The weights and biases determine the features to be emphasized or ignored in the input image. During training, the weights and biases are optimized to perform a given task on a given dataset. In Mask R-CNN, ResNet50 is used to extract the main features. An increase in width and height indicates an increase in depth and a decrease in resolution of the convolutional layer in ResNet50. The feature map created by the CNN is transferred to the FPN in the next step.[18] Since most CNNs reduce the size of the input image during feature extraction, the FPN recovers the spatial information that may be lost during the reduction. The feature maps extracted by the FPN are recovered to the five feature maps P5 to P1 with different sizes.

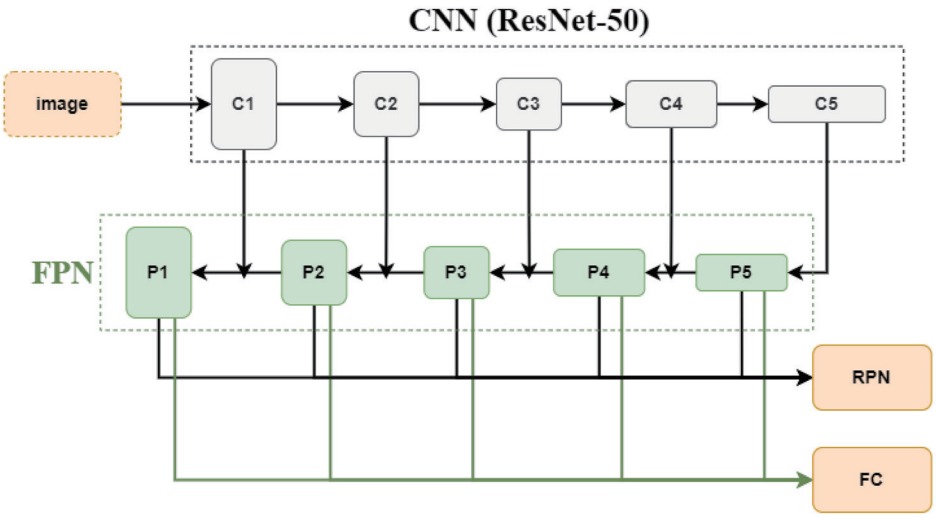


Fig. 3. (Color online) FPN network architecture.

Both Mask R-CNN and Faster-R-CNN use the same RPN. Under the preset conditions, three anchor scales are used. Each scale contains almost all the targets in the training process, but sometimes the training model leads to unnecessary calculations. Faster-R-CNN uses ROI pooling. The role of ROI pooling is to pool the corresponding region in the feature map into a fixed size according to the position coordinates of the preselected box for subsequent classification and bounding box regression operations. Since the position of the preselected box is usually obtained from the model regression, it is generally a floating-point number, and the pooled feature map requires a fixed size. In the discretization process, coordinates are rounded to integers, which leads to inaccuracy of the feature map. ROI Align removes the quantization operation and uses bilinear interpolation to obtain the image values on the pixel points with floating-point coordinates, which preserves the position of the decimal point so that features align correctly with the input. As shown in Fig. 4, ROI Align divides the candidate area into $k \times k$ cells, calculates four fixed coordinate positions in each cell, computes the values of these four positions by bilinear interpolation, and then performs the maximum pooling operation.

The bilinear interpolation performed by ROI Align is shown in Fig. 5.[19] The last interpolation point is P, and the four fixed coordinate positions A_{11}, A_{12}, A_{21}, and A_{22} in the cell are known. B_1 is obtained from A_{11} and A_{21} by linear interpolation. Similarly, B_2 is obtained from A_{12} and A_{22} by linear interpolation, as shown in Eqs. (1) and (2). P is obtained from B_1 and B_2 by linear interpolation, as shown in Eq. (3). The bilinear interpolation performed to find the coordinates (x, y) is given by Eq. (4).[20]

$$f(B_1) \approx \frac{x_2 - x}{x_2 - x_1} * f(A_{11}) + \frac{x - x_1}{x_2 - x_1} * f(A_{21}) \tag{1}$$

$$f(B_2) \approx \frac{x_2 - x}{x_2 - x_1} * f(A_{12}) + \frac{x - x_1}{x_2 - x_1} * f(A_{22}) \tag{2}$$

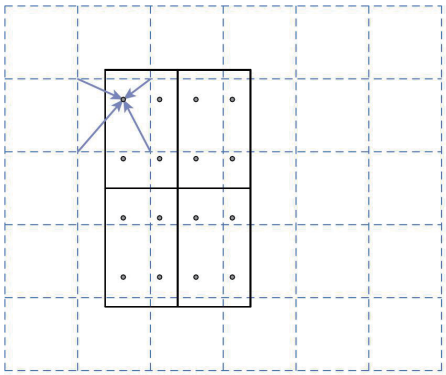


Fig. 4. (Color online) ROI Align.

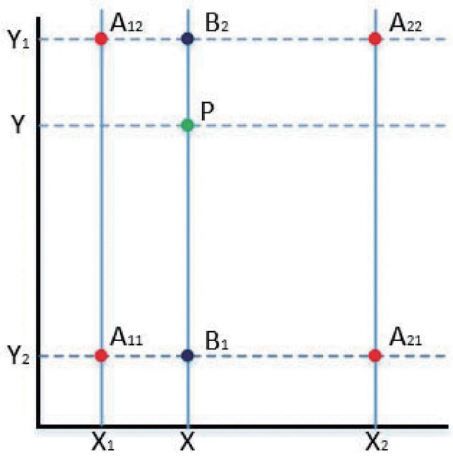


Fig. 5. (Color online) Bilinear interpolation in ROI Align.

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} * f(B_1) + \frac{y - y_1}{y_2 - y_1} * f(B_2) \tag{3}$$

$$\begin{aligned} f(x,y) \approx & \frac{f(A_{11})}{(x_2 - x_1)(y_2 - y_1)} * (x_2 - x)(y_2 - y) + \frac{f(A_{21})}{(x_2 - x_1)(y_2 - y_1)} * (x - x_1)(y_2 - y) \\ & + \frac{f(A_{12})}{(x_2 - x_1)(y_2 - y_1)} * (x_2 - x)(y - y_1) + \frac{f(A_{22})}{(x_2 - x_1)(y_2 - y_1)} * (x - x_1)(y - y_1) \end{aligned} \tag{4}$$

2.5 Gaussian filter

The Gaussian filter, also known as the Gaussian blur or Gaussian smoothing, is a commonly used method to create a smooth image or reduce image noise. It preserves multiple image information after processing.

