In this paper, we study a task of slope collapse detection (SCD) for river embankment and formulate it as the tasks of motion detection and image recognition. Specifically, we introduce an SCD method based on motion detection and image recognition technologies to help inspector attendants detect the slope collapse. In this method, we use the foreground motion detection algorithm to identify the slope collapse of the scene of the river embankment. Since the moving targets in the foreground may not only be the slope collapse but also maybe some biology, we further use the image feature extraction and image recognition technology to recognize the foreground motion area, thus eliminating the influence of the biology on the detection results. Experimental results on the relevant scene data show that the proposed method can identify the slope collapse in real-time, and can effectively eliminate the motion interference of the biology, which has a high practical value.

1. Introduction

As an important part of flood control projects, river embankment can resist flooding, prevent tides and waves, and protect the safety of residents and agricultural production in the dike [1, 2]. It is necessary to manage and maintain the river embankment to make it always keeps solid. Slope collapse is one of the commonest deformed forms of the river embankment. Early detection and prevention of the occurrence of the slope collapse are the realistic requirements for guaranteeing the overall safety of river embankment. However, this inspection task mainly depends on manpower so far, which has the following drawbacks:

(a) High cost: the river embankment is usually in a large scale, with long lines and multiple types, which needs many persons with professional quality to inspect at the same time and leads to a huge staff cost.

(b) Low reliability: rely on manpower to inspect the river embankment is easily affected by natural scenes and traffic conditions. For example, during the flood period, it is often accompanied by the process of rainfall and strong convective weather, which is difficult to realize the full coverage of river prevention projects and timely capture the slope collapse only with the help of simple equipment.

(c) High risk of human safe: slope collapse always is accompanied by the water level rising and flow rate enhancing, which increases the risk of the safety of the inspection personnel.

With the development of computer and network technology, some areas have tried to use video monitoring to timely monitor the river embankment and to reduce the cost of manual inspection. However, they only used the video surveillance to simply monitor the river embankment and did not quantitatively analyze the video data, which leads to the waste of video data resources and does not inherently increase the efficiency of inspection.

In order to improve the intelligent and digital levels of the flood control projects, make full use of the existing video data resources, and reduce the cost of labor, this paper proposes a slope collapse detection method based on computer vision and machine learning technologies. Specifically, we formulate the slope collapse detection as the tasks of motion detection and image classification. To timely detect the motion foreground area, we perform the background modeling technology on the captured river
embankment surveillance video [3]. Since the detected motion foreground area may include some biology image like birds, dogs, humans, which brings interference on the slope collapse detection, we collect an image dataset from the flood control projects that contain slopes, humans, birds, and so on. Based on the dataset, we train a support vector machine model that can recognize slope collapse area [4]. In practical application, after detecting the motion foreground area, we further calculate the color histogram [5] and gradient histogram features [6] of the area, respectively. Then, we use the trained support vector machine model to classify the feature data samples to judge whether the detected motion foreground area is the slope collapse or not, thus, the influence of environmental noise on detection results can be eliminated effectively. The experimental results on the reality river embankment scene show that the proposed method can effectively identify the slope collapse area and eliminate the noise and other disturbance information under the complex environment.

In summary, the contributions of this paper are as follows:

1. We are the first work to try to use computer vision and machine learning technologies to perform automatic slope collapse detection, which improves detection efficiency and reduces manpower detection costs.
2. We contribute a slope image dataset, in which about 1000 slope images and 500 biology images are collected.
3. The proposed slope collapse detection method shows high performance in practical application and meets the requirements of the inspection task.

The remainder of this paper is organized as follows: the proposed method is described in Section 2. In Section 3, we present the experimental results on the collected dataset and real riverbank scene. We give some conclusions and future work of this paper in Section 4.

2. Proposed Method

In this section, we present the carefully designed slope collapse detection method by first illustrating the background modeling technique and then presenting the slope collapse recognition technology. Finally, the slope collapse recognition technology is described.

