Adaboost-based Crack Detection Method for Pavement

Shibing Wang¹, Fan Yang², Yunmei Cheng³, Yuanyuan Yang⁴ and Yuzeng Wang*

¹-³ School of Mechanical Engineering, University of Jinan, Jinan, Shandong, 250022, CN

⁴ School of Electrical Engineering, University of Jinan, Jinan, Shandong, 250022, CN

*Corresponding author’s e-mail: wang201907@gmail.com

Abstract. In highway pavement, pavement crack is the most common and harmful disease, which has seriously affected road quality. This article reviews the current texture feature processing algorithms and analyses the principle of Adaboost algorithm from aspects of image processing, texture feature extraction and recognition. And we use OpenCV to train classifiers and identify cracks in cement pavements and asphalt pavements from a software perspective. Finally, this paper uses Adaboost algorithm to detect the cracks on cement pavement and asphalt pavement and achieve the desired effect.

1. Introduction

Affected by the quality of road materials and the level of design and construction, there are often five common diseases such as cracks, track, looseness, frost heave, and sinking in highway pavements. Cracking is the mainly defect present in the road surface. These diseases have seriously affected the driving speed and safety. It also increases the car’s wear rate and shorten the service life of pavements[1][2]. That is why many works have been proposed to detect and classify pavement cracks. Most highways require regular inspections of the road surface quality. The detection process mainly depends on labour which is low efficiency and seriously restricts the next steps [3]. In the past thirty years, the detection of pavement cracks is still a hot topic. The reason is that the complicated and varied road environment (oil spots, object shadows, reticle, etc.) has a great influence on image recognition. In many scholars’ research and engineering practice, the main task is to recognition the cracks in high-quality pavement images [4]. And line scan camera is used in image acquisition which puts forward higher requirements for lighting conditions. In this paper, image acquisition is carried out by using an area-array camera, which reduces the requirement of the camera environment. In addition, machine learning has been extensively applied in many areas, including road inspection and monitoring. A road surface image can be divided into blocks which could be subsequently described by different features and classified by some advanced machine learning algorithms [5]. We use area-array cameras and Adaboost learning algorithms to realize the identify and classify road cracks

2. Methodology

In this paper, the pavement crack detection system is based on statistical learning method, as shown in figure 1. The whole system can be divided into two procedure, sample training procedure and image detection procedure. In the sample training procedure, we used the Adaboost learning algorithm proposed in [6] to extract and train the Haar feature texture from the positive and negative samples of road cracks. The trained pavement crack classifier is called by the detection procedure. After the
system starts to detect, the input image will be preprocessed first, such as white balance, histogram balance, etc. Then, the preprocessed image will be scanned by the search window with different scaling coefficients, and the Haar feature vector in the window will be extracted in the scanning process. The extracted vector is input into the trained crack classifier to determine whether there are cracks in the scanning window.

Figure 1. General block diagram of a crack detection system based on statistical learning method

2.1. Haar Features
The Adaboost algorithm can be divided into the following three parts: (1) By using Haar feature to detect the crack texture, the integral image method can be used to calculate the Haar feature value in the image quickly, (2) The AdaBoost algorithm was trained to obtain a large number of weak classifiers, which were combined into strong classifiers according to different weighted overlapping values, (3) All the strong classifiers obtained by training are cascaded to form a cascading hierarchical classifier. This structure can effectively improve the detection speed of the classifier.

Figure 2. The set of Haar Features for Adaboost features

The principle of Haar feature operator, as in (1), is to use the sum of pixels in the white area of the feature block minus the sum of pixels in the black area.

\[
featureValue(x) = weight_{white} \sum_{p \in white} p - weight_{black} \sum_{p \in black} p
\]  

(1)
Where $featureValue$ is the Haar feature, and $weight_{\text{white}}$ is the weight factor of the white area, and the $weight_{\text{black}}$ is the weight factor of the black area.