Equation (5) is the formula of the specific Gaussian filter function, where σ is the standard deviation. If σ is smaller, then the center coefficient of the generated template becomes larger and the surrounding coefficients become smaller; thus, the smoothing effect on the image is not obvious. In contrast, when σ is larger, then the difference between the individual coefficients of the generated template is not significant, making the template more similar to the mean template, and the smoothing effect on the image is more obvious.

k is the size of the required filter template, and a common template size is a $(2k + 1) \times (2k + 1)$ matrix with an odd number of rows and columns. The minimum template size is 3×3 when $k = 1$.

$$H_{x,y} = \frac{1}{2\pi\sigma^2} \times \exp\left(-\frac{(x - k - 1)^2 + (y - k - 1)^2}{2\sigma^2} \right) \tag{5}$$

Taking a 3 × 3 Gaussian filter template as an example, x and y represent the numbers of rows and columns, respectively, and the coordinates of each position (x, y) are shown in Fig. 6. The coordinates of each position are substituted into $H_{x,y}$, each value obtained is arranged according to the position, and the filter kernel is obtained.

The following is a demonstration of a 3 × 3 Gaussian filter template calculation. When the standard deviation σ is 1.5, by substituting this value into Eq. (5), the results in Table 1 are normalized to those in Table 2.

2.6 HE

HE, which can evenly distribute the color intensity of image pixels, can give an image richer colors and improve its contrast, preventing the image from being overexposed or too dim. HE is often used to optimize overexposed or dim images. Its main purpose is to evenly map the color intensity of the original image pixels to the entire color range to obtain a uniformly distributed color intensity image. In this paper, we modified the original Mask R-CNN process by adding an HE step to the image set to improve the detection recognition of the graphs.

Equation (6) is the formula used for HE,[21] where x is the current pixel value, cdf represents the cumulative distribution function, S is the greyscale value (0–255), and L and W are the height and width of the image, respectively.

$$h(x) = round\left(\frac{cdf(x) - cdf_{min}}{(L \times W) - cdf_{min}}\right)(S-1) \tag{6}$$

In Fig. 7, the left side is the color intensity range before equalization, and the right side is the color intensity range after equalization. The x-axis of the greyscale histogram shows the

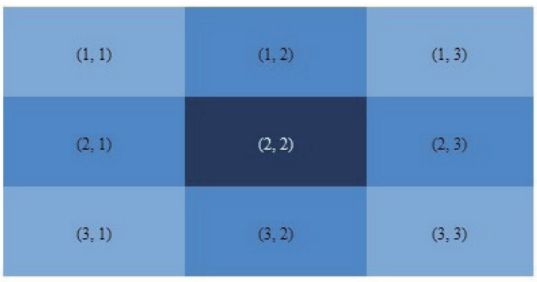


Fig. 6. (Color online) 3 × 3 Gaussian filter template.

Table 1
Gaussian filter with σ = 1.5, k = 1.

0.0453542	0.0566406	0.0453542
0.0566406	0.0707355	0.0566406
0.0453542	0.0566406	0.0453542

Table 2
Gaussian filter (normalized) with σ = 1.5, k = 1.

0.0947416	0.118318	0.0947416
0.118318	0.147761	0.118318
0.0947416	0.118318	0.0947416

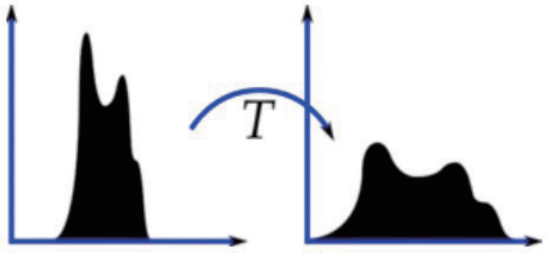
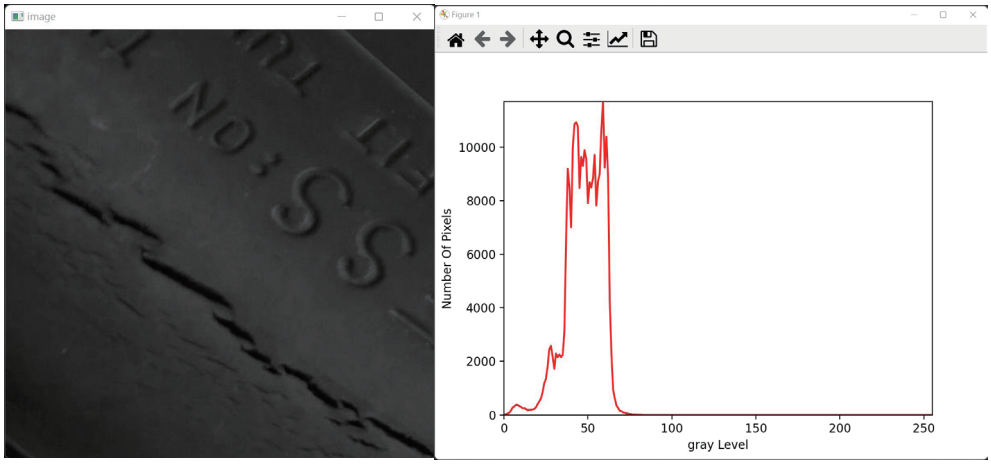


Fig. 7. (Color online) Color intensity range before and after HE.