2.1. Background Modeling for Detecting the Motion

Detecting the motion from the slope video data in real-time is the prime task of our method. In this paper, we use the background subtraction model for motion detection [7]. The basic idea of background subtraction is to establish a mathematical background model to approximate the simulated background, and then make a difference between the image of the current frame and the background model. Among the difference results, those pixel values whose change more than the threshold value are identified as foreground pixels, while the changes of pixel values less than the threshold value are identified as background pixels. The commonly adopted background subtraction models include the K-nearest neighbor model [8], mixed Gaussian model [9], codebook model [10], and Vibe (“atmosphere” field) model [11]. Among these methods, we use the mixed Gaussian model to detect the motion foreground.

The basic principles of the mixed Gaussian model are as follows:

1. As shown in Figure 1, we first define \( K(K>2) \) the single Gaussian models
2. Input the pixels into these single Gaussian models, respectively
3. If the pixels satisfy the mixed Gaussian distribution, then they are considered as the background pixels; otherwise, they are regarded as the foreground pixels

This process can be formulated as follows:

\[
P(X_t) = \sum_{i=1}^{K} \omega_{k,t} \times \eta(X_t, \mu_{k,t}, \sigma^2_{k,t}),
\]

where \( K \) is the number of the single Gaussian models, \( \omega_{k,t} \) is the weight of the \( k \)-th Gaussian model at time \( t \), \( \eta(X_t, \mu_{k,t}, \sigma^2_{k,t}) \) represents the probability density function of the \( k \)-th Gaussian model at time \( t \), and \( \mu_{k,t} \) and \( \sigma^2_{k,t} \) represent the mean and variance of the probability density function of the \( k \)-th Gaussian model at time \( t \), respectively.

To improve the reliability of the background model, we sort the \( K \) Gaussian distributions by descending order according to their corresponding values of \( \omega_{k,t}/\sigma_{k,t} \), thus the Gaussian distributions most likely to describe the background changes are arranged at the front of the sorted sequence, while the distributions caused by other unstable factors, such as environmental noise, are arranged behind the sequence.

Then, we take the first \( B \) Gaussian distributions from the sequence as the reliable background distributions:

\[
B = \arg \min_b \left( \sum_{k=1}^{b} \omega_{k,t} \geq T \right),
\]

where \( T \) is the background threshold that controls the number of background distributions.

The updating rules of the mixed Gaussian model are as follows:

First, update the weights. The weight of the \( K \) Gaussian distributions at time \( t+1 \) is

\[
\omega_{k,t+1} = (1-\alpha)\omega_{k,t} + \alpha M,
\]

where \( \alpha \) is the learning rate; we set it to 0.5. When the pixel values satisfy the \( k \)-th Gaussian distribution, \( M=1 \); otherwise, \( M=0 \).
For the mean and variance, we only update the Gaussian distributions that the pixel values satisfied as follows:

\[
\mu_{k,t+1} = (1 - \rho)\mu_{k,t} + \rho X_{t+1}, \quad (4)
\]

\[
\sigma^2_{k,t+1} = (1 - \rho)\sigma^2_{k,t} + \rho (X_{t+1} - \mu_{k,t+1})^T (X_{t+1} - \mu_{k,t+1}), \quad (5)
\]

where \( \rho \) is the update rate of the weight and \( \mu_{k,t} \) represents the mean vector of the \( K \) Gaussian distributions at time \( t \).

2.2. Image Feature Extraction. The foreground motion objective captured by the background subtraction model may be the biology images like human, bird, and dog which interferes the slope collapse detection. To determine whether the foreground motion object is slope collapse, we further extract the color histogram features and gradient histogram features of the objective.

2.3. Color Histogram Features. As a simple and effective feature descriptor based on statistical characteristics, the color histogram is widely used in many image retrieval systems [5, 12–14]. It describes the proportion of different colors in the whole image and does not care about the spatial position of each color; in other words, it is particularly good for describing the images that are difficult to automatically segment.