2.2. Integral Image

P. Viola[7] proposes an integral image algorithm to calculate the gray sum of rectangles. The principle of the integral image method is to use the principle of arrays to save the calculated characteristic pixel values in advance and call them directly at the next repeated calculation. The Haar feature is accelerated by the integral image method. The definition of the integral image is shown in formula (2).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Where $ii(x, y)$ is the integral value at the integral image point $(x, y)$ and $i(x', y')$ is the grayscale value of the image at the point $(x', y')$. Using the integral image method does not need to recalculate the sum of the pixel gray values in the rectangle each time when calculating the rectangular features. Instead, it only needs to index the integral image values of several corresponding points in the rectangle to calculate the rectangular feature values. The integral image only needs to traverse the image to quickly find the rectangular feature value. Any rectangular feature can be obtained by adding or subtracting, which improves the calculation speed of image feature values.

2.3. Classifier for sub-image

The Adaboost algorithm is implemented in an iterative manner. The first problem that the algorithm needs to solve is the design of weak classifiers, that is, different positive sample sets are used to implement the training of the same trainer, and then the resulting weak classifiers are combined to obtain a strong classifier. The constraints of the composed cascaded classifiers will increase gradually. In this way, most non-crack images will be eliminated in the previous stages, improving the efficiency of the AdaBoost algorithm. The image that may contain cracks in the first stage of the cascaded classifier is passed to the second stage classifier for detection, which can significantly improve detection accuracy. Therefore, any feature value that does not conform to the classifier will cause the sub-image to be eliminated immediately. The process is shown in figure 3.

Compared with other learning algorithms, Adaboost algorithm has the following advantages:

1) The performance of the weak classifier can be slightly stronger than the random guess (correct rate > 50%) and does not require the weak classifier to have high performance. Therefore, the acquisition of weak classifiers becomes easier, thereby reducing the complexity of the algorithm and improving the detection efficiency.

2) In the parameters of the training process, only the number of rounds of training needs to be adjusted, and other parameters may not be modified.

3) In the multi-objective classification problem, it is very easy to generalize the Adaboost algorithm, and discrete values, numbers, texts, etc. can all be used as training data.

![Figure 3. Adaboost classifier detection process](image-url)
3. Training with OpenCV
In the experiment, the cement pavement cracks and asphalt pavement cracks were trained and studied respectively. Experimental simulation environment: Visual Studio 2015, win10 system, Core i7 processor, 8GB of running memory and CAMMC2588 array camera, NIKON 14mm lens and microEnable 5 vq8-cxp6d image acquisition card. Two types of pavement cracking training are as follows:

3.1. Selection of positive sample pictures
The horizontal crack screenshot frame is cut to a fixed ratio of 3:1, and the longitudinal crack is cut to a ratio of 1:3. The cut-out sample image requires that the crack is not blocked, the image is clear and less noise. The effective feature part accounts for 80% of the cropped image. After cropping, create a vec file that describes the positive sample in a uniform sample size of 90pix × 30pix.

3.2. Selection of negative sample pictures
The negative sample is also called a negative sample or background description file. The sample collection is much easier than the positive sample. The negative sample collected must not contain the characteristics of the positive sample. The negative sample size is larger than the positive sample. During the training process, the OpenCV training program will automatically extract a block of the same size as the positive sample from the negative sample image library as a negative sample. The negative sample preferably contains the background in the positive sample. In this way, when training the classifier, the non-target features in the positive samples can be removed quickly and timely. As the number of negative samples increases, the classifier training time becomes longer. The number and size of negative samples are proportional to the time of OpenCV training.

3.3. Pavement crack detection experiment
Image processing and pattern recognition algorithms are integrated in opencv_haartraining.exe provided by OpenCV. After the trained, the classifier is used to identify the cracks on the cement pavement and cracks on the asphalt pavement to test the performance of the Adaboost algorithm for crack detection. The experiment using C# language, in 2015 environment running Visual Studio, call CascadeClassifier function, as follows:

1) Load classifier and crack image to be detected
   • InitializeComponent();
   • haar = new CascadeClassifier(@"D:\cascade.xml");\Load classifier
   • frame = new Image<Bgr, byte>(@"D:\test.jpg");\Loading images
   • pictureBox1.Image = frame.ToBitmap();
2) Detect cracks
   • Rectangle[] cracks = haar.DetectMultiScale(frame, double scaleFactor, int minNeighbors, Size minSize, Size maxSize);
   \Finds rectangular in the given image
3) Mark cracks
   • CvInvoke.Rectangle(frame, crack, new Bgr(Color.Red).MCvScalar, 1);\Mark cracks with red rectangles
The experimental results are shown in figure 4 and figure 5.
It can be seen from the test results of the two groups that the red rectangular frame is distributed along the crack texture direction and has strong anti-interference ability.