(a) (b)

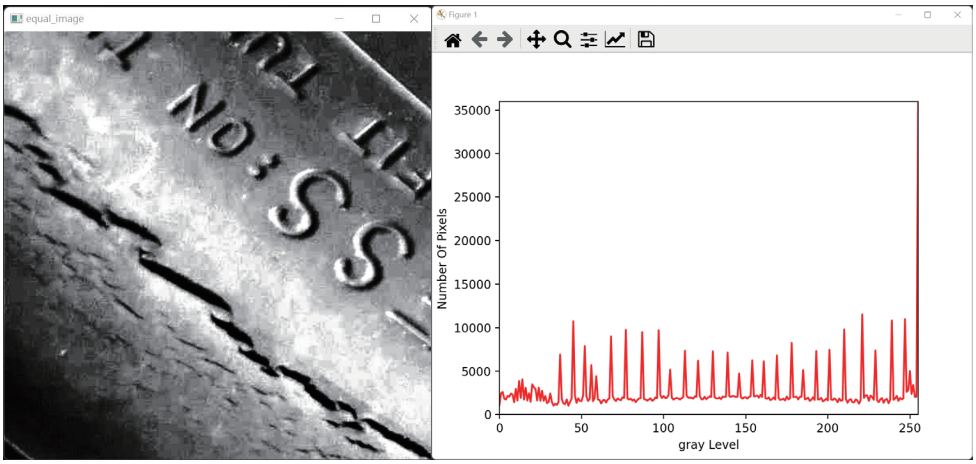
Fig. 8. (Color online) (a) Original image and (b) greyscale histogram.

brightness of the pixels (0–255) and the *y*-axis shows the number of pixels of a given brightness. The color intensity of the original image pixels is mapped uniformly over the entire color range.

Figure 8(a) is an image of a tire with cracks, and Fig. 8(b) shows that its brightness is concentrated around 50, indicating that it is a dark image. Figure 9(a) shows the HE results; the image is obviously brighter than that in Fig. 8(a). Figure 9(b) shows that the brightness is evenly distributed, with a maximum value appearing above a value of 250, indicating that the bright area accounts for a large proportion of the image.

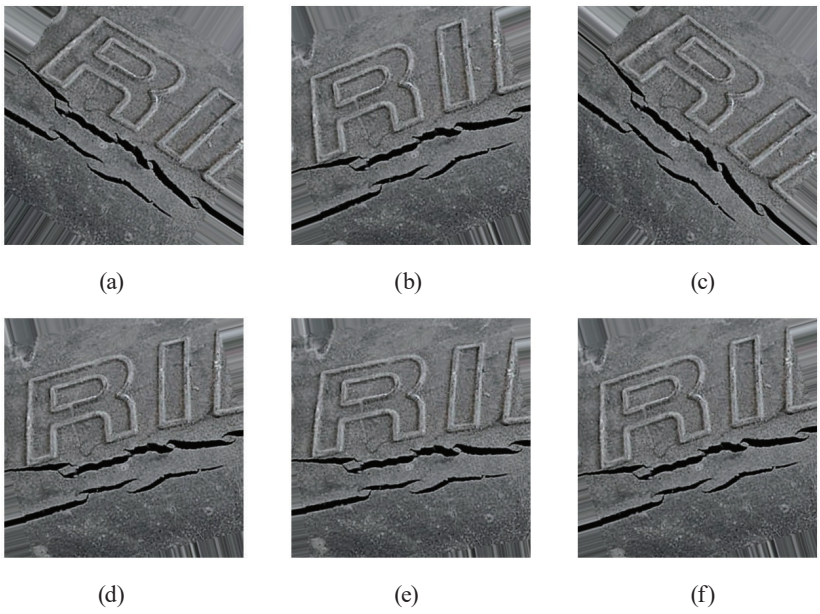
2.7 Data augmentation

The general reason for overfitting is that the training sample data is insufficient or the model is overtrained; thus, a model trained with the sample data has low prediction ability for data outside the sample. The purpose of data augmentation is to create more data by transforming the existing dataset by flipping, panning, adjusting the brightness, and adjusting the scale to create more data. Although the same images are used, the machine regards the transformed images as new images.



(a) (b)

Fig. 9. (Color online) (a) Image and (b) greyscale histogram after equalization.



(a) (b) (c)

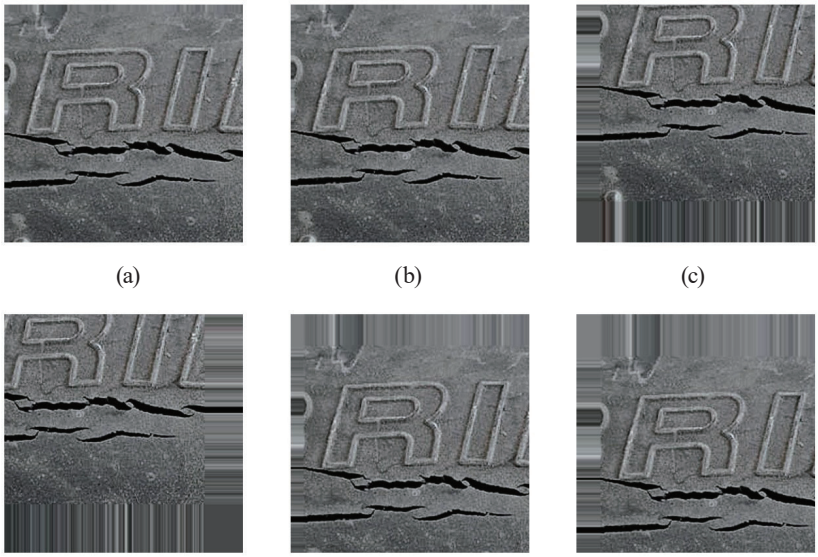
(d) (e) (f)

Fig. 10. Images randomly rotated by ±40°.

The tire dataset used in this paper is from a dataset of 554 tire cracks provided by Yaswanth Gali of Kaggle, which is a widely recognized machine learning and data science community with a large number of open datasets for scientific work. By preprocessing to exclude many cracks unrelated to the tire sidewall, the original 554 images were reduced to 234 images. To avoid overfitting and reduced prediction accuracy due to overfitting, the training dataset was augmented by horizontally flipping the images and other transformations.

Figures 10(a)–10(f) show randomly rotated images with different rotation angles generated to enhance the image effect with a maximum deviation of ±40°, which are used to simulate photography from different angles.