The commonly used color space is RGB, since most of the digital image is expressed in this color space. However, the RGB space structure does not conform to the subjective judgment of humans of color similarity. To solve this problem, the HSV color space [15], Luv color space [16], and Lab color space [17] are proposed. These color spaces are closer to people’s subjective perception of colors. The HSV color space has three components, each of which represents hue (\( H \)), saturation (\( S \)), and value (\( V \)). As shown in Figure 2, we use the cylinder to represent the HSV color space. The cross section of the cylinder can be thought of as a polar coordinate system. Among the system, \( H \), \( S \), and \( V \) are represented by the polar angles in polar coordinates, the length of the polar axis in polar coordinates, and the height of the axis of the cylinder, respectively. In this work, we choose the HSV color space to calculate the color histogram features because it has the following advantages:

(a) The \( H \) and \( S \) components of the space are relatively independent of the \( V \) component, which is easier to distinguish numerically.

(b) The \( H \) and \( S \) components are closely related to the way that humans perceive color, which is more consistent with the interpretation of characteristics.

Many studies show that the \( V \) component almost has no effect on the recognition of color, \( H \) component has the highest color feature recognition degree, and \( S \) component represents the degree of color close to the spectral color. Since the river embankment is in an open-air environment, the illumination varies greatly within one day; therefore, we only use the \( H \) and \( S \) components as color histogram features.

Since the size of the foreground region is different, the calculated size of the surrounding rectangle is also different. To ensure the value range of the features, we conduct minimax standardization on the two histograms of \( H \) and \( S \) components and carry out the linear transformation on the original data. The transformation function is shown as follows:

\[
x^* = \frac{x - \min}{\max - \min}, \quad (7)
\]

where \( x^* \) is the transformed output value, \( x \) is the original data, and \( \max \) and \( \min \) represent the maximum and minimum values in a single sample.

Finally, the two histograms are concatenated together to synthesize the HS feature.

2.4. Gradient Histogram Features. Gradient histogram is a feature descriptor for object detection in the computer vision domain [18–22]. The feature is constructed by calculating and counting the gradient information of the image. As shown in Figure 3, the calculation procedures of gradient histogram are as follows:

(1) Image gray processing.
(2) Using the gamma correction method to standardize the color space of the image.
(3) Calculating the gradient for each pixel of the image.
(4) Dividing the image into several small cells, such as 8×8 pixels.
(5) Through counting the gradient histogram of each cell, obtaining the descriptor of each cell.
(6) Using several cells to form a block (for example, 2×2 cells), and the HOG feature descriptor of a block can be obtained by connecting the feature descriptors of all cells in a block in series.
(7) The gradient feature descriptor of the image can be obtained by concatenating the feature descriptors of all blocks.

2.5. Slope Collapse Recognition Based on Support Vector Machine. After obtaining the features of the foreground motion objective, we use the support vector machine (SVM) to recognize it. SVM was proposed by Cortes and Vapnik in 1995, which has shown many unique advantages in solving small samples, nonlinear, and high-dimensional pattern recognition [23–27]. The basic principle of SVM learning is to find the separation hyperplane which can divide the training data set correctly and have the largest geometric interval. As shown in Figure 4, the solid line between the two dashed lines is the separation hyperplane. For a linearly
separable dataset, there are infinitely many such hyperplanes (i.e., perceptions), but the separation hyperplane with the largest geometric interval is unique. According to the different cases of hyperplane segmentation, support vector machines can be divided into the following cases:

(a) Hard-margin SVM (linear separable support vector machines): when the training data is linearly separable, a linearly separable support vector machine can be obtained by hard interval maximization.

(b) Soft-margin SVM: when the training data is approximately linearly separable, a linear support vector machine can be obtained by soft interval maximization.

(c) Nonlinear SVM: when the training data are linearly inseparable, a nonlinear support vector machine can be obtained by kernel method and soft interval maximization.