4. Evaluation and conclusion

4.1. Evaluation

By simulation, the parameters of training are shown in table 1.

The recognition accuracy rate is calculated according to formula 3,

\[
\text{accuracy rate} = \frac{\sum_{i=1}^{n} l_k}{l_c} \times 100\%
\]  

(3)

Where \( l_k \) is the length of the longest rectangular edge and the \( l_c \) is the length of crack. The \( n \) is the number of rectangular.

In the experiment, different scaleFactor and minNeighbors have significant influence on detection accuracy for both transverse and longitudinal crack detections. minSize and maxSize control the size of the detection rectangle (see section 3 for the use of each parameter). The detectMultiScale function setting parameters are shown in table 2.

| Crack Types       | numPos | numNeg | Training Time(h) | numStages | Detection Time(ms) | accuracy rate (%) |
|-------------------|--------|--------|------------------|-----------|-------------------|-------------------|
| Concrete Pavement | Transverse | 1500   | 2000             | 104h      | 19                | 376.8             | 83.03             |
|                   | Longitudinal | 1500   | 2000             | 118h      | 19                | 293.3             | 92.52             |
| Asphalt Pavements | Transverse | 1800   | 2400             | 152h      | 16                | 322.4             | 87.36             |
|                   | Longitudinal | 1000   | 2000             | 124h      | 16                | 286.5             | 85.71             |
Table 2. DetectMultiScale function parameter settings

|               | scaleFactor | minNeighbors | minSize    | maxSize     |
|---------------|-------------|--------------|------------|-------------|
| Concrete Pavement | 4.5         | 1            | (150,150)  | (1000,1000) |
| Asphalt Pavements | 3.4         | 1            | (200,200)  | (1000,1000) |

4.2. Conclusion

Pavement crack detection is an important part of the pavement crack assessment system. After accurately identifying the crack area, the crack width and length can be processed in the next step. In this paper, we use the Haar feature and Adaboost algorithm to realize the recognition of horizontal and longitudinal cracks on cement and asphalt pavement. The inspection rectangle can cover more than 83% of the crack length.

On the basis of the above work, we will identify the parameters such as the width and length of the crack and optimize the Adaboost algorithm to adapt to the hardware acceleration technology of FPGA to realize the rapid recognition and extraction of road cracks.

Acknowledgments

We would like to thank Shandong transportation planning and design institute for providing experimental vehicles and experimental sites for our experiments. At the same time, thanks to China Daheng Group, Inc. Beijing Image Vision Technology Branch for providing us with image acquisition hardware.

References

[1] H Oliveira, PL Correia. (2013) Automatic Road Crack Detection and Characterization. IEEE Transactions on Intelligent Transportation Systems, vol.14, no.1: 155-168.
[2] P Hu., (2013) The Digital Image Identification System of Highway Pavement Disease Application Research, Chongqing Jiao tong University Chongqing, China.
[3] A Tedeschi, F Benedetto. (2017) A real-time automatic pavement crack and pothole recognition system for mobile Android-based device, Advanced Engineering Informatics, vol.32: 11-25.
[4] D Zhang, Q Li, Y Chen, M Cao, L He, B Zhang. (2017) An Efficient and Reliable Coarse-to-fine Approach for Asphalt Pavement Crack Detection, Image and Vision Computing, vol.57, no.3:130-146.
[5] D Zhang, Q Q Li, Y C, M Cao, L He, B L, Zhang. (2017) An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection, Image and Vision Computing, vol.57:130-146.
[6] P Viola , M Jones. (2004) Robust Real-Time Face Detection, International Journal of Computer Vision, vol.57, no.2: 137-154.
[7] T Ojala , M Pietikäinen , T Mäenpää, (2000) Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.7:971-987.