Random shifts can randomly generate enhanced images with horizontal and vertical differences between their centers. The images are randomly shifted by their length and width with a maximum offset of 20% as shown in Figs. 11(a)–11(f). Random shears are used to randomly generate enhanced images without moving the vertical axis, as shown in Figs. 12(a) and 12(b). The images in Fig. 13 were generated by shearing the image counterclockwise by different angles. Random zooms can randomly generate different zoom ratios to enhance the diversity of images, as shown in Figs. 14(a)–14(f). Finally, random flips can randomly generate horizontal and vertical flips to enhance the diversity of images, as shown in Figs. 15(a)–15(f).



(a) (b) (c)

(d) (e) (f)

Fig. 11. Images randomly shifted by up to 20%.

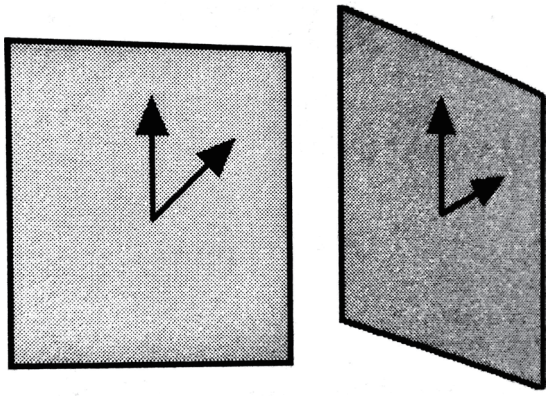
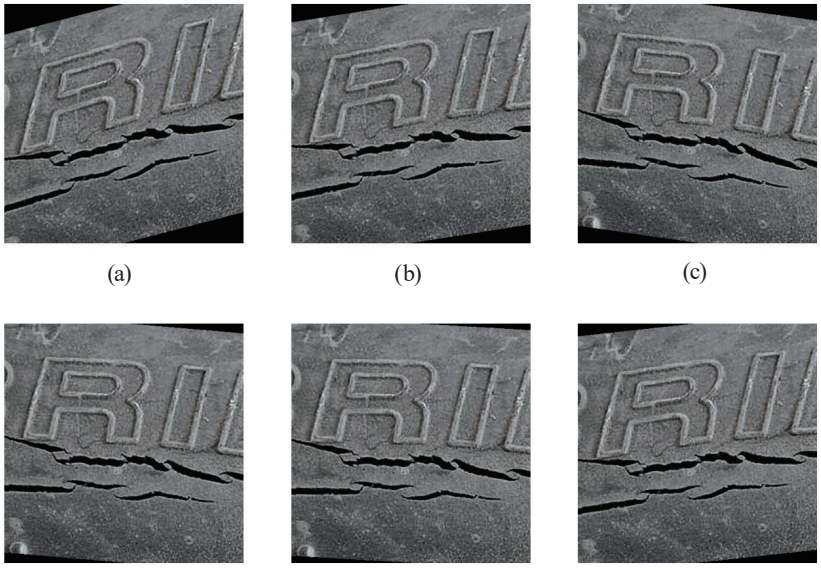


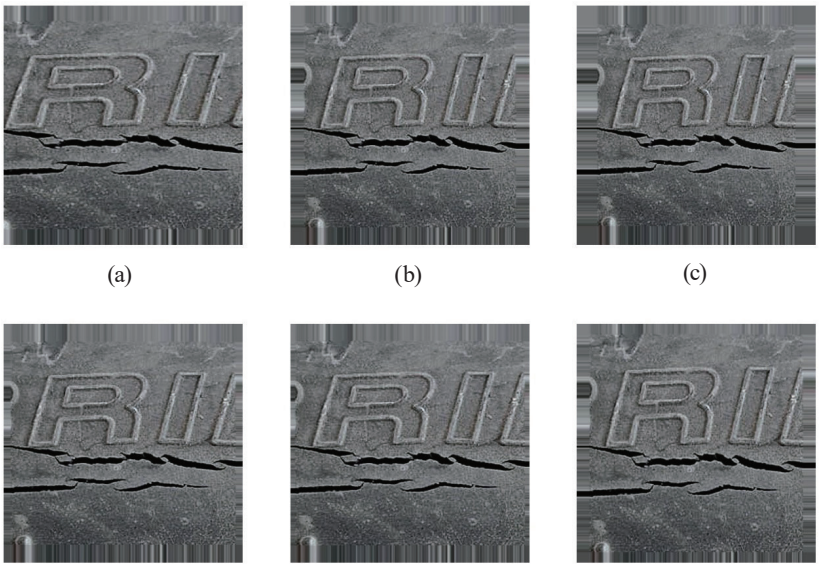
Fig. 12. (a) Images before and (b) after random shears.



(a) (b) (c)

(d) (e) (f)

Fig. 13. Randomly sheared images.



(a) (b) (c)

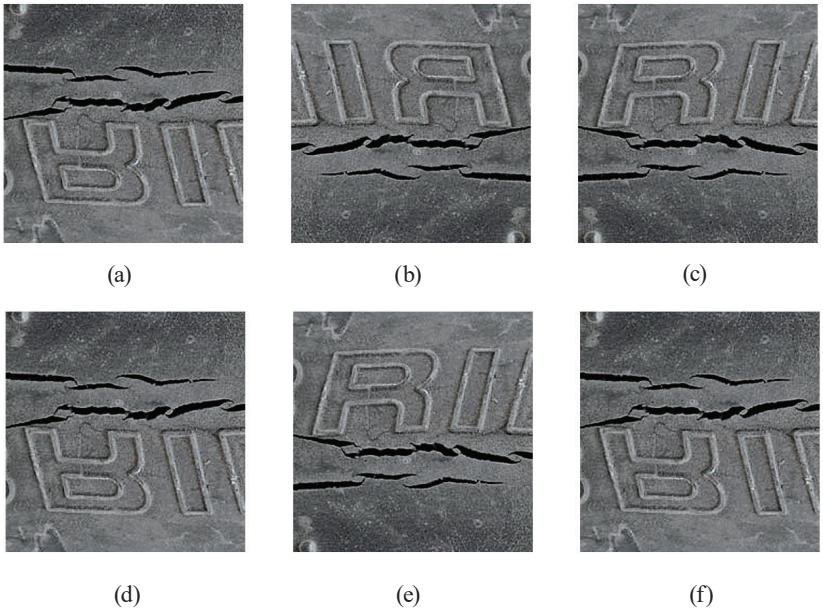
(d) (e) (f)

Fig. 14. Randomly zoomed images.