Since the collected dataset is linearly inseparable, we use the nonlinear SVM to deal with it.

3. Experiment

In this section, we present the implementation details and experimental results of the proposed method. The flow chart of the implementation process is shown in Figure 5.

3.1. Motion Detecting Results. We use the mixed Gaussian model to detect the foreground motion. Figure 6 shows the simulation result of foreground motion detection. The top of Figure 6 is the simulated slope collapse scene; we pull the baffle to simulate the slope collapse. The bottom is the foreground motion detection result, in which the white areas are the detected motion areas. From the bottom of Figure 6, we can observe that the motion areas are detected by the mixed Gaussian model efficiently.

3.2. Slope Collapse Area Recognition

3.2.1. Dataset. After detecting the foreground motion, we need to identify whether the motion areas are the slope collapse. As mentioned above, we use the SVM model to recognize the detected motion objectives. To this end, a dataset that contains 1000 slope images and 500 biology images is collected from the river embankment. Figure 7 displays some instances of the slope and biology images from the collected datasets. We randomly select 800 slope images and 400 biology images as the training set and use the rest of the 300 images for testing.

3.2.2. Feature Extraction. We carry the color histogram and gradient histogram features extraction on the collected images. Since the value range of a pixel in the computer is 0~256 and the value range of H component is 0~360, we regulate the value range of H component from 0~360 into 0~180. The value range of S component is 0~255, we concatenate the histograms of H and S components to form the final color histogram feature (HS) with the dimension of 435.

Because the dimension of the feature of the directional gradient histogram is related to the size of the image, it is necessary to normalize the image into the same size before extracting the gradient histogram features. Through statistical analysis of the samples’ size, the following results are obtained in Table 1:

Based on the statistical results, we resize the image into 64×128 pixels, with an aspect ratio of 1:2. We set the block size, block stride, cell size, and bin of the gradient histogram to 16×16, 8×8, 8×8, and 9, respectively. The final dimension of the gradient histogram feature (HOG) is set to 3780.

3.2.3. Training and Optimization. We conduct the experiment on the framework of sklearn [28] and use the linear kernel, polynomial kernel, and Gaussian kernel as the kernel functions of the SVM model, respectively. The parameters of the three kernels are set in Table 2, where the C is penalty parameter, Logspace is a function of python, and Gamma and Degree are the hyperparameters of SVM. We use the grid search method to optimize the parameters of the SVM. The precision, recall, and f1-score are adopted as evaluation metrics for evaluating the performance of the SVM model [29].

3.3. Recognition Results. To verify the effectiveness of the feature selection, we first only use the color histogram features (HS) to train Gaussian kernel SVM. Then, the color histogram and gradient histogram features (HOGs) are combined to train the Gaussian kernel SVM. The experimental results on the dataset are shown in Table 3. From Table 3, we can see that the combined features achieve a higher recognition rate.

To validate which kernel is effective and better to recognize the slope, we use the color histogram and gradient histogram features to train the linear kernel SVM, polynomial kernel SVM, and Gaussian kernel SVM, respectively. The experiment results of the three SVMs on the collected dataset are shown in Table 4, in which we can observe that the Gaussian kernel SVM beats all other two compared SVM models and achieves the highest performance. Therefore, we choose Gaussian kernel SVM as the classifier to recognize the slope.

We further implement some extra experiments to evaluate the practicability of the proposed method. The experiments were carried out under different scenes, including slope collapse simulation and the real slope collapse event. Figure 8 shows the slope collapse detection results. From the figure, we can see the slope collapse areas are detected accurately, which further demonstrates the effectiveness of the proposed model.
Figure 4: Separation hyperplane.

Input Background modeling → Feature extraction → SVM → Output

Figure 5: Flow chart of the implementation.

Figure 6: Simulation result of motion detecting.