3. Experimental Results

3.1 Experimental data

Figure 16 shows part of the dataset of healthy tire sidewalls, and Fig. 17 shows part of the dataset of tire sidewalls with cracks.



(a) (b) (c)

(d) (e) (f)

Fig. 15. Randomly flipped images.

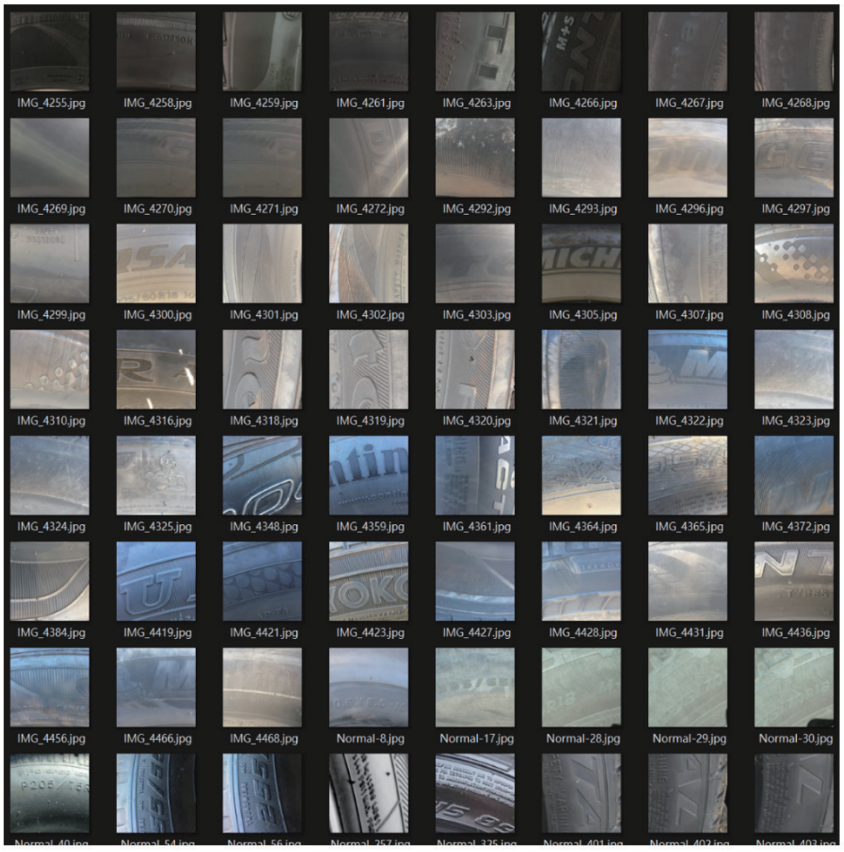


Fig. 16. (Color online) Healthy tire sidewalls.

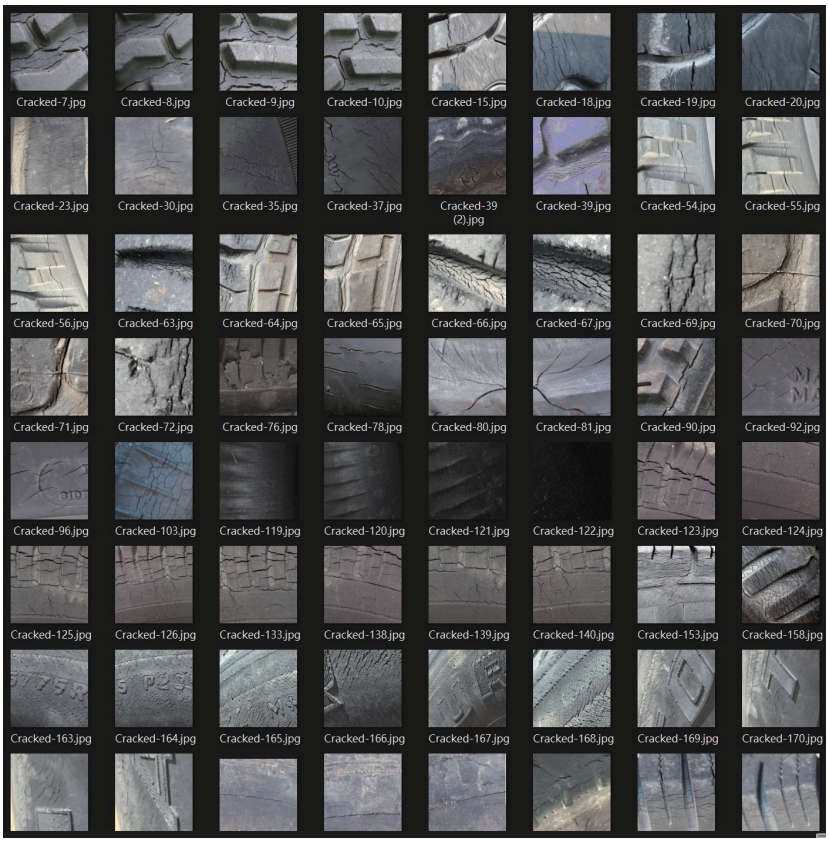


Fig. 17. (Color online) Tires with cracks in the sidewall.

3.2 Data augmentation results

In the machine learning, the entire dataset cannot be used for training, and some of the datasets must be retained as test data to evaluate the final performance of the model. Holdout validation is a static method of dividing datasets into training sets, validation sets, and test sets according to a fixed ratio. In this study, holdout validation is used with the dataset divided into a fixed 8:1:1 ratio of training, validation, and test sets as shown in Fig. 18.[22]

Mean accuracy (mAP) is currently the most commonly used metric to measure the performance of an object detection model. It is based on the sub-metrics of the confusion matrix, union intersection (IoU), recall, and precision. A confusion matrix has four attributes: true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs). Object detection mainly requires manual labeling, which is usually marked as a rectangular bounding box called the ground truth, and the bounding box generated by deep learning identification is called the predicted bounding box. IoU is equal to the area of the intersection of two bounding boxes divided by the area of the union, and this value is between 0 and 1. If IoU is greater than a threshold (usually 0.5), the target is considered as a TP; otherwise, it is considered as an FP. A higher IoU indicates that the predicted bounding box coordinates are very similar to the ground-truth box coordinates. The proportion of predicted targets that are actually targets is called the

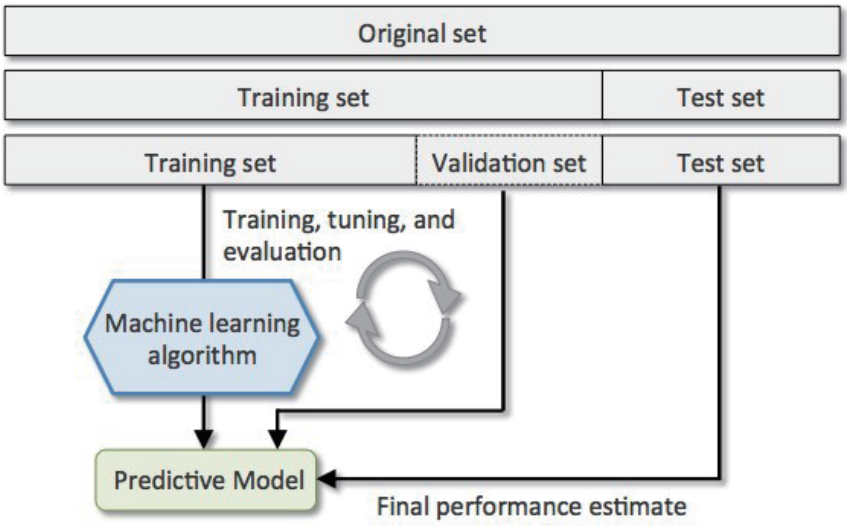


Fig. 18. (Color online) Workflow of holdout method.

precision [Eq. (7)], which indicates whether the result is accurate when the model predicts the target. The actual proportion of targets that are correctly predicted to be targets is called the recall [Eq. (8)], which indicates the ability of the model to find the targets. AP is the area under the precision–recall curve. Since precision and recall are both between 0 and 1, AP is also between 0 and 1. As shown in Eq. (9), AP divides recall into 11 points, namely {0, 0.1, ..., 0.9, 1.0}, and finds the maximum precision of these points for averaging. mAP is the average of the AP of each class, which is expressed as Eq. (10).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.2, \ldots, 1\}} P(r) \tag{9}$$

$P(r)$: interpolated precision that takes maximum precision over all recalls greater than r.

$$mAP = \frac{1}{k} \sum_{i}^{k} AP_i \tag{10}$$

After data augmentation, the number of images in the dataset was increased from 234 to 5850. The original training set was expanded from 188 to 4680, the validation set was expanded

Table 3
Comparison table of mAP before and after data augmentation.

	Training	Validation	Test	*mAP*
Before data augmentation	188	23	23	0.56
After data augmentation	4680	585	585	0.82

from 23 to 585, the test set was expanded from 23 to 585, and the mAP value was increased from 0.56 to 0.82. The results in Table 3 show that mAP of the training data was greatly improved from 0.56 to 0.82. This demonstrates that data augmentation effectively enhances mAP and does not cause overfitting due to insufficient data.

3.3 Experimental results

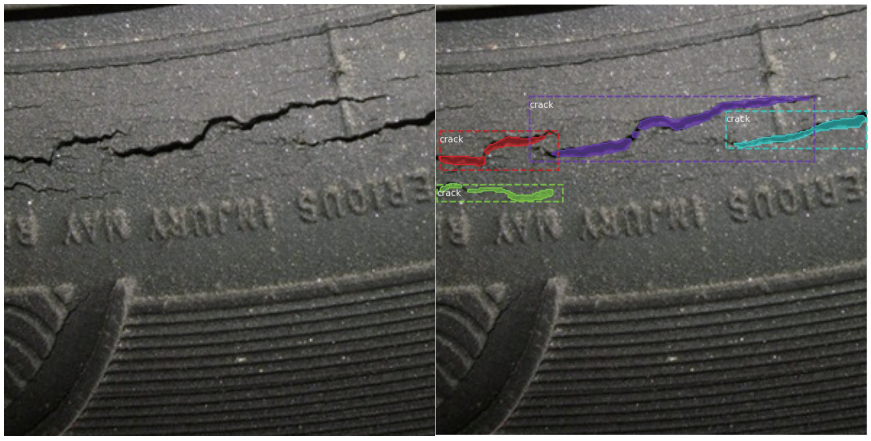
The neural network of supervised learning must be annotated with images before training, so that the trained model knows the location of the recognition target and can perform classification. VGG Image Annotator (VIA) is an open-source image annotation tool developed by the Visual Geometry Group. It can annotate rectangles, circles, ellipses, polygons, points, and lines. In this study, we use VIA to annotate images before training the model. Because of the irregular shape of the cracks, we use polygons to mark the target area, so that the target area and the background are completely separated. This allows better differentiation between the target area and the background during model training.

Figures 19–21 show images of tire cracks from the test datasets that have different brightnesses, shapes, and sizes. The experimental results show that Mask R-CNN successfully detected multiple cracks. As shown in Fig. 19, all cracks were detected without FPs in a bright environment, and detection was not affected by the text on the sidewalls. Figure 20 shows that among the complex cracks, some small cracks were not precisely segmented due to the resolution limitation. In Fig. 21, the original image of the tire sidewall was dark, and the brightness of many cracks was too close to that of the background, so even when HE was used, the detection performance was limited. Although the positioning accuracy was high, the actual segmentation effect was poor.