Figure 7: Samples of images from the collected dataset.
Table 1: Statistical analysis of the samples’ size.

| Aspect ratio          | Proportion (%) |
|-----------------------|----------------|
| 1:1.3~1:1.5           | 17.5           |
| 1:1.5~1:2             | 21.7           |
| 1:2~1:2.5             | 27.8           |
| 1:2.5~1:3             | 22.4           |
| Others                | 10.6           |

Table 2: Parameters setting of the three kernels.

| Linear kernel              | Parameter | Range   |
|----------------------------|-----------|---------|
|                           | C         | Logspace (-2,4,20) |
| Polynomial kernel          | Parameters | Logspace (-2,2,20) |
|                           | C         | 2, 3, 4, 5 |
|                           | Gamma (γ) | Logspace (-2,2,20) |
| Gaussian kernel            | Parameters | Logspace (-2,2,20) |
|                           | C         | Logspace (-2,4,20) |
|                           | Gamma (1/2δ²) | Logspace (-2,2,20) |

Table 3: Results on different feature selections.

| Category | Precision | Recall | F1-score |
|----------|-----------|--------|----------|
| HS       | Biology   | 1.00   | 0.86     | 0.93     |
|          | Slope     | 0.84   | 1.00     | 0.92     |
|          | Avg/total | 0.93   | 0.92     | 0.92     |
| HS + HOG | Biology   | 1.00   | 0.98     | 0.97     |
|          | Slope     | 0.96   | 1.00     | 1        |
|          | Avg/total | 0.98   | 0.99     | 0.98     |

Table 4: Results on different feature selections.

| Category | Precision | Recall | F1-score |
|----------|-----------|--------|----------|
| Linear kernel SVM | Biology   | 0.86   | 1.00     | 0.92     |
|            | Slope     | 1.00   | 0.83     | 0.91     |
|            | Avg/total | 0.93   | 0.92     | 0.92     |
| Polynomial kernel SVM | Biology   | 0.95   | 0.98     | 0.97     |
|            | Slope     | 0.99   | 0.96     | 0.92     |
|            | Avg/total | 0.97   | 0.97     | 0.94     |
| Gaussian kernel SVM | Biology   | 1.00   | 0.98     | 0.97     |
|            | Slope     | 0.96   | 1.00     | 1        |
|            | Avg/total | 0.98   | 0.99     | 0.98     |
4. Conclusion

To improve the level of information and intelligence of water conservancy and reduce the cost of labor, a slope collapse identification method based on computer vision and machine learning technology was proposed. The method used the foreground motion detection algorithm to identify the foreground motion in real-time. In order to eliminate the influence of other environmental disturbances on the detection results, the image feature extraction technology and the machine learning technology were combined to recognize the foreground region. The experimental results on slope collapse simulation and the real slope collapse event showed that the proposed method can identify the collapse area accurately, which has high practical value in the water conservancy domain. Future work will focus on using deep learning technology to detect the slope collapse.

Data Availability

A dataset that contains 1000 slope images and 500 biology images is collected from the river embankment.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the grants of the enterprise fund of Yellow River Engineering Consulting Co (No. 2019-ky11).