Image classification determines whether the input image contains a specific object. Object localization is further marked by a bounding box. By using ground truth and predicted bounding boxes, indices of precision and recall can help us evaluate the performance of the model. In this case, precision means the proportion of all cracks predicted by the system that are indeed cracks, and recall means the proportion of all actual cracks that are correctly predicted by the system. Precision and recall are influenced by each other. Ideally, both should be high, but in general, if the precision is high, the recall is low, and if the recall is low, the precision is high. Nine, four, and two cracks were detected in the images in Figs. 19–21, respectively. The cracks in each image are given a letter in alphabetical order, and their localization precision, localization recall, classification precision, and classification recall are calculated. Table 4 shows the performance evaluation results of the trained Mask R-CNN model as percentages (higher is better). These four evaluations are all high, showing that our proposed system model achieves good results.

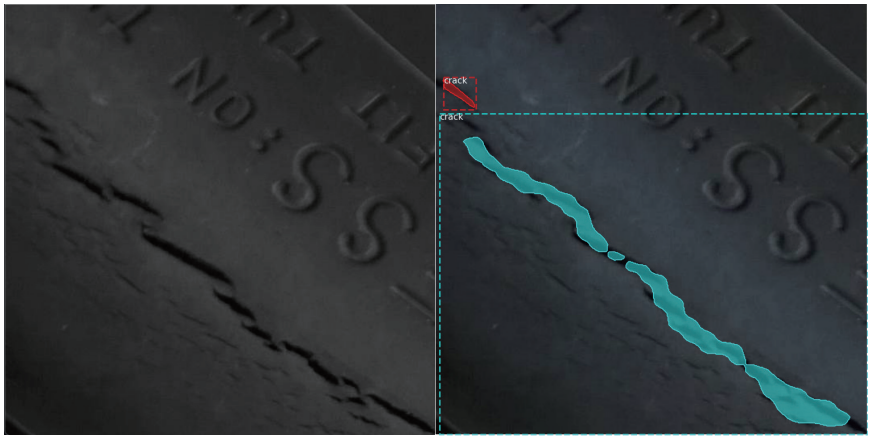
(a) (b)

Fig. 19. (Color online) Crack detection result of trained Mask R-CNN in bright-light environment (I). (a) Original image and (b) crack detection.



(a) (b)

Fig. 20. (Color online) Crack detection result of trained Mask R-CNN in bright-light environment (II). (a) Original image and (b) crack detection.



(a) (b)

Fig. 21. (Color online) Crack detection results of trained Mask R-CNN in low-light environment. (a) Original image and (b) crack detection.

Table 4
Evaluation results.

		Localization precision	Localization recall	Classification precision	Classification recall
Fig. 19 cracks	A	92.63	100	86.28	92.35
	B	98.32	86.48	87.52	81.53
	C	100	100	93.48	87.56
	D	100	100	89.63	88.25
	E	94.86	92.38	81.26	80.29
	F	100	100	93.51	95.21
	G	100	100	82.42	81.65
	H	97.42	95.07	86.39	84.28
	I	100	100	91.54	86.76
	Average	98.13	97.1	88	86.43
Fig. 20 cracks	A	90.27	93.58	91.32	83.27
	B	100	98.48	86.59	94.63
	C	97.56	100	91.43	87.59
	D	99.04	100	95.06	82.65
	Average	96.71	98.01	91.1	87.03
Fig. 21 cracks	A	88.53	91.43	92.41	84.59
	B	82.48	97.02	77.12	92.2
	Average	85.5	94.22	84.76	88.39
	Overall average	93.44	96.44	87.95	87.28

Table 5
Comparison of evaluation results.

	Modified Mask R-CNN	Original Mask R-CNN	Original Faster-R-CNN
mAP (%)	91	82	73
Localization precision (%)	93.4	90.1	89.7
Training time (h)	10.5	11	8
Test time (s)	5.5	10.2	4.5

3.4 Comparison of evaluation results

To demonstrate the effectiveness of the system, we compared the proposed algorithm with the original Mask R-CNN and the original Faster-R-CNN, and the experimental results are shown in Table 5. The test environment for this experiment is as follows: CPU: Intel i7-9700@4.5G, RAM: DDR4 16G@2666 MHz, GPU: Nvidia GeForce GTX1050 2G, OS: Windows 10 Professional (x64), programming language: Python 3.6.

The training time of the proposed modified Mask R-CNN is close to that of the original Mask R-CNN but slower than that of Faster-R-CNN. For model detection, it is much faster than the original Mask R-CNN network but still slightly slower than Faster-R-CNN. However, it has the highest performance for mAP and localization precision, as shown in Fig. 22. This result shows that our HE preprocessing of the images can indeed improve the performance of the system model.

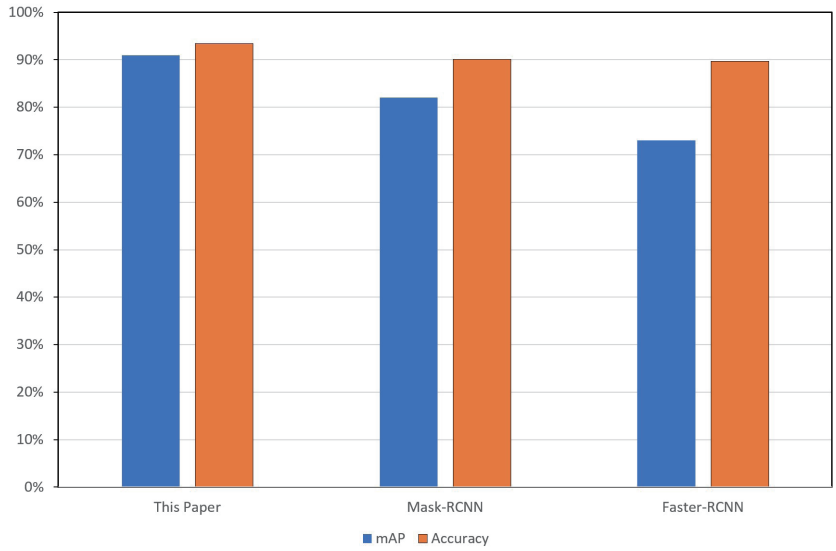


Fig. 22. (Color online) Histograms of evaluation metrics.