References

[1] L. Busato, J. Boaga, L. Peruzzo et al., “Combined geophysical surveys for the characterization of a reconstructed river embankment,” Engineering Geology, vol. 211, pp. 74–84, 2016.
[2] K. Van Looy, O. Honnay, and B. Bossuyt, “The effects of river embankment and forest fragmentation on the plant species richness and composition of floodplain forests in the meuse valley, Belgium,” Belgian Journal of Botany, vol. 136, no. 2, pp. 97–108, 2003.
[3] T. Bouwmans, F. El Baf, and B. Vachon, “Background modeling using mixture of Gaussians for foreground detection - a survey,” Recent Patents on Computer Science, vol. 1, no. 3, pp. 219–237, 2008.
[4] A. Christmann and I. Steinwart, Support Vector Machines Applications, Springer, New York, NY, USA, 2014.
[5] S. Sergyan, “Color histogram features based image classification in content-based image retrieval systems,” in Proceedings of the 2008 6th international symposium on applied machine intelligence and informatics, pp. 221–224, IEEE, Herlany, Slovakia, January 2008.
[6] J. A. Rodriguez and F. Perronnin, “Local gradient histogram features for word spotting in unconstrained handwritten documents,” in Proceedings of the 1st ICFHR, pp. 7–12, Shenzhen, China, October 2008.
[7] M. Piccardi, “Background subtraction techniques: a review,” in Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), pp. 3099–3104, IEEE, The Hague, Netherlands, October 2004.
[8] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, “A spatiotemporal correlitive k-nearest neighbor model for short-term traffic multistep forecasting,” Transportation Research Part C: Emerging Technologies, vol. 62, pp. 21–34, 2016.
[9] A. Komárek and E. Lesaffre, "Generalized linear mixed model with a penalized Gaussian mixture as a random effects distribution," *Computational Statistics & Data Analysis*, vol. 52, no. 7, pp. 3441–3458, 2008.

[10] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[11] O. Barnich and M. Van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2010.

[12] P. Liu, J.-M. Guo, K. Chammenonthai, and H. Prasetyo, “Fusion of color histogram and LBP-based features for texture image retrieval and classification,” *Information Sciences*, vol. 390, pp. 95–111, 2017.

[13] J. Mas and G. Fernandez, "Video shot boundary detection based on color histogram TRECVID," 2003.

[14] H. Ali, M. I. Lali, M. Z. Nawaz, M. Sharif, and B. A. Saleem, "Symptom based automated detection of citrus diseases using color histogram and textural descriptors," *Computers and Electronics in Agriculture*, vol. 138, pp. 92–104, 2017.

[15] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *Proceedings of the International Conference on Image Processing*, IEEE, New York, NY, USA, September 2002.

[16] H. B. Kekre and S. D. Thepade, "Image blending in vista creation using Kekre’s LUV color space," in *Proceedings of the SPIT-IEEE Colloquium and International Conference*, pp. 04-05, Sardar Patel Institute of Technology, Andheri, Mumbai, India, January 2008.

[17] D. Margulis, *Photoshop LAB Color: The Canyon Conundrum and Other Adventures in the Most Powerful Colorspace*, Peachpit Press, Berkeley, CA, USA, 2005.

[18] D. Sen and S. K. Pal, "Gradient histogram: thresholding in a region of interest for edge detection," *Image and Vision Computing*, vol. 28, no. 4, pp. 677–695, 2010.

[19] W. Zuo, L. Zhang, and C. Song, "Texture enhanced image denoising via gradient histogram preservation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1203–1210, Portland, OR, USA, June 2013.

[20] C. Shu, X. Ding, and C. Fang, "Histogram of the oriented gradient for face recognition," *Tsinghua Science and Technology*, vol. 16, no. 2, pp. 216–224, 2011.

[21] W. Bao, Z. Huang, C. A. Yuan, and D. S. Huang, "Pupylation sites prediction with ensemble classification model," *International Journal of Data Mining and Bioinformatics*, vol. 18, no. 2, pp. 91–104, 2017.

[22] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[23] W. Bao, D. Wang, and Y. Chen, "Classification of protein structure classes on flexible neutral tree," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, pp. 1122–1133, 2016.

[24] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines Neural networks for signal processing VII," in *Proceedings of the 1997 IEEE signal processing society workshop*, pp. 276–285, IEEE, Banff, Canada, July 1997.

[25] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," *Advances in Neural Information Processing Systems*, vol. 11, pp. 368–374, 1999.

[26] K. S. Durgesh and B. Lekha, "Data classification using support vector machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1–7, 2010.

[27] Z. Xuegong, “Introduction to statistical learning theory and support vector machines,” *Acta Automatica Sinica*, vol. 26, no. 1, pp. 32–42, 2000.

[28] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] W. Bao, "Multi-features prediction of protein translational modification sites," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1453–1460, 2017.