4. Conclusions

In this study, we propose a modified Mask R-CNN algorithm for the detection of tire cracks. Unlike the traditional Mask R-CNN algorithm, we add HE to the data preprocessing of the image set to improve the recognition ability and use data augmentation to expand the data of the image set fivefold to avoid overfitting. The experimental results show that mAP of the training data is improved by 46.4%. In the tire crack detection experiment, good results were obtained for both the precision and recall evaluation metrics, and about half of the localization evaluation metrics reached 100% under different light scenarios. Our proposed method was also compared with the original Mask R-CNN and Faster-R-CNN and achieved the best results in both mAP and localization accuracy, thus demonstrating the good performance of the proposed method.

There are still some shortcomings of the proposed method. For example, although HE can improve the problem of too dark or too bright images, it may not be possible to separate objects from the background if the object tones are too similar to the background color. In addition, since open-source datasets are for machine learning training under ideal conditions, it is sometimes impossible to accurately segment cracks without obvious damage. In the future, we hope to increase the accuracy of crack segmentation by increasing the diversity of the dataset and improving the edge contours used in image enhancement.

Acknowledgments

Author Contributions: J.-C. Cheng participated in the design and performed the investigation, methodology, formal analysis, supervision, original manuscript preparation, and revision of the manuscript; C.-Y. Xiao helped with software design and analysis of the data. Both authors read and approved the final manuscript.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1 Tire Plunger Testing System: https://www.asiamachinery.net/supplier/product_details.asp?ProID=1378&SupID=152 (accessed November 2022).
2 Inspection Systems for the Tire Industry: https://www.micro-epsilon.com/download/products/cat--systems--rubber-tire--en.pdf (accessed November 2022).
3 P. Behroozinia, S. Taheri, and R. Mirzaeifar: Sage **18** (2019) 390. https://doi.org/10.1177/1475921718756602
4 Y. Zhang, L. Wang, X. Jiang, Y. Zeng, and Y. Dai: Robotica **40** (2022) 38. https://doi.org/10.1017/S0263574721000369
5 M. I. Hussain, S. Azam, M. A. Rafique, A. M. Sheri, and M. Jeon: IEEE Trans. Veh. Technol. **71** (2022) 5971. https://doi.org/10.1109/TVT.2022.3161378
6 Z. Wei, F. Zhang, S. Chang, Y. Liu, H. Wu, and Z. Feng: Sensors **22** (2022) 2542. https://doi.org/10.3390/s22072542
7 R. Girshick, J. Donahue, T. Darrell, and J. Malik: Proc. 2014 IEEE Computer Vision and Pattern Recognition Conf. (IEEE, 2014) 580–587. https://doi.org/10.1109/CVPR.2014.81
8 R. Girshick: Fast R-CNN (2015). https://doi.org/10.48550/arXiv.1504.08083
9 S. Ren, K. He, R. B. Girshick, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2017) 1137. https://doi.org/10.1109/TPAMI.2016.2577031
10 K. He, G. Gkioxari, P. Dollár, and R. Girshick: Proc. 2017 IEEE Computer Vision Conf. (IEEE, 2017) 2980–2988.https://doi.org/10.1109/ICCV.2017.322.
11 A. Mikołajczyk and M. Grochowski: Proc. 2018 Int. Interdisciplinary PhD Workshop (IEEE, 2018) 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338
12 A. S. Paste and S. Chickerur: Proc. 2019 2nd Intelligent Computing, Instrumentation and Control Technologies Conf. (IEEE, 2019) 191–196. https://doi.org/10.1109/ICICICT46008.2019.8993224
13 G. Zhu, Z. Piao, and S. C. Kim: Proc. 2020 Artificial Intelligence in Information and Communication Conf. (IEEE, 2020) 070–072. https://doi.org/10.1109/ICAIIC48513.2020.9065216
14 P. K. Saha, S. Ahmed, T. Ahmed, H. Islam, A. Imran, A. Z. M. T. Kabir, and A. M. Mizan: Proc. 2021 Intelligent Technologies Conf. (IEEE 2021) 1–6. https://doi.org/10.1109/CONIT51480.2021.9498505
15 R. Rohitaa, S. Shreya, and R. Amutha: Proc. 2021 Fourth Electrical, Computer and Communication Technologies Conf. (IEEE, 2021) 1–5. https://doi.org/10.1109/ICECCT52121.2021.9616703
16 Q. Zhang, X. Chang, and S. B. Bian: IEEE Access **8** (2020) 6997. https://doi.org/10.1109/ACCESS.2020.2964055
17 S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He: Proc. 2017 Computer Vision and Pattern Recognition Conf. (IEEE, 2017) 5987–5995. https://doi.org/10.1109/CVPR.2017.634
18 T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and Proc. 2017 IEEE Computer Vision and Pattern Recognition (IEEE, 2017) 936–944. https://doi.org/10.1109/CVPR.2017.106
19 K. He, G. Gkioxari, P. Dollár, and R. Girshick: IEEE Trans. Pattern Anal. Mach. Intell. **42** (2020) 386. https://doi.org/10.1109/TPAMI.2018.2844175
20 J. Shi, Y. Zhou, and W. X. Q. Zhang: Proc. 2019 Chinese Control Conf. (IEEE, 2019) 8519–8524. https://doi.org/10.23919/ChiCC.2019.8866278
21 Histogram Equalization: https://www.nxp.com/docs/en/application-note/AN4318.pdf (accessed November 2022).
22 Cross-Validation: https://blog.csdn.net/lanchunhui/article/details/50522424 (accessed November 2022).

About the Authors

Jui-Chuan Cheng received his B.S. degree from Feng Chia University, Taiwan, in 1984, his M.S. degree from National Cheng Kung University, Taiwan, in 2003, and his Ph.D. degree from National Kaohsiung University of Applied Science and Technology, Taiwan, in 2012. He has been an associate professor at National Kaohsiung University of Science and Technology since 2019. His research interests include applications of microcontrollers, intelligent algorithms, and digital signal processing. (eagle@nkust.edu.tw)

Chih-Ying Xiao received his B.S. degree from Shu-Te University, Taiwan, in 2019 and his M.S. degree from National Kaohsiung University of Science and Technology, Taiwan, in 2022. He has been an engineer at Taiwan Semiconductor Manufacturing Co. since 2022. His research interests focus on signal and image processing. (F109152129@nkust.edu.tw)